GOVERNMENT OF TAMILNADU

పదకొండవ తరగతి

HIGHER SECONDARY FIRST YEAR

కంప్యూటర్ సైన్స్

COMPUTER SCIENCE

TELUGU MEDIUM

Department of School Education

Untouchability is Inhuman and a Crime

11th_Computer_Science_Introduction_Telugu.indd 1 30-03-2023 18:12:22

Government of Tamil Nadu

First Edition - 2022

Revised Edition - 2023

Reprint - 2024

(Published Under New Syllabus)

Content Creation

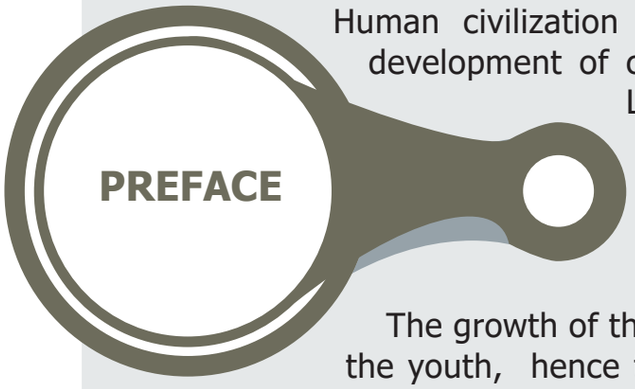
State Council of Educational
Research and Training
© SCERT 2022

Printing & Publishing

Tamil NaduTextbook and Educational
Services Corporation

www.textbooksonline.tn.nic.in

11th_Computer_Science_Introduction_Telugu.indd 2 30-03-2023 18:12:22



Human civilization achieved the highest peak with the development of computer known as "Computer era". Literate are those who have the knowledge in using the computer whereas others are considered illiterate inspite of the other degrees obtained.

The growth of the nation at present lies in the hands of the youth, hence the content of this book is prepared in such a way so as to attain utmost knowledge considering the future needs of the youth.



- This book does not require prior knowledge in computer Technology
- Each unit comprises of simple activities and demonstrations which can be done by the teacher and also students.
- Technical terminologies are listed in glossary for easy understanding
- The " Do you know?" boxes enrich the knowledge of reader with additional information
- Workshops are introduced to solve the exercises using software applications
- QR codes are used to link supporting additional
- materials in digital form

11th_Computer_Science_Introduction_Telugu.indd 3 30-03-2023 18:12:22

CAREER GUIDANCE AFTER 12TH

COURSES	COLLEGES/ UNIVERSITIES	PROFESSION
B.E / B.Tech	All University and their affiliated Colleges and Self financing Colleges in India and Abroad.	Software Engineer, Hardware Engineer, Software Development, Healthcare Section, IT & ITEs
Science and Humanities		
B.Sc (Computer Science) BCA B.Sc (Maths, Physics, Chemistry, Bio-Chemistry, Geography, journalism, Library Sciences, Political Science, Travel and Tourism)	All University and their affiliated Colleges and Self financing Colleges in India and Abroad.	Government Job and Private Company BPO, Geologist, Journalist
LAW		
LLB B.A+LLB B.Com BBM+LLB BBA+LLB	All University and their affiliated Colleges and Self financing Colleges in India and Abroad.	Lawyer, Legal Officer, Govt Job
CA	The Institute of Chartered Accountant of India (ICAI)	CA Private and Govt.
Diploma	Government Polytechnic and Self-financing colleges	Junior Engineer (Government and Private)
Commerce Courses		
B.com-Regular, B.com-Taxation & Tax Procedure, B.com-Travel &Tourism, B.com-Bank Management, B.com-Professional, BBA/BBM-Regular, BFM- Bachelors in Financial Markets, BMS-Bachelors in Management Studies, BAF- Bachelors in Accounting & Finance, Certified Stock Broker & Investment Analysis, Certified Financial Analyst, Certified Financial Planner, Certified Investment Banker	All University and their affiliated Colleges and Self financing Colleges in India and Abroad.	Private Organization , Government ,Banking sectors and prospects for self – employment.

11th_Computer_Science_Introduction_Telugu.indd 4 30-03-2023 18:12:22

COURSES	COLLEGES/ UNIVERSITIES	PROFESSION
Management Courses		
Business Management Bank Management Event Management Hospital Management Human Resource Management Logistics Management	All University and their affiliated Colleges and Self financing Colleges in India and Abroad.	Private Organization , Government ,Banking sectors and prospects for self – employment.
LAW		
LLB B.A+LLB B.Com BBM+LLB BBA+LLB	All University and their affiliated Colleges and Self financing Colleges in India and Abroad.	Lawyer, Legal Officer, Private Organization , Government, Banking sectors and prospects for self – employment
CA-Chartered Accountant CMA-Cost Management Accountant. CS-Company Secretary (Foundation)	The Institute of Chartered Accountant of India (ICAI)	CA, Private Organization, Government ,Banking sectors and prospects for self – employment.
Science and Humanities		
B.Sc.Botany B.Sc.Zoology B.Sc.Dietician & Nutritionist B.Sc.Home Science B.Sc.Food Technology B.Sc.Dairy Technology B.Sc. Hotel Management B.Sc. Fashion Design B.Sc. Mass Communication B.Sc. Multimedia B.Sc. -3D Animation	All University and their affiliated Colleges and Self financing Colleges in India and Abroad	Government Job and Private Company BPO, Geologist, Journalist
LAW		
LLB B.A+LLB B.Com BBM+LLB BBA+LLB	All University and their affiliated Colleges and Self financing Colleges in India and Abroad.	Lawyer, Legal Officer, Govt Job
CA	The Institute of Chartered Accountant of India (ICAI)	CA Private and Govt.
Diploma	Government Polytechnic and Self-financing colleges	Junior Engineer (Government and Private)

11th_Computer_Science_Introduction_Telugu.indd 5 30-03-2023 18:12:22

విషయసూచిక

అధ్యాయము	పాఠ్యశీర్షిక	పుట సంఖ్య	నెల
	విభాగం I – కంప్యూటర్ యొక్క ప్రాథమిక లక్ష్యాలు		
1	కంప్యూటర్ గురించి పరిచయము	1	జూన్
2	సంఖ్యా వ్యవస్థలు	14	జూన్
3	కంప్యూటర్ వ్యవస్థ	42	జూన్
4	ఆపరేటింగ్ సిస్టమ్ యొక్క సైద్ధాంతిక అంశాలు	52	జూలై
5	విండోస్ ఆపరేటింగ్ సిస్టమ్‌తో పనిచేయుట	61	జూలై
	విభాగం II- అల్గోరిథమిక్ సమస్య పరిష్కారం		
6	స్పెసిఫికేషన్ మరియు సంగ్రహణ	80	జూలై
7	కూర్పు మరియు విభజన	92	ఆగస్టు
8	పునరుక్తి మరియు పునరావృతం	106	ఆగస్టు
	విభాగం III – C++ పరిచయం		
9	C++ పరిచయం	119	సెప్టంబరు
10	ఫ్లో ఆఫ్ కంట్రోల్	158	సెప్టంబరు
11	ఫంక్షన్స్	186	అక్టోబరు
12	అరేస్ మరియు నిర్మాణాలు	212	అక్టోబరు
	విభాగం IV - C++ తో ఆబ్జెట్ ఓరియంటెడ్ ప్రోగ్రామ్		
13	ఆబ్జెక్టు ఓరియంటడ్ ప్రోగ్రామింగ్ సాంకేతిక పద్ధతులు పరిచయం	234	అక్టోబరు
14	క్లాస్ మరియు లక్ష్యాలు	240	నవంబరు
15	ఫాలిమార్ఫిజం	256	నవంబరు
16	ఇన్‌హెరిటెన్స్	267	నవంబరు
	విభాగం V – కంప్యూటర్ ఎథిక్స్ మరియు సైబర్ భద్రత		
17	కంప్యూటర్ ఎథిక్స్ మరియు సైబర్ భద్రత	282	డిసెంబరు
18	తమిళ కంప్యూటరింగ్	293	డిసెంబరు
	ప్రాక్టికల్	305	

W0J9P

E - book

HIWV1

Assessment

11th_Computer_Science_Introduction_Telugu.indd 6 30-03-2023 18:12:23

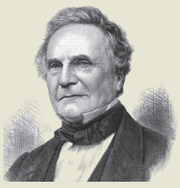
అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశము అభ్యసించిన తరువాత విద్యార్థులు నేర్చుకొను అంశములు.

- కంప్యూటర్ గురించి నేర్చుకొనుట
- వివిధ తరాల కంప్యూటర్‌ల గురించి నేర్చుకొనుట
- కంప్యూటర్ల యొక్క ప్రాథమిక కార్యకలాపాలను అర్థం చేసుకొనుట.
- కంప్యూటర్ల భాగాలు మరియు వాటి విధులను గురించి తెలుసుకొనుట.
- కంప్యూటర్ బూటింగ్ (booting) గురించి తెలుసుకొనుట

మీకు తెలుసా?

కంప్యూటర్ పితామహుడు



చార్లెస్ బాబేజ్ అనునతడు కంప్యూటర్ పితామహుడిగా పరిగణింపబడుచున్నారు. 1837 లో అనల్టికల్ ఇంజన్ అనునవి ఇతని యొక్క ఆవిష్కరణలు. ఈ విశ్లేషణాత్మక యంత్రాలలో Arithmetic Logic Unit (ALU) basic flow control మరియు అనుబంధ integrated మెమోరీలు ఉంటాయి. ఇది మొదటి సాధారణ కంప్యూటర్‌ను అభివృద్ధి చేయడానికి ఉపయోగించిన భావన.

506BD

కంప్యూటర్ గురించి పరిచయము

1.1 కంప్యూటర్ గురించి పరిచయము (Introduction to Computers)

కంప్యూటర్‌లు మన నిత్యజీవితంలో ప్రతి చోట కనిపిస్తూ ఉంటాయి, అవి విద్యావ్యవస్థ, పరిశోధనలు, పర్యాటక మరియు రవాణా రంగము, వాతావరణ సూచనలు, సాంఘీక కార్యకలాపాలు, ఇ – కార్యకలాపాలు మొదలగు రంగాలలో ఇవి ఉపయోగపడుచున్నవి. కంప్యూటర్‌లు తమ ఖచ్చితత్వము మరియు వేగముతో మన జీవితములో చాలా మార్పులను ప్రభావితపరుస్తున్నవి. ఇది నమ్మదగిన అంశం. ప్రస్తుతం కంప్యూటర్ లేకుండా ఏ సంస్థ పని చేయలేదు. నిజానికి వివిధ సంస్థలు కాగితము లేనివిగా మారి ఉన్నాయి. కంప్యూటర్లు కొన్ని సంవత్సరాలుగా కాలుక్కులేటర్ నుండి పోర్ట్‌బుల్ కంప్యూటర్ వరకు అభివృద్ధి చెందాయి.

లెక్కలను త్వరగాను మరియు ఖచ్చితంగాను పూర్తి చేయడానికి కంప్యూటర్ పరిశ్రమ ప్రారంభించి అభివృద్ధి చెందినది. మానవుల ద్వారా గణించడం నిదానంగాను మరియు కొన్ని లోపాలను కలిగి ఉండును. అందువలన త్వరగా గణించుటకు అబాకస్ పరికరం నుండి మొదలై ప్రస్తుతం అతివేగంగా గణించే కంప్యూటర్ సాధనాల వరకు ఈ రంగం అభివృద్ధి చెందినది.

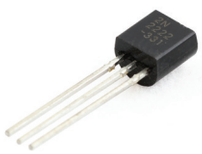
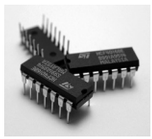
1.2 కంప్యూటర్‌ల యొక్క తరాలు (Generations of Computers)

కంప్యూటర్ పరిశ్రమ యొక్క అభివృద్ధి సాంకేతిక అభివృద్ధిపై ఆధారపడినది. కంప్యూటర్ యొక్క వివిధ దశలు ఆధారంగా వివిధ తరములుగా వర్గీకరించబడుచున్నది.

వ.సం	తరము	కాలము	ప్రధాన భాగం	ఉపయోగములు / డిమెరిట్స్
1	**మొదటి తరం**	1940 నుండి 1956	 **వ్యాక్యూమ్ ట్యూబ్**	• పరిమాణంలో పెద్దది. • ఎక్కువ విద్యుత్ శక్తి వినియోగం. • అధిక వేడి వలన పనిచేయకపోవడం. • యాంత్రిక భాషగా ఉపయోగించబడినది.

11th Computer Science_Chapter-1 Telugu.indd 1 10-03-2023 13:01:48

పంచ్‌కార్డులు ఉపయోగించారు

First Generation Computers - ENIAC , EDVAC , UNIVAC 1 ENIAC weighed about 27 tons, size 8 feet × 100 feet × 3 feet and consumed around 150 watts of power				
2	**రెండవ తరము**	1956 నుండి 1964	 **ట్రాన్సిస్టర్లు**	• మొదటి తరములో పోల్చితే చిన్నది • మొదటి తరములో పోలిస్తే తక్కువ విద్యుచ్ఛక్తిని వినియోగించెదరు. • తక్కువ వేడిని కలిగియుండును. • మొదటి ఆపరేటింగ్ సిస్టమ్ అభివృద్ధి • బ్యాచ్ ప్రాసెసింఇఇగ్ మరియు మల్టీ ప్రోగ్రామింగ్ చేయబడును. • యాంత్రిక భాష అలాగే అసెంబ్లీ భాష ఉపయోగించబడినవి.
రెండవ తరం కంప్యూటరు IBM 1401, IBM 1620, UNIVAC 1108				
3	**మూడవ తరము**	1964 నుండి 1971	 **సర్క్యూట్(IC)**	• తక్కువ విద్యుచ్ఛక్తిని వినియోగించును • ఉన్నత స్థాయి భాషలు ఉపయోగించబడినది. • కంప్యూటర్లు చిన్నవిగా ఉండినవి, ఖచ్చితత్వము మరియు వేగమైనవి.
మూడవ తరం కంప్యూటర్ IBM 360 సీరీస్ హనివెల్ 6000 సీరీస్				
4	**నాల్గవ తరము**	1971 నుండి 1980	 **Microprocessor** Very Large Scale Integrated Circuits (VLSI)	• చిన్నది మరియు వేగవంతమైనది • మైక్రో కంప్యూటర్ సీరీస్ IBM మరియు APPLE అభివృద్ధి చేయబడినది. • పోర్టబుల్ కంప్యూటర్లు పరిచయం చేయబడును.
5	**ఐదవ తరం**	1980 నుండి ఇప్పటి వరకు	 **Ultra Large Scale Integration (ULSI)**	• సమాంతర ప్రాసెసింగ్ • సూపర్ కండక్టర్లు • కంప్యూటర్ల పరిమాణం భారీగా తగ్గించబడినది మరియు గ్రాఫిక్స్‌లును చిత్రాలను గుర్తించగలదు. • కృత్రిమ పరిజ్ఞానం మరియు అత్యధిక ప్రమాణాలను పరిచయం చేయడం. • అధిక సంక్లిష్టమైన సమస్యలను పరిష్కార సామర్థ్యం కలిగియుండును. • high complex మరియు decision making తో కూడిన logical reasoning సమస్యలను సాధించును.

11th Computer Science_Chapter-1 Telugu.indd 2 10-03-2023 13:01:48

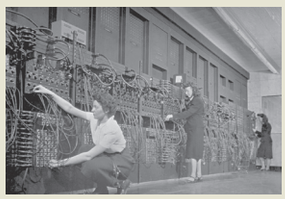
6	**ఆరవ తరము**	భవిష్యత్తులో		• సమాంతర మరియు పంపిణి కంప్యూటింగ్ • ఈ కంప్యూటర్లు తెలివైనవి వేగవంతమైనవి మరియు చిన్నవి • రోబోటిక్స్ అభివృద్ధి • సహజ భాషా ప్రాసెసింగ్ • Development of Voice Recognition Software

పట్టిక 1.1 కంప్యూటర్లు తరములు

మొదటి డిజిటల్ కంప్యూటర్



ఈ ENIAC (Electronic Numerical Integrator And Calculator) ను J. Presper Eckert మరియు John Mauchly ఇరువురు Pennsylvania విశ్వవిద్యాలయంలో కనుగొనడం జరిగినది. మరియు దీని నిర్మాణాన్ని 1943 లో ప్రారంభించి 1946 వరకు ముగింపు కాలేదు. ఇది 1800 చదరపు అడుగులు ఆక్రమించి మరియు 18000 vacuum tubes ఉపయోగించి దాని బరువు దాదాపు 50 టన్నులు. ENIAC అనునది పూర్తి సామర్థ్యంతో పనిచేయు మొట్టమొదటి digital computer.



1.3 ఆరవ తరం కంప్యూటింగ్ (Sixth Generation Computing)

దీని Artificial Neural Networks ఆధారంగా ఆరవ తరం కంప్యూటర్లు intelligent కంప్యూటర్‌లుగా నిర్వచించడం జరిగినది. ఈ ఆరవ తరం కంప్యూటర్ యొక్క సమూలమైన మార్పు అనునది విపరీతమైన Wide Area Network యొక్క పెరగుదల Artificial Intelligence లో Natural Language Processing (NLP) అనునది ఒక భాగము. ఇది మానవుని భాషను అర్థం చేసుకోవడానికి కంప్యూటర్‌కు నైపుణ్యాన్ని ఇచ్చును.



ఆప్టికల్ అక్షరం గుర్తింపు (ఆప్టికల్ గ్రాఫిమ్ రికగ్నిషన్) యంత్రము Indus Script ఉన్నాయి. క్షుణ్ణంగా నేర్చుకొని అభివృద్ధి చేయబడింది. నెట్‌వర్క్‌లు కృత్రిమ ఉపక్షేత్రం ఇంటెలిజెన్స్.

ఇవ్వబడిన photographలు scanner లేదా ఏదైనా చిత్రాలు అనునవి Indus Valley Civilization కళా ఖండానికి ప్రోత్సాహం వంటిది. ఇవి చిత్రాల నుండి inscriptions ను గుర్తిస్తాయి. Indus Script లో పూర్తిగా 417 చిహ్నాలు / రూపాంతరాలు / అక్షరాలు ఉన్నాయి మరియు మిషన్ తెలుసుకోవడానికి నిపుణుల స్థాయి స్థితిని పొందెందుకు కేవలం 3700+ డేటాలు ఉన్నాయి.





1.4. డేటా మరియు సమాచారం (Data and Information)

కంప్యూటర్ అంటే ఏమి అని మనందరికి తెలుసు. ఒక ఎలక్ట్రానిక్ పరికరం, ఇచ్చిన సూచనలను input గా తీసుకొని అతి వేగంగా పరిష్కరించి కావలసిన output ను డేటాను నిల్వచేయడం వాతావరణాన్ని అంచనా వేయడం విమానయాన రైల్వే బుక్కింగు కొరకు లేదా

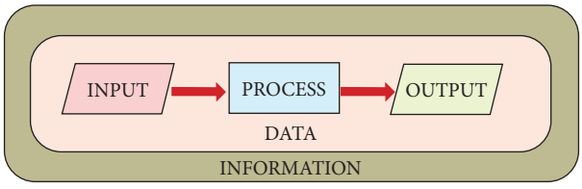
11th Computer Science_Chapter-1 Telugu.indd 3 10-03-2023 13:01:49

సినిమాలకు టికెట్లను బుక్ చేయడం కంప్యూటర్ ఆటలను ఆడడం వంటి విభిన్న పనులకు కంప్యూటర్లు చాలా పని చేస్తాయి.

డేటా (Data) : డేటా అనేది కమ్యూనికేషన్ ఇంటర్ ప్రిటేషన్ లేదా ప్రాసెసింగ్‌కు అనువైన ముడి వాస్తవాల యొక్క ప్రాసెస్ చేయడం

ఉదాహరణకు 134, 16 'కవిత', 'C' అనునది డేటా. ఇది ఒక అర్థవంతమైన సందేశాన్ని ఇవ్వదు.

సమాచారం (Information): సమాచారం అనునది వాస్తవాల సేకరణ నుండి ఒక ఖచ్చితమైన తీర్మానాలను చేయవచ్చును. సరళమైన మాటలలో డేటా అనునది అర్థవంతమైన ఆదేశించిన (లేదా) నిర్మాణాత్మక సమాచారాన్ని ఇవ్వడానికి ప్రాసెస్ చేయబడిన ముడి సేకరణ అని చెప్పగలము. ఉదాహరణకు కవితకు 16 సం॥రాల వయస్సు ఈ సమాచారం కవిత గురించి మరియు కొన్ని అర్థాన్ని తెలియజేస్తుంది. డేటాను సమాచారంగా మార్చడాన్ని డేటా ప్రాసెసింగ్ అని పిలుస్తారు.



పటము 1.1 డేటా మరియు సమాచారం

ఒక కంప్యూటర్ ఒక ఎలక్ట్రానిక్ పరికరం ఒక ముడి డేటాను ప్రాసెస్ చేయునిది. వినియోగదారుల నుండి (input) గా తీసుకొని సూచనలను సమితి నియంత్రణలో (program) ప్రాసెస్ చేసి ఫలితాన్ని (output) ఇస్తుంది మరియు భవిష్యత్తు ఉపయోగం కోసం నిల్వ (save) చేసుకొంటుంది.

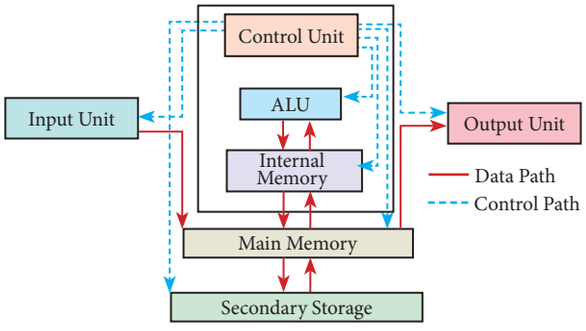
1.5 కంప్యూటర్ యొక్క భాగాలు (Components of a Computer)

కంప్యూటర్ అనునది హార్డ్‌వేర్ మరియు సాఫ్ట్‌వేర్‌ల కలయిక హార్డ్‌వేర్ అనునది ఒక కంప్యూటర్ యొక్క భౌతిక భాగం (motherboard) జ్ఞాపకశక్తి, పరికరాలు, మానిటర్; కీబోర్డ్ మొదలైనవి. సాఫ్ట్‌వేర్ అనునది కార్యక్రమాల (లేదా) సూచనల సమ్మేళనము హార్డ్వేరు మరియు సాఫ్ట్‌వేర్ కలిసి కంప్యూటర్ సిస్టమ్‌గా పని చేయడానికి ఉపయోగిస్తారు.



పటము 1.2 కంప్యూటర్

మొదటి కంప్యూటర్ యొక్క క్రియాత్మక భాగాలను పరిశీలిస్తాయి. కంప్యూటర్‌కు ఇచ్చిన ప్రతి పని (Input- Process) అవుట్‌పుట్ సైకిల్ (IPO cycle) లను అనుసరిస్తుంది. దీనికి నిర్ధిష్టమైన (Input) అవసరం. ఈ ఇన్‌పుట్‌ను (Process) చేసి కావలసిన (Output) ను ఉత్పత్తి చేస్తుంది.
ఇన్‌పుట్ యూనిట్ ఇన్‌పుట్‌ను తీసుకుంటుంది. CPU డేటాను ప్రాసెసింగ్ చేస్తుంది. (Output) ల యూనిట్ (Output) ని ఉత్పత్తి చేస్తుంది. ప్రాసెసింగ్ సమయంలో మెమోరీ యూనిట్ డేటా మరియు సూచనలను (Hold) చేస్తుంది.



పటము 1.3 కంప్యూటర్ యొక్క భాగాలు

1.5.1 ఇన్‌పుట్ యూనిట్ (Input Unit)

ఇన్‌పుట్ యూనిట్ అనేది ఏ విధమైన (data) నైనా అందించడానికి ఉపయోగిచబడుతుంది. ఇది తదుపరి ప్రాసెసింగ్ కొరకు మెమోరీ యూనిట్‌లో నిల్వచేయబడుతుంది. ఉదాహరణకు కీబోర్డ్, మౌస్ మొదలగునవి.

1.5.2 సెంట్రల్ ప్రాసెసింగ్ యూనిట్ (Central Processing Unit)

CPU అనేది సాఫ్ట్‌వేర్ సూచనలను వివరించే మరియు అమలు చేసే ప్రధాన భాగం. ఇది ఆపరేషన్‌ను కూడా నియంత్రిస్తుంది. మెమోరీ

11th Computer Science_Chapter-1 Telugu.indd 4 10-03-2023 13:01:50

ఇన్‌పుట్ మరియు అవుట్‌పుట్ యూనిట్ల వంటి అన్ని ఇతర భాగాల్లో ఇది బైనరీ డేటాను ఇన్‌పుట్‌గా అంగీకరిస్తుంది. సూచనల ప్రకారము డేటాను ప్రాసెస్ చేస్తుంది మరియు ఫలితాన్ని అవుట్‌పుట్‌గా అందిస్తుంది.

CPU మూడు భాగాలను కలిగి ఉంది. అవి : కంట్రోల్ యూనిట్, అర్థమెటిక్ మరియు లాజిక్ యూనిట్ (ALU) మరియు మెమోరీ యూనిట్

1.5.2.1 అంకగణితం మరియు లాజిక్ యూనిట్ (Arithmetic and Logic Unit)

ALU అనేది CPU లో ఒక భాగం. ఇక్కడ డేటాపై వివిధ కంప్యూటింగ్ విధులు నిర్వహించబడతాయి. ALU అదనంగా కూడిక, వ్యవకలనం, గుణకారం, విభజన మరియు తార్కిక కార్యకలాపాలు వంటి అంకగణిత కార్యకలాపాలను నిర్వహిస్తుంది. ఒక ఆపరేషన్ యొక్క ఫలితం CPU యొక్క అంతర్గత మెమోరీలో నిల్వచేయబడుతుంది. ALU యొక్క తార్కిక కార్యకలాపాలు కంప్యూటర్ యొక్క నిర్ణయాత్మక సామర్థ్యాన్ని ప్రోత్సాహిస్తుంది.

1.5.2.2 నియంత్రణ యూనిట్ (Control Unit)

CPU మెమోరీ మరియు I/O పరికరాల మధ్య డేటా ప్రవాహాన్ని కంట్రోల్ యూనిట్ నియంత్రిస్తుంది. ఇది కంప్యూటర్ యొక్క మొత్తం ఆపరేషన్‌ను కూడా నియంత్రిస్తుంది.

1.5.3 అవుట్‌పుట్ యూనిట్ (Output Unit)

అవుట్‌పుట్ యూనిట్ ఏదైనా హార్డ్‌వేర్ వినియోగదారులకు అర్థమయ్యే రూపంలో సమాచారాన్ని అంధించే భాగం. ఉదాహరణకు మానిటర్, ప్రింటర్ మొదలగునవి.

1.5.4 మెమోరీ యూనిట్ (Memory Unit)

మెమరీ యూనిట్ రెండు రకాలుగా ఉంటుంది. అవి ప్రాథమిక మెమరీ మరియు సెకండరీ మెమరీ ప్రాథమిక మెమరీని ఉపయోగించి సూచనలు అమలు చేయడానికి సిద్ధంగా ఉన్న ప్రోగ్రామ్‌లు మరియు డేటాను తాత్కాలికంగా నిల్వచేయడానికి ఉపయోగపడుతుంది. సెకండరీ మెమరీను ఉపయోగించి డేటాను శాశ్వతంగా నిల్వచేయబడుతుంది.

ప్రాథమిక మెమరీ అస్థిరంగా ఉంటుంది. అనగా విద్యుత్ సరఫరా స్విచ్ ఆఫ్ అయినప్పుడు content ను కోల్పోతుంది. రాండమ్ యాక్సెస్ మెమరీ (RAM) ప్రధాన మెమరీకి ఉదాహరణ. సెకండరీ మెమరీ అస్థిరంగా ఉండదు. అంటే విద్యుత్ సరఫరా స్విచ్ ఆఫ్ చేయబడిన తర్వాత కూడా content అందుబాటులో ఉంటుంది. హార్డ్‌డిస్క్, CD-ROM మరియు DVD ROM సెకండరీ మెమరీకి ఉదాహరణలు.

1.5.5 ఇన్‌పుట్ మరియు అవుట్‌పుట్ పరికరాలు (Input and Output Devices)

ఇన్‌పుట్ పరికరాలు (Input Devices):

(1) కీబోర్డ్ (Keyboard): కీబోర్డ్ (వైర్లు / వైర్‌లెస్ / వర్చువల్) అనేది నేడు ఉపయోగించే అత్యంత సాధారణ ఇన్‌పుట్ పరికరం. అక్షరాలు, సంఖ్యలు మరియు ప్రత్యేక అక్షరాల కోసం వ్యక్తిగత కీలను సమిష్టిగా అక్షర కీలు అంటారు. ఈ కీబోర్డ్ లేఅవుట్ అసలైన టైప్ రైటర్ యొక్క కీబోర్డ్ నుండి తీసుకోబడింది. కీబోర్డ్‌పై టైపు చేయడం ద్వారా డేటా మరియు సూచనలు కంప్యూటర్‌కు ఇన్‌పుట్‌గా ఇవ్వబడతాయి. ఆల్ఫాబెట్ మరియు న్యూమరిక్ కీలు కాకుండా, విభిన్న విధులను నిర్వహించడానికి ఇది ఫంక్షన్ కీలను కూడా కలిగి ఉంది. కీబోర్డ్‌లో క్యారెక్టర్ కీలు మాడిఫైయర్ కీలు, సిస్టమ్ మరియు GUI కీలు ఎంటర్ మరియు ఎడిటింగ్ కీలు, ఫంక్షన్ కీలు, నావిగేషన్ కీలు, న్యూమరిక్ కీప్యాడ్ మరియు లాక్ కీలు వంటి విభిన్న కీలు అందుబాటులో ఉన్నాయి.



పటము 1.4 కీబోర్డ్

(2) మౌస్ (Mouse): మౌస్ (వైర్డ్ / వైర్‌లెస్) అనేది డిస్ప్లే స్క్రీన్‌పై కర్సర్ యొక్క కదలికలను

11th Computer Science_Chapter-1 Telugu.indd 5 10-03-2023 13:01:50

నియంత్రించడానికి ఉపయోగించే పాయింటింగ్ పరికరం. ఇది చిహ్నాలు, మెనూలు, కమాండ్ బటన్‌లను ఎంచుకోవడానికి లేదా కంప్యూటర్‌లో ఏదైనా activate చేయడానికి ఉపయోగించవచ్చు. కొన్ని మౌస్ చర్యలు తరలింపు, క్లిక్, డబుల్ క్లిక్, కుడి క్లిక్, డ్రాగ్ మరియు డ్రాప్.

వివిధ రకాల మౌస్ అందుబాటులో ఉన్నాయి. మెకానికల్ మౌస్, ఆప్టికల్, లేజర్ మౌస్, ఎయిర్ మౌస్, 3D మౌస్ స్పర్శ మౌస్, ఎర్గోనామిక్ మౌస్ మరియు గేమింగ్ మౌస్

మౌస్‌ని ఎవరు కనుగొన్నారు?



ఈ కంప్యూటర్ మౌస్‌ని బిల్ ఇంగ్లీష్ సహాయముతో డగ్లస్ ఏంజెల్ బార్ట్ అనునతను 1960 లో కనుగొనడం జరిగింది. 1970, నవంబరు 17 న పేటెంట్ హక్కులు ఇవ్వబడినవి.

C8OW1

(3) స్కానర్ (Scanner): స్కానర్లు కంప్యూటర్ మెమరీలోకి నేరుగా సమాచారాన్ని నమోదుచేయడానికి ఉపయోగించబడుతాయి. ఈ పరికరం జిరాక్స్ మెషిన్‌లా పనిచేస్తుంది. స్కానర్ ఏదైనా సమాచారాన్ని లేదా ఫోటోగ్రాఫ్‌లతో సహా డిజిటల్ పార్మాట్‌లోకి మారుస్తుంది. దీనిని కంప్యూటర్ ద్వారా మార్చవచ్చు.



పటము 1.5 స్కానర్

(4) వేలిముద్ర స్కానర్ (Fingerprint Scanner): ఫింగర్ ప్రింట్ స్కానర్ అనేది కంప్యూటర్ భద్రత కోసం ఉపయోగించే వేలిముద్ర గుర్తింపు పరికరం, బయోమెట్రిక్ సాంకేతికతను ఉపయోగించే వేలిముద్ర గుర్తింపు ఫీచర్‌తో అమర్చబడి ఉంటుంది. ఫింగర్ ప్రింట్ రీడర్ / స్కానర్ అనేది పాస్‌వర్డ్‌లను ఉపయోగించకుండా భద్రత కోసం చాలా సురక్షితమైన మరియు అనుకూలమైన పరికరం. ఇది మోసానికి గురవుతుంది మరియు కష్టంగా ఉంటుందని గుర్తుంచుకోండి.



పటము 1.6 వేలిముద్ర స్కానర్

(5) ట్రాక్ బాల్ (Track Ball): ట్రాక్ బాల్ మౌస్ యొక్క తలకిందులుగా ఉండే డిజైన్‌ను పోలి ఉంటుంది. వినియోగదారుడు బంతిని నేరుగా తరలిస్తారు అయితే పరికరం స్థిరంగా ఉంటుంది. వినియోగదారు స్క్రీన్ కదలికలను చేయడానికి బాల్‌ను వివిధ దశల్లో త్రిప్పుతారు.



పటము 1.7 ట్రాక్‌బాల్ (Track Ball)

(6) రెటీనా స్కానర్ (Retinal Scanner): ఇది ఒక వ్యక్తి యొక్క రెటీనా స్కానర్‌లపై ప్రత్యేకమైన నమూనాలను ఉపయోగించే బయోమెట్రిక్ టెక్నిక్ అయిన రెటినా స్కాన్‌ను నిర్వహిస్తుంది.



పటము 1.8 రెటీనా స్కానర్

(7) లైట్ పెన్ (Light Pen): లైట్ పెన్ ఒక పాయింటింగ్ పరికరం, పెన్ ఆకారంలో మరియు మానిటర్‌కి కనెక్ట్ చేయబడింది. లైట్‌పెన్ యొక్క కొన లైట్ సెన్సిటివ్‌గా ఉంటుంది. ఇది స్క్రీన్ నుండి వచ్చే కాంతిని గుర్తిస్తుంది. ఇది కంప్యూటర్‌ను స్క్రీన్‌పై ఉన్న స్థానాన్ని గుర్తించేలా చేస్తుంది. లైట్‌పెన్‌లు స్క్రీన్ నేరుగా డ్రాయింగ్ యొక్క ప్రయోజనాన్ని కలిగి



పటము 1.9 లైట్‌పెన్

11th Computer Science_Chapter-1 Telugu.indd 6 10-03-2023 13:01:50

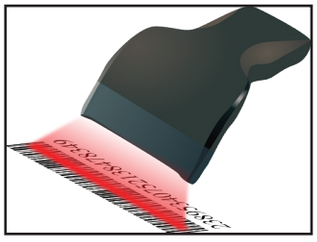
ఉంటుంది. కానీ దీనిని ఉపయోగించడం కష్టమవుతుంది మరియు ఖచ్చితత్వం కలిగి ఉండదు.

(8) ఆప్టికల్ క్యారెక్టర్ రీడర్ (Optical Character Reader): ఇది ఒక పరికరం OCR తో లేదా కాగితంపై వ్రాసిన అక్షరాలు వినియోగదారు పుస్తకం నుండి పేజీని స్కాన్ చేయవచ్చు. అక్షరాలు మరియు విరామ చిహ్నాలుగా కంప్యూటర్ గుర్తించి స్టోర్ చేయును. స్కాన్ చేసిన పత్రాన్ని వర్డ్ ప్రాసెసర్ ఉపయోగించి సవరించవచ్చు.



పటము 1.10 ఆప్టికల్ క్యారెక్టర్ రీడర్

(9) బార్ కోడ్ / QR కోడ్ రీడర్ (Bar Code / QR Code Reader): బార్ కోడ్ అనేది వివిధ మందం కలిగిన పంక్తులలో ముద్రించబడిన నమూనా. బార్ కోడ్ రీడర్ తదుపరి ప్రాసెసింగ్ కోసం కంప్యూటర్‌కు ప్రసారం చేసే బార్‌కోడ్‌పై సమాచారాన్ని స్కాన్ చేస్తుంది. సిస్టమ్ వేగంగా మరియు లోపం లేకుండా కంప్యూటర్‌లోకి ప్రవేశాన్ని రీడర్‌కి ఇస్తుంది.



పటము 1.11 బార్ కోడ్ రీడర్

QR (త్వరిత ప్రతిస్పందన) కోడ్ : QR కోడ్ అనేది ద్వి డైమెన్షన్ బార్ కోడ్, దీనిని కెమెరా ద్వారా చదవవచ్చు మరియు ఇమేజ్‌ని ఇంటర్‌పర్డ్ చేయడానికి ప్రాసెస్ చేయవచ్చు.

(10) వాయిస్ ఇన్‌పుట్ సిస్టమ్‌లు (Voice Input Systems): మైక్రోఫోన్ వాయిస్ ఇన్‌పుట్ పరికరంగా పనిచేస్తుంది. ఇది వాయిస్ డేటాను క్యాప్చర్ చేసి కంప్యూటర్‌కు పంపుతుంది. ప్రసంగంతో పాటు మైక్రోఫోన్‌ను ఉపయోగించడం పూర్తిగా అందించవచ్చు.



పటము 1.12 వాయిస్ ఇన్‌పుట్ సిస్టమ్

(11) డిజిటల్ కెమరా (Digital Camera): చిత్రాలు మరియు వీడియోలను నేరుగా డిజిటల్ రూపంలో క్యాప్చర్ చేస్తుంది. ఇది CCD (చార్జ్ కపుల్డ్ డివైస్) ఎలక్ట్రానిక్ చిప్‌ని ఉపయోగిస్తుంది. కాంతి చిప్‌లపై లెన్స్ ద్వారా పడినపుడు ఇది కాంతి కిరణాలను డిజిటల్ ఆకృతిలోకి మారుస్తుంది.



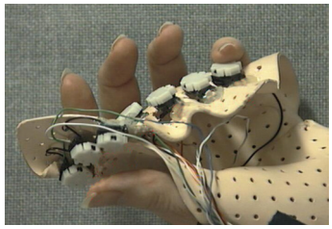
పటము 1.13 డిజిటల్ కెమరా

(12) టచ్ స్క్రీన్ (Touch Screen): Touch screen అనేది వేలిని ఉపయోగించి కంప్యూటర్‌తో ఇంటరాక్ట్ అవ్వడానికి వినియోగదారుని అనుమతించే డిస్‌ప్లే పరికరం. గ్రాఫికల్ యూజర్ ఇంటర్‌ఫేస్ (GUI) ని నావిగేట్ చేయడానికి మౌస్ లేదా కీబోర్డుకు ప్రత్యామ్నాయంగా ఇది చాలా ఉపయోగకరంగా ఉంటుంది. టచ్ స్క్రీన్‌లు కంప్యూటర్లు, ల్యాప్‌టాప్‌ల, మానిటర్లు స్మార్ట్ ఫోన్‌లు, టాబ్లెట్‌లు, క్యాష్ రిజిస్టర్లు పరికరాలతో స్క్రీన్ ఉపయోగించబడుతాయి. కొన్ని టచ్‌స్క్రీన్‌లు టచ్ సెన్సిటివ్ ఇన్‌పుట్‌ని ఉపయోగించకుండా వేలి ఉనికిని పసిగట్టడానికి ఇన్‌ఫ్రారెడ్ కిరణాల గ్రిడ్‌ను ఉపయోగిస్తాయి.



పటము 1.14 టచ్ స్క్రీన్

(13) కీయర్ (Keyer) : కీయర్ అనేది ఒకటి లేదా అంతకంటే ఎక్కువ స్విచ్‌లను నొక్కడం ద్వారా చేతిలో సిగ్నలింగ్ చేసే పరికరం. ఆధునిక కీయర్‌లు పెద్ద సంఖ్యలో స్విచ్‌లను కలిగి ఉంటాయి. కానీ



పటము 1.15 కీయర్

11th Computer Science_Chapter-1 Telugu.indd 7 10-03-2023 13:01:51

సాధారణంగా పూర్తి సైజు కీబోర్డ్ వంటి అనేకం లేదు. సాధారణంగా ఈ సంఖ్య 4 మరియు 50 మధ్య ఉంటుంది. "no board" ఉన్న కీబోర్డ్ నుండి కీయర్ భిన్నంగా ఉంటుంది కానీ కీలు క్లస్టర్‌లో అమర్చబడి ఉంటాయి.

అవుట్‌పుట్ పరికరాలు (Output Devices)

(1) మానిటర్ (Monitor): మానిటర్ అనేది సమాచారాన్ని ప్రదర్శించడానికి సాధారణంగా ఉపయోగించే అవుట్‌పుట్ పరికరం. ఇది టీవి లాగా ఉంది. మానిటర్‌పై చిత్రాలు ఏర్పడతాయి. అనేక చిత్ర అంశాలతో (PIXELS) పిక్సెల్స్ మానిటర్‌లు మైనోక్రోమ్ కావచ్చు. ఇది (text) టెక్స్ట్ పటం 1.16 మానిటర్ లేదా ఇమేజ్‌లను బ్లాక్ అండ్ వైట్‌లో ప్రదర్శిస్తుంది లేదా రంగులో ఉండవచ్చు ఇది ఫలితాలను బహుళ రంగులలో ప్రదర్శిస్తుంది. CRT (కాథోడ్ రే ట్యూబ్) LCD (లిక్విడ్ క్రిస్టల్ డిస్ప్లే) మరియు LED (లైట్ ఎమిటింగ్ డయోడ్స్) వంటి అనేక రకాల మానిటర్లు అందుబాటులో ఉన్నాయి. మానిటర్ VGA (వీడియో గ్రాఫిక్స్ అర్రే) కార్డ్‌తో పని చేస్తుంది. వీడియో గ్రాఫిక్స్ కార్డ్ స్క్రీన్‌తో కమ్యూనికేట్ చేయడానికి కీబోర్డ్‌కు సహాయపడుతుంది. ఇది కంప్యూటర్ మరియు డిస్ప్లే మానిటర్ మధ్య ఇంటర్ ఫేస్‌గా పనిచేస్తుంది. సాధారణంగా ఇటీవల మదర్‌బోర్డులు అంతర్నిర్మిత వీడియో కార్డును కలిగి ఉంటాయి.



పటము 1.16 మానిటర్

మొదటి కంప్యూటర్ మానిటర్ జిరాక్స్ ఆల్టో కంప్యూటర్ సిస్టమ్‌లో భాగం. ఇది మార్చి 1, 1973 న విడుదలైంది.

(2) ప్లాటర్ (Plotter): ప్లాటర్ అనేది పేపర్‌లపై గ్రాఫికల్ అవుట్‌పుట్‌ను ఉత్పత్తి చేయడానికి ఉపయోగించే అవుట్‌పుట్ పరికరం. ఇది చిత్రాలను గీయడానికి సింగిల్ కలర్ లేదా మల్టీ కలర్ పెన్నులను ఉపయోగిస్తుంది.



పటము 1.17 ప్లాటర్

(3) ప్రింటర్లు (Printers): కాగితాలపై సమాచారాన్ని ముద్రించడానికి ప్రింటర్లు ఉపయోగించబడుతాయి. ప్రింటర్లు రెండు ప్రధాన వర్గాలుగా విభజించబడ్డాయి.

- ఇంపాక్ట్ ప్రింటర్లు
- నాన్ ఇంపాక్ట్ ప్రింటర్లు

ఇంపాక్ట్ ప్రింటర్లు (Impact Printers)

ఈ ప్రింటర్‌లు రిబ్బన్‌పై సుత్తులు లేదా పిన్‌లను కొట్టడంతో ముద్రిస్తాయి. ఈ ప్రింటర్లు యాంత్రిక ఒత్తిడి ఉపయోగించి (కార్బన్ ఫిగర్ 1.18 ఇంపాక్ట్ ప్రింటర్ బహుళ భాగాల కాగితాలపై ఉపయోగించి ముద్రించవచ్చు. ఉదాహరణకు డాట్ మ్యాట్రిక్స్ ప్రింటర్లు మరియు లైన్ మ్యాట్రిక్స్ ప్రింటర్లును ఇంపాక్ట్ ప్రింటర్లు అందురు.



పటము 1.18 ఇంపాక్ట్ ప్రింటర్

నిర్ణీత సంఖ్యలో పిన్‌లు లేదా వైర్‌లను ఉపయోగించి ముద్రించే డాట్ మ్యాట్రిక్స్ ప్రింటర్ ప్రతి డాట్ ఒక చిన్న లోహపు కడ్డీ ద్వారా ఉత్పత్తి చేయబడుతుంది. దీనిని వైర్ లేదా పిన్ అని కూడా పిలుస్తారు. ఇది ఒక చిన్న విద్యుదయస్కాంతం లేదా సోలనోయిడ్ యొక్క శక్తితో నేరుగా లేదా చిన్న మీటల సమితి ద్వారా పనిచేస్తుంది. ఇది సాధారణంగా ఒక సమయంలో టెక్స్ట్ యొక్క ఒక లైన్ ప్రింట్ చేస్తుంది. ఈ ప్రింటర్‌ల ప్రింటింగ్ వేగం 30 నుండి 1550 CPS (సెకనుకు అక్షరం) వరకు ఉంటుంది.

11th Computer Science_Chapter-1 Telugu.indd 8 10-03-2023 13:01:51

లైన్ మ్యాట్రిక్స్ ప్రింటర్లు ప్రింటింగ్ కోసం ఫిక్స్‌డ్ ప్రింట్ హెడ్‌ని ఉపయోగిస్తాయి. ప్రాథమికంగా ఇది పేజీ మొత్తం డాట్ వరుసను ముద్రిస్తుంది. కానీ ఇది డాట్ లైన్ ముద్రించడం ద్వారా లైన్ స్టేట్మెంట్‌ను నిర్మిస్తుంది. లైన్ ప్రింటర్లు నిమిషానికి 1000 లైన్‌ల కంటే ఎక్కువ ప్రింటింగ్ చేయగలవు ఫలితంగా గంటకు వేల పేజీలు ఉంటాయి. ఈ ప్రింటర్లు బహుళ భాగాలపై (కార్బన్ పేపర్‌లను ఉపయోగించి) ముద్రించడానికి యాంత్రిక ఒత్తిడిని కూడా ఉపయోగిస్తాయి.

నాన్ ఇంపాక్ట్ ప్రింటర్లు (Non-Impact Printers)

ఈ ప్రింటర్లు ప్రింటింగ్ కోసం స్ట్రైకింగ్ మెకానిజంను ఉపయోగించవు. వారు ఎలెక్ట్రోస్టాటిక్ లేదా లేజర్ టెక్నాలజీని ఉపయోగిస్తారు. ఈ ప్రింటర్లు నాణ్యత మరియు వేగం ఇంపాక్ట్ ప్రింటర్లు కంటే మెరుగ్గా ఉన్నాయి. ఉదాహరణకు లేజర్ ప్రింటర్లు మరియు ఇంక్‌జెట్ ప్రింటర్లు నాన్ ఇంపాక్ట్ ప్రింటర్లు.

లేజర్ ప్రింటర్లు (Laser Printers)

లేజర్ ప్రింటర్లు ఎక్కువగా ఫోటోకాపియర్లు ఉపయోగించే సాంకేతికతతో పనిచేస్తాయి. ఇది ప్రింటర్ లోపల డ్రమ్ అంతటా లేజర్ పుంజం స్కాన్‌ను ముందుకు వెనుకకు స్కాన్ చేస్తుంది. ఒక నమూనాను రూపొందిస్తుంది. ఇది చాలా మంచి నాణ్యత గల గ్రాఫిక్ చిత్రాలను రూపొందించగలదు. లేజర్ ప్రింటర్ యొక్క ముఖ్య లక్షణాలలో ఒకటి వాటి రిజల్యూషన్ అంగుళానికి ఎన్ని డాట్‌లు (DPI) అందుబాటులో ఉన్న రిజల్యూషన్ పరిధి సుమారు 1200 dpi ఇది దాదాపు నిమిషానికి 100 పేజీలు (PPM) ప్రింట్ చేయగలదు.



పటము 1.19 లేజర్ ప్రింటర్

ఇంక్‌జెట్ ప్రింటర్లు (Inkjet Printers):

ఇంక్‌జెట్ ప్రింటర్లు రంగు టోన్‌లను సృష్టించడానికి మెజెంటా పసుపు మరియు సియాన్ ఇంక్‌లను కలిపి రంగు కాట్రిడ్జ్‌లను ఉపయోగిస్తారు. మోనోక్రోమ్ అవుట్‌పుట్ కోసం బ్లాక్ కాట్రిడ్జ్ కూడా ఉపయోగించబడుతుంది. ఇంక్‌జెట్ ప్రింటర్లు కాగితంపై అయానైజ్డ్ ఇంక్‌ను స్ప్రే చేయడం ద్వారా పనిచేస్తాయి. ఇంక్‌జెట్ ప్రింటర్ల వేగం సాధారణంగా 1-20 PPM (నిమిషానికి పేజీ) వరకు ఉంటుంది.



పటము 1.20 ఇంక్‌జెట్ ప్రింటర్

వారు ఇంక్‌ను వేడి చేయడం ద్వారా కాల్చే సాంకేతికతను ఉపయోగిస్తారు. తద్వారా అది బబుల్స్‌గా కాగితంపై పేలుతుంది. ఎలక్ట్రానిక్ సర్క్యూట్‌లచే నియంత్రించబడే చిన్న విద్యుత్ ప్రవాహాలను ఉపయోగించి పైజో ఎలక్ట్రిసిటిని ఉపయోగిస్తుంది. జెట్ వేగంతో ఇంక్‌ని వ్యాప్తి చేయడానికి ఎలక్ట్రానిక్ సర్క్యూట్‌లు ప్రింటర్ లోపల ఉపయోగించబడతాయి. ఇంక్‌జెట్ ప్రింటర్ ప్రతి సెకనుకు పేపర్‌పై మిలియన్ల చుక్కల కొద్దీ ఇంక్‌లను వ్యాపింపజేస్తుంది.

స్పీకర్లు (Speakers:)



పటము 1.21 స్పీకర్లు

స్పీకర్లు వాయిస్ అవుట్‌పుట్ (ఆడియో) ఉత్పత్తి చేస్తాయి. సాఫ్ట్‌వేర్‌తో పాటు స్పీచ్ సింథసిస్ స్పీకరును ఉపయోగించి కంప్యూటర్ వాయిస్ అవుట్‌పుట్‌ను అందించగలదు. విమానయాన సంస్థలు, పాఠశాలలు, బ్యాంకులు, రైల్వే స్టేషన్లు మొదలైన ప్రదేశాలలో ఇది సర్వసాధారణంగా మారింది.

మల్టీ మీడియా ప్రొజెక్టర్లు (Multimedia Projectors):

పెద్ద స్క్రీన్‌పై కంప్యూటర్ అవుట్‌పుట్‌ను ఉత్పత్తి చేయడానికి మల్టీ మీడియా ప్రొజెక్టర్లు

11th Computer Science_Chapter-1 Telugu.indd 9 10-03-2023 13:01:51

ఉపయోగించబడతాయి. సమావేశ మందిరాలలో లేదా తరగతి గదులలో ప్రదర్శనలను ప్రదర్శించడానికి ఇవి ఉపయోగింబడతాయి.



పటము 1.22 మల్టీమీడియా ప్రొజెక్టర్

1.6 బూటింగ్ కంప్యూటర్ (Booting of computer)

ఆపరేటింగ్ సిస్టమ్ (OS) అనేది కంప్యూటర్‌ను పనిచేసేలా చేసే ప్రాథమిక సాఫ్ట్‌వేర్. కంప్యూటర్ స్విచ్ ఆన్‌చేసినపుడు దాని RAMలో సమాచారం ఉండదు. అదే సమయంలో ROM లో POST (పవర్ ఆన్ సెల్ఫ్ టెస్ట్) అని పిలువబడే ముందుగా వ్రాసిన ప్రోగ్రామ్ మొదట అమలు చేయబడుతుంది. ఈ ప్రోగ్రామ్ RAM కీబోర్డ్ మొదలైన పరికరాలు సరిగ్గా కనెక్ట్ చేయబడి ఆపరేట్ చేయడానికి సిద్ధంగా ఉన్నాయో లేదో తనిఖీ చేస్తుంది. ఈ పరికరాలు సిద్ధంగా ఉంటే BIOS (బేసిక్ ఇన్‌పుట్ అవుట్‌పుట్ సిస్టమ్) అమలు చేయబడుతుంది. ఈ ప్రక్రియను బూటింగ్ అంటారు. ఈ తరువాత 'బూట్‌స్ట్రాప్' 'లోడర్' అనే ప్రోగ్రామ్ హార్డ్‌డిస్క్ నుండి మెయిన్ మెమరిలోకి OS ని బదిలీ చేస్తుంది. ఇపుడు OS లోడ్ అవుతుంది. Windows/ Linux, మొదలైనవి అమలు చేయబడుతుంది బూటింగ్ ప్రక్రియ రెండు రకాలు.

1) కోల్డ్ బూటింగ్ (Cold Booting)
2) వెచ్చని బూటింగ్ (Warm Booting)

కోల్డ్ బూటింగ్ (Cold Booting): సిస్టమ్ ప్రారంభ స్థితి నుండి ప్రారంభమైనపుడు అంటే అది స్విచ్ ఆన్ చేయబడినప్పుడు మేము దానిని కోల్డ్ బూటింగ్ లేదా హార్డ్ బూటింగ్ అని పిలుస్తాము. యూసర్ పవర్ బటన్‌ను నొక్కినపుడు బూటింగ్ ప్రక్రియను ప్రారంభించడానికి సూచనలు ROM నుండి చదవబడతాయి.

వార్మ్ బూటింగ్ (Warm Booting): సిస్టమ్ పున:ప్రారంభించబడినపుడు లేదా రీసెట్ బటన్ నొక్కినప్పుడు మేము దానిని వార్మ్ బూటింగ్ లేదా సాఫ్ట్ బూటింగ్ అని పిలుస్తాము. సిస్టమ్ ప్రారంభ స్థితి నుండి ప్రారంభించబడదు మరియు ఈ సందర్భంలో అన్ని నిర్ధారణ పరీక్షలు నిర్వహించాల్సిన అవసరం లేదు. డేటా సరిగ్గా నిల్వ చేయబడినందున డేటా నష్టం మరియు సిస్టమ్ దెబ్బతినే అవకాశాలు ఉన్నాయి.

గుర్తుంచుకోవలసిన అంశాలు

- చుట్టూ ఎక్కడ చూసినా కంప్యూటర్లే కనిపిస్తున్నాయి.
- ఇది సూచనల సేట్ ప్రకారం ఇన్‌పుట్‌ను ప్రాసెస్ చేసే ఎలక్ట్రానిక్ పరికరం. దానిని అందించబడింది. మరియు కావలసిన అవుట్‌పుట్‌ను చాలా వేగంగా ఇస్తుంది. అభివృద్ధి యొక్క వివిధ దశల ఆధారంగా కంప్యూటర్లను ఆరు వేర్వేరు తరాలుగా విభజించవచ్చు.
- కంప్యూటర్ అనేది హార్డ్‌వేర్ మరియు సాఫ్ట్‌వేర్ కలయిక.
- హార్డ్‌వేర్ అనేది కంప్యూటర్ యొక్క భౌతిక భాగం.
- ఇన్‌పుట్ యూనిట్ అనేది కంప్యూటర్‌కు ఏ విధమైన డేటాను అందించడానికి ఉపయోగించబడుతుంది.
- CPU సాఫ్ట్‌వేర్‌ను అన్వయిస్తుంది మరియు అమలు చేస్తుంది. సూచనలు: ALU అనేది వివిధ కంప్యూటింగ్ ఫంక్షన్‌లు ఉన్న CPU లో ఒక డేటాపై ప్రదర్శించబడుతుంది.
- CPU మెమరీ మరియు I/O పరికరాల మధ్య డేటా ప్రవాహాన్ని కంట్రోల్ యూనిట్ నియంత్రిస్తుంది.
- అవుట్‌పుట్ యూనిట్ అనేది ఏదైనా హార్డ్‌వేర్ భాగం ఇది వినియోగదారు అర్థమయ్యే రూపంలో ఒకరు లేదా అంతకంటే ఎక్కువ మంది వ్యక్తులకు సమాచారాన్ని అందజేస్తుంది. మెమరీ యూనిట్ రెండు రకాలుగా ఉంటుంది. అవి ప్రైమరీ మెమరీ మరియు సెకండరీ మెమరీ.

11th Computer Science_Chapter-1 Telugu.indd 10 10-03-2023 13:01:51

కృత్యము

విద్యార్థి కృత్యము

1. కంప్యూటర్ల వర్గీకరణను వివరించండి.
2. 1993, 1995, 2005, 2008, 2016 సంవత్సరాల్లో ఉపయోగించిన మదర్‌బోర్డ్ పేర్లు RAM సామర్థ్యం వివరాలను ఇవ్వండి.
3. ఈ అధ్యాయంలో ఇవ్వని ఇన్‌పుట్ మరియు అవుట్‌పుట్ (input and output) పరికరాలు ఏవి?

మూల్యాంకనం

విభాగం – అ

సరైన సమాధానాన్ని ఎన్నుకొనుము

1. మొదటి తరం కంప్యూటరులో ఉపయోగించినది
   (అ) వాక్యూమ్ ట్యూబ్‌లు
   (ఆ) ట్రాన్సిస్టర్లు
   (ఇ) ఇంటిగ్రేటెడ్ సర్క్యూట్‌లు
   (ఈ) మైక్రోప్రాసెసర్లు
2. అస్థిర జ్ఞాపక శక్తికి పేరు
   (అ) ROM (ఆ) PROM
   (ఇ) RAM (ఈ) EPROM
3. క్రింది వానిలో అవుట్‌పుట్ పరికరాన్ని గుర్తించండి.
   (అ) Keyboard (ఆ) Memory
   (ఇ) Monitor (ఈ) Mouse
4. ఇన్‌పుట్ పరికరాన్ని గుర్తించండి.
   (అ) Printer (ఆ) Mouse
   (ఇ) Plotter (ఈ) Projector
5. ..... కోసం అవుట్‌పుట్ పరికరాన్ని ఉపయోగించబడుతాయి.
   (అ) Thermal printer
   (ఆ) Plotter
   (ఇ) Dot matrix
   (ఈ) inkjet printer
6. ATM మిషీన్‌లో క్రింది వాటిలో ఏది ఉపయోగించబడుతుంది?
   (అ) Touch Screen (ఆ) speaker
   (ఇ) Monitor (ఈ) Printer
7. సిస్టమ్ రీస్టార్ట్ చేసినప్పుడు ఏ రకం బూటింగ్ ఉపయోగించబడుతుంది?
   (అ) Warm booting
   (ఆ) Cold booting
   (ఇ) Touch boot
   (ఈ) Real boot
8. POST ని విస్తరించండి.
   (అ) Post on self Test
   (ఆ) Power on Software Test
   (ఇ) Power on Self Test
   (ఈ) Power on Self Text
9. క్రింది వానిలో ఏది main memory?
   (అ) ROM (ఆ) RAM
   (ఇ) Flash drive (ఈ) Hard disk
10. IC లలో ఏ తరం కంప్యూటర్లు ఉపయోగించబడును?
    (అ) మొదటి (ఆ) రెండవ
    (ఇ) మూడవ (ఈ) నాల్గవ

విభాగం – ఆ

లఘు సమాధానములిమ్ము

1. కంప్యూటర్ అంటే ఏమిటి?
2. డేటా మరియు సమాచారాల మధ్య గల తేడాలను వివరించుము.
3. CPU యొక్క భాగాలు వివరించుము?
4. ALU యొక్క విధులు ఏమి?
5. నియంత్రణ యూనిట్ యొక్క విధులు వ్రాయండి.
6. మెమొరి యొక్క విధులు వ్రాయండి.
7. ఇన్‌పుట్ మరియు అవుట్‌పుట్ యూనిట్‌ల మధ్య తేడా వ్రాయండి.
8. ప్రైమరీ మరియు సెకండరీ మెమొరీకి మధ్య తేడా వ్రాయండి.

11th Computer Science_Chapter-1 Telugu.indd 11 10-03-2023 13:01:52

విభాగం - ఇ

సంక్షిప్త సమాధానము వ్రాయుము

1. కంప్యూటర్ యొక్క లక్షణాలు ఏమిటి?
2. కంప్యూటర్ అప్లికేషన్లను వ్రాయండి.
3. ఇన్‌పుట్ పరికరం అంటే ఏమిటి? రెండు ఉదాహరణలు ఇవ్వండి.
4. ఏదైనా మూడు అవుట్‌పుట్ పరికరాల పేర్లు వ్రాయండి.
5. ఆప్టికల్ మరియు లేజర్ మౌస్‌ల మధ్య భేదం తెలుపుము.
6. ఇంపాక్టు ప్రింటర్‌పై లఘు వ్యాసం వ్రాయండి.
7. ఆరవ తరం లక్షణాలను వ్రాయండి.
8. మానిటర్ యొక్క ముఖ్యమైన లక్షణాలను వ్రాయండి.

విభాగం - ఈ

సవివరంగా వివరించండి

1. కంప్యూటర్ యొక్క ప్రాథమిక భాగాలు చక్కని రేఖాచిత్రంతో వివరించండి.
2. వివిధ తరాల కంప్యూటర్ల గురించి చర్చించండి.
3. క్రింది వాటిని వివరించండి.

   అ. Inkjet Printer ఆ. Multimedia projector ఇ. Bar code / QR code Reader

సూచన పుస్తకాలు (References)

(1) Fundamentals of Computers – V. Rajaraman – PHI Publications

(2) Computer Science text book – NCERT, New Delhi

అంతర్జాల వనరులు

(1) www.wikipedia.org

(2) https://www.computerhope.com/jargon/c/computer.htm

కేస్ స్టడీ

Prepare a comparative study of various computers of past and present with respect to speed, memory, size, power consumption and other features

11th Computer Science_Chapter-1 Telugu.indd 12 10-03-2023 13:01:52

పదకోశము

Computer	It is an electronic device that processes the input according to the set of instructions provided to it and gives the desired output at a very fast rate.
Vacuum tube	Vacuum tubes contain electrodes for controlling electron flow and were used in early computers as a switch or an amplifier.
Transistors	The transistor ("transfer resistance") is made up of semi-conductors. It is a component used to control the amount of current or voltage used for amplification/modulation of an electronic signal.
Punched cards	Punch cards also known as Hollerith cards are paper cards containing several punched or perforated holes that were punched by hand or machine to represent data.
Machine Language	Machine language is a collection of binary digits or bits that the computer reads and interprets.
Assembly language	An assembly language is a low-level programming language.
Integrated Circuits	The IC is a package containing many circuits, pathways, transistors, and other electronic components all working together to perform a particular function or a series of functions.
Microcomputer	Micro computer is used to describe a standard personal computer.
High-level languages	A high-level language is a computer programming language that isn't limited by the computer, designed for a specific job, and is easier to understand.
Natural Language Processing (NLP)	Natural Language Processing is a method used in artificial intelligence to process and derive meaning from the human language.
Robotics	Robot is a term coined by Karel Capek in the 1921 to play RUR (Rossum's Universal Robots). It is used to describe a computerized machine designed to respond to input received manually or from its surroundings.
Nanotechnology	Nanotechnology is an engineering, science, and technology that develops machines or works with one atom or one molecule that is 100 nanometers or smaller.
Bioengineering	A discipline that applies engineering principles of design and analysis to biological systems and biomedical technologies

11th Computer Science_Chapter-1 Telugu.indd 13 10-03-2023 13:01:52

అధ్యాయము 2

సంఖ్యా వ్యవస్థలు (NUMBER SYSTEMS)

అభ్యసన లక్ష్యాలు

- కంప్యూటర్, డేటాను ఎలా అర్థం చేసుకొంటుంది మరియు మెమరీలో ఎలా నిల్వచేస్తుంది అని తెలుసుకుందాం.
- వివిధ డేటా ప్రాతినిథ్యాలు మరియు బైనరీ అంకగణితము గురించి తెలుసుకొనుట.
- వివిధ రకాల సంఖ్యా వ్యవస్థలను ఒకటి నుండి మరొక దానికి మార్చుటను గురించి తెలుసుకొనుట.

2.1 పరిచయం (Introduction)

డేటా (దత్తాంశము) అను పదము డేటమ్ అను పదము నుండి గ్రహించబడినది. వాస్తవ విషయం అనునది దీని అర్థం. ఈ డేటా ప్రజల గురించి మరియు కొన్ని వస్తువుల గురించి ఉండును.

ఉదాహరణ:

పేరు, వయస్సు, తరగతి, మార్కులు మరియు సబ్జెక్టు కొన్ని డిఫైన్ వేరియబుల్స్‌గా ఉండనివ్వండి. ఇప్పుడు ఈ వేరియబుల్స్‌లో ప్రతిదానికి ఒక విలువను కేటాయించండి.

Name	=	Rajesh
Age	=	16
Class	=	XI
Mark	=	65
Subject	=	Computer Science

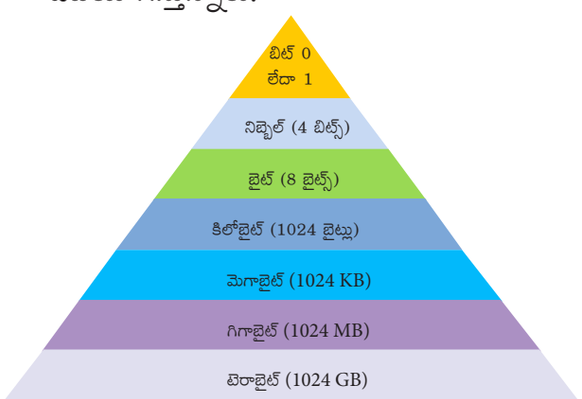
పటము 2.1 డేటాకు ఉదాహరణ

పైన ఇవ్వబడిన ఉదాహరణలో ఐదు చరరాశులలో ఇవ్వబడిన మార్కులు దత్తాంశము (Data) అంటారు. ఈ దత్తాంశము క్రియాశీలమైనప్పుడు రాజేష్ వయస్సు 16, పన్నెండవ తరగతి చదువు చున్నాడని అతను కంప్యూటర్ సైన్స్ పాఠంలో 65 మార్కులు పొందియున్నాడు అనే సమాచారం మనకు అందుతుంది.

2.2 డేటా ప్రాతినిథ్యాలు (Data Representations)

కంప్యూటర్ డేటాను 0 (పూజ్యము) మరియు '1' ఒకటి రూపములో నిర్వహించును. సంఖ్య అక్షరము మరియు ప్రత్యేక చిహ్నము ఏ రకంగా ఉండినను వాటిని కంప్యూటర్ అర్థం చేసుకోగలిగే '0' పూజ్యము మరియు '1' ఒకటి అను సంఖ్యా రూపములో మార్చబడాలి. కంప్యూటర్లకు అవగాహన పొందే '0' మరియు '1' లను యంత్ర భాష (**Machine language**) అని పిలువబడుచున్నది. 0 మరియు 1 లను ఆంగ్లములో ద్వి సంఖ్య అంకెలు (**Bi**nary Digit**s**)(BIT) అని పిలువబడుచున్నది. అందువలన డేటా ప్రాతినిథ్యం యొక్క అధ్యయనానికి కంప్యూటర్ ముఖ్యం.

- ఒక **బిట్, బైనరీ** అంకె యొక్క చిన్న రూపం ఇది '0' లేదా '1' కావచ్చు. ఇది ప్రాథమికమైనది కంప్యూటర్ డేటా యూనిట్.
- ఒక **నిబ్బెల్** (**nibble**) అనేది 4 బిట్ల అంకెల సమాహారం (బైనరీ అంకెలు)
- 8 బిట్ల సేకరణను బైట్ అంటారు. ఒక బైట్ అనునది కంప్యూటర్‌ను కొలిచే మెమరీ పరిమాణము యొక్క ప్రాథమిక యూనిట్.
- **పదము పొడవు** (Word length) అనునది కంప్యూటర్ (CPU) చే ప్రాసస్ చేయబడిన బిట్ సంఖ్యను సూచించును. ఉదాహరణకు ఒక పదం పొడవు 8 బిట్స్, 16 బిట్స్,32 బిట్స్ మరియు 64 బిట్స్ కలిగియుండును. ప్రస్తుత కంప్యూటర్లలో 32 బిట్స్ లేదా 64 బిట్స్ ఉపయోగిస్తున్నారు.



చిత్రం 2.2 డేటా ప్రాతినిథ్యం

11th Computer Science_Chapter-2 Telugu.indd 14 30-03-2023 18:02:08

కంప్యూటర్ మెమరీ – ప్రైమరీ మెమరీ మరియు సెకండరీ మెమరీ ఇది సాధారణంగా కిలోబైట్ (KB) లేదా మెగాబైట్ (MB) స్టోరేజ్ అనునది ప్రాతినిధ్యం వహిస్తుంది. దశాంశ వ్యవస్థలో 1 కిలో 1000 ని సూచిస్తుంది. అనగా 10^3 బైనరీ సిస్టము 1 కిలో బైట్ 1024 బైట్‌ల 2^{10} సూచించును. క్రింది పట్టికలో వివిధ మెమరీ పరిమాణములు సూచించును.

పట్టిక 2.1 మెమరీ పరిమాణము (2^10 ను 2 ఘాతం 10 అని చదవాలి)

పేరు	సూక్ష్మ రూపము	పరిమాణము
కిలో	K	2^10 = 1,024
మెగా	M	2^20 = 1,048,576
గిగా	G	2^30 = 1,073,741,824
టెరా	T	2^40 = 1,099,511,627,776
పెటా	P	2^50 = 1,125,899,906,842,624
ఎక్సా	E	2^60 = 1,152,921,504,606,846,976
జిట్టా	Z	2^70 = 1,180,591,620,717,411,303,424
యొట్టా	Y	2^80 = 1,208,925,819,614,629,174,706,173

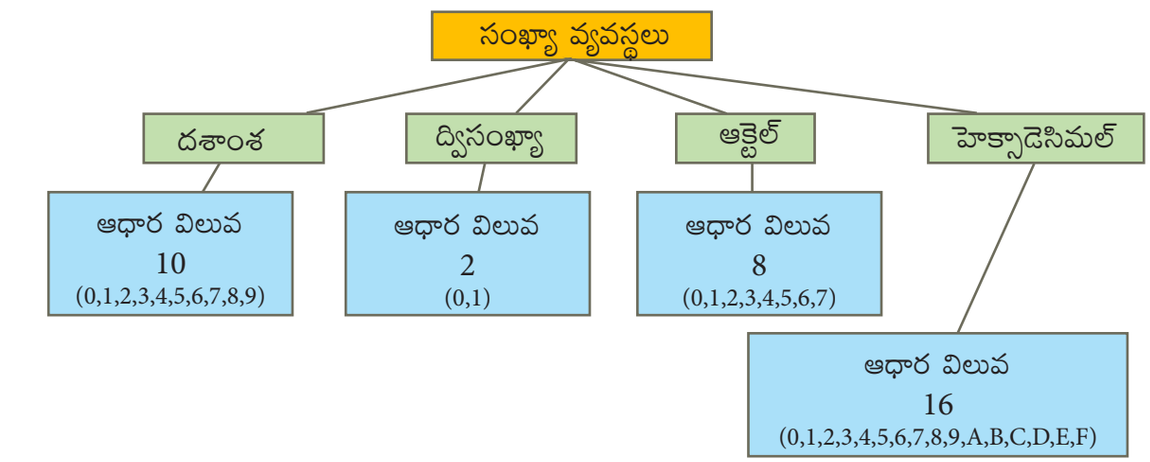
9GLYA

పై పట్టికలో ఇవ్వబడిన వివరములు బైట్లుగా ఉపయోగించబడుతాయి. అక్షర సమితి మరియు సంఖ్యలను సూచించుటకు వివిధ రకముల కోడింగ్ పథకములు ఉపయోగించబడుతాయి. ఇది **అమెరికన్ స్టాండర్డ్ కోడ్ ఫర్ ఇన్‌ఫర్‌మేషన్ ఇంటర్‌చేంజ్** (ASCII). ఇది అత్యంత సాధారణంగా ఉపయోగించే కోడింగ్ పథకము 0 మరియు 127 మధ్య ఉన్న ప్రతి బైనరీ విలువ ఒక నిర్దిష్ట అక్షరము సూచించడానికి ఉపయోగపడును. ఖాళీ స్థలము కోసం ASCII విలువ 32 మరియు సంఖ్య 0 యొక్క ASCII 48. లోయర్ కేస్ (lower case) ఆల్ఫాబీట్‌ల కోసం ASCII పరిధి 97 నుండి 122 వరకు ఉంటుంది. మరియు (upper case) ఆల్ఫాబీట్‌ల కోసం ASCII విలువల పరిధి 65 నుండి 90 వరకు ఉండును.

మీకు తెలుసా?

కంప్యూటర్ యొక్క వేగము అది ఒకసారి ప్రాసెస్ చేయగల బిట్‌ల సంఖ్యపై ఆధారపడి ఉండును. ఉదాహరణకు 64 బిట్ కంప్యూటర్ ఒక ఆపరేషన్‌లో 64 బిట్ సంఖ్యలను ప్రాసెస్ చేయును. అయిన 32 బిట్ కంప్యూటర్ 64 బిట్‌ల సంఖ్యలను చిన్న ముక్కలుగా విడదీసి నెమ్మదిగా చేస్తుంది.

2.3 వివిధ రకముల సంఖ్యా వ్యవస్థలు (Different Types of Number Systems)



పటము 2.3 సంఖ్యా వ్యవస్థలు

11th Computer Science_Chapter-2 Telugu.indd 15 30-03-2023 18:02:08

సంఖ్యా వ్యవస్థ అనునది సంఖ్యలను సూచించే మార్గము. సాధారణంగా నిత్యజీవితములో దశాంశ వ్యవస్థను ఉపయోగిస్తారు. ఇతర సంఖ్యా వ్యవస్థలైన ద్వి సంఖ్యా, ఆక్టెల్, హెక్సాడెసిమల్ కూడా కలవు. ప్రతి సంఖ్యా వ్యవస్థ దాని **ఆధార విలువ** లేదా **రాడిక్స్** (Radix) ద్వారా ప్రత్యేకంగా గుర్తించబడుతుంది. లేదా Base అనేది ప్రతి Number System లోని అంకెల సంఖ్య అగును. **రాడిక్స్** లేక **ఆధార విలువ** సాధారణ సంఖ్యా వ్యవస్థ యొక్క మూలముగా ఉండును.

2.3.1 దశాంశ సంఖ్యా వ్యవస్థ (Decimal Number System)

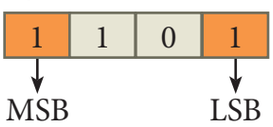
ఇందులో 0,1,2,3,4,5,6,7,8,9 (10 స్థానములు) ఉండును. మన నిత్యజీవితంలో ఉపయోగించే పురాతన మరియు అత్యంత ప్రజాదరణ పొందిన దాని స్థానానికి సంబంధించి Relative గా ఉంటుంది. దీని అర్థం సంఖ్యలోని ప్రతి అంకెను 10 తో గుణిస్తే ఆ సంఖ్య స్థానానికి సంబంధించిన ఘాతం పెంచబడుతుంది.

ఉదాహరణ

$$(123)_{10} = 1x10^2 + 2x10^1 + 3x10^0$$
$$= 100 + 20 + 3$$
$$= (123)_{10}$$

2.3.2 ద్వి సంఖ్యా వ్యవస్థ (Binary Number System)

ద్వి సంఖ్యా వ్యవస్థలో రెండు అంకెలు మాత్రమే ఉండును. అవి 0 మరియు 1. ద్వి సంఖ్యా వ్యవస్థలో సంఖ్యల స్థానము 2 మరియు స్థానము విలువ 2 యొక్క గుణకములుగా ఉండును. (2'S Power Positional weight method) ఇది ఎడమ భాగములో బిట్ అనునది గరిష్ట విలువను కలిగి యుండును. (**M**ost **S**ignificant **B**it - MSB) వలన గరిష్ట విలువ కలిగిన బిట్ అందురు. ఇది అతి పెద్ద స్థాన విలువను కలిగి ఉండును. ఎడమ చేతి వైపుగల బిట్ కనీస విలువ గుర్తించిన బిట్ (**L**east **S**ignificant **B**it - LSB) మరియు కనీస విలువను కలిగియుండును.



ఉదాహరణ

ఉదాహరణ : $(1101)_2$ అను ద్వి సంఖ్యా శ్రేణికి సరిపోవు దశాంశ స్థానవిలువ ఎంత?

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= 8 + 4 + 0 + 1$$
$$= (13)_{10}$$

2.3.3 అష్టాంశ సంఖ్యా వ్యవస్థ (Octal Number System)

అష్టాంశ సంఖ్యా వ్యవస్థలో స్థాన సంఖ్యలు 0,1,2,3,4,5,6 మరియు 7 (8 స్థానములు) ఉండును. ప్రతి అష్టాంశ సంఖ్యా స్థానము విలువ ఆధారముగ ఉండును.

ఉదాహరణ

$(547)_8$ అనే అష్టాంశ సంఖ్యకు సరియైన దశాంశ సంఖ్య

$$(547)_8 = 5\times8^2 + 4\times8^1 + 7\times8^0$$
$$= 5\times64 + 4\times8 + 7\times1$$
$$= 320 + 32 + 7$$
$$= (359)_{10}$$

2.3.4 ద్వి అష్టాంశ సంఖ్యా వ్యవస్థ (Hexadecimal Number System)

ద్వి అష్టాంశ సంఖ్య యొక్క ఆధారము 16 గా ఉండును. ద్వి అష్టాంశ సంఖ్య ప్రతినిధ్యము వహించు లేదా ద్వి అష్టాంశ దశాంశము ద్విసంఖ్యామాన సూక్ష్మ రూపములో ఉండును. ఈ వ్యవస్థ విషయానుశ్రద్ధగా పరిగణలోకి తీసుకొనును. ఇందులో 16 చిహ్నాలు ఉపయోగించబడును. అవి 0 నుండి F వరకు గల సంఖ్య స్థానములు ద్విఅష్టాంశములుగా ఉండును. మొదటి 10 స్థానములు దశాంశ స్థానము వలె ఉండును. 0 నుండి 9 మరియు మిగిలిన 6 చిహ్నాలు ఆంగ్ల అక్షర మాలలోని మొదటి 6 అక్షరములను శ్రేణిగా A నుండి F వరకు తీసుకొంటారు. ఇక్కడి A గాను 10 ను B ను 11 ను C ను 12 D ను 13 E ను 14 ను మరియు F ను 15 ను గా తీసుకోందురు.

11th Computer Science_Chapter-2 Telugu.indd 16 30-03-2023 18:02:08

పట్టిక 2.2 ద్వి సంఖ్యా అష్టాంశ ద్వి అష్టాంశ (హెక్సా) సరియగు దశాంశ సంఖ్యలు

దశాంశము	ద్వి సంఖ్య	అష్టాంశ	ద్వి అష్టాంశము
0	0000	000	0000
1	0001	001	0001
2	0010	002	0002
3	0011	003	0003
4	0100	004	0004
5	0101	005	0005
6	0110	006	0006
7	0111	007	0007
8	1000	010	0008
9	1001	011	0009
10	1010	012	A
11	1011	013	B
12	1100	014	C
13	1101	015	D
14	1110	016	E
15	1111	017	F

ఉదాహరణ

ద్వి అష్టాంశ శ్రేణి $(25)_{16}$ నకు సరిపోవు దశాంశము

$$(25)_{16} = 2\times16^1 + 5\times16^0$$
$$= 32+5$$
$$= (37)_{10}$$

శిక్షణ శిబిరం (Workshop)

1. క్రింది ఇవ్వబడిన సంఖ్యలు ఏ సంఖ్యా వ్యవస్థకు చెందినవి.

వ.సంఖ్య	సంఖ్య	సంఖ్యా వ్యవస్థ
1	$(1010)_{10}$	దశాంశ సంఖ్యా వ్యవస్థ
2	$(1010)_{2}$	
3	$(989)_{16}$	
4	$(750)_{8}$	
5	$(926)_{10}$	

2. క్రింద ఇవ్వబడిన సంఖ్యలు సరి లేక తప్పా అని సరిచూడుము. తప్పు అయిన కారణమును తెలుపుము

వ.సం	ప్రవచనాలు	అవును / కాదు	కారణము (సరియైనది)
1.	786 అనునది ద్వి అష్టాంశ సంఖ్య		
2.	101 అనునది ద్వి సంఖ్య		
3.	ద్వి అష్టాంశ 7		

2.4 సంఖ్యా వ్యవస్థ మార్పిడిలు (Number System Conversions)

2.4.1 దశాంశ నుండి ద్వి సంఖ్యకు మార్పుచేయుట (Decimal to Binary Conversion)

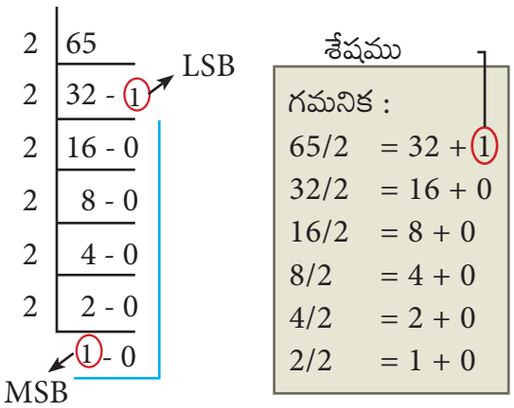
సాధారణంగా రెండు రకములైన పద్ధతులు కలవు.

పద్ధతి 1: దశాంశము నుండి ద్వి సంఖ్యమానము మార్చుటకు 2చే పునరావృత్త భాగాహారము పద్ధతిని ఉపయోగిస్తారు. ఏదైనా దశాంశ సంఖ్యను 2 చే భాగిస్తే 0 లేదా 1 మిగిలి ఉంటుంది. 2 తో పునరావృత్తం అయ్యే విభజన 0 లు మరియు 1 ల క్రమాన్ని వదిలిస్తుంది మరియు అది దశాంశ సంఖ్యకు సమానమైన బైనరి అవుతుంది. దశాంశ సంఖ్య N ను ద్వి సంఖ్య రూపంలోకి మార్చుట అవసరము అనుకుందాం. N ను '2' ద్వారా భాగించడం దశాంశ వ్యవస్థ. ఈ దశాంశ వ్యవస్థలో N ను 2 చే భాగించినపుడు భాగఫలము N1 మరియు శేషము R1 వచ్చును. అప్పుడు శేష విలువలు 0 లేక 1 అయ్యే వరకు ప్రక్రియ పునరావృతం అగును. గుణకం 0 లేదా 1 అయినపుడు ఇది చివరి మిగిలిన విలువ అవుతుంది. పొందిన చివరి మిగిలిన విలువల నుండి పొందిన మొదటి మిగిలిన విలువ వరకు తుది సమాచారాన్ని రాయండి.

ఉదాహరణ

$(65)_{10}$ కు సమానమైన బైనరీ సంఖ్యగా మార్చండి.

11th Computer Science_Chapter-2 Telugu.indd 17 30-03-2023 18:02:09



$(65)_{10} = (1\ 0\ 0\ 0\ 0\ 0\ 1)_2$

పద్ధతి 2 : 2 యొక్క ఘాత పద్ధతి

దశాంశ సంఖ్యను ద్వి సంఖ్యామానంలో మార్చుటకు 2 యొక్క ఘాతములను జోడించి ఆ సంఖ్య యొక్క మొత్తము విలువ పొందుటకు అవసరమైన బిట్టును జోడించడం ద్వారా దశాంశ సంఖ్య ద్వి సంఖ్యగా మారును.

అ) 65 కు సమానంగా లేక దానికన్నా చిన్నదైన 2 యొక్క అతి పెద్ద ఘాతమును కనుగొనుము.

$65_{10} > 64_{10}$

ఆ) 64 ను బిట్ 1 గా అమర్చుము మరియు ఇవ్వబడిన విలువ నుండి 64 ను తీసివేయుము.

65-64=1

ఇ) శేష విలువ కన్నా 32 పెద్దది. కావున 32 యొక్క బిట్టును '0' కి సెట్ చేయండి.

ఈ) శేష విలువ కన్నా 16 పెద్దది. 16 యొక్క బిట్టును 0 కి అమర్చి వ్రాయుము.

ఉ) శేష విలువ కన్నా 8 పెద్దది 8 యొక్క బిట్టును '0' కి అమర్చి వ్రాయుము.

ఊ) శేష విలువ కన్నా 4 పెద్దది కావున 4 యొక్క బిట్టును '0' కి అమర్చి వ్రాయుము.

ఎ) శేష విలువ కన్నా 2 పెద్దది కావున 2 యొక్క బిట్టును '0' కి అమర్చి వ్రాయుము.

ఏ) శేష విలువ కన్నా 1 పెద్దది కావున 1 యొక్క బిట్టును '0' కి అమర్చి వ్రాయుము.

1-1=0

మార్పిడి ప్రక్రియ పూర్తి అయినది $65_{10} = (1000001)_2$

ఉదాహరణ: మార్పిడి దశలను ఈ క్రింది విధంగా ఇవ్వబడినది.

ఇవ్వబడిన సంఖ్య : 65

సమానమైన లేక ఘాతము 2 కన్నా తక్కువ గల విలువ అనునది : 64

(1) 65 - 64 = 1

(2) 1 - 1= 0

యొక్క ఘాతము 2	64	32	16	8	4	2	1
ద్వి సంఖ్య	1	0	0	0	0	0	1

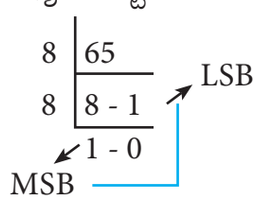
$65_{10} = (1000001)_2$

2.4.2 దశాంశ సంఖ్యను అష్టాంశ సంఖ్యగా మార్చుట (Decimal to Octal Conversion)

ఒక దశాంశ సంఖ్యను అష్టాంశ సంఖ్యగా మార్చుటకు '8' యొక్క ఆవర్తన భాగాహార పద్ధతిని ఉపయోగించి కనుగొనవచ్చు. ఈ పద్ధతి 2.4.1 లో నేర్చిన విధంగా అదే పద్ధతులను ఉపయోగించవలెను. కాని ఈ పద్ధతిలో ఇవ్వబడిన సంఖ్యను 8 చే ఆవర్తన భాగాహార పద్ధతిలో భాగించవలెను.

ఉదాహరణ

$(65)_{10}$ అను సంఖ్యను అష్టాంశ రూపంలో మార్చుము



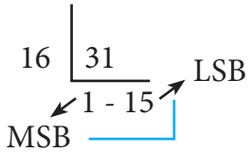
$(65)_{10} = (1\ 0\ 1)_8$

2.4.3 దశాంశ సంఖ్యను ద్వి అష్టాంశ సంఖ్యగా మార్చుట (Decimal to Hexadecimal Conversion)

ఒక దశాంశ సంఖ్యను ద్వి అష్టాంశ సంఖ్యగా మార్చుటకు 16 యొక్క ఆవర్తన భాగాహార పద్ధతిని ఉపయోగించవచ్చు. ఈ పద్ధతిని 2.41 లో నేర్చుకొన్న విధంగా అదే పద్ధతిలో కాని ఈ పద్ధతిలో ఇవ్వబడిన సంఖ్యను 16 తో భాగాహారం చేయవలెను.

ఉదాహరణ

$(31)_{10}$ ను కు సమానమైన హెక్సాడెసిమల్‌నంబర్‌గా మార్చడం



$(16)_{10} = (1F)_{16}$((పట్టిక 2.2 F=15 ను చూడుము)

11th Computer Science_Chapter-2 Telugu.indd 18 30-03-2023 18:02:09

2.4.4 భిన్న దశాంశ సంఖ్యలను ద్వి సంఖ్యగా మార్చుట (Conversion of fractional Decimal to Binary)

ఈ పద్ధతి ద్వారా ఇటువంటి దశాంశ భిన్నాలను మార్చుటకు **2 ను పునరావృత గుణకారంగా** ఉపయోగించాలి.

2 చేత ఆవర్తన గుణకారం చేయు సోపానాలు:

సోపానము 1: దశాంశ భిన్నమును 2 తో గుణించండి మరియు పూర్ణాంక భాగమును గమనించండి. పూర్ణాంక భాగము 0 లేక 1 అగును.

సోపానము 2: సోపానము 1 లో లభించిన పూర్ణాంక భాగాన్ని విస్మరించండి పూర్ణాంక భాగము ఇవ్వబడిన భిన్న దశాంశ సంఖ్య నుండి తీసివేసి మిగిలిన భిన్న దశాంశ సంఖ్యను మరలా 2 చే గుణించండి. అదే భిన్నం పునరావృతం అయ్యేవరకు లేదా ముగిసే వరకు సోపానం 1 ని పునరావృతం చేయండి.

సోపానము 3: ఫలితంగా వచ్చే పూర్ణాంక భాగం '0' మరియు 1 ల క్రమాన్ని ఏర్పరుస్తుంది. ఇది దశాంశ భిన్నానికి సమానమైన బైనరీ అవుతుంది.

సోపానము 4: ఫలితముగా లభించిన అన్ని పూర్ణాంకములను పై నుండి క్రిందకు వ్రాయవలెను.

ఉదాహరణ	పూర్ణాంక భాగము	
$0.2 \times 2 = 0.4$	0	(ఫలితముగా లభించిన మొదటి పూర్ణాంకము)
$0.4 \times 2 = 0.8$	0	
$0.8 \times 2 = 1.6$	1	
$0.6 \times 2 = 1.2$	1	
$0.2 \times 2 = 0.4$	0	(చివరిగా లభించిన ఫలిత పూర్ణాంక భాగము)

గమనిక: భిన్నం పునరావృతం అవుతుంది లబ్ధం అనునది మొదటి సోపానం వలె ఉంటుంది.

Fractional బైనరీ నంబర్‌ను పొందడానికి ఇంటీజర్ పార్ట్‌ను పై నుండి క్రిందికి రాయండి. అందువలన $(0.2)_{10} = (0.00110011\ldots)_2 = (0.00110011)_2$

శిక్షణ శిబిరం (Workshop)

3.క్రింద ఇవ్వబడిన దశాంశ సంఖ్యలను, ద్వి సంఖ్య, అష్టాంశ మరియు ద్విసంఖ్యలుగా మార్చుము.

1) 1920 2) 255 3)126

2.4.5 ద్వి సంఖ్య నుండి దశాంశ రూపంలో మార్చుట (Binary to Decimal Conversion)

ద్వి సంఖ్యలను దశాంశ సంఖ్యగా మార్చుటకు పొజిషినల్ నొటేషన్ పద్ధతిని ఉపయోగిస్తాము.

సోపానం 1: బైనరీ సంఖ్యలు మరియు 2 యొక్క ఘాతాల జాబితాను కుడి నుండి ఎడమకు రాయండి.

సోపానం 2: డిజిట్స్ కోసం రాసిన ప్రతి positional notation కోసం ఇప్పుడు సమానమైన బరువులను వ్రాయుము.

సోపానం 3: ప్రతి అంకెను దాని సంబంధిత బరువు (weight) తో గుణించండి.

సోపానం 4: అన్ని విలువలను జోడించండి.

పట్టిక 2.3 స్థాన సంజ్ఞామానం మరియు బరువు

స్థాన సంజ్ఞామానం	బరువు
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32

స్థాన సంజ్ఞామానం	బరువు
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1024

ఉదాహరణ

$(111011)_2$ ను సమాన దశాంశ సంఖ్యలుగా మార్చుము.

బరువు	32	16	8	4	2	1
స్థాన సంజ్ఞామానం (2 యొక్క ఘాతాలు)	2^5	2^4	2^3	2^2	2^1	2^0
ఇవ్వబడిన ద్వి సంఖ్య	1	1	1	0	1	1

11th Computer Science_Chapter-2 Telugu.indd 19 30-03-2023 18:02:09

$32+16+8+0+2+1 = (59)_{10}$

$(111011)_2 = (59)_{10}$

2.4.6 ద్వి సంఖ్యా నుండి అష్టాంశ సంఖ్యలుగా మార్చుట (Binary to Octal Conversion)

సోపానం 1:ఇవ్వబడిన ద్వి సంఖ్యలను కుడి నుండి ఎడమకు 3 బిట్స్ (Bits) లోకి సమూహపరచుట.

సోపానం 2: మీరు 3 బిట్‌ల సమూహాన్ని ఎక్కువగా వదిలివేసేందుకు ముందు 0 ను జోడించవచ్చును. group 3 బిట్స్ కంటే తక్కువ కలిగి ఉంటుంది.

సోపానం 3: 2 యొక్క ఘాతము స్థాన బరువు పద్ధతిని ఉపయోగించి సమానమైన అష్టాంశ విలువగా మార్చుము.

అష్టాంశ సంఖ్యలు మరియు వాటి ద్వి సంఖ్య సమానమైనది పట్టిక 2.4

అష్టాంశ	ద్వి సంఖ్య సమానమైనది
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

ఉదాహరణ

$(11010110)_2$ అను సంఖ్యను అష్టాంశ సమాన సంఖ్యగా మార్చుట.

సోపానం 1: ఇవ్వబడిన ద్వి సంఖ్యలను కుడి నుండి ఎడమకు 3 బిట్లుగా మార్చుము

011 010 110

గమనిక: ఎడమ ఒకే సమూహము 3 బిట్ల కన్నా తక్కువ కలిగివుండును. కనుక 3 బిట్ల సమూహమును చేయుటకు దీని ఎడమ వైపున 0 జోడించవలయును.

సోపానం-2: ప్రతి విలువకు సమానమైన అష్టాంశ సమూహమును కనుగొనుట.

$\underbrace{011}_{3}\ \underbrace{010}_{2}\ \underbrace{110}_{6}$

$(11010110)_2 = (326)_8$

2.4.7 ద్వి సంఖ్య నుండి ద్వి అష్టాంశ సంఖ్యలుగా మార్చుట (Binary to Hexadecimal Conversion)

సోపానం 1: ఇవ్వబడిన సంఖ్యను కుడి నుండి ఎడమకు 4 బిట్లుగా సమూహ (Group) పరచండి.

సోపానం 2: ఎడమ వైపు అత్యధికంగా Group లో 4 bits కంటే తక్కువ ఉండే 4 బిట్‌ల గ్రూపును రూపొందించడానికి దాని ముందు '0' ను జోడించండి.

సోపానం 3: 2 యొక్క పవర్ పొజిషినల్ weight పద్ధతిని ఉపయోగించి సమానమైన హెక్సాడెసిమల్ విలువగా మార్చండి.

ఉదాహరణ:

$(1111010110)_2$ అను సంఖ్యను ద్వి అష్టాంశ సంఖ్యగా మార్చుము

సోపానం 1: ఇవ్వబడిన ద్వి సంఖ్యలను కుడి నుండి ఎడమనకు 4 బిట్లుగా సమూహపరచండి.

0011 1101 0110

గమనిక: 4 బిట్ల Group గా మార్చడానికి ఎడమ వైపు గల అత్యధిక సమూహానికి 0 ను జోడించబడినది.

$\underbrace{0011}_{3}\ \underbrace{1101}_{D}\ \underbrace{0110}_{6}$

$(1111010110)_2 = (3D6)_{16}$

2.4.8 పాక్షిక బైనరీ సంఖ్యను దాని సమానమైన డెసిమల్‌గా మార్చండి (Conversion of fractional Binary to Decimal equivalent)

క్రింద ఇవ్వబడిన సోపానాలను ఉపయోగించి పాక్షిక బైనరీ సంఖ్యలను దానికి సమానమైన దశాంశ సంఖ్యలుగా మార్చుట.

11th Computer Science_Chapter-2 Telugu.indd 20 30-03-2023 18:02:09

సోపానం 1: ఇవ్వబడిన అంతర్గత భాగంలో ద్వి సంఖ్యల నుండి సమగ్ర దశాంశ సమానమైనదిగా మార్చుట. (పటము 2.4.5 చూడుము)

సోపానం 2: దశాంశ భాగము నుండి ద్వి సంఖ్యా సంఖ్యలకు సమానమైన స్థాన సంజ్ఞామానమును ఉపయోగించి క్రింద ఇవ్వబడిన సోపానములను ఉపయోగించుము.

సోపానం 2.1: ద్వి సంఖ్యా విధానమును భిన్న భాగముగా మార్చుము.

సోపానం 2.2: అన్ని అంకెలను 2^{-1}, 2^{-2}, 2^{-3}..... 2^{-n} ను 2 యొక్క ఫలితాలుగాను ఎడమ నుండి కుడికి రాయండి. ఇప్పుడు దానికి సమానమైన బరువులను రాయండి.

సోపానం 2.3: ఒక్కొక్క అంకెను దాని సంబంధిత బరువులో గుణించుము.

సోపానం 2.4: సోపానం 2.3 లోని విలువలు వచ్చేంత వరకు జోడించండి.

పట్టిక 2.5 స్థాన సంజ్ఞామానం మరియు బరువు

స్థాన సంజ్ఞామానం	బరువు
2^{-1} (1/2)	0.5
2^{-2} (1/4)	0.25
2^{-3} (1/8)	0.125
2^{-4} (1/16)	0.0625
2^{-5} (1/32)	0.03125
2^{-6} (1/64)	0.015625
2^{-7} (1/128)	0.0078125

సోపానం 3: చివరి సమాధానము పొందడానికి సమగ్ర భాగాన్ని దశాంశ బిందువుతో చేర్చి వ్రాయవలయును.

ఉదాహరణ

ఇవ్వబడిన $(11.011)_2$ ద్వి సంఖ్యను దశాంశ సమానమైన పూర్ణాంక భాగము వ్రాయుము $(11)_2 = 3$ (పట్టిక 2.5 ను చూడుము)

2^1	2^0		2^{-1}	2^{-2}	2^{-3}
↑	↑		↑	↑	↑
1	1	.	0	1	1

$3 + . (0\times0.5 + 1\times0.25 + 1\times0.125)$
$= 3.375$
$(11.011)_2 = (3.375)_{10}$

శిక్షణ శిబిరం (Workshop)

4. క్రింద ఇవ్వబడిన ద్వి సంఖ్యా సంఖ్యలకు సమానమైన దశాంశము, అష్టాంశము మరియు ద్వి అష్టాంశముగా మార్చుము.
1) 101110101 2) 1011010 3) 101011111

2.4.9. అష్టాంశ సంఖ్యను దశాంశ సంఖ్యగా మార్చుట (Octal to Decimal Conversion)

అష్టాంశ సంఖ్యకు దశాంశ సంఖ్యగా మార్చుటకు స్థాన సంజ్ఞామానమును పద్ధతిని ఉపయోగించవచ్చును.

1. అష్టాంశ అంకెలు మరియు 8 యొక్క ఘాతముల జాబితాను కుడి నుండి ఎడమకు రాయండి.
2. ప్రతి అంకె యొక్క పొజిషల్ నొటిషన్‌కు దానికి సమానమైన బరువును రాయండి
3. అన్ని అంకెలను సంబంధిత బరువుతో గుణించాలి.
4. అన్ని విలువలను జోడించండి.

ఉదాహరణ

$(1265)_8$ ను సమాన దశాంశ సంఖ్యగా మార్చుము.

బరువు	512	64	8	1
స్థాన సంజ్ఞామానం	8^3	8^2	8^1	8^0
ఇచ్చిన సంఖ్య	1	2	6	5

$(1265)_8 = 512\times1 + 64\times2 + 8\times6 + 1\times5$
$= 512 + 128 + 48 + 5$
$(1265)_8 = (693)_{10}$

11th Computer Science_Chapter-2 Telugu.indd 21 30-03-2023 18:02:09

2.4.10 అష్టాంశమును ద్వి సంఖ్యగా మార్చుట (Octal to Binary Conversion)

ఇవ్వబడిన సంఖ్యలోని ప్రతి అష్టాంశ అంకెకు స్థాన సంజ్ఞామానం ఉపయోగించి దాని 3 అంకెలు ద్వి సంఖ్య సమానమై వ్రాయుము.

ఉదాహరణ

$(6213)_8$ కు సమానమైన ద్వి సంఖ్యా సంఖ్యలుగా మార్చుము.

6	2	1	3
↓	↓	↓	↓
110	010	001	011

$(6213)_8 = (110010001011)_2$

శిక్షణ శిబిరం (Workshop)

5. క్రింద ఇవ్వబడిన అష్టాంశ సంఖ్యలను ద్వి సంఖ్యా సంఖ్యలుగా మార్చుము.
(అ) 472 (ఆ) 145 (ఇ) 347
(ఈ) 6247 (ఉ) 645

2.4.11 ద్వి అష్టాంశమును దశాంశముగా మార్చుట (Hexadecimal to Decimal Conversion)

స్థాన సంజ్ఞామానమును ఉపయోగించి ద్వి అష్టాంశమును దశాంశముగా మార్చవచ్చును.

1. ద్వి అష్టాంశ అంకెలు మరియు 16 యొక్క ఘాతముల జాబితాను కుడి నుండి ఎడమకు వ్రాయండి.
2. ప్రతి అంకె కోసం వ్రాసిన ప్రతి స్థాన సంజ్ఞామానం కోసం సమానమైన బరువును వ్రాయవలయును.
3. ప్రతి అంకెను సంబంధిత బరువుతో గుణించుము
4. తుది విలువ కొరకు అన్ని విలువను జోడించండి.

ఉదాహరణ:

$(25F)_{16}$ నకు సమానమైన దశాంశ సంఖ్యగా మార్చుము.

బరువు	256	16	1
స్థాన సంజ్ఞామానం	16^2	16^1	16^0
ఇచ్చిన సంఖ్య	2	5	F(15)

$(25F)_{16} = 2\times256 + 5\times16 + 15\times1$
$= 512 + 80 + 15$
$(25F)_{16} = (607)_{10}$

2.4.12 ద్వి అష్టాంశము నుండి ద్వి సంఖ్యగా మార్చుట (Hexadecimal to Binary Conversion)

ఇవ్వబడిన సంఖ్యకు స్థాన సంజ్ఞామానం పద్ధతిని ఉపయోగించి హెక్సాడెసిమల్ అంకెకు సమానమైన 4 bit ల బైనరీగా మార్చండి.

ఉదాహరణ:

$(8BC)_{16}$ కు సమాన ద్వి సంఖ్యా సంఖ్యగా మార్చుము

8	B	C
↓	↓	↓
1000	1011	1100

$(8BC)_{16} = (100010111100)_2$

శిక్షణ శిబిరం (Workshop)

6. క్రింద ఇవ్వబడిన ద్వి అష్టాంశ సంఖ్యలను ద్వి సంఖ్యా సంఖ్యలుగా మార్చుట.
(అ) A6 (ఆ) BE
(ఇ) 9BC8 (ఈ) BC9

2.5 సైన్డ్ నంబర్స్ కొరకు బైనరీ రెప్రజెంటేషన్స్ (Binary Representation for Signed Numbers)

కంప్యూటర్లు సానుకూలము మరియు ప్రతికూల సంఖ్యలను నిర్వహించగలదు. ప్రతికూల ద్వి సంఖ్యా సంఖ్యలను సూచించే సరళమైన పద్ధతిని సైన్డ్ యోగ్నిట్యూడ్ అందురు. సైన్డ్ యోగ్నిట్యూడ్ పద్ధతిలో ఎడమ ఒక బిట్ (left most bit) లేక చాలా ముఖ్యమైన బిట్ (MSB) దీనిని సైన్ బిట్ లేక పారిటీ బిట్ అందురు.

11th Computer Science_Chapter-2 Telugu.indd 22 30-03-2023 18:02:09

కంప్యూటర్లలోని సంఖ్యలు వివిధ మార్గాల్లో సూచించబడును. అవి

- సంతకం చేసిన పరిమాణం (Signed Magnitude representation)
- 1 యొక్క పూరక (1's Complement)
- 2 యొక్క పూరక (2's Complement)

2.5.1 సైన్డ్ చేసిన పరిమాణ ప్రాతినిధ్యం (Signed Magnitude representation)

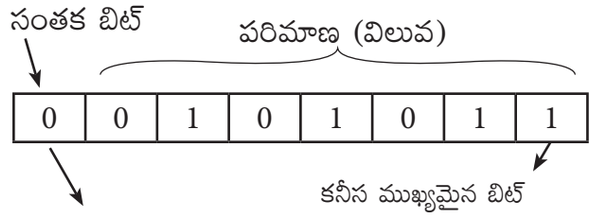
ఒక మొత్తం సంఖ్యల విలువను దీని యందు ఉపయోగించే గుర్తు ద్వారా నిర్ణయించవచ్చును. ఆ సంఖ్యకు '+' గుర్తు లేక సంకేతం లేక అది సానుకూలముగా పరిగణించబడు సంఖ్య. '–' గుర్తు వుంటే అది ప్రతికూలముగా పరిగణించబడును.

ఉదాహరణ:

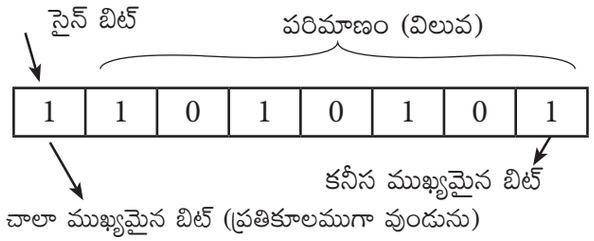
+43 లేక 43 సానుకూల సంఖ్య

–43 అనునది ఒక ప్రతికూల సంఖ్య

సైన్ చేసిన ద్వి సంఖ్యా ప్రాతినిధ్యంలో ఎడమ వైపు బిట్‌ను సైన్డ్ బిట్‌గా పరిగణిస్తారు. అప్పుడు ఈ బిట్ '0' అయితే ఇది సానుకూల సంఖ్య మరియు అది 1 అయితే ఇది ప్రతికూల సంఖ్య అగును. అందువలన సైన్ చేసిన ద్వి సంఖ్యా 8 బిట్లను కలిగి ఉంటుంది. విలువలను నిల్వ చేయడానికి 7 బిట్స్ మాత్రమే ఉపయోగించబడును మరియు ఎడమ '1' బిట్ గుర్తు కోసం ఉపయోగపడు +43 అనునది క్రింది విధంగా సూచించబడును.



-43 అనునది క్రింద విధముగా సూచించదు.



2.5.2. 1 యొక్క పూరక ప్రాతినిధ్యం (1's Complement representation)

సైన్ చేసిన సంఖ్యలను సూచించుటకు ఇది సులభమైన విధానము. ఇది ప్రతికూల సంఖ్యల కోసము మాత్రమే వుండే సంఖ్య. ఇది MSB యొక్క పూరకం 1.

ఒక సంఖ్య 1 యొక్క పూరకం కనుగొనుటకు ఈ క్రింది సోపానమును పాటించుము.

సోపానం 1: ఇవ్వబడిన దశాంశ సంఖ్యలకు సమాన ద్వి సంఖ్యా సంఖ్యలుగా మార్చుట.

సోపానం 2: మార్చబడిన ద్వి సంఖ్యా 8 బిట్లు (bits) గా ఉన్నదా అని సరిచూడవలయును. 8 బిట్లుకు తక్కువగా వుండిన ముందుగా 0 లను చేర్చి 8 బిట్లుగా మార్చవలయును.

సోపానం 3: అన్ని బిట్లును తల క్రిందులుగా మార్చుము (అది 1 అనునది 0 గాను 0 అనునది 1 గాను మార్చవలయును)

ఉదాహరణ:

$(-24)_{10}$ కోసం 1– యొక్క పూరక కనుగొనుము.

ఇచ్చిన సంఖ్య	ద్వి సంఖ్యా సంఖ్య	1's యొక్క పూరక
$(-24)_{10}$	00011000	11100111

2.5.3 2 యొక్క పూరక ప్రాతినిధ్యం (Complement representation)

ప్రతికూల సంఖ్య కోసం 2 యొక్క పూరక పద్ధతి ఈ క్రింది విధంగా వుండును.

అ. ద్వి సంఖ్యా క్రమంలోని అన్ని బిట్లను తలక్రిందులు చేయగా (అనగా 1 ని 0 గాను 0 ను 1 గాను)

ఆ. తరువాత కనీసం ముఖ్యమైన బిట్ గుండా 1 ని మొత్తము (LSB). ఫలితమునకు అత్యల్పంగా 1 ని జోడించండి.

ఉదాహరణ:

$(-24)_{10}$ను 2 యొక్క పూరక ప్రాతినిధ్యం

11th Computer Science_Chapter-2 Telugu.indd 23 30-03-2023 18:02:09

+24 కు సమాన ద్వి సంఖ్యా సంఖ్యలు :	11000
8 బిట్ ఫార్మాట్	00011000
1 యొక్క పూరక	11100111
LSB నకు 1 జోడించండి	+1
2 యొక్క పూరక -24:	**11101000**

శిక్షణ పట్టిక (Workshop)

7. ఈ క్రింద ఇవ్వబడిన దశాంశ సంఖ్యలను 1 యొక్క పూరక మరియు 2 యొక్క పూరకములలో రాయండి.
(అ) 22 (ఆ) -13 (ఇ) -65 (ఈ) -46

2.6 ద్వి సంఖ్యా అంకగణితము (Binary Arithmetic)

దశాంశ సంఖ్యలు లాగానే ద్వి సంఖ్యా సంఖ్యలు కూడా కూడిక వ్యవకలనము, గుణకారము మరియు భాగాహారము వంటి గణనలను అనుమతిస్తుంది. ఈ భాగము ద్విసంఖ్యా సంఖ్యల యొక్క కూడిక మరియు తీసివేత పద్ధతులను గురించి మాత్రమే వ్యవహరిస్తుంది.

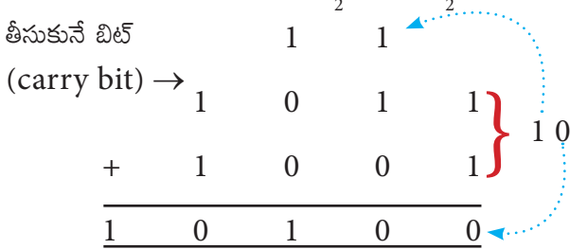
2.6.1 ద్వి సంఖ్యా కూడిక (Binary Addition)

ద్వి సంఖ్యా సంఖ్యలు సులభంగా కూడుటకు క్రింది ఇవ్వబడిన కూడిక పట్టిక ఉపయోగపడును.

A	B	మొత్తం (A + B)	తీసుకునే బిట్
0	0	0	-
0	1	1	-
1	0	1	-
1	1	0	1

1 + 1 = 10 లో 0 ని మాత్రము పరిగణించుకొని 1 యొక్క పూరక స్థానముతో తీసుకొనవలెను. కాబట్టి దీనిని తీసుకునే బిట్ (carry bit)గా పరిగణిస్తారు.

ఉదాహరణ: కూడుము: $1011_2 + 1001_2$



$1011_2 + 1001_2 = 10100_2$

ఉదాహరణ: ద్వి సంఖ్యా రూపములో కూడుట: $23_{10} + 12_{10}$

సోపానము 1: 23 మరియు 12 లను ద్వి సంఖ్యా రూపంలో మార్చుట.

23_{10}

2 యొక్క ఘాతములు	16	8	4	2	1
ద్వి సంఖ్యా సంఖ్యలు	1	0	1	1	1

$23_{10} = 00010111_2$

12_{10}

2 యొక్క ఘాతములు	8	4	2	1
ద్వి సంఖ్యా సంఖ్యలు	1	1	0	0

$12_{10} = 00001100_2$

సోపానము 2: 23 మరియు 12 ల ద్విసంఖ్యా రూపములో కూడుట.

తీసుకేళ్లు బిట్ (carry bit) →			1	1	1		
$23_{10} = 0$	0	0	1	0	1	1	1
$12_{10} = 0$	0	0	0	1	1	0	0
$35_{10} = 0$	**0**	**1**	**0**	**0**	**0**	**1**	**1**

2.6.2 ద్వి సంఖ్యా వ్యవకలనం (Binary Substraction)

ద్వి సంఖ్యా వ్యవకలనం పట్టిక

A	B	(తేడా) వ్యత్యాస విలువ (A-B)	రుణ తీసుకొను విలువ (Borrow)
0	0	0	0
1	0	1	0
1	1	0	0
0	1	1	1

11th Computer Science_Chapter-2 Telugu.indd 24 30-03-2023 18:02:09

0 నుండి 1 ని వ్యవకలనం చేసినపుడు తదుపరి బిట్ (MSB) నుండి 1 ని అప్పుగా తీసుకొనుము. ఆ విధంగా 1 ని అప్పుగా పొందునపుడు దాని విలువ ముందు అత్యంత ముఖ్యమైన బిట్ 1 అయిన దానిని '0' గా మార్చవలయును. అప్పుకు 0 లు 1 అయ్యే వరకు ముఖ్యమైన బిట్ 0 అయిన దాని ముందు ముఖ్యమైన బిట్ 1 గా ముఖ్యమైన బిట్ 1 ని కలిగి ఉన్న మరింత ముఖ్యమైన బిట్ నుండి రుణం తీసుకోవాలి మరియు దానిని 0 మరియు 0 సెలతో భర్తీ చేయాలి. తీసివ్రాయుము (వ్యవకలనం చేయుము)

ఉదాహరణ: తీసివేయుము (వ్యవకలనం చేయుము) $1001010_2 - 10100_2$

	0	1	10	0	10	
	~~1~~	~~0~~	0	~~1~~	0	1	0
(-)			1	0	1	0	0
		1	1	0	1	1	0

ఉదాహరణ: ద్వి సంఖ్యా రూపంలో కూడుము : $(-21)_{10} + (5)_{10}$

సోపానము 1: -21 మరియు 5 లను ద్వి సంఖ్యా రూపంలోనికి మార్చుము.

21_{10}

2 యొక్క ఘాతములు	16	8	4	2	1
ద్వి సంఖ్యా సంఖ్యలు	1	0	1	0	1

$21_{10} = 00010101_2$

5_{10}

2 యొక్క ఘాతములు	4	2	1
ద్వి సంఖ్యా సంఖ్యలు	1	0	1

$5_{10} = 00000101_2$

సోపానము 2:

21_{10}	0	0	0	1	0	1	0	1
1 యొక్క విలువ	1	1	1	0	1	0	1	0
2 యొక్క పూరక	1	1	1	0	1	0	1	1

సోపానము 3:

–21 మరియు 5 యొక్క ద్వి సంఖ్యా కూడిక

Carry bit				1	1	1	1	
-21_{10}	1	1	1	0	1	0	1	1
5_{10}	0	0	0	0	0	1	0	1
-16_{10} (Result)	1	1	1	1	0	0	0	0

శిక్షణ పట్టిక (Workshop)

8. క్రింద ఇవ్వబడిన ద్వి సంఖ్యా గణనను జరుపుము.

(A) $10_{10} + 15_{10}$ (B) $-12_{10} + 5_{10}$

(C) $14_{10} - 12_{10}$ (D) $(-2_{10}) - (-6_{10})$

2.7 మెమరీలోని అక్షరాలను సూచించడం (Representing Characters in Memory)

పరిచయంలో ప్రాతినిధ్యం వహించినట్లుగా ఒక కంప్యూటర్‌కు ఇవ్వబడిన అన్ని ఇన్‌పుట్ డేటా అర్థమయ్యే విధంగా ఉండవలెను. సాధారణంగా కంప్యూటర్‌లో 26 ఆంగ్ల పెద్ద అక్షరాలు (uppercase) 26 చిన్న అక్షరాలు (lowercase) 0 నుండి 9 అంకెలు మరియు ప్రత్యేక అక్షరాలు. దీనిని అక్షర సమితి (character set) అందురు. ఈ అక్షర సమితుల మొత్తం సంఖ్యల ద్వారా మాత్రమే సూచించబడును. అక్షర సమితిలోని అన్ని అక్షరాలకు సాధారణ ఎన్కోడింగ్ వ్యవస్థకు అవసరము. కంప్యూటర్ కోసం అనేక ఎన్కోడింగ్ వ్యవస్థలు ఉపయోగపడును. అవి:

- BCD – Binary Coded Decimal
- EBCDIC – Extended Binary Coded Decimal Interchange Code
- ASCII – American Standard Code for Information Interchange
- Unicode
- ISCII - Indian Standard Code for Information Interchange

11th Computer Science_Chapter-2 Telugu.indd 25 30-03-2023 18:02:10

2.7.1 ద్వి సంఖ్యా కోడెడ్ దశాంశం (Binary Coded Decimal (BCD))

ఈ ఎన్కోడింగ్ వ్యవస్థ ప్రస్తుతం వాడుకలో లేదు. ఈ పద్ధతి 2^6 బిట్ ఎన్కోడింగ్ వ్యవస్థ. ఇది $2^6 = 64$ అక్షరాలను మాత్రమే ఈ పద్ధతిలో నిర్వహించగలదు.

2.7.2 సమాచార మార్పిడి కోసం అమెరికన్ ప్రామాణిక కోడ్ (American Standard Code for Information Interchange - (ASCII)

ఇది యునైటెడ్ స్టేట్స్ (USA) చేత ఆమోదించబడిన అత్యంత ప్రాచుర్యం కలిగిన ఎన్కోడింగ్ వ్యవస్థ. చాలా కంప్యూటర్లు ఈ వ్యవస్థను ఉపయోగిస్తాయి. ఈ ఎన్కోడింగ్ వ్యవస్థ ఆంగ్ల భాషలోని అక్షరాలను మాత్రమే నిర్వహించగలదు అని తెలుసుకొనుము. 2^7 బిట్‌ను ఈ వ్యవస్థ నిర్వహించగలదు. దీని అర్థం 128 అక్షరాలు ఈ వ్యవస్థలో ప్రతి అక్షరానికి వ్యక్తిగత సంఖ్య వుండును (ఉంటుంది) (అనుబంధం చూడండి)

ASCII -8 అనే కొత్త ఎడిషన్ 2^8 బిట్ పద్ధతిలో 256 అక్షరాలను నిర్వహించగలదు 0 నుండి 255 ప్రత్యేక సంఖ్యల వరకు సూచించబడును.

ఆంగ్ల పెద్ద అక్షరం 'A' కి సమానమైన ASCII గుర్తు సంఖ్య 65. దీనికి సమానమైన 7 బిట్ ASCII యొక్క ద్వి సంఖ్యా ప్రాతినిధ్యం 01000001 అగును. అలాగే ASCII-8 యొక్క ద్వి సంఖ్యా ప్రాతినిధ్యం 01000001

2.7.3 విస్తరించిన ద్వి సంఖ్యా కోడెడ్ దశాంశ ఇంటర్ చేంజ్ కోడ్ (Extended Binary Coded Decimal Interchange Code - EBCDIC)

ఇది 8 బిట్ ప్రాతినిధ్యంలో ASCII Code ని పోలి ఉంటుంది. ఈ కోడింగ్ వ్యవస్థ అంతర్జాతీయ వ్యాపార యంత్రము (International Business Machine-IBM) చేత రూపొందించబడినది. ఈ కోడింగ్ సిస్టమ్ 256 అక్షరాలను నిర్వహించగలదు. ASCII లోని ఇన్‌పుట్ కోడ్‌ను EBCDIC సిస్టమ్‌గా మార్చును మరియు దీనికి విరుద్ధంగా మార్చవచ్చును.

2.7.4 Indian Standard Code for Information Interchange (ISCII)

ISCII అనునది భారతీయ స్థానిక భాషల స్వభావాన్ని నిర్వహించే వ్యవస్థ. ఇది 8 బిట్ కోడింగ్ వ్యవస్థ అందువలన 256 (2^8) అక్షరాలను నిర్వహించగలదు. ఈ వ్యవస్థ భారతదేశంలోని ఎలక్ట్రానిక్ విభాగంచే 1986-88 వ సం॥లో రూపొందించబడినది మరియు Bureau of Indian Standards (BIS) చేత గుర్తింపబడినది. ప్రస్తుతం ఈ కోడింగ్ పద్ధతి యూనికోడ్ Unicode పద్ధతిలో కలుపబడినది.

2.7.5 యూనికోడ్ (Unicode)

ఈ కోడింగ్ విధానము చాలా ఆధునిక కంప్యూటర్లలలో ఉపయోగించబడుతుంది. ASCII తరువాత ప్రసిద్ధ కోడింగ్ పద్ధతి Unicode అగును. ASCII 256 అక్షరాలను మాత్రమే సూచిస్తుంది. అందువలన ఆంగ్లము మరియు ఐరోపా భాషలను మాత్రమే ASCII నిర్వహించగలదు. ముఖ్యంగా తమిళం, మలయాళం, తెలుగు, కన్నడం వంటి భాషలకు ASCII ప్రాతినిధ్యం వహించలేని పరిస్థితి ఏర్పడింది. అందువల్ల యూనివర్సల్ లాంగ్వేజ్‌ల కోడింగ్ సిస్టమ్‌ను నిర్వహించడానికి. Unicode నిర్వహించబడినది. ఇది 16 బిట్ కోడ్‌ను మరియు 65536 అక్షరాలను నిర్వహించగలదు.

యూనికోడ్ పథకం 16 బిట్ కోడ్ పద్ధతిలో సూచించబడుచున్నది. తమిళం, తెలుగు, మలయాళం మరియు కన్నడం యొక్క యూనికోడ్ సూచిక పట్టిక 2.6 లో ఇవ్వబడినది.

11th Computer Science_Chapter-2 Telugu.indd 26 30-03-2023 18:02:10

పట్టిక 2.6

తమిళం యొక్క యూనికోడ్ పట్టిక

	0B8	0B9	0BA	0BB	0BC	0BD	0BE	0BF
0		ஐ 0B90		ர 0BB0	ீ 0BC0	ௐ 0BD0		௰ 0BF0
1				ற 0BB1	ு 0BC1			௱ 0BF1
2	ஂ 0B82	ஒ 0B92		ல 0BB2	ூ 0BC2			௲ 0BF2
3	ஃ 0B83	ஓ 0B93	ண 0BA3	ள 0BB3				௳ 0BF3
4		ஔ 0B94	த 0BA4	ழ 0BB4				௴ 0BF4
5	அ 0B85	க 0B95		வ 0BB5				௵ 0BF5
6	ஆ 0B86			ஶ 0BB6	ெ 0BC6		௦ 0BE6	௶ 0BF6
7	இ 0B87			ஷ 0BB7	ே 0BC7	ௗ 0BD7	க 0BE7	௷ 0BF7
8	ஈ 0B88		ந 0BA8	ஸ 0BB8	ை 0BC8		உ 0BE8	௸ 0BF8
9	உ 0B89	ங 0B99	ன 0BA9	ஹ 0BB9			ங 0BE9	௹ 0BF9
A	ஊ 0B8A	ச 0B9A	ப 0BAA		ொ 0BCA		ச 0BEA	௺ 0BFA
B					ோ 0BCB		ந் 0BEB	
C		ஜ 0B9C			ௌ 0BCC		ஂ 0BEC	
D					் 0BCD		எ 0BED	
E	எ 0B8E	ஞ 0B9E	ம 0BAE	ா 0BBE			அ 0BEE	
F	ஏ 0B8F	ட 0B9F	ய 0BAF	ி 0BBF			கூ 0BEF	

మలయాళం యొక్క యూనికోడ్ పట్టిక

	0D0	0D1	0D2	0D3	0D4	0D5	0D6	0D7
0	ം 0D00	ഐ 0D10	ഠ 0D20	ര 0D30	ീ 0D40		ൠ 0D60	൰ 0D70
1	ഁ 0D01		ഡ 0D21	റ 0D31	ു 0D41		ൡ 0D61	൱ 0D71
2	ഃ 0D02	ഒ 0D12	ഢ 0D22	ല 0D32	ൂ 0D42		ൢ 0D62	൲ 0D72
3	അ 0D03	ഓ 0D13	ണ 0D23	ള 0D33	ൃ 0D43		ൣ 0D63	൳ 0D73
4		ഔ 0D14	ത 0D24	ഴ 0D34	ൄ 0D44	ഄ 0D54		൴ 0D74
5	ആ 0D05	ക 0D15	ഥ 0D25	വ 0D35		യ 0D55		൵ 0D75
6	ആ 0D06	ഖ 0D16	ദ 0D26	ശ 0D36	െ 0D46	ഴ 0D56	൦ 0D66	൶ 0D76
7	ഇ 0D07	ഗ 0D17	ധ 0D27	ഷ 0D37	േ 0D47	ൗ 0D57	൧ 0D67	൷ 0D77
8	ഈ 0D08	ഘ 0D18	ന 0D28	സ 0D38	ൈ 0D48	ഩ 0D58	൨ 0D68	൸ 0D78
9	ഉ 0D09	ങ 0D19	ഩ 0D29	ഹ 0D39		ഺ 0D59	൩ 0D69	൹ 0D79
A	ഊ 0D0A	ച 0D1A	പ 0D2A	ഺ 0D3A	ൊ 0D4A	഻ 0D5A	൪ 0D6A	ൺ 0D7A
B	ഋ 0D0B	ഛ 0D1B	ഫ 0D2B	഻ 0D3B	ോ 0D4B	഼ 0D5B	൫ 0D6B	ൻ 0D7B
C	ഌ 0D0C	ജ 0D1C	ബ 0D2C	഼ 0D3C	ൌ 0D4C	ഽ 0D5C	൬ 0D6C	ർ 0D7C
D		ഝ 0D1D	ഭ 0D2D	ഽ 0D3D	് 0D4D	ാ 0D5D	൭ 0D6D	ൽ 0D7D
E	എ 0D0E	ഞ 0D1E	മ 0D2E	ാ 0D3E	◌ 0D4E	ൠ 0D5E	൮ 0D6E	ൾ 0D7E
F	ഏ 0D0F	ട 0D1F	യ 0D2F	ി 0D3F	൏ 0D4F	ൟ 0D5F	൯ 0D6F	ൿ 0D7F

11th Computer Science_Chapter-2 Telugu.indd 27 30-03-2023 18:02:10

పట్టిక 2.6

తెలుగు యొక్క యూనికోడ్ పట్టిక

	0C0	0C1	0C2	0C3	0C4	0C5	0C6	0C7
0	◌ంౕ 0C00	ఐ 0C10	త 0C20	ర 0C30	◌ీ 0C40		ౠ 0C60	
1	◌ఁ 0C01		డ 0C21	ఱ 0C31	◌ు 0C41		ౡ 0C61	
2	◌ం 0C02	ఒ 0C12	ఢ 0C22	ల 0C32	◌ూ 0C42		◌ౢ 0C62	
3	◌ః 0C03	ఓ 0C13	ణ 0C23	ళ 0C33	◌ృ 0C43		◌ౣ 0C63	
4		ఔ 0C14	త 0C24	ఴ 0C34	◌ౄ 0C44			
5	అ 0C05	క 0C15	థ 0C25	వ 0C35		◌ఽ 0C55		
6	ఆ 0C06	ఖ 0C16	ద 0C26	శ 0C36	◌ె 0C46	◌ౖ 0C56	౦ 0C66	
7	ఇ 0C07	గ 0C17	ధ 0C27	ష 0C37	◌ే 0C47		౧ 0C67	
8	ఈ 0C08	ఘ 0C18	న 0C28	స 0C38	◌ై 0C48	◌ౘ 0C58	౨ 0C68	౸ 0C78
9	ఉ 0C09	ఙ 0C19		హ 0C39		◌ౙ 0C59	౩ 0C69	౹ 0C79
A	ఊ 0C0A	చ 0C1A	ప 0C2A		◌ొ 0C4A	ఀ 0C5A	౪ 0C6A	౺ 0C7A
B	ఋ 0C0B	ఛ 0C1B	ఫ 0C2B		◌ో 0C4B		౫ 0C6B	౻ 0C7B
C	ఌ 0C0C	జ 0C1C	బ 0C2C		◌ౌ 0C4C		౬ 0C6C	౼ 0C7C
D		ఝ 0C1D	భ 0C2D	ఽ 0C3D	◌్ 0C4D		౭ 0C6D	౽ 0C7D
E	ఎ 0C0E	ఞ 0C1E	మ 0C2E	◌ా 0C3E			౮ 0C6E	౾ 0C7E
F	ఏ 0C0F	ట 0C1F	య 0C2F	◌ి 0C3F			౯ 0C6F	౿ 0C7F

కన్నడం యొక్క యూనికోడ్ పట్టిక

	0C8	0C9	0CA	0CB	0CC	0CD	0CE	0CF
0	◌ಀ 0C80	ಐ 0C90	ಠ 0CA0	ರ 0CB0	◌ೀ 0CC0		ೠ 0CE0	
1	◌ಁ 0C81		ಡ 0CA1	ಱ 0CB1	◌ು 0CC1		ೡ 0CE1	☒ 0CF1
2	◌ಂ 0C82	ಒ 0C92	ಢ 0CA2	ಲ 0CB2	◌ೂ 0CC2		◌ೢ 0CE2	◌◌ 0CF2
3	◌ಃ 0C83	ಓ 0C93	ಣ 0CA3	ಳ 0CB3	◌ೃ 0CC3		◌ೣ 0CE3	
4		ಔ 0C94	ತ 0CA4		◌ೄ 0CC4			
5	ಅ 0C85	ಕ 0C95	ಥ 0CA5	ವ 0CB5		◌ೕ 0CD5		
6	ಆ 0C86	ಖ 0C96	ದ 0CA6	ಶ 0CB6	◌ೆ 0CC6	◌ೖ 0CD6	೦ 0CE6	
7	ಇ 0C87	ಗ 0C97	ಧ 0CA7	ಷ 0CB7	◌ೇ 0CC7		೧ 0CE7	
8	ಈ 0C88	ಘ 0C98	ನ 0CA8	ಸ 0CB8	◌ೈ 0CC8		೨ 0CE8	
9	ಉ 0C89	ಙ 0C99		ಹ 0CB9			೩ 0CE9	
A	ಊ 0C8A	ಚ 0C9A	ಪ 0CAA		◌ೊ 0CCA		೪ 0CEA	
B	ಋ 0C8B	ಛ 0C9B	ಫ 0CAB		◌ೋ 0CCB		೫ 0CEB	
C	ಌ 0C8C	ಜ 0C9C	ಬ 0CAC	◌಼ 0CBC	◌ೌ 0CCC		೬ 0CEC	
D		ಝ 0C9D	ಭ 0CAD	ಽ 0CBD	◌್ 0CCD		೭ 0CED	
E	ಎ 0C8E	ಞ 0C9E	ಮ 0CAE	◌ಾ 0CBE		ೞ 0CDE	೮ 0CEE	
F	ಏ 0C8F	ಟ 0C9F	ಯ 0CAF	◌ಿ 0CBF			೯ 0CEF	

11th Computer Science_Chapter-2 Telugu.indd 28 30-03-2023 18:02:10

(అపెండిక్స్) అనుసంధానం

సమాచారం మార్పిడి కోసం అమెరికన్ ప్రమాణిక కోడ్

American Standard Code for Information Interchange (ASCII)

(కొన్ని నిర్ధిష్ట అక్షరాలు మాత్రమే)

ఆంగ్ల వర్ణమాల (Alphabets)

ఆంగ్ల వర్ణమాలలు	దశాంశ సంఖ్యలు	ద్వి సంఖ్యా సంఖ్యలు (8 బిట్)	అష్టాంశ సంఖ్యలు	ద్వి అష్టాంశ సంఖ్యలు
A	65	01000001	101	41
B	66	01000010	102	42
C	67	01000011	103	43
D	68	01000100	104	44
E	69	01000101	105	45
F	70	01000110	106	46
G	71	01000111	107	47
H	72	01001000	110	48
I	73	01001001	111	49
J	74	01001010	112	4A
K	75	01001011	113	4B
L	76	01001100	114	4C
M	77	01001101	115	4D
N	78	01001110	116	4E
O	79	01001111	117	4F
P	80	01010000	120	50
Q	81	01010001	121	51
R	82	01010010	122	52
S	83	01010011	123	53
T	84	01010100	124	54
U	85	01010101	125	55
V	86	01010110	126	56
W	87	01010111	127	57
X	88	01011000	130	58
Y	89	01011001	131	59
Z	90	01011010	132	5A
a	97	01100001	141	61
b	98	01100010	142	62
c	99	01100011	143	63
d	100	01100100	144	64

11th Computer Science_Chapter-2 Telugu.indd 29 30-03-2023 18:02:10

e	101	01100101	145	65
f	102	01100110	146	66
g	103	01100111	147	67
h	104	01101000	150	68
i	105	01101001	151	69
j	106	01101010	152	6A
k	107	01101011	153	6B
l	108	01101100	154	6C
m	109	01101101	155	6D
n	110	01101110	156	6E
o	111	01101111	157	6F
p	112	01110000	160	70
q	113	01110001	161	71
r	114	01110010	162	72
s	115	01110011	163	73
t	116	01110100	164	74
u	117	01110101	165	75
v	118	01110110	166	76
w	119	01110111	167	77
x	120	01111000	170	78
y	121	01111001	171	79
z	122	01111010	172	7A

సంఖ్యలు (Numerals)

ఆంగ్ల వర్ణ మాలలు	దశాంశ సంఖ్యలు	ద్వి సంఖ్యా సంఖ్యలు (8 బిట్)	అష్టాంశ సంఖ్యలు	ద్వి అష్టాంశ సంఖ్యలు
0	48	00110000	60	30
1	49	00110001	61	31
2	50	00110010	62	32
3	51	00110011	63	33
4	52	00110100	64	34
5	53	00110101	65	35
6	54	00110110	66	36
7	55	00110111	67	37
8	56	00111000	70	38
9	57	00111001	71	39

11th Computer Science_Chapter-2 Telugu.indd 30 30-03-2023 18:02:10

ప్రత్యేక చిహ్నాలు (Special Characters)

ఆంగ్ల వర్ణ మాల	దశాంశ సంఖ్యలు	ద్వి సంఖ్యా సంఖ్యలు (8 బిట్)	అష్టాంశ సంఖ్యలు	ద్వి అష్టాంశ సంఖ్యలు
Blank	32	00100000	40	20
!	33	00100001	41	21
"	34	00100010	42	22
#	35	00100011	43	23
$	36	00100100	44	24
%	37	00100101	45	25
&	38	00100110	46	26
'	39	00100111	47	27
(	40	00101000	50	28
)	41	00101001	51	29
*	42	00101010	52	2A
+	43	00101011	53	2B
,	44	00101100	54	2C
-	45	00101101	55	2D
.	46	00101110	56	2E
/	47	00101111	57	2F
:	58	00111010	72	3A
;	59	00111011	73	3B
<	60	00111100	74	3C
=	61	00111101	75	3D
>	62	00111110	76	3E
?	63	00111111	77	3F
@	64	01000000	100	40
[	91	01011011	133	5B
\	92	01011100	134	5C
]	93	01011101	135	5D
^	94	01011110	136	5E
_	95	01011111	137	5F
`	96	01100000	140	60
{	123	01111011	173	7B
|	124	01111100	174	7C
}	125	01111101	175	7D
~	126	01111110	176	7E

11th Computer Science_Chapter-2 Telugu.indd 31 30-03-2023 18:02:10

మూల్యాంకనం

UB5MI

విభాగం - అ

సరియైన సమాధానమును ఎన్నుకొనుము

1. కంప్యూటర్ యొక్క CPU చే ప్రాసెస్ చేయబడిన బిట్లు సంఖ్యను ఇది సూచించును.
   (అ) బైట్ (ఆ) నిబ్బెల్ (ఇ) పద పొడవు (ఈ) బిట్
2. ఒక కిలోబైట్ అనునది ఎన్ని బైట్‌లను కలిగి ఉండును?
   (అ) 1000 (ఆ) 8 (ఇ) 4 (ఈ) 1024
3. ASCII విస్తరణ వ్రాయుము
   (అ) American School Code for Information Interchange
   (ఆ) American Standard Code for Information Interchange
   (ఇ) All Standard Code for Information Interchange
   (ఈ) American Society Code for Information Interchange
4. 2^50 అనునది దేనిని సూచిస్తుంది?
   (అ) కిలో (Kilo) (ఆ) తేరా (Tera) (ఇ) బిటా (Peta) (ఈ) జీట్టా (Zetta)
5. Binary ద్వి సంఖ్యా కోడెడ్ దశాంశ వ్యవస్థలో ఎన్ని అక్షరాలను నిర్వహణ చేయవచ్చును?
   (అ) 64 (ఆ) 255 (ఇ) 256 (ఈ) 128
6. 1101_2 నకు సమానమైన ద్వి అష్టాంశ దశాంశము ఏది?
   (అ) F (ఆ) E (ఇ) D (ఈ) B
7. 00100110 నకు 1 యొక్క పూరక ఏమిటి?
   (అ) 00100110 (ఆ) 11011001 (ఇ) 11010001 (ఈ) 00101001
8. క్రింది వాటిలో ఏది అష్టాంశ సంఖ్య కాదు?
   (అ) 645 (ఆ) 234 (ఇ) 876 (ఈ) 123

విభాగం - ఆ

లఘు సమాధానములిమ్ము

1. డేటా అనగానేమి?
2. 1 - యొక్క పూరక విధానము వ్రాయుము.
3. $(46)_{10}$ నకు సమానమైన ద్వి సంఖ్యామానంలో మార్చుము.
4. $(28)_{10}$. కు 1 యొక్క పూరక సంఖ్య వీలుకానిది. కారణము తెలుపుము.
5. మెమరీలోని అక్షరాలను సూచించే ఎన్‌కోడింగ్ వ్యవస్థలను జాబితా చేయండి.

విభాగము - ఇ

సంక్షిప్త సమాధానము వ్రాయుము

1. సంఖ్యా వ్యవస్థలో (radix) ఆధారము అనగానేమి? ఉదాహరణమునిమ్ము.
2. ద్వి సంఖ్యా వ్యవస్థ గురించి ఒక సంపుటిక వ్రాయుము.
3. $(150)_{10}$ ను ద్వి సంఖ్యామానంలో మార్చుము తర్వాత ద్వి సంఖ్యామానం సంఖ్య అష్టాంశములోకి మార్చుము.
4. ISCII గురించి ఒక సంపుటి వ్రాయుము.
5. కూడుము (అ) $-22_{10}+15_{10}$ (ఆ) $20_{10}+25_{10}$

విభాగము - ఈ

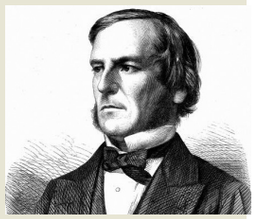
విపులముగా సమాధానములిమ్ము

1. అ) దశాంశ భిన్నమును ద్వి సంఖ్యామానములోనికి మార్చు పద్ధతి గురించి వివరింపుము.
   ఆ) $(98.46)_{10}$ ను ద్వి సంఖ్యామానములోనికి మార్చుము
2. క్రింద ఇవ్వబడిన దశాంశ సంఖ్యలకు 1 యొక్క పూరకము మరియు 2 యొక్క పూరకము కనుగొనుము.
   అ) -98 ఆ) -135
3. అ) కూడుము $1101010_2+101101_2$ ఆ) తీసివేయుము $1101011_2 - 111010_2$

11th Computer Science_Chapter-2 Telugu.indd 32 30-03-2023 18:02:10

2.8 పరిచయము

బూలియన్ ఆల్‌జీబ్రా అనేది డిజిటల్ కంప్యూటర్‌లో డిజిటల్ సర్క్యూట్‌లను రూపొందించడానికి ఉపయోగించే గణితశాస్త్ర విభాగం. ఇది డిజిటల్ సర్క్యూట్ యొక్క ఇన్‌పుట్ మరియు అవుట్‌పుట్‌ల మధ్య సంబంధాన్ని వివరిస్తుంది. ఈ బీజగణితంకు యొక్క ప్రాథమిక సూత్రాలను ప్రతిపాదించిన ఆంగ్ల గణిత శాస్త్రజ్ఞుడు జార్జ్ బూలే గౌరవార్థం బూలియన్ బీజగణితం అను పేరు పెట్టబడింది.



మీకు తెలుసా?

జార్జ్ బూలే (1815 – 1864) దిగువ తరగతి కుటుంబంలో జన్మించినాడు. అతను ప్రాథమిక విద్యను మాత్రమే పూర్తిచేసాడు. అయినప్పటికి అతను తన యుక్త వయస్సులో ఎటువంటి సహాయం లేకుండా అత్యంత అధునాతన గణితమును మరియు వివిధ భాషలను జోడించారు. అతను 16 ఏండ్ల వయస్సులో బోధించడం ప్రతిపాదించాడు మరియు 19 సం।7 వయస్సులో తన సొంత పాఠశాలను ప్రారంభించాడు. అతని 20 ల మధ్య నాటికి తన కాలంలోని చాలా ముఖ్యమైన గణిత సూత్రాలపై పట్టు సాధించాడు.

2.8.1 ద్వి సంఖ్య విలువల పరిమాణాలు:

ప్రతిరోజు మనం తార్కిక నిర్ణయాలు తీసుకోవాలి.

1. నేను ప్రతిరోజు కంప్యూటర్ సైన్స్ పుస్తకము పాఠశాలకు తీసుకెళ్ళలా? అవును / కాదు
2. 8-10 = 10 అనునది సరైన సమాధానమా? అవును / కాదు
3. భారత రాజధాని చెన్నై. అవును / కాదు
4. నేను నిన్న ఏమి చేప్పాను?

పైన చెప్పిన మూడు ప్రశ్నలకు సమాధానము ఒప్పు (అవును) లేదా తప్పు (కాదు) కావచ్చు. కాని నాల్గవ ప్రశ్నకు మనము ఒప్పు లేదా తప్పు అని చెప్పలేము. అందువల్ల ఒప్పు లేదా తప్పు అని నిర్ణయించగల వాక్యాలను “Logical Statement” లేదా “Truth Functions” అని పిలుస్తారు. ఒప్ప లేదా తప్పు ఫలితాలను నిజ విలువలు అని పిలుస్తారు. నిజ విలువలు తార్కిక స్థిరాంకము 1 మరియు 0 మధ్య చిత్రీకరించబడినది. 1 అనగా ఒప్పు 0 అనగా తప్పు ఈ సత్య విలువలను నిల్వ చేయగల వేరియబుల్‌ను ‘లాజికల్ వేరియబుల్’ లేదా ‘బైనరీ నాల్యూడ్ వేరియబుల్స్’ లేదా ‘బూలియన్ వేరియబుల్స్’ అని పిలుస్తారు. ఎందుకంటే ఇది ఒప్పు లేదా తప్పు యొక్క రెండు విలువలలో ఒకదాన్ని నిల్వచేయగలవు.

2.8.2 లాజికల్ ఆపరేషన్స్

బూలియన్ బీజగణితము వేరియబుల్స్ మరియు ఆపరేషన్లను (ఫంక్షన్లు) ఉపయోగించుకొంటుంది. ప్రాథమిక తార్కిక కార్యకలాపాలు AND, OR మరియు NOT వీటిని ప్రతీకగా డాట్ (.) ప్లస్ (+) మరియు ఓవర్‌బోర్ / సింగిల్ అపోస్ట్రోఫి ద్వారా సూచిస్తారు. ఈ చిహ్నాలను లాజికల్ ఆపరేటర్లు అని కూడా పిలుస్తారు.

2.8.3 సత్య పట్టిక (Truth Table):

truth table అనేది లాజిక్ వేరియబుల్ లేదా స్టేట్‌మెంట్ యొక్క సాధ్యమయ్యే అన్ని విలువలను, అలాగే truth విలువల యొక్క ఇచ్చిన కలయిక యొక్క అన్ని సాధ్యమైన ఫలితాలతో పాటుగా సూచిస్తుంది.

2.8.4 AND ఆపరేటర్

AND ఆపరేటర్ బూలియన్ బీజగణితంలో డాట్ (.) ఆపరేటర్ ద్వారా నిర్వచించబడింది. ఇది సాధారణ బీజగణితంలో గుణకారం మాదిరిగానే ఉంటుంది. AND ఆపరేటర్ రెండు లేదా అంతకంటే ఎక్కువ ఇన్పుట్ చరరాశులను మిళితం చేస్తుంది. తద్వారా ఇన్‌పుట్‌లు నిజమైతే మాత్రమే అవుట్‌పుట్‌లు నిజమవుతుంది. 2-ఇన్పుట్ మరియు ఆపరేటర్ కోసం truth table ఈ క్రింది విధంగా చూపబడింది.

11th Computer Science_Chapter-2 Telugu.indd 33 30-03-2023 18:02:11

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

పై 2-ఇన్‌పుట్ AND ఆపరేషన్ ఇలా వ్యక్తీకరించబడింది. Y = A . B

2.8.5 OR ఆపరేటర్



OR ఆపరేటర్‌ను సూచించడానికి కూడిక గుర్తు (+) ఉపయోగించబడుతుంది OR ఆపరేటర్ రెండు లేదా అంతకంటే ఎక్కువ ఇన్‌పుట్ వేరియబుల్స్ (చరరాశులు) మిళితం చేస్తుంది. తద్వారా కనీసం ఒక నిజమైతే అవుట్‌పుట్ నిజం. 2-ఇన్‌పుట్ OR ఆపరేటర్ సత్య పట్టిక క్రింది విధంగా చూపబడినది.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

పై 2-ఇన్‌పుట్ OR ఆపరేషన్‌ను ఇలా వ్యక్తీకరించబడినది Y = A + B.

2.8.6 NOT ఆపరేటర్

NOT ఆపరేటర్‌కు ఒక ఇన్‌పుట్ మరియు ఒక అవుట్‌పుట్ ఉంది. ఇన్పుట్ నిజం లేదా తప్పుగా ఉండును. అవుట్‌పుట్ ఎల్లప్పుడు వ్యతిరేకంగా ఉంటుంది అనగా NOT ఆపరేటర్ ఇన్పుట్లను విలోమం చేస్తుంది. A అనునది ఇన్‌పుట్ వేరియబుల్ మరియు Y అనునది అవుట్‌పుట్ అయిన NOT ఆపరేటర్ కోసం సత్యపట్టిక క్రింది విధంగా చూపబడినది.

A	Y
0	1
1	0

NOT ఆపరేటర్ బీజగణిత ప్రాతినిధ్యం వహించే బూలియన్ వ్యక్తీకరణ Y = A

ఉదాహరణ:

బూలియన్ సమీకరణాన్ని పరిగణించండి.

D = A + (B . C)

A అనునది 1 (సత్యము), లేదా (B. C) అనునది 1, అయితే D అనునది (1) కి సమానం B = 0 మరియు C = 1.

అనగా D అనునది 0 కి సమానం (అసత్యము).

ప్రాథమిక తార్కిక ప్రమేయాలైన AND, OR మరియు NOT వంటి ఇతర తార్కిక ఆపరేటర్లు అనగా NAND మరియు NOR

2.8.7 NAND ఆపరేటర్

NAND ఆపరేటర్ అనునది NOT మరియు AND ల కలయిక AND ఆపరేటర్ యొక్క అవుట్‌పుట్‌ను విలోమం చేయడం ద్వారా NAND ఉత్పత్తి అవుతుంది. NAND ఫంక్షన్ యొక్క బీజగణిత వ్యక్తీకరణం

$Y = \overline{A . B}$

NAND ఫంక్షన్ యొక్క సత్య పట్టిక క్రింది విధంగా చూపబడింది

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

A NAND B = NOT (A AND B)

2.8.8 NOR ఆపరేటర్

NOR అనేది NOR మరియు OR ల కలయిక OR ఆపరేటర్ యొక్క అవుట్‌పుట్‌ను విలోమం చేయడం ద్వారా NOR ఉత్పత్తి అవుతుంది. NOR ప్రమేయాల యొక్క బీజగణిత వ్యక్తీకరణ

$Y = \overline{A . B}$

NOR ప్రమేయం యొక్క సత్య పట్టిక క్రింద చూపబడినది.

11th Computer Science_Chapter-2 Telugu.indd 34 30-03-2023 18:02:11

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

A NOR B = NOT (A OR B)

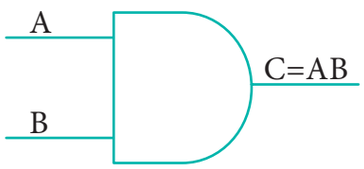
2.9 ప్రాథమిక లాజిక్ గేట్లు (Basic Logic Gates):

గేట్ అనేది ఒక ప్రాథమిక ఎలక్ట్రానిక్ సర్క్యూట్. ఇది అవుట్‌పుట్ సిగ్నల్‌ను ఉత్పత్తి చేయడానికి ఒకటి లేదా అంతకంటే ఎక్కువ సిగ్నల్స్ పై పనిచేస్తుంది. AND, OR మరియు NOT అనే మూడు ప్రాథమిక గేట్‌లు ఉన్నాయి. NAND, NOR, XOR మరియు XNOR వంటి ఇతర లాజిక్ గేట్లు ప్రాథమిక గేట్ల నుండి తీసుకోబడిన గేట్లు. NAND మరియు NOR గేట్లను యూనివర్సల్ గేట్స్ అని పిలుస్తారు. ఎందుకంటే వాటి ద్వారా ప్రాథమిక లాజిక్ గేట్లను గ్రహించవచ్చు.

2.9.1 AND గేట్

AND గేట్ రెండు లేదా అంతకంటే ఎక్కువ ఇన్పుట్ సిగ్నల్స్ కలిగి ఉంటుంది మరియు అవుట్‌పుట్ సిగ్నల్‌ను ఉత్పత్తి చేస్తుంది.

రెండు ఇన్పుట్లు "true" అయినప్పుడు మాత్రమే అవుట్‌పుట్ నిజము లేకపోతే అవుట్‌పుట్ తప్పు మరో మాటలో రెండు ఇన్‌పుట్‌లు 1 అయితే అవుట్‌పుట్ 1 అవుతుంది. లేకపోతే **అవుట్‌పుట్ 0 AND** గేట్ యొక్క అవుట్‌పుట్ వేరియబుల్ C ద్వారా సూచించబడుతుంది. ఇక్కడ A మరియు B రెండు బూలియన్ వేరియబుల్స్ బూలియన్ బీజగణితములో ఒకే వేరియబుల్ విలువ 0 లేదా 1 విలువలలో దేనినైనా తీసుకొనవచ్చు. AND గేట్ యొక్క లాజికల్ చిహ్నము.



పటము 2.4 *AND* గేట్ యొక్క తార్కిక చిహ్నము.

చర్యకు ప్రతికగా AND గేట్ యొక్క బూలియన్ ఫంక్షన్ రాయడం ద్వారా ఉంటుంది.

C = A AND B

బూలియన్ బీజగణితంలో గుణకారము AND ఆపరేషన్ కోసం నిలుస్తుంది. కాబట్టి AND గేట్ యొక్క అవుట్‌పుట్

C = A . B లేదా

కేవలం C = AB

'C అనునది A మరియు B లకు సమానము' అని దీనిని చదవగా దీని నుండి రెండు ఇన్‌పుట్ వేరియబుల్స్ ఉన్నందున సత్య పట్టికకు నాలుగు ENTRY లు ఉంటాయి. ఎందుకంటే ఇన్‌పుట్‌లు సాధ్యమే ఎందుకనగా 00, 01, 10 మరియు 11 ఉదాహరణముగా రెండు ఇన్పుట్ అయినచో

C = A . B
= 0 . 0
= 0

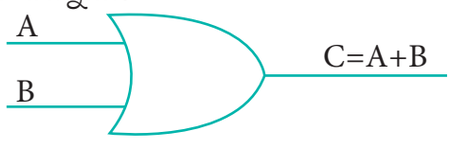
అండ్ AND గేట్ యొక్క సత్య పట్టిక

ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

పట్టిక 2.7 అండ్ *AND* గేట్ యొక్క సత్య పట్టిక

2.9.2 OR గేట్

OR గేట్ లాజికల్ ఇన్‌క్లూసివ్ OR వంటి ప్రవర్తన నుండి దాని పేరును పొందినది. ఇవ్వబడిన రెండు ఇన్‌పుట్‌లలో ఏదైనా ఒకటి లేక రెండు ఇన్‌పుట్‌లు సత్యముగా ఉన్నప్పుడు మాత్రమే అవుట్‌పుట్ నిజమగును. రెండు ఇన్‌పుట్‌లు అసత్యమయిన అవుట్‌పుట్‌లు అసత్యమగును. ఇంకో విధముగా చెప్పాలంటే ఒకటి లేక రెండు ఇన్‌పుట్‌లు 1 అయితే మాత్రమే అవుట్‌పుట్ 1 అవుతుంది. లేకపోతే అవుట్‌పుట్ 0 ఇదియే OR గేట్ యొక్క తార్కిక చిహ్నము.



పటము 2.5 *OR* గేట్ యొక్క తార్కిక చిహ్నము

11th Computer Science_Chapter-2 Telugu.indd 35 30-03-2023 18:02:11

OR గేట్ అవుట్‌పుట్ అనునది

$$C = A \text{ OR } B$$

OR ప్రమేయం సూచించుటకు కూడిక గుర్తు (+) ఉపయోగిస్తారు. కావున

$$C = A + B$$

పై వివరణను C అనునది A OR B కి సమానము అని చదవవలెను.

ఉదాహరణకు రెండు ఇన్‌పుట్‌లు ఒకటి అయిన

$$C = A + B = 1 + 1 = 1$$

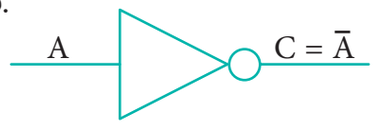
OR గేట్ యొక్క సత్య పట్టిక

ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

పట్టిక 2.8 *OR* గేట్ యొక్క సత్య పట్టిక

2.9.3 NOT గేట్

NOT గేట్ అనగా తార్కిక విలోమము అని పిలువబడే ఒకే ఒక ఇన్‌పుట్ ఉండును. ఇది తార్కిక స్థితిని తారుమారు చేయును. ఇంకో విధంగా చెప్పాలంటే అవుట్‌పుట్ C ఇన్‌పుట్ యొక్క పూరకముగా ఉండును. NOT గేట్ తార్కిక చిహ్నము.



పటము 2.6 *NOT* యొక్క తార్కిక చిహ్నము.

NOT గేట్ యొక్క బూలియన్ ప్రమేయము

$$C = \text{NOT } A$$

బూలియన్ బీజగణితములో పై గీత overbar అనునది NOT గేట్ పనిని తెలుపును. కావున

$$C = \overline{A}$$

దీనిలో C అనునది NOT A కు సమానము లేదా C అనునది A యొక్క పూరకమునకు సమానము అని చదువవలెను.

A యొక్క విలువ '0' అయితే

$$C = \overline{0} = 1$$

ఇంకో విధంగా చెప్పాలంటే A విలువ 1 అయితే

$$C = \overline{1} = 0$$

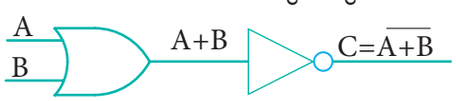
NOT గేట్ యొక్క సత్య పట్టిక

ఇన్‌పుట్	అవుట్‌పుట్
A	C
1	0
0	1

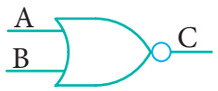
పట్టిక 2.9 *NOT* గేట్ సత్య పట్టిక

2.9.4 NOR గేట్

NOR గేట్ వలయము ఒక OR గేట్‌ను అనుసరించే ఒక విలోమ గేటు (NOT గేటు)ను బట్టి ఏర్పడును. ఇవ్వబడిన రెండు ఇన్‌పుట్‌లు అసత్యము అయిన మాత్రమే ఈ గేట్లు సత్యము అనే అవుట్‌పుట్‌లు ఇచ్చును. లేనిచో అసత్య అవుట్‌పుట్‌లు ఇచ్చును. మరొక విధంగా చెప్పాలంటే 1 అనే అవుట్‌పుట్‌గా పొందుటకు రెండు ఇన్‌పుట్‌ల విలువ 0 గా ఉండవలెను లేకపోతే దాని అవుట్‌పుట్‌లు 0 అవుతుంది. NOR గేటు యొక్క తర్క వలయము



పటము 2.7 *NOR* గేటు యొక్క తర్క వలయము



పటము 2.8 *NOR* యొక్క తర్క చిహ్నము

NOR గేటు యొక్క అవుట్‌పుట్

$$C = (\overline{A + B})$$

దీనిలో C అనేది NOT యొక్క A OR B లేదా C అనేది పూరకము .A OR B. చెప్పవచ్చు. ఉదాహరణకు రెండు ఇన్‌పుట్‌లు 0 అయిన

$$C = (\overline{0 + 0}) = \overline{0} = 1$$

NOR గేటు యొక్క సత్య పట్టిక

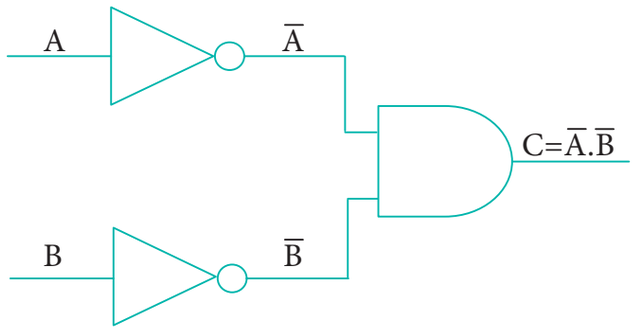
ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

పట్టిక 2.10 *NOR* గేటు యొక్క సత్య పట్టిక

11th Computer Science_Chapter-2 Telugu.indd 36 30-03-2023 18:02:11

2.9.5 బబుల్డ్ AND గేటు

బబుల్డ్ AND గేటు యొక్క తర్క వలయము

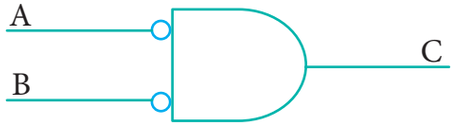


పటము 2.9 బబుల్డ్ *AND* గేటు యొక్క తర్క వలయము

పైన ఇవ్వబడిన వలయములో ఇన్‌పుట్ విలువలను తలక్రిందులుగా మార్చి AND గేటులో తీసుకొను అవుట్‌పుట్

$C = (\overline{A} . \overline{B})$ గా ఇచ్చును.

ఇన్‌పుట్‌లకు బబుల్డ్ వేయడం ద్వారా ఈ వలయమును క్రింద ఇవ్వబడిన విధంగా మార్చి గీయవచ్చును. దీనిలో బబుల్డ్‌లు తలక్రిందులుగా సూచించును.



పటము 2.10 బబుల్డ్ *AND* గేటు యొక్క లాజిక్ చిహ్నము

దీనిని బబుల్డ్ AND గేటు అని సూచిస్తాము. లాజిక్ వలయము యొక్క అన్ని ఇన్‌పుట్ సాధ్యాలను విశ్లేషిస్తాయి.

A = 0 మరియు B = 0 అయిన $C=(\bar{0}.\bar{0}) = 1.1 = 1$

A = 0 మరియు B = 1 అయిన $C=(\bar{0}.\bar{1}) = 1.0 = 0$

A = 1 మరియు B = 0 అయిన $C=(\bar{1}.\bar{0}) = 0.1 = 0$

A = 1 మరియు B = 1 అయిన $C=(\bar{1}.\bar{1}) = 0.0 = 0$

సత్య పట్టిక

ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

బబుల్డ్ AND గేటు యొక్క అవుట్‌పుట్ NOR గేటు యొక్క అవుట్‌పుట్ ఒకే విధంగా ఉండును. NOR గేటును బబుల్డ్ AND గేటు ద్వారా సూచించవచ్చును. ఒకదానికొకటి మార్చి అమర్చవచ్చును.

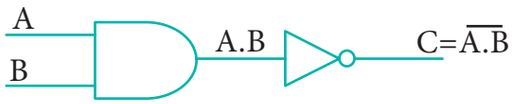
కావున

$$\overline{(A + B)} = \overline{A} . \overline{B}$$

ఈ సమీకరణము డీ మోర్గాన్ యొక్క మొదటి సిద్ధాంతమగును.

2.9.6 NAND గేటు

AND గేటు యొక్క అవుట్‌పుట్ NOT గేటుకు పంపించి పొందే అవుట్‌పుట్‌ను NAND గేటు నిర్వహించును. దీని తర్క కృత్యము AND గేటు యొక్క అవుట్‌పుట్ తలక్రిందులుగా పొందు ఇన్‌పుట్‌లు అన్ని సత్యము అయిన దాని అవుట్‌పుట్ అసత్యముగా నుండును. లేనిచో దాని అవుట్‌పుట్ సత్యముగా నుండును. దానికి బదులుగా ఇన్‌పుట్‌లు అన్ని 1 అయిన NAND గేటు యొక్క అవుట్‌పుట్ 0 గా ఉండును లేనిచో దీని అవుట్‌పుట్ 1 గా ఉండును. NAND గేటు యొక్క తర్క వలయము



పటము 2.11 *NAND* గేటు యొక్క లాజిక్ వలయం NAND గేటు యొక్క లాజిక్ సూచిక క్రింద ఇవ్వబడినది.



పటము 2.12 *NAND* గేటు యొక్క లాజిక్ సూచిక

NAND గేటు యొక్క అవుట్‌పుట్

$$C = \overline{(A . B)}$$

"C" అనునది NOT యొక్క A AND B" కి సమానం లేదా "C" అనేది పూరకము A AND B కి సమానం అని చదవండి.

ఉదాహరణకు రెండింటి ఇన్‌పుట్‌లు 1 అయితే

$$C = \overline{(1 . 1)} = \overline{1} = 0$$

11th Computer Science_Chapter-2 Telugu.indd 37 30-03-2023 18:02:11

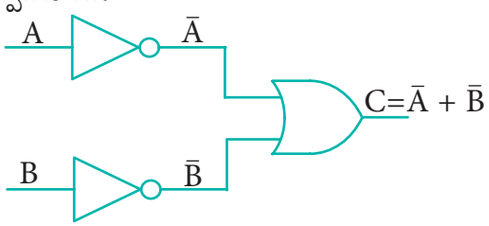
NAND గేట్ యొక్క సత్య పట్టిక

ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

పట్టిక 2.11 *NAND* గేట్ సత్య పట్టిక

2.9.7 బబుల్డ్ OR గేట్

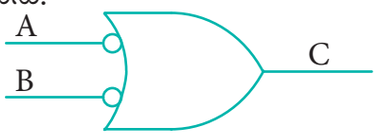
బబుల్డ్ OR గేటు యొక్క లాజిక్ వలయము ఇవ్వబడినది.



పటము 2.13 బబుల్డ్ *OR* గేటు యొక్క లాజిక్ వలయము

ఈ వలయము యొక్క అవుట్‌పుట్ $C = \bar{A} + \bar{B}$ అని వ్రాయవచ్చును.

పై సర్క్యూట్‌ని ఇన్‌పుట్‌లోని బబుల్డ్‌గా మరలా గీయవచ్చును. ఇందులో బబుల్స్ విలోమంను సూచించును.



పటము 2.14 బబుల్డ్ *OR* గేటు యొక్క లాజిక్ చిహ్నము

మేము దీనిని బబుల్డ్ OR గేటు అని సూచిస్తాము. బబుల్డ్ చేయబడిన OR కోసం సత్య పట్టిక

ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

పట్టిక 2.12 బబుల్డ్ *OR* గేట్ యొక్క సత్య పట్టిక

బబుల్డ్స్ OR గేటు మరియు NAND గేట్ యొక్క నిజమైన జాబితాల పోల్చినపుడు రెండు ఒకే లాగ ఉంటాయి. కావున రెండు సర్క్యూట్‌లు ఒకదాని నుండి మరొక దానికి మార్చవచ్చును.

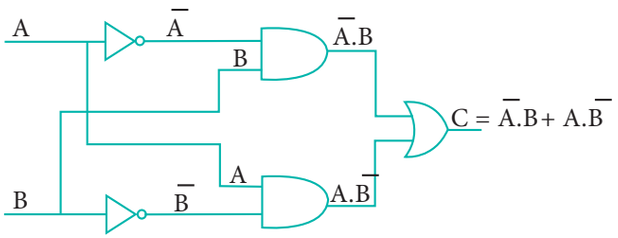
కావున,

$$\overline{(A \cdot B)} = \bar{A} + \bar{B}$$

ఇది డీ మోర్గన్ (De Morgan) రెండవ సిద్ధాంతంను స్థాపిస్తుంది.

2.9.8 XOR గేట్

XOR గేట్ అనునది "either/or" లాగానే పనిచేయును. ఈ రెండింటి ఔట్‌పుట్ "true" అయితే ఇన్‌పుట్‌లు "true" కావు. ఔట్‌పుట్ "false" అయినపుడు, రెండు ఇన్‌పుట్‌లు "false" అగును లేదా రెండు ఇన్‌పుట్‌లు "true" అగును. ఇవ్వబడిన సర్క్యూట్‌ను పరిశీలిస్తే, రెండు ఇన్‌పుట్‌లు వేర్వేరుగా ఉన్నప్పుడు, ఔట్‌పుట్ "1". అయితే, ఇన్‌పుట్‌లు ఒకురకంగా ఉన్నప్పుడు ఔట్‌పుట్ విలువ "0". XOR గేట్ యొక్క లాజిక్ సర్క్యూట్ క్రింది విధంగా ఉండును.



పటము 2.15 *XOR* గేట్ యొక్క తర్క వలయము

XOR గేట్ యొక్క అవుట్‌పుట్

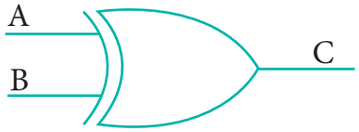
ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

పట్టిక 2.13 *XOR* గేటు యొక్క సత్య పట్టిక

exclusive - OR కృత్యము, బులియన్ గణితములో వృత్తములో గల సంకలన గుర్తు $\oplus$ ద్వారా దీనిని "encircled plus" గా సూచిస్తారు.

$$C = A \oplus B$$

XOR గేటు యొక్క తర్క చిహ్నము క్రింద ఇవ్వబడినది



పటము 2.16 *XOR* గేటు యొక్క తర్క చిహ్నము

11th Computer Science_Chapter-2 Telugu.indd 38 30-03-2023 18:02:11

2.9.9 NOR గేటు

XNOR (exclusive - NOR) గేటు అనేది XOR గేటు నుండి వచ్చు అవుట్‌పుట్‌ను విలోమముగా మార్చి ఇచ్చును. రెండు ఇన్‌పుట్లు సమానంగా ఉన్నప్పుడు దీని అవుట్‌పుట్ సత్యముగా ఉండును లేక ఇన్‌పుట్లు సత్యముగా "true" ఉండును. లేక ఇన్‌పుట్లు వేర్వేరుగా ఉండునప్పుడు అవుట్‌పుట్ అసత్యము "false" గా ఉండును. దాని బదులు ఇన్‌పుట్లు సమానము అయినప్పుడు అవుట్‌పుట్ 1 అవుతుంది. లేకపోతే అవుట్‌పుట్ 0 అగును. XNOR గేటు యొక్క తర్క వలయము అనునది



పటము 2.17 *XNOR* గేటు యొక్క తర్క వలయము

XNOR గేటు యొక్క అవుట్‌పుట్ అనునది NOT యొక్క XOR

$$C = \overline{A \oplus B}$$

$$= \overline{A} \cdot B + A \cdot \overline{B}$$

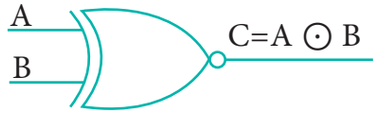
$$= AB + \overline{A}\ \overline{B}$$

(De Morgan's సిద్ధాంతమును ఉపయోగించి)

బూలియన్ గణితంలో $\odot$ అనునది XNOR అగును.

$$C = A \odot B$$

దీనిని తర్క చిహ్నము



పటము 2.18 *XNOR* గేట్ యొక్క తర్క చిహ్నము

XNOR గేట్ యొక్క సత్యపట్టిక క్రింద ఇవ్వబడినది.

ఇన్‌పుట్		అవుట్‌పుట్
A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

పట్టిక 2.14 *XNOR* గేట్ యొక్క సత్య *Truth* పట్టిక

తర్క గేట్‌ల కలయికను ఉపయోగించి సంక్లిష్ట కార్యకలాపాలు చేయవచ్చు. ఈ సిద్ధాంతములో ఒకే పరికరములో కలిసి అమర్చగల గేట్ల సంఖ్యకు పరిమితం లేదు. కావున ఆచరణలో ఇవ్వబడిన భౌతిక స్థలంలో అమర్చబడిన గేట్ల సంఖ్య పరిమితముగా ఉండును. లాజిక్ (తర్క) గేట్లు శ్రేణులు డిజిటల్ ఇంటిగ్రేటెడ్ వలయాల్లో కనిపిస్తాయి.

బులియన్ బీజగణిత సిద్ధాంతములు

తత్సమము

$A + 0 = A$

$A \cdot 1 = A$

పూరకము

$A + \overline{A} = 1$

$A \cdot \overline{A} = 0$

స్థిత్యంతర

$A + B = B + A$

$A \cdot B = B \cdot A$

సహచర

$A + (B + C) = (A + B) + C$

$A \cdot (B \cdot C) = (A \cdot B) \cdot C$

విభాగ

$A \cdot (B + C) = A \cdot B + A \cdot C$

$A + (B \cdot C) = (A + B) \cdot (A + C)$

శూన్య మూలకము

$A + 1 = 1$

$A \cdot 0 = 0$

వ్యత్యయము

$\overline{(\overline{A})} = A$

ఇన్‌పుట్ అవుట్‌పుట్‌లు సమానము

$A + A = A$

$A \cdot A = A$

$A + (A \cdot B) = A$

$A \cdot (A + B) = A$

3వ విభాగ

$A + \overline{A} \cdot B = A + B$

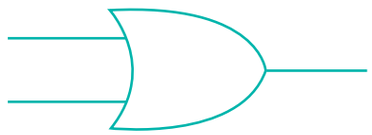
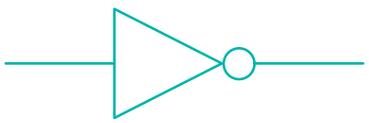
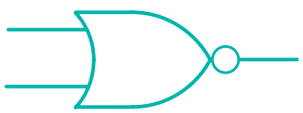
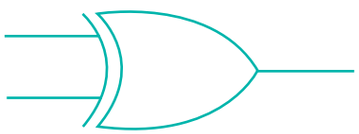
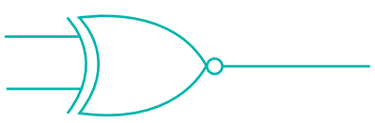
డీమోర్గాన్

$\overline{A + B} = \overline{A} \cdot \overline{B}$

$\overline{(A \cdot B)} = \overline{A} + \overline{B}$

11th Computer Science_Chapter-2 Telugu.indd 39 30-03-2023 18:02:11

పట్టిక 2.15
తర్క గేట్లు యొక్క విభజన సత్య పట్టికలు

తర్క గేట్లు	చిహ్నము	సత్య పట్టిక
AND		A | B | AB 0 | 0 | 0 0 | 1 | 0 1 | 0 | 0 1 | 1 | 1
OR		A | B | A + B 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 1
NOT		A | $\overline{A}$ 0 | 1 1 | 0
NAND		A | B | $\overline{A\,B}$ 0 | 0 | 1 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0
NOR		A | B | $\overline{A + B}$ 0 | 0 | 1 0 | 1 | 0 1 | 0 | 0 1 | 1 | 0
XOR		A | B | $A \oplus B$ 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0
XNOR		A | B | $\overline{A \oplus B}$ 0 | 0 | 1 0 | 1 | 0 1 | 0 | 0 1 | 1 | 1

11th Computer Science_Chapter-2 Telugu.indd 40 30-03-2023 18:02:11

మూల్యాంకనం

విభాగము – అ

సరియైన సమాధానమును ఎన్నుకొనుము

1. ఇందులో ఒకటి లేక అంతకంటే ఎక్కువ సంకేతాలపై పనిచేయు ప్రాథమిక ఎలక్ట్రానిక్ వలయం ఏది?
   (అ) బూలియన్ బీజగణితం (ఆ) గేటు
   (ఇ) ప్రాథమిక గేట్లు (ఈ) ఉత్పన్నమైన గేట్లు
2. ఇందులో ఏ గేటును తర్క విలోమము అని అంటారు?
   (అ) AND (ఆ) OR
   (ఇ) NOT (ఈ) XNOR
3. A + A = ?
   (అ) A (ఆ) O
   (ఇ) 1 (ఈ) A
4. NOR దేని యొక్క కలయిక?
   (అ) NOT(OR) (ఆ) NOT(AND)
   (ఇ) NOT(NOT) (ఈ) NOT(NOR)
5. NAND ను _____ గేటు అని పిలుస్తారు.
   (అ) ప్రాథమిక గేటు (ఆ) ఉత్పన్నమైన గేటు
   (ఇ) తర్క గేటు (ఈ) యూనివర్సల్ గేటు

విభాగము – ఆ

లఘు సమాధానములిమ్ము

1. బూలియన్ బీజగణితం అనగానేమి?
2. NAND గేటు గురించి చిన్న గమనిక వ్రాయుము.
3. XOR గేటు యొక్క సత్య పట్టికను గీయుము.
4. సహచర ధర్మములను వ్రాయుము.
5. ఉత్పన్న గేట్లు అనగానేమి?

విభాగము – ఇ

సంక్షిప్త సమాధానము వ్రాయుము

1. ప్రాథమిక గేటు యొక్క సత్య పట్టికను (truth table) వ్రాయుము.
2. XNOR గేటుపై ఒక చిన్న సూచికను వ్రాయుము.
3. NAND మరియు NOR గేట్లు యూనివర్సల్ గేట్లు అని ఎందుకు పిలువబడును? కారణము తెలుపుము.
4. XOR గేటు యొక్క సత్య పట్టికను (truth table) వ్రాయుము.
5. డీ మోర్గాన్ నియమమును వ్రాయుము.

విభాగము – ఈ

విపులముగా సమాధానములిమ్ము

1. ప్రాథమిక గేట్లును వ్యక్తీకరణలో మరియు సత్య పట్టికను (truth table) ద్వారా వివరింపుము
2. NAND మరియు NOR గేట్లు ఉపయోగించి AND మరియు OR గేట్లును ఎలా విశ్లేషించగలరు?
3. ఉత్పన్న గేటును వ్యక్తీకరణ మరియు సత్య పట్టికను (truth table) ద్వారా వివరింపుము.

11th Computer Science_Chapter-2 Telugu.indd 41 30-03-2023 18:02:11

అధ్యాయము 3

కంప్యూటర్ వ్యవస్థ
(Computer Organisation)

అభ్యసనా లక్ష్యాలు
(Learning Objectives)

- కంప్యూటర్ భాగాలు మరియు వాటి అంతర్గత సంబంధములను తెలుసుకొంటారు.
- ప్రొసెసర్లు మరియు వాటి లక్షణాలు తెలుసుకొనుట.
- మెమరీ పరికరాల యొక్క ప్రాముఖ్యతను కంప్యూటర్‌లో వాటి యొక్క పాత్రలను తెలుసుకొనుట
- RAM, ROM లను అన్వేషించి వాటి మధ్య గల బేధాలను తెలియజేయుట.
- కేచ్ (cache) మెమరీ గురించి తెలుసుకోవడం మరియు కంప్యూటర్ పనితీరును ఎలా మెరుగు పరుచునో తెలుసుకొనుట.
- ద్వితీయ పరికరాల గూర్చి మరియు వాటి వినియోగం (ఉపయోగం) గూర్చి తెలుసుకొనుట
- పోర్టులు మరియు ఇంటర్‌ఫేస్‌లను ఉపయోగించి, బాహ్య పరికాలను ఎలా అను సంధానించబడునని తెలుసుకొనుట.

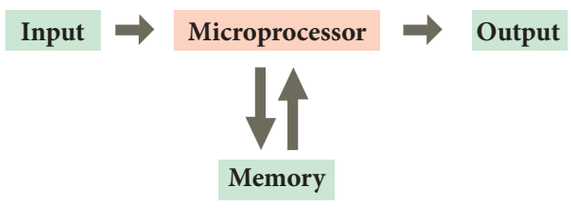
3.1 పరిచయము (Introduction)

కంప్యూటర్ వ్యవస్థ అనునది కంప్యూటర్ హార్డ్‌వేర్ పరికరాలను గూర్చి వివరించునది అగును. ఇందులో ఇన్‌పుట్ / అవుట్‌పుట్ పరికరాలు, కేంద్రీకృత వ్యవస్థ (CPU) నిల్వ ఉంచు పరికరాలు మరియు ప్రాథమిక మెమరీ మొదలగునవి కలవు. ఇది కంప్యూటర్ యొక్క వివిధ రకాల భాగాలు ఎలా పనిచేస్తుదని బట్టి ఉండును. ఇది ఒక కంప్యూటర్ యొక్క అన్ని హార్డువేర్ సాధనాలు ఎలా పనిచేయును అనుదానిని బట్టి పైగా వివిధ రకాల భాగాల యొక్క అనుసంధానాన్ని బట్టి వివరించును. ఆర్కిటెక్చురల్ స్పెసిఫికేషన్ అమలు చేయడానికి అని ఎలా పరిస్పరం అనుసంధానించబడి ఉన్నాయి అననే దానితో కూడా ఇదే వ్యవహరిస్తుంది. కంప్యూటర్ ఆర్గనైజేషన్ అనే పదం కంప్యూటర్ ఆర్కిటెక్చర్ అనే పదాన్ని పోలి ఉంటుంది. కాని కంప్యూటర్ ఆర్కిటెక్చర్ కంప్యూటర్ రూపకల్పనలో ఇంజనీరింగ్ పరిశీలనతో వ్యవహరిస్తుంది (అనుసరిస్తుంది) అదే సమయంలో కంప్యూటర్ వ్యవస్థ అనునది ప్రోగ్రామర్లలకు హార్డువేర్ పరకాలను గూర్చి పారదర్శకముగా వివరించును.

3.2 మైక్రో ప్రొసెసర్ల ప్రాథమికాలు
(Basics of Microprocessors)

కంప్యూటర్ యొక్క ప్రధాన భాగం CPU అగును. ఇది కంప్యూటర్ యొక్క అన్ని పనులు నిర్వహించును. ఇది ఇంటిగ్రేటడ్ సర్క్యూట్ అయిన మైక్రో ప్రొసెసర్లల్యూరా గ్రహించబడును. 1970 సం॥లో మొట్ట మొదట మైక్రో ప్రోససర్లు ప్రవేశపెట్టబడినది. మొదటి మైక్రో ప్రొసెసర్ 4004 ను ఇంటల్ సంస్థ అభివృద్ధి చేసినది.

మైక్రో ప్రొసెసర్ అనునది ప్రోగ్రామబల్ బహుళార్థ సాధన సిలికాన్ చిప్ అగును. ఇది గడియారపు నాడీ స్పందనచే నడుపబడుతుంది. ఇది బైనరీ డేటాను ఇన్‌పుట్‌గా స్వీకరించి ప్రొసెస్ చేసిన తర్వాత మెమరీయందు నిల్వ ఉంచుతుంది. సూచనల ప్రకారం అవుట్‌పుట్‌ను అందిస్తుంది. ప్రాథమిక మైక్రోప్రొసెసర్ వ్యవస్థ పటము 3.11 లో చూపబడినది



పటము 3.1 మైక్రోప్రోసెసర్ ప్రాథమిక వ్యవస్థ

మైక్రో ప్రొసెసర్ 3 ప్రధాన యూనిట్లలో రూపొందించబడినది. అవి వరుసగా:

11th Computer Science_Chapter-3 Telugu.indd 42 30-03-2023 18:06:00

- **అంకగణిత మరియు తార్కిక యూనిట్ (ALU):** కంప్యూటర్ సూచనలు ఆధారంగా అంకగణితం మరియు తార్కిక సూచనలను నిర్వహించడానికి ఉపయోగపడును.

- **నియంత్రణ విభాగము :** కంప్యూటర్ యొక్క మొత్తం కార్యకలాపాల సంకేతాల (సిగ్నల్స్) ద్వారా నియంత్రించబడును.

- **రిజిస్టర్లు (ఇంటర్నల్ మెమరీ) :** ఇది ప్రోసెసర్‌ను అమలు చేయడానికి సూచనలు మరియు డేటాను నిల్వచేయుటకు ఉపయోగపడును.

మైక్రోప్రొసెసర్ యొక్క లక్షణాలు

ఒక మైక్రోప్రొసెసర్ యొక్క క్రియాశీల సామర్థ్యం క్రింద పేర్కొనిన లక్షణాల ఆధారంగా రూపొందును.

అ) గడియార వేగం (Clock speed)

ఆ) సూచనల సమితి (Instruction set)

ఇ) పద పరిమాణం (Word size)

అ) గడియార వేగం (Clock Speed)

ప్రతి మైక్రో ప్రోసెసర్‌లో ఒక **అంతర్గత గడియారం** ఉంటుంది. ఇది వేగాన్ని సూచనలను నియంత్రించి అమలు చేస్తుంది. ఒక మైక్రో ప్రోసెసర్ సూచనలను అమలు చేసే వేగాన్ని **"గడియార వేగం"** అందురు. గడియార వేగాన్ని MHz (మెగా హెర్ట్జ్) లేదా GHz (గిగా హెర్ట్జ్) తో కొలుస్తారు.

వేగం కొలత

మీకు తెలుసా?

Hertz : పౌనపున్యమును కొలుచుటకు ఉపయోగించే ప్రామాణిక కొలత Hz అగును. ఒక సెకనులో చుట్ల సంఖ్యను బట్టి పౌన:పున్యమును కొలుస్తారు. ఒక సెకనులో ఒక చుట్టుకు సమానము ఒక Hz అగును.

ధ్వని తరంగాలు, కాంతి తరంగాలు మరియు రేడియో తరంగాలు వంటి పౌన:పున్యములను కొలవడానికి ఉపయోగపడు ప్రమాణము Hz అగును.

ఉదాహరణకు

మానవుని వినికిడి సుమారు 20 Hz నుండి 20,000 Hz వరకు ధ్వని తరంగములకు కలిగియుంటుంది. 20 Hz దగ్గరగా ఉన్న ధ్వని తరంగాలు తక్కువ పిచ్ కలిగి ఉంటాయి. వాటిని "బ్రస్ ఫ్రీ క్వేన్స్‌లు"" అంటారు. 5,000 Hz కంటే ఎక్కువ ఉన్న ధ్వని తరంగాలు అధిక పిచ్ కలిగి ఉంటాయి. వీటికి ట్రబుల్ పౌన:పుణ్యాలు అంటారు. కాబట్టి తరంగ పౌనఃపుణ్యాలను కొలవడానికి Hz లను ఉపయోగపడును. ఇది కంప్యూటర్ ప్రోసెసింగ్ వేగమునకు కొలవడానికి కూడా ఉపయోగపడుతుంది. ఉదాహరణకు ప్రతి CPU ఒక నిర్ధిష్ట గడియారం వేగంలో గణించబడుతుంది. ఇది ప్రతి సెకనులో ప్రోసెసర్ నిబంధన చక్రాలను చేయగలదో సంఖ్యారూపంలో సూచిస్తుంది. ఆధునిక ప్రోసెసర్లు సెకనుకు మిలియన్లు లేదా బిలియన్లు సూచనలను చేయగలవు. కాబట్టి గడియార వేగం సాధారణంగా మోగాహెర్ట్స్ లేదా జిగా హెర్ట్స్‌లతో కొలువబడును.

ఆ) సూచనల సమితి (Instruction Set)

డేటాపై ఆపరేషన్ చేయడానికి కంప్యూటర్‌కు యివ్వబడే ఆదేశాన్ని "సూచన" గా పిలుస్తారు. మైక్రోప్రోసెసర్‌ని అమలు చేయడానికి రూపొందించబడిన యంత్ర స్థాయి సూచనల ప్రాథమిక సమితిని "సూచన సమితి" అంటారు. ఈ సూచన సమితి క్రింది రకాల కార్యకలాపాలను నిర్వహించును.

- సమాచార బదిలీ (మార్పిడి)
- అంకగణిత కార్యకలాపాలు
- తార్కిక కార్యకలాపాలు
- నియంత్రణ కదలిక (control / flow)
- ఇన్‌పుట్ / అవుట్‌పుట్

ఇ) పద పరిమాణం (Word Size)

- ఒకే సూచనలో ప్రోసెసర్ ద్వారా ప్రోసెస్ చేయగల బిట్ల సంఖ్యను దాని "పద పరిమాణం" అంటారు. మైక్రోప్రోసెసర్ ద్వారా ప్రాప్యత చేయగల RAM మొత్తాన్ని **పద పరిమాణం** నిర్ణయిస్తుంది

11th Computer Science_Chapter-3 Telugu.indd 43 30-03-2023 18:06:00

3.3 CPU మరియు మెమరీ మధ్య డేటా కమ్యూనికేషన్ (Data communication between CPU and memory)

సెంట్రల్ ప్రాసెసింగ్ యూనిట్ (సిపియు) లో మెమరీ డేటా రిజిస్టర్ (MAR) మరియు మెమరీ అడ్రసు రిజిస్టర్ (MDR) లను కలిగి ఉండును. మెమరీకి మరియు CPU కి మధ్య బదిలీ చేయబడిన డేటాలు మెమరీ డేటా నందు పొందపరచబడి ఉండును. **ప్రోగ్రామ్ కౌంటర్** (PC) అనునది CPU లో ప్రత్యేక రిజిస్టర్. ఇది ఎల్లప్పుడూ తదుపరి సూచలను చిరునామాను ఉంచుతుంది. CPU యొక్క అంకగణితం మరియు తార్కిక యూనిట్ మెమరీ చిరునామాను మెమరీ అడ్రసు రిజిస్టర్ నుండి పొందుతుంది.

కంప్యూటర్‌లోని అంతర్గత భాగాల మధ్య కమ్యూనికేషన్ (సమాచారం కోసం) ఉపయోగించి తీగల సమూహమును బస్ అని అందురు.

RAM లోని పదము మెమరీ డేటా రిజిస్టర్ వలె అదే పరిమాణంలో (.. సంఖ్య) కలిగి ఉంటుంది. ఇంటల్ 8085 అనునది ఒక 8 బిట్స్ ప్రాసెసర్. దీని యొక్క MDR, RAM లోని పదము రెండూ 8 బిట్లను కలిగి ఉండును.

రీడ్ ఆపరేషన్ అనునది పదము నుండి డేటాను (bits) ను మెమరీ డేటా రిజిస్టర్‌కు బదిలీచేయును. రైట్ ఆపరేషన్ అనునది డేటా (bits) ను మెమరీ డేటా రిజిస్టర్ నుండి పదమునకు బదిలీ చేయును.

మీకు తెలుసా?

తీగ చివర 5V వర్తించబడితే ఇంకొక (మరొక) తీగ చివరి 5V ని అందుకొంటుంది. అదే విధంగా బస్‌లను వైయర్ (తీగలుగాను) మరియు బైనరీ డేటాలను వోల్టేజ్ (5V ని 1 మరియు 0V as 0) బస్‌లు సరళంగా డేటాను వోల్టేజ్‌గా ఒక చివరి నుండి మరొక చివరకు మార్చగలవు.

3.4 మైక్రో ప్రోసెసర్ల రకాలు (Types of Microprocessors)

మైక్రో ప్రోసెసర్లను ఈ క్రింది అభి లక్షణాల ఆధారంగా వర్గీకరించవచ్చును.

- ప్రోసెస్ చేయగల డేటా యొక్క వెడల్పు
- సూచనల సమితి (instruction set)

3.4.1 మైక్రో ప్రోసెసర్ డేటా వెడల్పు ఆధారంగా వర్గీకరణ (Classification of Microprocessors based on the Data Width)

డేటా వెడల్పు ఆధారంగా మైక్రో ప్రోసెసర్లు సూచనలను అమలు చేయును. మైక్రో ప్రోసెసర్లను ఈ క్రింది విధంగా వర్గీకరించవచ్చును.

- 8– బిట్స్ మైక్రో ప్రోసెసర్
- 16– బిట్స్ మైక్రో ప్రోసెసర్
- 32– బిట్స్ మైక్రో ప్రోసెసర్
- 64– బిట్స్ మైక్రో ప్రోసెసర్

3.4.2 సూచనల సమితి ఆధారంగా మైక్రో ప్రోసెసర్ వర్గీకరణ (Classification of Microprocessors based on Instruction Set)

సూచనల సమితి యొక్క పరిమాణము మైక్రో ప్రోసెసర్ వర్గీకరణలో ముఖ్యముగా పరిశీలింపదగినది. సూచనల సమితి ఆధారంగా మైక్రో ప్రోసెసర్‌ను రెండు విధాలుగా వర్గీకరించిరి.

- రెడ్యూస్డ్ ఇన్‌స్ట్రక్షన్ సెట్ కంప్యూటర్స్ (RISC)
- కాంప్లెక్స్ ఇన్‌స్ట్రక్షన్ సెట్ కంప్యూటర్స్ (CCC)

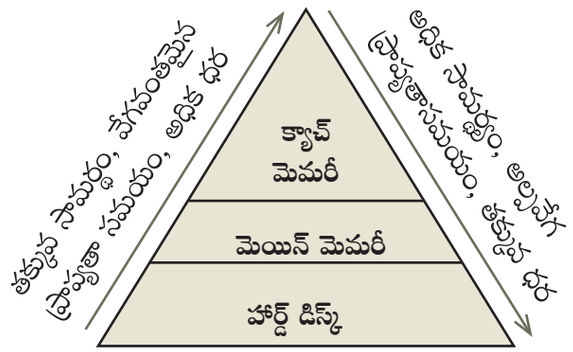
RISC ప్రోసెసర్స్‌కి ఉదాహరణలు పెన్‌టియమ్ IV ఇంటెల్ P6, AMD K6 మరియు K7 CISC ప్రోసెసర్స్‌కి ఉదాహరణలుగా ఇంటెల్ 386 & 486 పెన్‌టియమ్ II & III మరియు మొటరోలా 68000

3.5 మెమరీ పరికరాలు (Memory Devices)

మెమరీ అనేది మానవుని మెదడుతో సమానమైనది. ఇది డేటాను మరియు సూచనలను

11th Computer Science_Chapter-3 Telugu.indd 44 30-03-2023 18:06:00

నిల్వచేయును. కంప్యూటర్ మెమరీ అనేది కంప్యూటర్‌లో డేటాను మరియు సూచనలను నిల్వ చేయు స్థలము. మెమరీలో నిల్వ చేయబడిన డేటా మరియు సూచనలను రెండు విధాలుగా పేర్కొనవచ్చును. (చదవడం, వ్రాయడం) అవి నిరంతర మరియు యాదృచ్ఛిక ప్రాప్యత అగును. నిరంతర ప్రాప్యతలో ప్రారంభం నుండి చివరి వరకు క్రమబద్ధమైన పద్ధతి కనబడుతుంది. కానీ యాదృచ్ఛిక ప్రాప్యతలో మెమరీ యొక్క ఏదైనా బైట్ నావిగేట్ చేయకుండా నేరుగా యాక్సస్ చేయవచ్చును. మునుపటి బైట్ల ద్వారా క్రింది చిత్రములో చూపించిన విధంగా మెమరీ పరికరాలు భిన్న రీతిలో అమర్చబడి వుంటాయి. సామర్థ్యం వేగం మరియు వ్యయం ఆధారంగా క్రమపరచబడి ఉన్నది.



పటము 3.6 మెమరీ సోపాన క్రమం

3.5.1 రాండమ్ – యాక్సస్ మెమరీ (Random-Access Memory (RAM))

ప్రధాన మెమరీని రాండమ్ – యాక్సస్ మెమరీ అని కూడా పిలుస్తారు. ఇది కంప్యూటర్లలో ఇంటిగ్రేటెడ్ సర్క్యూట్స్ (ICs) రూపంలో అందుబాటులో కలదు. కంప్యూటర్లలో దీనిని ఆపరేటింగ్ సిస్టమ్, అప్లికేషన్ ప్రోగ్రామ్ మరియు తాత్కాలిక డేటాలందు పొందుపరచెదరు. దీనిని కంప్యూటర్లో ప్రాసెస్ ద్వారా యాక్సెస్ చేయవచ్చును. మెమరీలో నిల్వ చేయబడిన అతిచిన్న సమాచారాన్ని బిట్ అంటారు మెమరీ ఒకే సమయంలో 8 బిట్ల వరకు యాక్సస్ చేయగలదు. దీన్ని బైట్ అంటారు.

RAM ఒక తాత్కాలిక మెమరీ. దీనిలో నిల్వచేసిన సమాచారం శాశ్వతం కాదని అర్థము. కంప్యూటర్‌కు విద్యుత్ సరఫరా ఆపివేయబడినప్పుడు RAM లో ఉన్న మొత్తం డేటా చెరిపిచేయబడును. ఇది ఆపరేషన్లను చదవడానికి మరియు వ్రాయడానికి అనుమతిస్తుంది.

3.5.2 RAM నందు రకాలు

RAM నందు రెండు ప్రాథమిక రకాలు కలవు

- డైనమిక్ RAM (DRAM)
- స్టాటిక్ RAM (SRAM)

డేటాను పొందుపరచుటకు ఈ రెండు రకాల RAM లు సాంకేతిక దృక్పథంలో వేరుపరచబడును. డైనమిక్ ప్రాథమిక మెమరీను తరచూ నవీనపరచ వలసి ఉంటుంది. కాని స్టాటిక్ ప్రాథమిక మెమరీను తక్కువ సార్లు నవీన పరచవలసి ఉంటుంది. కాబట్టి ఇది వేగంగా పనిచేస్తుంది. స్టాటిక్ RAM, డైనమిక్ RAM కంటే ఖరీదైనది.

3.5.3 రీడ్ ఓన్లీ మెమరీ (Read Only Memory (ROM))

రీడ్ ఓన్లీ మెమరీ అనునది కంప్యూటర్‌లోని ప్రత్యేక మెమరీ. ఇది తయారీ సమయంలో ముందే రికార్డు చేసిన డేటా. దీనిని సవరించుటకు వీలుకాదు. నిల్వచేసిన ప్రోగ్రామ్‌లు కంప్యూటర్ ప్రారంభములో మరియు డయగ్నస్టిక్స్ ROMs లో లభిస్తుంది. క్లిష్టమైన ప్రోగ్రామ్స్‌ని అంటే కంప్యూటర్ బూట్స్ వంటి వాటిని ROM నిల్వ చేసుకుంటుంది. ఒకసారి ROM చిప్‌లో డేటాను పొందుపరచిన దానిని మార్పు చేయుటకు తొలగించుటకు వీలులేదు. కేవలం చదువగలగుతాము. కంప్యూటర్ ఆపివేయబడిననూ ROM లోని విషయాలు యధావిధిగా ఉంటుంది. కాబట్టి ROM ని స్థిర మెమరీగా పిలవబడుచున్నది.

3.5.3.1 ప్రోగ్రామబుల్ రీడ్ ఓన్లీమెమరీ (PROM)

ప్రోగ్రామబుల్ రీడ్ ఓన్లీ మెమరీ కూడా అస్థిరత కలిగిన మెమరీ ఒకసారి మాత్రమే దీనిపై కూడా డేటాను నమోదు చేయవచ్చు. ఇది ఎప్పటికి నిలిచి ఉంటుంది. ప్రధాన మెమరీలాగ కాకుండా కంప్యూటర్ ఆపివేసినపుడు PROM లో వాటి విషయాలు అలానే నిలిచి ఉంటుంది. ROM కి భిన్నంగా వ్యవహరిస్తుంది PROM

PROM, ROM తో పోల్చినచో వైవిధ్యంగా ఉండును. ఇది ఒక ఖాళీ మెమరీగా తయారవుతుంది. కానీ తయారీ ప్రక్రియలో ROM ప్రోగ్రామ్ చేయబడును. PROM చిప్‌లో డేటాను

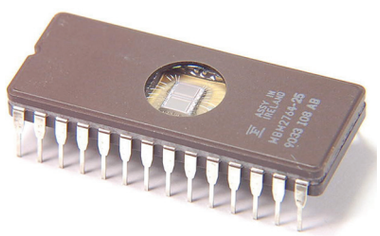
11th Computer Science_Chapter-3 Telugu.indd 45 30-03-2023 18:06:00

వ్రాయుటకు PROM ప్రోగ్రామ్ లేక PROM బర్నర్ ఉపయోగించబడుతుంది. PROM ప్రోగ్రామ్ చేయు విధానమును PROM బర్నింగ్ అంటారు.

3.5.3.2 ఎరేజబుల్ ప్రోగ్రామబుల్ రీడ్ ఓన్లీ మెమరీ (EPROM)

ఎరేజబుల్ ప్రోగ్రామబుల్ రీడ్ ఓన్లీ మెమరీ ఒక ప్రత్యేక రకం మెమరీ గల PROM గా వ్యవహరిస్తుంది. అయితే అతినీల లోహిత కిరణాలను ఉపయోగించి సమాచారాన్ని తొలగించవచ్చు. కానీ అతినీల లోహిత కిరణాలు కాంతికి బహిర్గతం అగు వరకు EPROM దాని సమాచారాన్ని నిలుపుకుంటుంది. అతినీల లోహిత కిరణాలు సమాచారాన్ని తొలగించడం వలన EPROM ప్రోగ్రాం మరలా చేయుటకు సాధ్యపడుతుంది.

EPROM, PROM కి భిన్నంగా వ్యవహరిస్తుంది. EPROMs ఒకేసారి మాత్రమే వ్రాయుటకు వీలు కలదు మరియు దానిని తొలగించలేము. PROM యొక్క సమాచారాన్ని నవీనకరించిన సంస్కర భర్తీ చేయుటకు విషయాలను తొలగించవచ్చును. ఉత్పత్తిదారులు అమ్మకపు ముందు EPROM లు వ్యక్తిగత కంప్యూటర్లలో విస్తృత్తంగా ఉపయోగించబడుతున్నాయి.



పటము 3.7 ఎరాజబుల్ ప్రోగ్రామబుల్ రీడ్ ఓన్లీ మెమరీ

మీకు తెలుసా?

చాలా వరకు EPROM చిప్స్ ఉపరితలం వద్ద పారదర్శక ప్రాంతాన్ని కలిగి ఉంటాయి. ఇవి స్టికర్‌లతో కప్పబడి ఉండును. దానిని తొలగిస్తే సూర్యకాంతిలోని అతినీల లోహిత కిరణాలు విషయాలను చెరిపి వేయును.

3.5.3.3 ఎలక్ట్రికల్లీ ఎరేజబుల్ ప్రోగ్రామబుల్ రీడ్ ఓన్లీ మెమరీ (EEPROM)

ఎలక్ట్రికల్లీ ఎరేజబుల్ ప్రోగ్రామబుల్ రీడ్ ఓన్లీ మెమరీ ఒక ప్రత్యేకమైన మెమరీ PROM రకం అగును. దీనిలో ఉన్న మెమరీ విద్యుత్ సమాచారాన్ని విద్యుత్ చెల్లించడం ద్వారా తొలగించవచ్చును. కానీ ఈ EEPROM ఇతర PROM లాగా విద్యుత్ నిలిపివేయునప్పుడు డేటా తొలగించబడదు. ఇతర ROM రకాలతో పోలిస్తే EEPROM నెమ్మదిగా పనిచేసే మెమరీ అగును.

3.5.4 క్యాచ్ మెమరీ (Cache Memory)

క్యాచ్ మెమరీ అనునది అతివేగవంతమైన మరియు ఖరీదైన మెమరీ. మెమరీలోని సమాచారాన్ని వేగంగా తిరిగి పొందుటకు ఉపయోగపడే మెమరీ అగును. దీని విలువ అదన ఖర్చుతో కూడినది. అందువల్ల CPU ప్రధాన మెమరితో పోలిస్తే క్యాచ్ మెమరీ పరిమాణంలో చిన్నది. క్యాచ్ మెమరీ లేని యెడల CPU ప్రతిసారి డేటాను ప్రధాన మెమరీ నుండి పొందుతుంది. అందువల్ల ఇది ఎక్కువ సమయాన్ని తీసుకుంటుంది. క్యాచ్ మెమరీని పరిచయం చేయుటకు ముఖ్య కారణం ఈ మెమరీ తరచుగా అవసరము మరియు చాలా వేగవంతమైనది. కావలసిన డేటా సేకరించబడుతుంది. ఇది వేగవంతమైన ప్రతిస్పందనను మరియు డేటాను నిల్వ చేస్తుంది. మెమరీ రీడ్ / రైట్ సమయాన్ని ఆధాచేయుటకు ఉపయోగపడు ప్రాప్యతకు త్వరగా ప్రతి స్పందన తెలియజేస్తుంది. యంగ్ మెమరీ / చదువుట / వ్రాయుట / లకు ఎంత వేగముగా సమాధానమును పంపుటకు వీలు అగును. (తెలియజేయును) పటము 3.8 CPU మరియు ప్రధాన మెమరీ మధ్య క్యాచ్ మెమరీ అమరికను తెలియజేస్తుంది.



పటము 3.8 క్యాచ్ మెమరీ అమరిక

11th Computer Science_Chapter-3 Telugu.indd 46 30-03-2023 18:06:00

3.6 ద్వితీయ స్టోరేజ్ పరికరాలు (Secondary Storage Devices)

ఒక కంప్యూటర్ సాధారణంగా పరిమితమైన ప్రాథమిక మెమరీని కలిగి ఉంటుంది. ఇది ఖరీదైనది మరియు అస్థిరతను కలిగియుండును. డేటా మరియు ప్రోగ్రామ్‌లను శాశ్వతంగా నిల్వచేయడానికి ద్వితీయ నిల్వ పరికరాలు ఉపయోగించబడతాయి. ద్వితీయ నిల్వ పరికరాలు ప్రధాన మెమరీకి సహాయక స్టోరేజ్ పనిచేస్తాయి. మరియు ఇవి ప్రకృతిలో అస్థిరత్వ స్వభావమును కలిగి ఉండును. ద్వితీయ నిల్వను బ్యాక్ ఆప్ స్టోరేజ్ అని కూడా పిలుస్తారు.

3.6.1 హార్డ్ డిస్క్ (Hard Disk)

హార్డ్ డిస్క్ అనేది అయస్కాంతత్వ డిస్క్ దీనిలో మీరు డేటాను స్టోరేజ్ చేయవచ్చు. ప్రతి డిస్క్ ఒక జత హెడ్స్ ద్వారా యాక్సెస్ చేయబడి డిస్క్‌లు ఒక దానిపై ఒకటి అమర్చబడిన అమరికను కలిగి ఉండును. హార్డ్ డిస్క్‌లు ఒకటి లేదా ఇరు వైపుల డిస్క్‌లను కలిగి ఉంటుంది.

3.6.2 కంపాక్ట్ డిస్క్ (Compact Disc (CD))

ఒక CD లేదా CD ROM 1.2 మిల్లీ మీటర్ల మందంతో పాలీకార్బొనేట్ అనే ప్లాస్టిక్ పదార్థంతో తయారవుతుంది. ఉపరితలం మందపాటి అల్యూమినియం లేదా బంగారం పూత పై భాగములో పూయబడి ఉండును. CD లో డేటా చిన్న ఇండెంటేషన్లుగా నిల్వ చేయబడుటను (pits) అందురు. పాలికార్బొనేట్ వాటి పొర పైభాగంలో అచ్చు వేయబడిన స్పైరల్ ట్రాక్స్‌లో ఎన్కోడ్ చేయబడినది. (pits) రెండు గుంటల మధ్య ఉన్న ప్రాంతాలను "ల్యాండ్స్" అంటారు. CD ప్లేయర్‌లో వుండే మోటారు CD ప్లేయర్‌ని తిరిగేటట్లు చేస్తుంది. సాధారణ CD-ROM సామర్థ్యం 700MB



పటము 3.9 కంపాక్ట్ డిస్క్

3.6.3 డిజిటల్ వెర్సటైల్ డిస్క్ (DVD)

ఒక DVD (డిజిటల్ వైర్సటైల్ డిస్క్ (లేదా) డిజిటల్ వీడియో డిస్క్) అనునది 4.7 GB డేటాను నిల్వ చేయగల ఒక ఆప్టికల్ డిస్క్. ఇది CD నిల్వ ప్రదేశం కంటే ఆరు రెట్లు ఎక్కువ నిల్వ ప్రదేశం కలిగి ఉండును. చలన చిత్రాలను మంచి నాణ్యతతో నిల్వచేయడానికి DVD లను తరచుగా ఉపయోగిస్తారు. CD ల మాదిరిగానే DVD లు లేజర్ ద్వారా చదువబడును.

ప్రతి డిస్క్ ఒకటి లేదా రెండు వైపులు కలిగిఉండును. మరియు ప్రతి వైపు ఒకటి లేదా రెండు పొరల డేటా వుంటుంది. భుజాలు మరియు పొరల సంఖ్య అది ఎంత వరకు నిల్వ చేసుకొనగలదో నిర్ణయిస్తుంది. సాధారణంగా రెండు పొరలు గల్గిన వైపు బంగారు రంగులో ఒక పొర గల్గిన వైపు వెండి రంగులో ఉంటాయి.



చిత్రము 3.10 డిజిటల్ వెర్సటైల్ డిస్క్

3.6.4 ఫ్లాష్ మెమరీ పరికరములు (Flash Memory Devices)

ఫ్లాష్ మెమరీ ఒక ఎలక్ట్రానిక్ (ఘన – స్థితి) అస్థిర కంప్యూటర్ నిల్వ మాధ్యమం. దీనిని విద్యుత్‌చే చెరిపివేయబడవచ్చు మరల ప్రోగ్రామ్ పునరుత్పత్తి చేయబడవచ్చు. EEPROM లేదా EPROM ఫ్లాష్ మెమరీకి ఉదాహరణలు పెన్‌డ్రైవ్‌లు, మెమరీ కార్డ్‌లు మొదలైనవి. ఫ్లాష్ మెమరీలు వ్యక్తిగత కంప్యూటర్లలో, వ్యక్తిగత డిజిటల్ కెమెరాలు మరియు మొబైల్ ఫోన్లలో వేగవంతమైన ప్రాప్యత సమయాన్ని అందిస్తుంది. మెమరీలో అక్షరమును చదవడానికి లేదా వ్రాయడానికి తీసుకున్న సమయాన్ని యాక్సెస్ సమయం అంటారు. ఫ్లాష్ మెమరీ సామర్థ్యం 1 గిగాబైట్ల (GB) నుండి 2 టెరాబైట్ల (TB) వరకు ఉంటుంది. ఫ్లాష్ మెమరీ నమూన చిత్రము 3.11 లో చూపబడింది.

11th Computer Science_Chapter-3 Telugu.indd 47 30-03-2023 18:06:01



పటము 3.11 ఫ్లాష్ మెమరీ

3.6.5 బ్లూ – రే – డిస్క్ (Blu-Ray Disc)

బ్లూరే అనేది DVD కి సమానమైన అధిక సాంద్రత కలిగిన ఆప్టికల్ డిస్క్ . ప్లేస్టేషన్ ఆటలకు మరియు హై – డెఫినేషన్ (HD) సినిమాలు ఆడటానికి ఉపయోగించు ఒక రకమైన బ్లూరే డిస్క్ అగును. రెండు పొరల బ్లూ రే డిస్క్ 50GB (గిగా బైట్లు) డేటాను నిల్వ చేయగలదు. డేటాను చదవడానికి మరియు వ్రాయడానికి DVD ఎరుపు లేర్‌ను ఉపయోగిస్తుంది. అయితే బ్లూ రే వ్రాయడానికి బ్లూ వైలెట్ లేజర్‌ను ఉపయోగిస్తుంది. అందువల్ల దీనిని బ్లూరే అందురు.



పటము 3.12 బ్లూ రేడిస్క్

3.7 పోర్ట్స్ మరియు ఇంటర్ఫేస్లు (Ports and Interfaces)

కంప్యూటర్ యొక్క మదర్‌బోర్డు చాలా I/O సాకేట్లను కలిగిఉండును. ఇవి కంప్యూటర్ వెనుక వైపు కనిపించే పోర్టులు మరియు ఇంటర్ఫేస్లుకు అనుసంధానించబడి ఉండును. (పటము 3.13) బాహ్య పరికరాలు పోర్ట్స్ మరియు ఇంటర్ఫేసులకు కనెక్టు చేయవచ్చు. వివిధ రకాలైన పోర్టులు క్రింద ఇవ్వబడినవి.

సీరియల్ పోర్ట్ : పాత కంప్యూటర్లలో కనిపించే బాహ్య పరికరాలను కనెక్ట్ చేయడానికి ఉపయోగించిరి.

సమాంతర పోర్ట్ : పాత కంప్యూటర్లతో కనిపించే ప్రింటర్లను కనెక్ట్ చేయడానికి ఉపయోగించిరి.

యు.ఎస్.బి. పోర్ట్ : కెమెరాలు, స్కానర్లు మొబైల్ ఫోన్లు, బాహ్య హార్డ్ డిస్క్లు మరియు ప్రింటర్లు వంటి బాహ్య పరికరాలను కంప్యూటర్‌కు కనెక్ట్ చేయడానికి ఉపయోగించిరి.

యూనివర్సల్ సీరియల్ బస్ (USB) ప్రమాణం యొక్క మూడవ ప్రధాన వర్షన్ యు ఎస్ బి 3.0 పటము 3.13 లో చూపిన విధంగా కంప్యూటర్లను ఇతర ఎలక్ట్రానిక్ గాడ్జెట్‌లతో అనుసంధానానికి ఉపయోగపడును. యు ఎస్ బి 3.0 5 గిగాబైట్ / సెకను వరకు డేటాను బదిలీ చేయగలదు. యు ఎస్ బి 3.1 యు ఎస్ బి 3.2 కూడా విడుదలయ్యాయి.



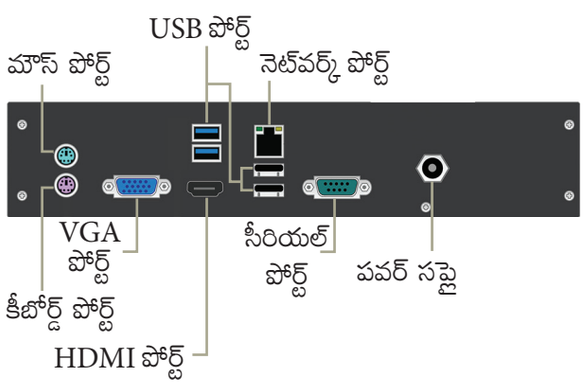
పటము 3.13 యు ఎస్ బి 3.0 పోర్టులు

VGA కనెక్టర్ : మానిటర్ లేదా LCD ప్రొజెక్టర్ వంటి ఏదైనా ప్రదర్శనా పరికరాన్ని కనెక్ట్ చేయడానికి ఉపయోగపడును.

ఆడియోప్లగ్స్: ధ్వని స్పీకర్లను, మైక్రో ఫోన్లను, హెడ్ ఫోన్లను కనెక్ట్ చేయడానికి ఉపయోగపడును.

పి ఎస్ 2 పోర్ట్ : మౌస్‌ను, కీ బోర్డును PC కి కనెక్ట్ చేయడానికి ఉపయోగిస్తారు.

SCSI పోర్ట్ : హార్డ్ డిస్క్ డ్రైవ్స్ మరియు నెట్‌వర్క్స్ కనెక్ట్ చేయడానికి

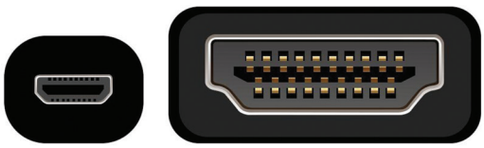


పటము 3.14 పోర్ట్ మరియు ఇంటర్ఫేస్లు

11th Computer Science_Chapter-3 Telugu.indd 48 30-03-2023 18:06:01

హై డెఫినేషన్ మల్టీమీడియా ఇంటర్ఫేసు (HDMI)

హైడెఫినేషన్ మల్టీమీడియా ఇంటర్ఫేస్ అనేది ఆడియో / వీడియో ఇంటర్ఫేస్. ఇది అన్‌కంప్రెస్డ్ వీడియో మరియు ఆడియో డేటాను వీడియో కంట్రోలర్ నుండి అనుకూల కంప్యూటర్ మానిటర్ ఎల్ సి డి ప్రొజెక్టర్ డిజిటల్ టెలివిజన్ మొదలైన వాటికి బదిలీ చేస్తుంది.



Micro HDMI HDMI

పటము 3.15 HDMI పోర్ట్

కృత్యము

విద్యార్థుల కృత్యము

- కంప్యూటర్ యొక్క భాగాలను గుర్తించండి
- బాహ్య పరికరాలను సంధానం చేయండి ఉదా: (ప్రింటర్ / ఎల్ సి డి ప్రొజెక్టర్)

ఉపాధ్యాయుల కృత్యము

- కంప్యూటర్ యొక్క భాగాలను చూపించటం
- విభిన్న ROM ICs లను ప్రదర్శించుట
- ఫ్లాష్ మెమరీని ప్రదర్శించటం
- పోర్టులు దాని యొక్క ఉపయోగాలను తెల్పుట

YV8EM

మూల్యాంకనము

భాగం – I

సరైన సమాధానం ఎన్నుకొనుము

1. ఈ క్రింది వాటిలో ఏది కంప్యూటర్ యొక్క మెదడు అని పిలువబడును?

   అ) ఇన్‌పుట్ పరికరాలు

   ఆ) అవుట్‌పుట్ పరికరాలు

   ఇ) మెమరీ పరికరం

   ఈ) మైక్రో ప్రాసెసర్

2. క్రింది వాటిలో మైక్రో ప్రాసెసర్ యూనిట్‌లో లేని భాగం ఏది?

   అ) ALU ఆ) నియంత్రణ యూనిట్

   ఇ) క్యాచ్ మెమరీ ఈ) రిజిస్టర్

3. ఒక పదం ఎన్ని బిట్స్‌ను కలిగివుండును?

   అ) 8

   ఆ) 16

   ఇ) 32

   ఈ) ఉపయోగించిన ప్రాసెసర్ ద్వారా

4. మెమోరీ అడ్రస్ రిజిస్టర్‌లో చిరునామా ఉంచబడినప్పుడు ఈ క్రింది వాటిలో ఏ పరికరం స్థానాన్ని గుర్తించును?

   అ) లొకేటర్ ఆ) ఎన్కోడర్

   ఇ) డీకోడర్ ఈ) మల్టీప్లెక్సర్

5. క్రింది వాటిలో CISC ప్రాసెసర్ ఏది?

   అ) ఇంటెల్ P6 ఆ) AMD K6

   ఇ) పెంటియమ్ III ఈ) పెంటియమ్ IV

6. వేగవంతమైన మెమరీ ఏది?

   అ) హార్డ్ డిస్క్ ఆ) ప్రధాన మెమరీ

   ఇ) క్యాచ్ మెమరీ ఈ) బ్లూ-రే డిస్క్

7. ఒకేసారి 8 బిట్స్ అడ్రస్ బస్ వున్న ప్రాసెసర్ ద్వారా ఎన్ని మెమరీ స్థానాలు గుర్తించబడును?

   అ) 28 ఆ) 1024

   ఇ) 256 ఈ) 8000

11th Computer Science_Chapter-3 Telugu.indd 49 30-03-2023 18:06:01

8. ఒక ప్రక్క మరియు ఒక అంతస్థు 12cm వ్యాసమును కలిగియున్న DVD యొక్క మొత్తం సామర్థ్యం ఎంత?

అ) 4.7 GB ఆ) 5.5 GB

ఇ) 4.8 GB ఈ) 2.2 GB

9. CD లో సూచించబడిన అతిచిన్న డేటా పరిమాణం ఏమిటి?

అ) బ్లాక్స్ ఆ) సెక్టార్స్

ఇ) పిట్స్ ఈ) ట్రాక్స్

10. ప్రదర్శనా పరికరాలను కంప్యూటర్‌కు కనెక్ట్ చేయడానికి ఉపయోగపడు పరికరాలు ఏవి?

అ) USB పోర్ట్ ఆ) Ps/2 పోర్ట్

ఇ) SCSI పోర్ట్ ఈ) VGA కనెక్టర్

భాగం – II

లఘు సమాధానమిమ్ము

1 మైక్రో ప్రాసెసర్ యొక్క లక్షణాలను ప్రభావితం చేసే పారామితులు ఏవి?

2 సూచనలు అనగానేమి?

3 ప్రోగ్రామ్ కౌంటర్ అనగానేమి?

4 HDMI అంటే ఏమిటి?

5 EPROM యొక్క కంటెంట్‌ను తొలగించడానికి ఏ మూలం ఉపయోగించబడుతుంది?

భాగం – III

సంక్షిప్త సమాధానమిమ్ము

1. భేదము తెల్పుము కంప్యూటర్ సంస్థ నుండి కంప్యూటర్ ఆర్క్‌టెక్చర్.
2. డేటా పరిమాణం ఆధారంగా మైక్రో ప్రాసెసర్‌ను వర్గీకరించండి.
3. సూచనల సమితి ఆధారంగా మైక్రో ప్రాసెసర్‌ను వర్గీకరించుము.
4. PROM మరియు EPROM కు భేదములను తెల్పుము.
5. కంప్యూటర్‌లలో గల ఇంటర్ఫేసస్ మరియు పోర్టును వ్రాయుము.
6. CD మరియు DVD లకు భేదములను తెల్పుము
7. ఫ్లాష్ మెమరీ మరియు EEPROM ను నీవు ఎలా వ్యత్యాసించెదవు?

భాగం – IV

వివరంగా సమాధానాలిమ్ము.

1. మైక్రో ప్రాసెసర్ యొక్క లక్షణాలను వివరించండి
2. ప్రాసెసర్ ద్వారా చదవడం మరియు వ్రాయడం ఆపరేషన్స్ ఎలా నిర్వహించబడుతాయి? వివరించండి.
3. యాక్సస్ సమయం ఆధారంగా మెమరీ పరికరాలను ఆరోహణ క్రమంలో అమర్చండి.
4. ROM లోని రకాలను వివరించండి.

11th Computer Science_Chapter-3 Telugu.indd 50 30-03-2023 18:06:01

పదకోశము

Computer hardware	కంప్యూటర్ యొక్క భౌతిక భాగాలు, CPU, మదర్ బోర్డు, మానిటర్ కీ బోర్డు మొదలగునవి
Intel	ఇంటెల్ కార్పొరేషన్ అనేది ఒక అమెరికన్ బహుళ జాతి సంస్థ మరియు సాంకేతిక సంస్థ. ఇది హార్డ్‌వేర్ తయారీలో పాల్గొంటుంది. ముఖ్యంగా మదర్ బోర్డు మరియు ప్రాసెసర్లు.
Silicon chip	సిలికాన్ చిప్ అనేది ఇంటిగ్రేటడ్ ఎలక్ట్రానిక్ సర్క్యూట్ల సమితి, సెమీ కండెక్టర్ పదార్థం యొక్క చిన్న ముక్క ఈ సిలికాన్ చిప్.
Multipurpose	బహుళార్థ సాధకం అనేది అనేక ఉపయోగాలు
Address bus	అడ్రస్ బస్ అనునది తీగల సమాహారం ఇది చిరునామాలను బిట్స్ రూపములో తీసుకొని పోవును
Data bus	డేటా బస్ అనునదే వైర్ల సేకరణ అగును. ఇవి డేటాలను బిట్స్‌లో తీసుకొని పోవును.
Control bus	సమాచారాన్ని ఒక అంశము నుంచి మరొక అంశమునకు తరలించటం. కంట్రోల్ బస్ అనేది తీగల సమాహారం ఇది కార్య క్రమాలను నియంత్రణ చేస్తుంది
Arithmetic operations	కూడిక , తీసివేత మొదలగు గణిత సంబందిత కార్యకలాపములు. ఉదా: AND, OR , NOT
Data Transfer	సమాచారాన్ని ఒక అంశము నుంచి మరొక అంశమునకు తరలించటం.
Logical operations	ఏక దిశాత్మక అంటే ఒకటే దిశ
Bidirectional	ద్వి దిశాత్మక అంటే రెండు దిశలు / మార్గాలు
Unidirectional	ఏక దిశాత్మక అంటే ఒక దిశ
Access time	యాక్సస్ టైమ్ అనేది అభ్యర్థన మధ్య సమయం ఆలస్యం / జాప్యం. ఎలక్ట్రానిక్ సిస్టమ్ యాక్సస్ పూర్తయింది లేదా అభ్యర్థించ డేటా తిరిగి ఇవ్వబడుతుంది

11th Computer Science_Chapter-3 Telugu.indd 51 30-03-2023 18:06:01

అధ్యాయము 4

ఆపరేటింగ్ సిస్టమ్ యొక్క సైద్ధాంతిక అంశాలు (Theoretical concepts of Operating System)

అభ్యసనా లక్ష్యాలు

- ✓ ఆపరేటింగ్ సిస్టమ్ యొక్క భావన మరియు వాటి రకాలను తెలుసుకోవడం
- ✓ ఆపరేటింగ్ సిస్టమ్ యొక్క ప్రాథమిక జ్ఞానాన్ని మరియు దాని విధులను తెలుసుకోవడం

4.1 సాఫ్ట్‌వేర్ పరిచయము (Introduction to Software)

సాఫ్ట్‌వేర్ ఒక సూచనల సమితి. ఇది నిర్దిష్ట పనిని సూచిస్తుంది. ఇది హార్డ్‌వేర్‌తో అనుసంధానమై కావలసిన అవుట్‌పుట్‌ను వెలువరిస్తుంది.

4.1.1 సాఫ్ట్‌వేర్‌లోని రకాలు (Types of Software)

సాఫ్ట్‌వేర్‌లోని రకాలను రెండు విధాలుగా వర్గీకరించవచ్చును.

1) అప్లికేషన్ సాఫ్ట్‌వేర్
2) సిస్టం సాఫ్ట్‌వేర్

అప్లికేషన్ సాఫ్ట్‌వేర్ (Application Software):

అనువర్తన సాఫ్ట్‌వేర్ అనునది ప్రోగ్రామ్ ఒక నిర్దిష్ట పనిని నిర్వహించడానికి కావలసిన ప్రోగ్రామ్స్ సమితి అగును. ఉదా: MS-word అనేది టెక్స్ట్ డాక్యుమెంట్లను రూపొందించడానికి ఉపయోగపడు అప్లికేషన్ సాఫ్ట్‌వేర్ అగును మరియు VLC ప్లేయర్ అనునది ఆడియో / వీడియో ఫైల్స్ మరియు మరెన్నో ఫైల్స్ ప్లే చేయుటకు ఉపయోగపడు అప్లికేషన్ సాఫ్ట్‌వేర్.

Q712S

సిస్టమ్ సాఫ్ట్‌వేర్ (System Software)

సిస్టమ్ సాఫ్ట్‌వేర్ అనేది కంప్యూటర్‌ను హార్డ్‌వేర్ మరియు అప్లికేషన్ ప్రోగ్రామ్‌లను అమలుచేయడానికి రూపొందించబడిన కంప్యూటర్ ప్రోగ్రామ్ అగును. ఉదా: ఆపరేటింగ్ సిస్టమ్ మరియు కంపైలర్

4.2 ఆపరేటింగ్ సిస్టమ్ పరిచయము (Introduction to Operating System (OS)

ఆపరేటింగ్ సిస్టమ్ అనునది వినియోగదారునికి కంప్యూటర్‌కు మధ్య ఇంటర్ఫేస్‌గా వ్యవహరించే ఒక సాఫ్ట్‌వేర్ సిస్టమ్ అగును.

ఇది ఇన్‌పుట్ అవుట్‌పుట్ మరియు డిస్క్ డ్రైవ్లు, ప్రింటర్లు మరియు ఎలక్ట్రాన్ గాడ్జెట్లు వంటి ఇతర పరిధీయ పరికరాలను నియంత్రించును. ఆపరేటింగ్ సిస్టమ్ యొక్క విధులు ఫైల్ మేనేజ్‌మెంట్, మెమరీ మేనేజ్‌మెంట్ ప్రాసెస్ మేనేజ్‌మెంట్ మరియు డివైస్ మేనేజ్‌మెంట్ మొదలగునవి ఆవరేటింగ్ సిస్టమ్ యొక్క విధులు అగును.

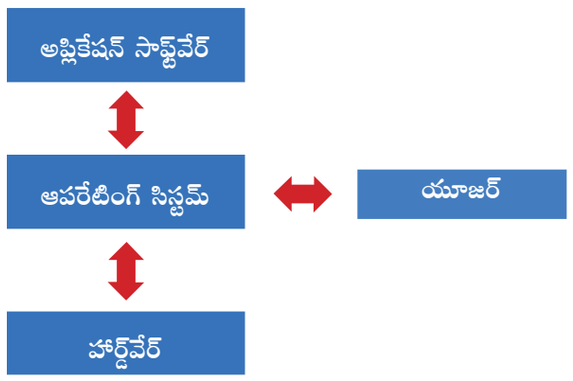
ఆపరేటింగ్ సిస్టమ్ లేకుండా కంప్యూటర్ దాని అన్ని వనరులను సమర్థవంతంగా నిర్వహించుటకు వీలుకాదు. కంప్యూటర్ స్విచ్ ఆన్ చేసినపుడు, ఆపరేటింగ్ సిస్టమ్ స్వయం చాలకంగా మెమరీలోనికి లోడ్ అవుతుంది.

11th Computer Science_Chapter-4 Telugu.indd 52 15-03-2023 11:46:50



పటము 4.1 ఆపరేటింగ్ సిస్టమ్

విండోస్, యునిక్స్ మరియు లినక్స్ మొదలగు ఆపరేటింగ్ సిస్టమ్‌లను వ్యక్తిగత కంప్యూటర్లు మరియు ల్యాప్‌టాప్‌లలో ఉపయోగించుదురు. మొబైల్ పరికరములు ఎక్కువగా 'ఆండ్రాయడ్ మరియు ioS ని మొబైల్ **OS** ఆపరేటింగ్ సిస్టమ్‌లో ఉపయోగించుదురు.



పటము 4.2 ఆపరేటింగ్ సిస్టమ్ మరియు యూజర్ యొక్క పరస్పర చర్య

ఆపరేటింగ్ సిస్టమ్ యొక్క ఉపయోగాలు (Uses of Operating Systems)

ఆపరేటింగ్ సిస్టమ్ యొక్క కొన్ని ఉపయోగాలు క్రింద ఇవ్వబడినవి:

ఆపరేటింగ్ సిస్టమ్ యొక్క ప్రధాన ఉపయోగాలు.

- వినియోగదారునికి కావలసిన సమాచారాన్ని కంప్యూటర్ అందించడానికి ఉపయోగపడుతుంది.
- కంప్యూటర్లకి వినియోగదారునికి మధ్య సులభమైన ప్రతిస్పందనకు ఉపయోగపడును.
- కంప్యూటర్‌ను ఆన్ చేయగానే విద్యుత్ సరఫరా ప్రారంభించగానే స్వయం చాలకంగా కంప్యూటర్ ఆపరేషన్ సిస్టమ్ ప్రారంభమగును (Booting)
- ఇన్‌పుట్ మరియు అవుట్‌పుట్ పరికరాలను నియంత్రించును.
- ప్రధాన మెమరీ వినియోగాన్ని నిర్వహించడం
- వినియోగదారుని ప్రోగ్రామ్‌లకు భద్రతను కలిగించును.

4.3 ఆపరేటింగ్ సిస్టమ్‌లోని రకాలు (Types of Operating System)

ప్రాసెసింగ్ సామర్థ్యాలను ఆధారంగా ఆపరేటింగ్ సిస్టమ్ క్రింది రకాలుగా వర్గీకరించబడును.

4.3.1 సింగిల్ యూజర్ ఆపరేటింగ్ సిస్టమ్ (Single User Operating Systems)

ఈ ఆపరేటింగ్ సిస్టమ్స్ ఒకే సమయంలో ఒక వినియోగదారుని ఒక పనిని మాత్రమే చేయడానికి అనుమతిస్తుంది. ఇది సింగిల్ యూజర్ మరియు సింగిల్ టాస్క్ ఆపరేటింగ్ సిస్టమ్ అని పిలువబడును. సింగిల్ యూజర్ మరియు సింగిల్ టాస్క్‌కు ఒక ఉదాహరణగా MS-DOS ను పేర్కొనవచ్చును.

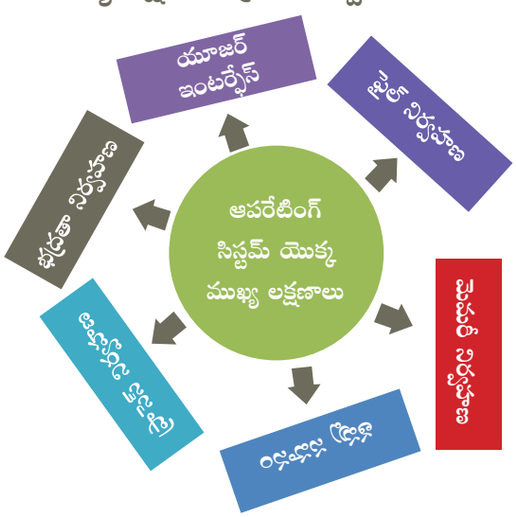
11th Computer Science_Chapter-4 Telugu.indd 53 15-03-2023 11:46:50

4.3.2 మల్టీయూజర్ ఆపరేటింగ్ సిస్టమ్ (Multi-user Operating Systems)

యాక్సెస్ చేయడానికి మరియు అప్లికేషన్స్‌ను అను వర్తకాలను మరియు ల్యాప్‌టాప్‌లలో బహుళ వినియోగదారులచే ఒకే సమయంలో ఉపయోగించబడు ఆపరేటింగ్ సిస్టమ్‌లు "మల్టీ యూజర్ ఆపరేటింగ్ సిస్టమ్‌లు". ఇది వినియోగదారులు ఒకరితో మరొకరు సంభాషించు కొనుటకు తోడ్పడును. బహుళ వినియోగదారు ఆపరేటింగ్ సిస్టమ్‌కు ఉదాహరణలు విండోస్, లినక్స్ మరియు యునిక్స్.

4.4 ఆపరేటింగ్ సిస్టమ్ యొక్క ముఖ్య లక్షణాలు (Key features of the Operating System)

వివిధ ముఖ్య లక్షణాలు క్రింద ఇవ్వబడినది



పటము 4.3 ఆపరేటింగ్ సిస్టమ్ యొక్క ముఖ్య లక్షణాలు

4.4.1 యూజర్ ఇంటర్‌ఫేస్ (User Interface (UI))

ఆపరేటింగ్ సిస్టమ్‌లోని ముఖ్యమైన లక్షణాలలో యూజర్ ఇంటర్‌ఫేస్ ఒకటి. ఇది UI వినియోగదారుడు కంప్యూటర్‌తో పరస్పర చర్య చేయగల ఏకైక మార్గం అగును. GUI (గ్రాఫికల్ యూజర్ ఇంటర్‌ఫేస్) ఆధారిత ఆపరేటింగ్ సిస్టమ్ యొక్క విజయానికి GUI ఒక ప్రధాన కారణం అగును. I/O అనేది విండో ఆధారిత వ్యవస్థ. ఇది pointing device ని నిర్దేశించడానికి లను కలిగియుండును. ఎంపిక చేయడానికి కావలసిన మరియు టెక్స్ట్‌లోకి ప్రవేశించడానికి ఒక కీబోర్డ్‌ను కలిగివుండును. దీని శక్తివంతమైన రంగులు వినియోగదారుని చాలా సలభంగా ఆకర్షించును.

ఇప్పుడు లినక్స్ పంపిణీ GUI ఆధారిత ఆపరేటింగ్ సిస్టమ్‌గా కూడా అందుబాటులో ఉంది. అనువర్తనం కోసం వినియోగదారుల ఇంటర్‌ఫేస్ రూపొందించబడినప్పుడు ఈ క్రింది అంశాలను పరిగణించవలయును.

1. వినియోగదారు ఇంటర్‌ఫేస్ వినియోగదారుని ఈ నైపుణ్యాన్ని దీర్ఘకాలము నిలుపుకోచేయవలయును.
2. వినియోగదారు యొక్క అవసరాలను బట్టి వినియోగదారు ఇంటర్‌ఫేస్ వినియోగదారుని అవసరములను సంతృప్తి పరచాలి.
3. వినియోగదారుని విలువైన సమయాన్ని వినియోగదారు ఇంటర్‌ఫేస్ పొదుపు చేయాలి.
4. ఒక ఉత్పత్తి యొక్క అంతిమ లక్ష్యం వినియోగదారుని సంతృప్తిపరచును. అదే విధంగా వినియోగదారుని ఇంటర్‌ఫేస్ వినియోగదారుని సంతృప్తిపరచడమే.
5. వినియోగదారుని యొక్క తప్పుల సంఖ్యను యూజర్ ఇంటర్‌ఫేస్ తగ్గించవలెను.

4.4.2 మెమరీ నిర్వహణ (Memory Management)

మెమరీ నిర్వహణ అనునది కంప్యూటర్ యొక్క ప్రధాన మెమరీని నియంత్రించడం, సమన్వయం

11th Computer Science_Chapter-4 Telugu.indd 54 15-03-2023 11:46:51

చేయడం, మొత్తం కంప్యూటర్ల పనితీరును ఆప్టిమైజ్ చేయడం మరియు వివిధ రన్నింగ్ ప్రోగ్రామ్‌లకు మెమరీ బ్లాక్‌ను (స్పేస్) కేటాయించడం అగును. మెమరీ నిర్వహణలో వినియోగదారు డిమాండ్ల ఆధారంగా వ్యక్తిగత ప్రోగ్రామ్‌లకు నిర్దిష్ట మెమరీ బ్లాక్‌ల కేటాయింపు వుంటుంది.

మెమరీ నిర్వహణ ప్రక్రియ యొక్క లక్ష్యం CPU వినియోగం మరియు ప్రధాన మెమరీ ద్వారా దాని వినియోగదారులకు కంప్యూటర్ ప్రతిస్పందన యొక్క వేగం రెండింటిని మెరుగుపరచడం అగును. ఈ కారణాల వల్ల కంప్యూటర్లు అనేక మెమరీ మేనేజ్‌మెంట్ స్కీమ్‌లతో మెమరీలో ఉంచాలి.

మెమరీ నిర్వహణకు సంబంధించిన క్రింది కార్యకలాపాలకు ఆపరేటింగ్ సిస్టమ్ బాధ్యత వహించును.

- మెమరీ యొక్క ఏ భాగం ప్రస్తుతం ఎవరిచే ఉపయోగింపబడుచున్నదను విషయమును పర్యవేక్షించడం.
- ఏ ప్రక్రియలను లేదా ప్రక్రియ భాగాలు మరియు డేటాను తరలించాలో నిర్ణయించటం. మెమరీ లోపల మరియు వెలుపల తరలించు ప్రక్రియలను మరియు డేటాలను నిర్ణయించడం.
- మెమరీ బ్లాక్‌లోని కేటాయింపులు మరియు డీ-కేటాయింపులు అవసరమైనవి. (చెత్తను సేకరించటం) అవసరమగు మెమరీ బ్లాక్‌లను కేటాయించడం మరియు తొలగించడం అగును. వ్యర్థాల సేకరణ

4.4.3 ప్రాసెస్ నిర్వహణ (Process management)

ప్రాసెస్ మేనేజ్‌మెంట్ అనేది ప్రక్రియలను సృష్టించడం మరియు తొలగించటం (ప్రోగ్రామ్) మరియు ప్రక్రియలు మరియు ఒకదానితో ఒకటి సంభాషించి సమకాలించడానికి విధానాలను అవసరమైన యాంత్రికలను కల్గిన దానిని "ప్రాసెస్ మేనేజ్‌మెంట్" అందురు.

ప్రింటర్ లేదా స్క్రీన్‌కు అవుట్‌పుట్ పంపడం వంటి సిస్టమ్ పనిని "ప్రాసెస్"అని కూడా పిలుస్తారు.

కంప్యూటర్లలో ప్రక్రియల సేకరణ రెండు వర్గాలుగా విభజించవచ్చు.

- సిస్టమ్ కోడ్ ద్వారా అమలుచేయడానికి ఆపరేటింగ్ సిస్టమ్ ప్రాసెస్‌లు. (ఆపరేటింగ్ సిస్టమ్ ప్రక్రియలు సిస్టమ్ కోడ్ ద్వారా అమలు చేయబడును
- వినియోగదారుని కోడ్ ద్వారా అమలు చేయబడిన యూజర్ ప్రాసెస్‌ర్లు అగును.

ఈ ప్రక్రియలన్నీ సమర్థవంతంగా ఒకే CPU తో ఏక కాలంలో అమలు చేయబడును.

ప్రధానంగా పని ప్రక్రియను కేటాయించడానికి క్రింది ఆల్గోరిథంలు ప్రాసెసర్‌లో ఉపయోగిస్తారు.

1. FIFO 2. SJF 3. రౌండ్ రాబిన్

4. ప్రాధాన్యత ఆధారంగా

ఫస్ట్‌ఇన్ అవుట్‌పుట్ షెడ్యూలింగ్ (FIFO (First In First Out) Scheduling)

ఈ ఆల్గోరిథం క్యూయింగ్ టెక్నిక్ మీద ఆధారపడి ఉంటుంది.

ఒక విద్యార్థి అతని / ఆమె గురువు నుండి గ్రేడ్ షీట్ పొందడానికి క్యూలో నిలబడి క్యూలో మొదటి

11th Computer Science_Chapter-4 Telugu.indd 55 15-03-2023 11:46:51

స్థానంలో నిలిచిన ఇతర విద్యార్థి మొదట అతని / ఆమె గ్రేడ్ షీట్ పొందాడు మరియు క్యూ FIFO నుండి బయలుదేరాడు. క్యూలో తదుపరి విద్యార్థి అనుసరించబడును. ఈ విధానాన్ని CPU అల్గోరిథం యొక్క ప్రాథమిక తర్కం అగును.
సాంకేతికంగా మొదట క్యూలోకి ప్రవేశించే ప్రక్రియ మొదట చేత అమలు చేయబడుతుంది. తరువాత ప్రక్రియలు క్యూ (రో) క్రమంలో అమలు చేయబడుతాయి.

SJF (చిన్నదైన ప్రక్రియ పని మొదట) షెడ్యూలింగ్: SJF (Shortest Job First)Scheduling

ఈ ఆల్గోరిథం CPU చేత అమలు చేయబడుతున్న పని (వర్క్) పరిమాణం ఆధారంగా పనిచేస్తుంది.

A మరియు B అనే రెండు పనులను పరిగణించండి.

1) A = 6 కిలోబైట్లు 2) B = 9 కిలో బైట్లు

మొదటి 'ఎ' పని కేటాయించబడును మరియు తరువాత పని 'బి' కేటాయించబడును.

రౌండ్ రాబిన్ షెడ్యూలింగ్ (Round Robin Scheduling)

రౌండ్ రాబిన్ (RR) షెడ్యూలింగ్ అనే ఆల్గోరిథం ముఖ్యంగా సమయం పంచుకునే వ్యవస్థలకోసం రూపొందించబడింది. పని ప్రక్రియలు వృత్తాకార పద్ధతి మరియు ప్రాసెసర్ సమయంను అనుసరించి కేటాయించబడుతాయి. ఉదాహరణకు A, B, C. మూడు పనులను తీసుకోండి. మొదటి పని A ని CPU కి కేటాయించి తరువాత పని C మరియు పని A, B, తరువాత B మరియు A, D మొదలైనవి మరలా మరియు ఇదే ప్రక్రియ కొనసాగించబడును.

ప్రాధాన్యత ఆధారంగా (Based On Priority)

ఇచ్చిన పని (ప్రక్రియ) ప్రాధాన్యత ఆధారంగా కేటాయించబడుతుంది. ఇతర పనుల కన్నా ఎక్కువ ప్రాధాన్యత గల పని చాలా ముఖ్యం. A మరియు B అనే రెండు పనులను తీసుకోండి. A యొక్క ప్రాధాన్యత 5 మరియు B యొక్క ప్రాధాన్యత 7 జాబ్ A కి ముందు జాబ్ B ప్రాసెసర్‌కు కేటాయించబడును.

4.4.4 భద్రతా నిర్వహణ (Security Management)

కంప్యూటర్ మరియు సాఫ్ట్‌వేర్ పరిశ్రమలో ప్రధాన సవాలు యూజర్ యొక్క చట్టబద్ధమైన డేటాను హ్యాకర్ల నుండి రక్షించడం. ఆపరేటింగ్ సిస్టమ్ వినియోగదారు ముగింపుకు మూడు స్థాయిలో సెక్యూరిటీలను అందించును.

అవి
(1) ఫైల్ యాక్సస్ స్థాయి
(2) సిస్టమ్ స్థాయి
(3) నెట్‌వర్క్ స్థాయి

ఇతర వ్యక్తులు సృష్టించిన ఫైల్‌లను యాక్సస్ చేయడానికి మీకు యాక్సస్ అనుమతి ఉండాలి. ఫైల్ యొక్క సృష్టికర్త లేదా సిస్టమ్ యొక్క నిర్వాహకుడు అనుమతులను మంజూరు చేయవచ్చు. బహుళ వినియోగదారు వాతావరణంలో పాస్‌వర్డ్ ద్వారా సిస్టమ్ స్థాయి భద్రత అందించబడుతుంది. విండోస్ మరియు లినక్స్ రెండూ పాస్‌వర్డ్ సౌకర్యాన్ని కలిగి ఉంటాయి.

నెట్‌వర్క్ భద్రత అనిర్వచనీయమైనది. కాబట్టి ప్రపంచం నలుమూలల నుండి ప్రజలు అలాంటి భద్రతను కల్పించడానికి ప్రయత్నిస్తారు.

11th Computer Science_Chapter-4 Telugu.indd 56 15-03-2023 11:46:51

పైన పేర్కొన్న అన్ని స్థాయి భద్రతలు ఆపరేటింగ్ సిస్టమ్ ద్వారా మాత్రమే అందించబడుతాయి.

4.4.5 తప్పు సహనం (Fault Tolerance)

ఆపరేటింగ్ సిస్టమ్స్ దృడంగా ఉండాలి. ఏదేని లోపం ఉన్నప్పుడు ఆపరేటింగ్ సిస్టమ్ క్రాష్ కాకూడదు. బదులుగా తప్పులను సరిదిద్దు సామర్థ్యమును కలిగి ప్రస్తుత వ్యవస్థ యొక్క సహాయ స్థితిని నిలుపుకోవాలి.

4.4.6 ఫైల్ నిర్వహణ (File Management)

ఫైల్ మేనేజ్‌మెంట్ అనేది OS యొక్క ముఖ్యమైన భాగము ఇది డేటా నిల్వ పద్ధతులను నిర్వహిస్తుంది. ఆపరేటింగ్ సిస్టమ్ కంప్యూటర్‌లోని ఫైల్స్, ఫోల్డర్‌లు మరియు డైరెక్టరీ సిస్టమ్స్‌లను నిర్వహిస్తుంది. FAT (ఫైల్ కేటాయింపు పట్టిక) ఫైల్ పేరు, రకం (టెక్స్ట్ లేదా బైనరీ) పరిమాణము, ప్రారంభ మరియు యాక్సస్‌ను నిల్వచేస్తుంది. ఆపరేటింగ్ సిస్టమ్ యొక్క ఫైల్ మేనేజర్ ఫైల్‌లను సృష్టించడానికి సవరించడానికి కాపీ చేయడానికి మెమరీని కేటాయించడానికి సహాయపడుతుంది మరియు FAT ని కూడా నవీకరిస్తుంది. భవిష్యత్ తరాల ఫైల్ సిస్టమ్ (NTFS) మరియు ఎక్స్‌ట్2 (లినక్స్) వంటి కొన్ని ఇతర ఫైల్ మేనేజ్‌మెంట్ టెక్నిక్‌లు అందుబాటులో ఉన్నాయి.

4.4.7 బహుళ ప్రక్రియ (మల్టీ ప్రాసెసింగ్) (Multi-Processing)

ఆపరేటింగ్ సిస్టమ్ యొక్క లక్షణాలలో ఇది ఒకటి. ఒకే రన్నింగ్ ప్రాసెస్ (పని) కోసం ఇది రెండు లేదా అంతకంటే ఎక్కువ ప్రాసెసర్లను కలిగివుండును. ప్రాసెసింగ్ సమాంతరంగా జరుగును కాబట్టి దీనిని సమాంతర ప్రాసెసింగ్ అందురు. ఎగ్జిక్యూషన్ సమాంతరంగా జరుగును కాబట్టి ఈ లక్షణం హైస్పీడ్ ఎగ్జిక్యూషన్ కోసం ఉపయోగించబడుతుంది. ఇది కంప్యూటింగ్ శక్తిని పెంచుతుంది.

4.4.8 సమయం – పంచుకోవడం (Time-sharing)

ఆపరేటింగ్ సిస్టమ్స్ యొక్క లక్షణాలలో ఇది ఒకటి. ఏక కాలంలో బహుళ ప్రక్రియలను అమలు చేయడానికి ఇది అనుమతిస్తుంది. ప్రతి పనికి నిర్ణీత సమయం కేటాయించబడుతుంది. ఈ సమయ విభజననే సమయ భాగస్వామ్యం అందురు. సమయం ముగిసిన తర్వాత లేదా ప్రక్రియ పూర్తయిన తర్వాత ప్రాసెసర్ వివిధ ప్రక్రియల మధ్య వేగంగా మారుతుంది.

ఉదాహరణకు పి1, పి2, పి3 అనే మూడు ప్రక్రియలు ఉన్నాయని అనుకోండి. ప్రతి ప్రక్రియకు 30, 40, 50 నిమిషములు కేటాయించబడును. ప్రాసెస్ P1 నిమిషాల్లో పూర్తయితే ప్రాసెసర్ అమలు కోసం తదుపరి ప్రాసెసర్ 20 ను తీసుకుంటుంది. P2 ప్రాసెస్ 40 నిమిషాల్లో పూర్తిచేయలేకపోతే ప్రస్తుత ప్రాసెస్ P2 నిలిపి వేయబడును. తదుపరి ప్రాసెసర్ P3 కి మార్చబడును.

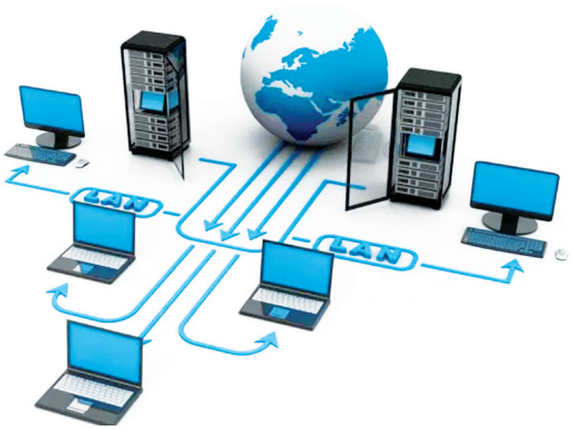
4.4.9 పంపిణీ ఆపరేటింగ్ సిస్టమ్స్ (Distributed Operating Systems)

ఇంటర్‌నెట్ / ఇంట్రానెట్ ఉపయోగించి ప్రపంచంలోని ఏ యంత్రంలోనైనా ఉన్న పంచుకోగల డేటా మరియు ఫైల్‌లను యాక్సెస్ చేయడానికి పంపిణీ ఆవరేటింగ్ సిస్టమ్ ఉపయోగించబడుతుంది. వినియోగదారులు తమ సొంత కంప్యూటర్లలో అందుబాట్లలో ఉన్నట్లుగా యాక్సస్ చేయవచ్చు.

పంపిణీ చేయబడిన ఆపరేటింగ్ సిస్టమ్ యొక్క ప్రయోజనాలు క్రింది విధంగా వివరించవచ్చును.

11th Computer Science_Chapter-4 Telugu.indd 57 15-03-2023 11:46:51

- ఒక ప్రదేశంలో ఉన్న వినియోగదారు నెట్‌వర్క్ ద్వారా మరొక ప్రదేశంలో లభించే అన్ని వనరులను ఉపయోగించుకోవచ్చు.
- అనేక కంప్యూటర్ వనరులను నెట్‌వర్క్‌తో సులభంగా కలపవచ్చును.
- వినియోగదారుల మధ్య పరస్పర అవగాహనను అభివృద్ధి చేయును.
- హోస్ట్ కంప్యూటర్‌లలో గల లోడ్‌ను తగ్గిస్తుంది.



పటము: *4.4* పంపిణీ ఆపరేటింగ్ సిస్టమ్

4.5 ప్రముఖ ఆపరేటింగ్ సిస్టమ్ (Prominent Operating Systems)

ప్రముఖ ఆపరేటింగ్ సిస్టమ్ OS క్రింది విధంగా ఉన్నాయి.

- యునిక్స్
- మైక్రోసాఫ్ట్ విండోస్
- లినెక్స్
- iOS
- ఆండ్రాయిడ్

ఆధునిక ఆపరేటింగ్ సిస్టమ్స్ గ్రాఫికల్ యూజర్ ఇంటర్ఫేస్ (GUI) ను ఉపయోగిస్తాయి. ఐకాన్లు, బటన్లు, మెనూలు మరియు ప్రతిదీ క్లిక్ చేయడానికి GUI మౌస్‌కు అనుమతిస్తుంది. గ్రాఫిక్స్ మరియు టెక్స్ట్ అంశాల కలయికను ఉపయోగించి తెరపై స్పష్టంగా ప్రదర్శించబడుతుంది.

విద్యార్థుల కార్యాచరణ (Student Activity)

ఆపరేటింగ్ సిస్టమ్ లోగో మరియు సరైన వివరణ మధ్య ఒక గీతను గీయుము

కమాండ్ లైన్ ఆపరేటింగ్ సిస్టమ్ ఓపెన్ సోర్స్ సాఫ్ట్‌వేర్ అభివృద్ధికి మరియు మొబైల్ ఫోన్ టెక్నాలజీ కోసం ఒక ఉదాహరణ అగును.	
పాపులర్ ఆపరేటింగ్ సిస్టమ్‌కు ఇది సంబంధం లేదు. ఇది ఆపిల్ ఉత్పత్తికారు.	
ఆపిల్ కంప్యూటర్లతో ఉపయోగించబడుతుంది మరియు క్లౌడ్ కంప్యూటర్‌తో బాగా పనిచేస్తుంది.	
ఆపిల్ ఐఫోన్ కోసం ఉపయోగింటానికి రూపొందించబడినది	
విశ్వవిద్యాలయాలు, కంపెనీలు, పెద్ద సంస్థలు మొదలైన వాటిలో బాగా ప్రసిద్ధి చెందిన ఆపరేటింగ్ వ్యవస్థ అగును.	
ఆపరేటింగ్ సిస్టమ్ వ్యక్తిగత కంప్యూటర్లలో ప్రసిద్ధి గాంచినది	

మూల్యాంకనం



భాగము - I

సరైన సమాధానాన్ని ఎన్నుకొనుము

1) ఆపరేటింగ్ సిస్టమ్ అనగా

అ) అప్లికేషన్ సాఫ్ట్‌వేర్

ఆ) హార్డ్‌వేర్

ఇ) సిస్టమ్ సాఫ్ట్‌వేర్

ఈ) విభాగాలు

11th Computer Science_Chapter-4 Telugu.indd 58 15-03-2023 11:46:54

2) ఆపరేటింగ్ సిస్టమ్ యొక్క ఉపయోగాన్ని గుర్తించండి

అ) మానవులు మరియు కంప్యూటర్ మధ్య సులభ పరస్పర చర్య

ఆ) ఇన్‌పుట్ మరియు అవుట్‌పుట్ పరికరాలను నియంత్రించడం

ఇ) ప్రధాన మెమరీ ఉపయోగాన్ని నియంత్రించుట

ఈ) పైవన్నీ

3) ఈ క్రింది వానిలో ఆపరేటింగ్ సిస్టమ్ యొక్క ఫంక్షన్స్ కానివి ఏది

అ) ప్రాసెస్ నిర్వహణ

ఆ) మెమరీ నిర్వహణ

ఇ) భద్రతా నిర్వహణ

ఈ) కంపైలర్

4) క్రింది వాటిలో వాణిజ్యపరంగా అంగీకారం పొందిన ఆపరేటింగ్ సిస్టమ్ వ్యవస్థ ఏది?

అ) WINDOWS ఆ) UBUNTU

ఇ) FEDORA ఈ) REDHAT

5) క్రింది వాటిలో ఏది మొబైల్ పరికరాలను మద్దతు ఇస్తుంది?

అ) విండోస్ 7 ఆ) లినక్స్

ఇ) BOSS ఈ) iOS

6) ఫైల్ మేనేజర్ _____ ను నిర్వహించును

అ) ఫైల్స్

ఆ) ఫోల్డర్స్

ఇ) డైరక్టరీ వ్యవస్థలు

ఈ) పైవన్నీ

7) పరస్పర ఆపరేటింగ్ సిస్టమ్ అందించునది

అ) గ్రాఫిక్స్ యూజర్ ఇంటర్ఫేస్ (GUI)

ఆ) డేటా పంపిణీ

ఇ) భద్రతా నిర్వహణ

ఈ) రియల్ టైం ప్రాసెసింగ్

8) సింగిల్ టాస్క్ ఆపరేటింగ్ సిస్టంకు ఒక ఉదాహరణ

అ)లినక్స్

ఆ) విండోస్

ఇ) ఎంఎస్‌డాస్

ఈ) యునిక్స్

9) ఏది లినక్స్ ద్వారా ఉపయోగించబడు ఫైల్ మేనేజర్.

అ) ext2

ఆ) NTFS

ఇ) FAT

ఈ) NFTS

భాగము – II

ఒకటి లేదా రెండు వాక్యాలలో సమాధానమిమ్ము

1. ఏదైనా రెండు ఆపరేటింగ్ సిస్టమ్ యొక్క ఉపయోగాలను జాబితా పరచండి.

2. మల్టీ యూజర్ ఆపరేటింగ్ సిస్టమ్ అనగానేమి?

3. GUI అనగానేమి?

4. ఆపరేటింగ్ సిస్టమ్‌లో అందుబాటులో ఉన్న భద్రతా నిర్వహణ లక్షణాలు ఏమిటి?

5. మల్టీ ప్రాసెసింగ్ అనగానేమి?

6. కంప్యూటర్లలో ఉపయోగించబడు వివిధ రకమైన ఆపరేటింగ్ సిస్టమ్స్ ఏవి?

11th Computer Science_Chapter-4 Telugu.indd 59 15-03-2023 11:46:56

భాగము - III

సంక్షిప్త సమాధానమిమ్ము

1. సమయం – పంచుకోవటం (Time - sharing) వల్ల కలిగే ప్రయోజనాలు – అప్రయోజనాలను తెల్పుము.
2. ఆపరేటింగ్ సిస్టమ్ యొక్క ముఖ్య లక్షణాలను జాబితా పరచుము.
3. మల్టీ ప్రాసెసింగ్ గురించి ఒక గమనికను వ్రాయుము.

భాగము - IV

విపులముగా సమాధానమిమ్ము

1. డిస్ట్రిబ్యూటెడ్ ఆపరేటింగ్ సిస్టమ్ యొక్క భావనలు, ప్రయోజనాలను వివరించుము.
2. ఆపరేటింగ్ సిస్టమ్ యొక్క యూజర్ ఇంటర్ఫేస్‌ను సృష్టించేటప్పుడు గమనించవలసిన అంశాలను జాబితా పరచుము.
3. ఆపరేటింగ్ సిస్టమ్‌లో ప్రాసెస్ మేనేజ్‌మెంట్ ఆల్గోరిథంను వివరించండి.

ప్రస్తావనలు

1. Silberschatz, galvin gagne, Operating System concepts – john wiley&sons, inc
2. Andrew S. Tanenbaum, modern Operating Systems – pearson publication
3. Andrew S. Tanenbaum, Operating Systems design and implementation, prentice hall publication
4. Tom anderson, Operating Systems: principles and practice, recursive books
5. Thomas W. Doeppner, Operating Systems in depth: design and programming, john wiley & sons, inc

11th Computer Science_Chapter-4 Telugu.indd 60 15-03-2023 11:46:56

అధ్యాయము 5

విండోస్ ఆపరేటింగ్ సిస్టమ్‌తో పనిచేయుట (Working with Windows Operating System)

అభ్యసన లక్ష్యాలు

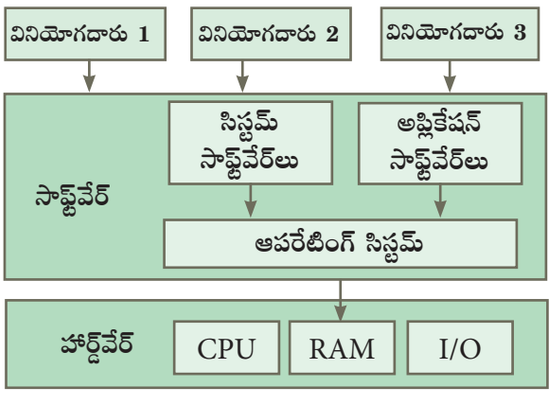
ఈ అధ్యాయంలోని భావనలను నేర్చుకొన్న తరువాత విద్యార్థులు తెలుసుకోవలసినది

- ఆపరేటింగ్ సిస్టమ్ యొక్క భావనలను తెలుసుకొనెదరు.
- విండోస్ ఆపరేటంగ్ సిస్టమ్ యొక్క సంస్కరణలను తెలుసుకొనెదరు.
- డెస్క్‌టాప్ మరియు విండో యొక్క అంశాలు వంటి భావనలను తెలుసుకొనెదరు.
- విండో డాక్యుమెంట్ అన్వేషించడం
- వివిధ రకాల చిహ్నాలను పోల్చడం
- విండోస్ డైరెక్టరీ నిర్మాణాన్ని అన్వేషించడం
- నిర్దిష్ట డ్రైవ్‌లలో ఫైళ్ళు మరియు ఫోల్డర్‌లను సృష్టించడంలో అభ్యసనం చేయడం
- ఫైళ్ళు మరియు ఫోల్డర్లను నిర్వహించడం
- కంప్యూటర్‌ని పద్ధతి ప్రకారం ప్రారంభించడానికి (start) మరియు షట్‌డౌన్ (shutdown) చేయు విధానాన్ని తెలుసుకొనుట.
- మెమొరీ నిర్వహణ
- ప్రక్రియ నిర్వహణ
- పరికర నిర్వహణ
- ఫైల్ నిర్వహణ
- భద్రతా నిర్వహణ
- మొత్తం సిస్టమ్ పనితీరును నియంత్రించుట

5.1 ఆపరేటింగ్ సిస్టమ్ యొక్క పరిచయం (Introduction to Operating System)

ఆపరేటింగ్ సిస్టమ్ (OS) అనునది సిస్టమ్ సాఫ్ట్‌వేర్. ఇది ఇతర సాఫ్ట్‌వేర్‌లతో సమాచారం అందించడానికి మరియు పనిచేయడానికి హార్డ్‌వేర్‌ను అనుమతిస్తుంది. ఇది వినియోగదారు మరియు హార్డ్‌వేర్ మధ్య ఇంటర్‌ఫేస్‌గా కూడా పనిచేస్తుంది. మరియు కంప్యూటర్ యొక్క మొత్తం విధులను నియంత్రిస్తుంది.

ఈ క్రింది కొన్ని ముఖ్యమైన ఆపరేటింగ్ సిస్టమ్ యొక్క విధులను మునుపటి అధ్యాయంలో చర్చించాము.



పటం 5.1 ఆపరేటింగ్ సిస్టమ్ యొక్క అవలోకనం

5.2 విండోస్ ఆపరేటింగ్ సిస్టమ్ పరిచయం (Introduction to Windows Operating System)

ప్రతి కంప్యూటర్‌కు పనిచేయడానికి ఒక ఆపరేటింగ్ సిస్టమ్ అవసరం. మైక్రోసాఫ్ట్ విండోస్ అనునది అత్యంత ప్రాచుర్యము పొందిన ఒక గ్రాఫికల్ యూజర్ ఇంటర్ఫేస్ (GUI). విండోస్‌లో ఏక కాలంలో బహుళ అనువర్తనాలను అమలు చేయడాన్ని మల్టీ టాస్కింగ్ (Multitasking) అంటారు. విండోస్ ఆపరేటింగ్ సిస్టమ్ కీబోర్డు మరియు మౌస్ వంటి రెండు ఇన్‌పుట్ పరికరాలను ఉపయోగిస్తుంది.

మౌస్ దానిని క్లిక్ చేయుట ద్వారా విండోస్‌తో సంభాషించడానికి ఉపయోగపడుతుంది. వర్ణమాలలు, సంఖ్యలు మరియు ప్రత్యేక అక్షరాలు నమోదు చేయడానికి కీబోర్డ్‌ని ఉపయోగిస్తారు.

11th Computer Science_Chapter-5 Telugu.indd 61 30-03-2023 18:28:34

విండోస్ ఆపరేటింగ్ సిస్టమ్ యొక్క కొన్ని విధులు:

- కంప్యూటర్‌లో అనువర్తనాలను (programs) యాక్సెస్ చేయుట (access) (వర్డ్ ప్రాసెసింగ్, ఆటలు, స్ప్రెడ్ షీట్లు, క్యాలికులేటర్లు మొదలైనవి)
- కంప్యూటర్‌లో ఏదైనా కొత్త ప్రోగ్రామ్‌ను లోడ్ చేయడం
- ముద్రణా యంత్రం, స్కానర్లు, మౌస్, డిజిటల్ కెమెరాలు మొదలైనటువంటి హార్డ్‌వేర్‌లను నిర్వహించుట.
- ఫైల్ నిర్వహణా కార్యకలాపాలు (ఉదాహరణకు, సృష్టించుట, సవరించుట, భద్రపరచుట, ఫైళ్ళు మరియు ఫోల్డర్లను తొలగించుట)
- మానిటర్ యొక్క స్క్రీన్ సేవర్స్ మొదలైనటువంటి కంప్యూటర్ సెట్టింగులను మార్చుట.

పట్టిక 5.1 ఆధారంగా విండోస్ ఆపరేటింగ్ సిస్టమ్ యొక్క రూపాంతరములను చూద్దాం.

5.4 విండోస్ యొక్క వివిధ సంస్కరణలు (Handling the mouse)

విండో ఆపరేటింగ్ సిస్టమ్ గురించి నేర్చుకునే ముందు మీరు మౌస్ మరియు దాని చర్యలు గురించి మరింత తెలుసుకొనవలెను.

5.3 విండోస్ యొక్క వివిధ సంస్కరణలు (Various versions of Windows)

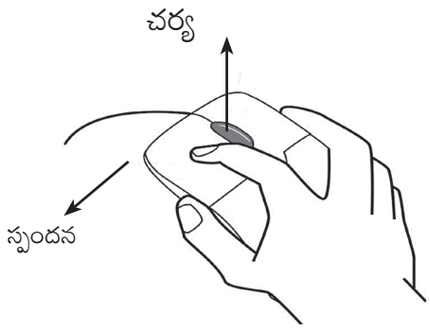
సంస్కరణలు	లోగో	సంవత్సరం	నిర్దిష్ట లక్షణాలు
Windows 1.x		1985	• ప్రాసెసర్ 16 బిట్‌లలో GUI యొక్క పరిచయం • మౌస్ ఇన్పుట్ పరికరంగా పరిచయం చేయబడింది
Windows 2.x		1987	• విండోలను కనిష్టీకరించడానికి లేదా పెంచడానికి మద్ధత్తునిస్తుంది. • కంట్రోల్ పేనెల్ ఫ్యీచర్ వివిధ సిస్టమ్ సెట్టింగులు మరియు అనుకూలీకరించే ఎంపికలతో పరిచయం చేయబడింది.
Windows 3.x		1992	• బహుళ విధి (multitasking) భావనను పరిచయం. • 256 రంగులు మరింత ఆధునికమైన రంగులు ఇంటర్ఫేస్‌కు రూపకల్పనను తెచ్చి పెట్టింది.
Windows NT		1993	• స్టార్ట్ బటన్, టాస్క్ బార్, విండోస్ ఎక్స్‌ప్లోరర్ మరియు స్టార్ట్ మెను పరిచయం • నెట్‌వర్క్‌లో సర్వర్‌లుగా పనిచేయడానికి రూపొందించబడింది.
Windows 95		1995	• 32 - బిట్ ప్రాసెసర్‌ను పరిచయం చేయబడింది మరియు మల్టీ టాస్కింగ్‌పై ఎక్కువ దృష్టి పెట్టారు.
Windows NT		1998	• DOS - ఆధారిత ఆటలు అదృశ్యమై విండోస్ ఆధారిత ఆటలు మెరుగుపడ్డాయి. • ప్లగ్ మరియు ప్లే ఫీచర్ ప్రవేశపెట్టబడింది.
Windows Me		2000	• ఆటోమెటిక్ సిస్టమ్ డయాగ్నొస్టిక్స్ మరియు రికవరీ సాధనాలు పరిచయం చేయబడినది.

11th Computer Science_Chapter-5 Telugu.indd 62 30-03-2023 18:28:36

Windows 2000		2000	• వ్యాపార డెస్క్‌టాప్ మరియు లాప్‌టాప్ వ్యవస్థల కోసం ఆపరేటింగ్ సిస్టమ్‌గా పనిచేసినది. • విండోస్ 2000 లో నాలుగు వెర్షన్లు విడుదల చేయబడింది.: Professional (వ్యాపార డెస్క్‌టాప్ మరియు లాప్‌టాప్ వ్యవస్థ) Server Webserver and an office server Advanced Server వ్యాపార ఉపయోగాల కోసం Data Centre Server అధిక ట్రాఫిక్ కంప్యూటర్ నెట్‌వర్క్‌ల కోసం.
Windows XP		2001	• 64- బిట్ ప్రాసెసర్ పరిచయం చేయబడినది. • మెరుగైన విండోస్ (appearance) ప్రదర్శన మరియు స్థిరమైన (themes) సంస్కరణను అందించింది.
Windows Vista		2006	• విండోస్ యొక్క రూపాన్ని మరియు అనుభూతిని నవీకరించారు.
Windows 7		2009	• బూటింగ్ సమయం మెరుగుపరచబడింది. ఏరో పీక్ (Aero Peek) పిన్నింగ్ ప్రోగ్రామ్స్ టు టాస్క్‌బార్ (pinning programs to taskbar) చేతిరాత గుర్తింపు (handwriting recognition) మొ॥ మరియు ఇంటర్‌నెట్ ఎక్స్‌పోరర్ (Internet Explorer) వంటి క్రొత్త వినియోగదారు ఇంటర్‌పేస్ కోసం పరిచయం చేయబడింది.
Windows 8		2012	• విండోస్ యొక్క మునుపటి సంస్కరణల కంటే విండోస్ 8 వేగంగా ఉంటుంది. • ప్రారంభ బటన్ తొలగించబడింది. • విండోస్ Windows 8 లో మల్టీకోర్ ప్రాసెసింగ్, సాలిడ్ స్టేట్ డ్రైవ్స్ (SSD) టచ్ స్క్రీన్లు మరియు ఇతర ప్రత్యామ్నాయ ఇన్పుట్ పద్ధతులు వంటి మంచి ప్రయోజనాలు పొందుతుంది. • మొబైల్ మరియు కంప్యూటర్ల రెండింటికి ఒకటే వేదికగా ఉపయోగపడింది.
Windows 10		2015	• ప్రారంభ బటన్ మళ్ళీ జోడించబడింది. • మల్టిపుల్ డెస్క్‌టాప్ / బహుళ డెస్క్‌టాప్ అనువర్తన నోటిఫికేషన్ కోసం మరియు శీఘ్రు చర్యల కోసం సెంట్రల్ నోటిఫికేషన్ • కార్టానా వాయిస్ యాక్టివేటెడ్ పర్సనల్ అసిస్టెంట్.

పట్టిక 5.1 విండోస్ ఆపరేటింగ్ సిస్టమ్ యొక్క చర్యలు (*Versions of Windows Operating System).*

ఈ క్రింద ఇవ్వబడినవి మౌస్ యొక్క చర్యలు:



పట్టిక 5.2 మౌస్ చర్యలు

చర్య	స్పందన
ఒక అంశాన్ని గుర్తించుట	మౌస్ పాయింటర్‌ను ఒక అంశంపై తరలించడం
క్లిక్	తెరపై ఉన్న ఒక అంశంపై మౌస్ పాయింటర్‌ను ఉంచి ఎడమ బటన్‌ని నొక్కి వెంటనే విడుదల చేయుట.
కుడి క్లిక్	ఒక అంశంపై మౌస్ పాయింటర్‌ను ఉంచి కుడి బటన్‌ను నొక్కి వదిలిన తరువాత వివిధ ఎంపికలతో కూడిన పాప్ అప్ మెనుని ప్రదర్శిస్తుంది.
డబుల్ క్లిక్	తెరపై ఉన్న అంశంపై మౌస్ పాయింటర్‌ను ఉంచి త్వరగా ఎడమ బటన్‌ని రెండుసార్లు నొక్కుట.

11th Computer Science_Chapter-5 Telugu.indd 63 30-03-2023 18:28:37

లాగుట మరియు వదలుట	ఒక అంశంపై మౌస్ పాయింటర్‌ను ఉంచి మౌస్ యొక్క ఎడమ బటన్ నొక్కి పట్టుకుని మనం కోరుకున్న స్థితికి చేరాక విడుదల చేయుట

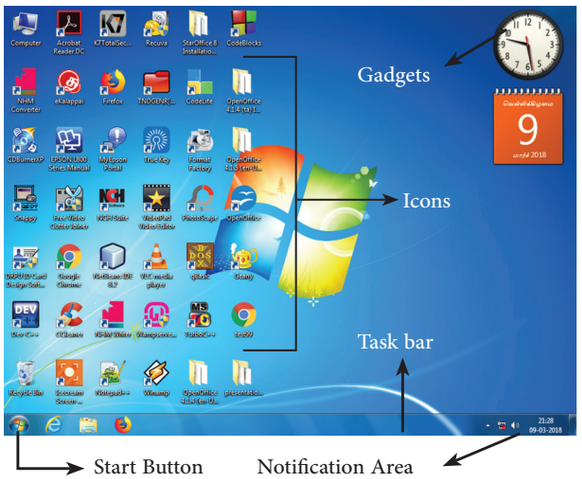
5.5 విండోస్ డెస్క్‌టాప్ (Windows Desktop)

విండోస్ యొక్క ప్రారంభ తెరను "డెస్క్‌టాప్" అని పిలుస్తారు.

మీ కంప్యూటర్ యొక్క డెస్క్‌టాప్ పటం 5.3 లో కనిపించే దానికి భిన్నంగా ఉండవచ్చు.

ఇందుకు కారణం డెస్క్‌టాప్ రూపాన్ని మార్చడానికి విండోస్ మిమ్మల్ని అనుమతిస్తుంది.

పటం 5.3, లో డెస్క్‌టాప్ ప్రారంభ బటన్, టాస్క్ బార్, నోటిఫికేషన్ ప్రాంతం, తేది మరియు సమయాన్ని చూపిస్తుంది.



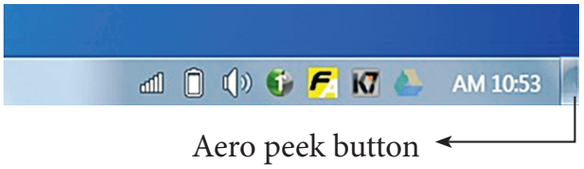
పటం 5.3 మైక్రోసాఫ్ట్ విండోస్ 7 డెస్క్‌టాప్ (Microsoft WindowsDesktop)

5.5.1 చిహ్నాలు (The Icons)

(The Icon) చిహ్నము అనునది విండో యొక్క అంశాలైన ఫైల్స్ ఫోల్డర్స్ సత్వర మార్గాలు మొదలగు రేఖా చిహ్నములగును. (GUI) ఆధారిత అనువర్తనములలో ఐకాన్స్ / చిహ్నాలు కీలక పాత్రను పోషిస్తాయి.

5.5.1.1 స్టాండర్డ్ చిహ్నాలు (ప్రామాణిక చిహ్నాలు) (Standard Icons)

విండోస్ (OS) opening Screen ని ఇన్‌స్టాల్ చేయునపుడు అప్రమేయంగా డెస్క్‌టాప్ పై అందుబాటులో ఉన్న చిహ్నాలను ప్రామాణిక చిహ్నాలు అంటారు. మై కంప్యూటర్, డాక్యుమెంట్స్ మరియు రీసైకిల్‌బిన్ లాంటివి అన్నీ విండోస్ (OS) లలో కనిపించు ప్రామాణిక చిహ్నాలు అగును.



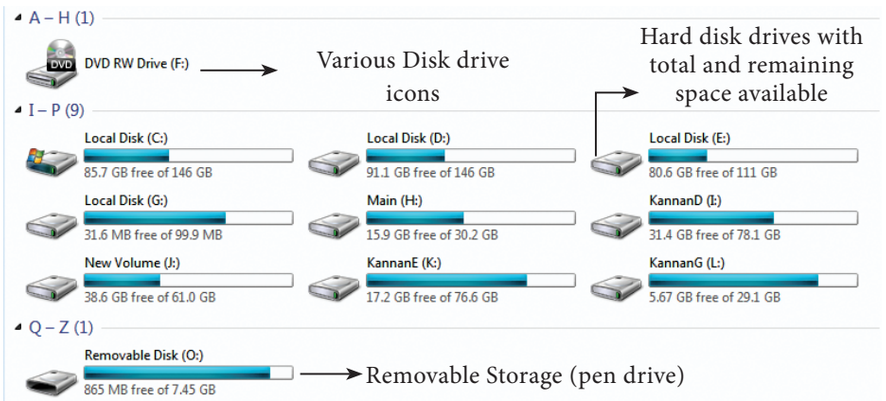
పటం 5.4 ఏరోపీక్ బటన్ (Aero peek button)

5.5.1.2 షార్ట్‌కట్ చిహ్నాలు (సత్వర మార్గ చిహ్నాలు) (Shortcut Icons)

ఏదైనా అనువర్తన లేదా ఫైల్ లేదా ఫోల్డర్ కోసం సత్వర మార్గ చిహ్నాలను సృష్టించవచ్చు. ఒక చిహ్నాన్ని రెండుసార్లు క్లిక్ చేయుట ద్వారా సంబంధిత అనువర్తన లేదా ఫైల్ లేదా ఫోల్డర్ తెరవబడుతుంది. (పటం 5.5)



పటం 5.5 వివిధ రకాల చిహ్నాలు (The types of Icons)



పటం 5.6 డిస్క్ డ్రైవ్ చిహ్నాలు

11th Computer Science_Chapter-5 Telugu.indd 64 30-03-2023 18:28:38

5.5.1.3 డిస్క్ డ్రైవ్ చిహ్నాలు (Disk drive icons)

డిస్క్ డ్రైవ్ చిహ్నాలు తద్రూపముగా 5 డిస్క్ డ్రైవ్ ఎంపికలను సూచిస్తుంది. (i) హార్డ్ డిస్క్ (ii) CD-ROM/DVD డ్రైవ్ (iii) పెన్ డ్రైవ్ (iv) (మొబైల్ స్మార్ట్‌ఫోన్, టాబ్లెట్) వంటి ఇతర తొలగించగల నిల్వలు అయిన (v) ఇతర సిస్టమ్‌తో మీ సిస్టమ్‌ని కనెక్ట్ చేసినప్పుడు వాటి యొక్క నెట్‌వర్క్ డ్రైవ్స్ ఉపయోగపడును.

మీకు తెలుసా?

Winkey + D ని నొక్కడం ద్వారా ఏదైనా అనువర్తనములో పనిచేయునపుడు ఏ సమయములోనైనను ఏరో పీక్ ఉపయోగించి ఎప్పుడైనా డెస్క్‌టాప్‌కు వెళ్ళవచ్చు టాస్క్‌బార్‌లో ఏరో పీక్ ఎక్కడ ఉంటుందో పటం 5.4 లో చూడవచ్చు.

5.6 విండో (The Window)

విండో అనునది ఒక అనువర్తనము లేదా ఒక పత్రము (document) లో సాధారణ దీర్ఘ చతురస్రాకార ప్రాంతం. ఇది తెరపై నిర్దిష్ట ప్రోగ్రామ్ కోసం సమాచారాన్ని ప్రదర్శించుటకు ఉపయోగపడు ప్రాంతమగును.

5.7 అనువర్తన విండో (Application Window)

ఇది కంప్యూటర్ తెరపై నిర్వచించిన సరిహద్దులలో ఉన్న ప్రాంతం, ఈ ప్రాంతంలో సమాచారం ప్రదర్శించబడుతుంది. ఇటువంటి విండోల యొక్క పరిమాణాన్ని మార్చవచ్చు, గరిష్టీకరించవచ్చు, కనిష్టీకరించవచ్చు, పక్కపక్కనే ఉంచవచ్చు, ఒక దానిపై ఒకటి ఉంచవచ్చు మొదలైనవి.

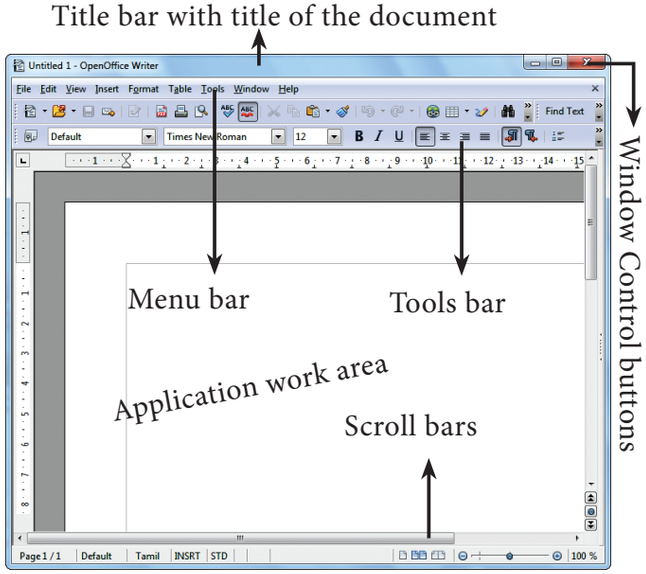
ఒక అనువర్తన విండో (Application Window) లో ఒక open application అంటే అవి ప్రస్తుత అనువర్తనములైన వర్డ్ (Word) లేదా పెయింట్ (Paint) ఉంటుంది. రెండు లేదా అంతకంటే ఎక్కువ విండోలు తెరవబడినపుడు వాటిలో ఒకటి మాత్రమే చురుకుగా ఉంటుంది మరియు మిగిలినవి క్రియారహితంగా ఉంటుంది. పటం 5.6 మరియు 5.7 లో ఓపెన్ ఆఫీస్ రైటర్ యొక్క అనువర్తన విండో మరియు డెస్క్‌టాప్‌లో తెరవబడిన బహుళ విండోస్‌ని (overlapped) చూడవచ్చు.

5.8 పత్రం విండో (Document Window)

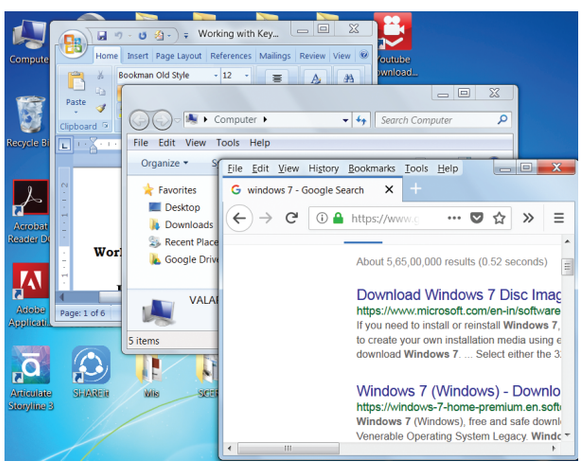
డాక్యుమెంట్ విండో (document window) అనునది డాక్యుమెంట్ యొక్క అంశాలను ప్రదర్శించడానికి ఉపయోగించే తెర యొక్క ఒక విభాగం. పటం 5.8 డాక్యుమెంట్ విండో యొక్క ఉదాహరణ అగును.

గమనిక

ఓపెన్ ఆఫీస్ రైటర్ ఓపెన్ ఆఫీస్ ఇంప్రెస్ లేదా ఓపెన్ ఆఫీస్ కాల్క్ మొదలైన వాటిని తెరవబడినప్పుడు తెరపై రెండు విండోలను కనుగొంటాము. పెద్ద విండోను అనువర్తన విండో అని అంటారు. ఈ విండో అనువర్తన ప్రోగ్రామ్‌తో కమ్యూనికేట్ చేయడానికి వినియోగదారుకు సహాయపడుతుంది. అనువర్తన విండో లోపల గల చిన్న విండోను పత్రం విండో (Document window) అని అంటారు. ఈ విండో టైప్ చేయడానికి సవరించుటకు గీయుటకు మరియు వచనం మరియు గ్రాఫిక్స్‌ని ఆకృతీకరించుటకు ఉపయోగించబడుతుంది.

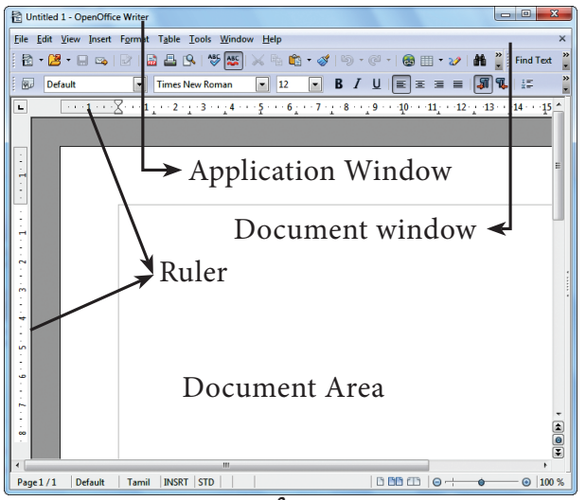


పటం 5.6 అనువర్తన విండో (Application Window)



పటం 5.7 బహుళ విండోలు తెరవబడిన డెస్క్‌టాప్ (Multiple Windows opened in Desktop)

11th Computer Science_Chapter-5 Telugu.indd 65 30-03-2023 18:28:38

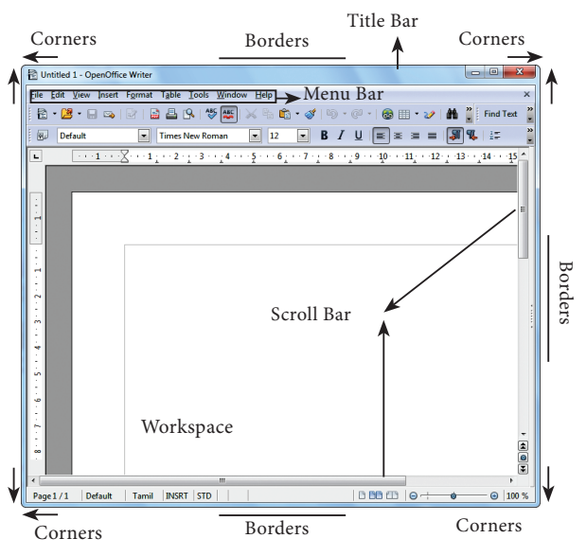


పటం 5.8 పత్రం విండో (Document Window)

5.9 విండో యొక్క అంశాలు (Elements of a window)

విండో యొక్క అంశాలను అర్థం చేసుకోవడానికి పటం 5.9 సహాయపడుతుంది.

5.9.1. టైటిల్ బార్ (Title Bar) శీర్షిక పట్టి : అనువర్తనం యొక్క పేరును మరియు తెరవబడిన పత్రం యొక్క పేరును శీర్షిక పట్టినీ (title bar) ప్రదర్శిస్తుంది. ఇందులో కనిష్టీకరించు గరిష్టీకరించు మరియు మూసివేయు బటన్‌లు ఉండును.

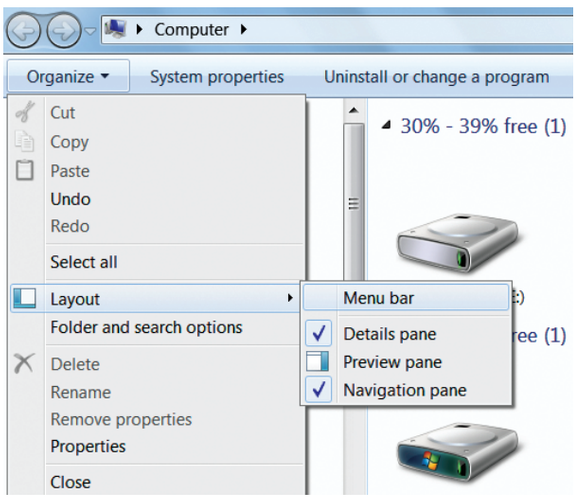


పటం 5.9 విండో యొక్క అంశాలు (The elements of a window)

5.9.2 మెనూ బార్ (Menu Bar)

మెను పట్టిక శీర్షిక పట్టి (title bar) క్రింద కనిపిస్తుంది. మెనుబార్‌లోని మెనూలను Alt కీ మరియు మెను శీర్షికలో అండర్‌లైన్ చేయబడిన అక్షరాన్ని నొక్కుట ద్వారా యాక్సెస్ చేయవచ్చు. అదనంగా Alt లేదా F10 బటన్ నొక్కుట ద్వారా, మెనూ బార్ యొక్క మొదటి మెనుపై దృష్టిని తెస్తుంది.

విండోస్ 7 లో మెను బార్ లేనప్పుడు **Organise** బటన్‌ను క్లిక్ చేసి డ్రాప్ డౌన్ మెను నుండి **Layout** ఎంపికను ఎంచుకొని / క్లిక్ చేసి మరియు ఆ జాబితా నుండి కావలసిన అంశాన్ని ఎంచుకోవచ్చు.



పటం 5.10 మెను బార్ ప్రదర్శన (To display Menu Bar)

మెనుబార్ లేనప్పుడు దానిని ఎలా ప్రదర్శింపచేయాలో తెలుసుకొనుటకు / అర్థం చేసుకొనుటకు పటం 5.10 సహాయపడుతుంది.

5.9.3 కార్య స్థలం (The Workspace)

డాక్యుమెంట్ విండో (document window) లో మీ పత్రం యొక్క వచనాన్ని నమోదు చేయడానికి లేదా టైప్ చేయుటకు గల ప్రాంతాన్ని కార్యస్థలం (workspace) అని అంటారు. పత్రం విండోలో కార్యస్థల ప్రాంతాన్ని పటం 5.9 సూచిస్తుంది.

5.9.4 స్క్రోల్ బార్స్ – కార్యస్థలాన్ని నిలువుగా లేదా అడ్డంగా స్క్రోల్ చేయుటకు స్క్రోల్ బార్స్ (Scroll bars) ఉపయోపడును. పటం 5.9 లో స్క్రోల్ బార్స్‌ని చూడవచ్చు.

5.9.5 మూలలు మరియు అంచులు (Corners and borders)

విండోస్ యొక్క పరిమాణాన్ని తగ్గించుటకు మరియు ఒక ప్రక్క నుండి ఇంకో ప్రక్కకి లాగుటకు విండో యొక్క మూలలు మరియు అంచులు సహాయపడును. ఒక అంచు లేదా మూలలపై మౌస్ పాయింటర్‌ను ఉంచినపుడు అది ఇరుప్రక్కల బాణపుగుర్తు ఉండునట్లుగా మారును. పటము 5.9 లో

11th Computer Science_Chapter-5 Telugu.indd 66 30-03-2023 18:28:38

చూపిన విధంగా ఇరుప్రక్కల బాణపుగుర్తును అంచు లేదా మూలలపై ఉంచి అది సూచించిన దిశవైపు లాగిన విండో తెర యొక్క పరిమాణం మనకు కావలసిన పరిమాణమునకు మారును. విండో యొక్క మూలలలో ఇరుప్రక్కల బాణపుగుర్తుని ఉంచి వికర్ణముగా లాగినపుడు విండో యొక్క పరిమాణం మార్చవచ్చు.

5.10.1 ప్రారంభ మెను (Start Menu)

విండోస్ తెర దిగువ ఎడమ చేతి మూలలో ప్రారంభ (Start) బటన్ ఉంటుంది. దానిని నొక్కినపుడు ప్రారంభ మెను కనిపిస్తుంది. ప్రారంభ మెనుని ఉపయోగించి మీరు ఏ అనువర్తనమునైనా ప్రారంభించవచ్చు.



పటము 5.11 ప్రారంభ మెను (Start Menu)

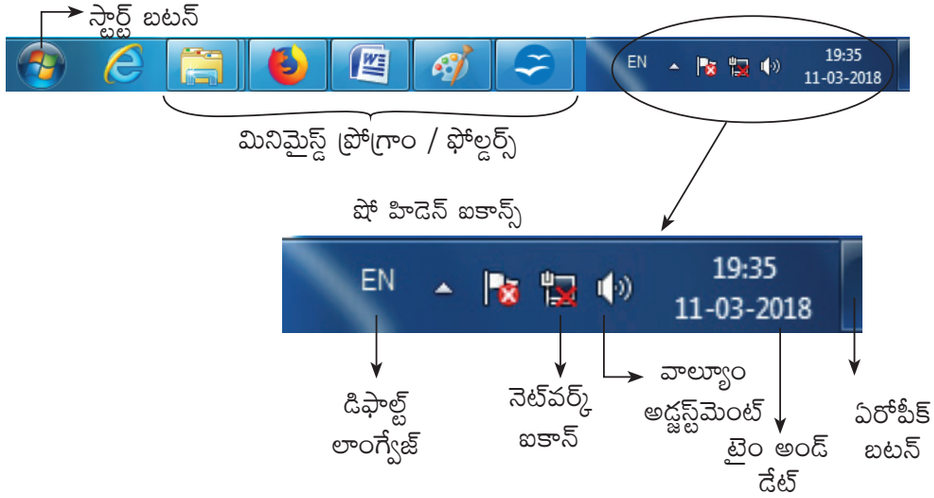
టాస్క్ బార్ (Taskbar)

తెర యొక్క దిగువ భాగమున టాస్క్‌బార్ అని పిలువబడే క్షితిజ సమాంతర పట్టీ ఉంటుంది. ఈ పట్టీలో (ఎడమ నుండి కుడి) ప్రారంభ బటన్ (Start button), వివిధ ప్రోగ్రామ్‌లకు సత్వర మార్గాలు (shortcuts), కనిష్టీకరించబడిన ప్రోగ్రామ్‌లు ఉండును. మరియు కుడి మూలలో సిస్టమ్ ట్రేని చూడవచ్చు, అందులో శబ్ద నియంత్రణ (volume control) నెట్‌వర్క్, తేది మరియు సమయం ఉండును. ప్రారంభ బటన్ తరువాత త్వరిత ప్రయోగ ఉపకరణ పట్టి (quick Launch Toolbar) ఉండును. ఇందులో తరుచుగా ఉపయోగించే అనువర్తనాల కోసం టాస్క్ ఉంటుంది.

5.10.2 కంప్యూటర్ చిహ్నం (Computer Icon)

ఈ చిహ్నాన్ని క్లిక్ చేయడం ద్వారా సిస్టమ్‌లో అమర్చిన డిస్క్ డ్రైవర్లను వినియోగదారు చూడవచ్చు. విండోస్ ("My computer") విస్టాలలో ఈ చిహ్నాన్ని దిస్ పీసి ("This PC") అని పిలుస్తారు. పటం 5.13 లో చూపిన విధంగా కంప్యూటర్ ఐకాన్ కార్యాచరణ విండోస్ యొక్క అన్ని సంస్కరణలలోను ఒకే విధంగా ఉంటుంది.

11th Computer Science_Chapter-5 Telugu.indd 67 30-03-2023 18:28:38



*పటం 5.12.*టాస్క్‌బార్

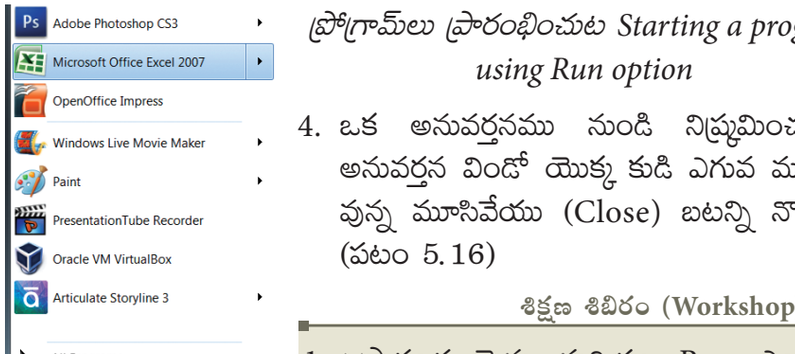


పటం 5.13 విండోస్ OS యొక్క సంస్కరణలలో కంప్యూటర్ చిహ్నం Computer icon in versions of Windows

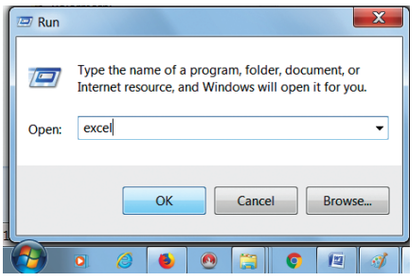
5.10.3. అనువర్తనాలను ప్రారంభించడం మరియు మూసివేయడం (Starting and Closing Applications)

మీ కంప్యూటర్‌లో వ్యవస్థాపించబడ్డ చాలా అనువర్తనాలు ప్రారంభ మెను ద్వారా అందుబాటులో ఉంటాయి. సిస్టమ్ సెటప్ (system setup) ను అనుసరించి ప్రారంభ మెనులోని అనువర్తనాలు వేరుబడును. ఒక అనువర్తనాన్ని ప్రారంభించడానికి:

1. Start బటన్‌ని నొక్కి All Programs లో మౌస్ ఉంచవలయును. ప్రోగ్రామ్ మెను కనిపిస్తుంది. (పటం 5.14)
2. మీరు ప్రారంభించాలనుకునే అనువర్తనము కలిగిన సమూహంపై మౌస్ పాయింటర్‌ను ఉంచి ఆ తరువాత అనువర్తన పేరును నొక్కండి.
3. ప్రారంభ మెనులో ఉన్న Run అనుదానిని నొక్కి మీకు కావలసిన అనువర్తన పేరును టైప్ చేసిన తర్వాత కూడా ఒక అనువర్తనమును తెరవవచ్చు.



పటం 5.14 ప్రారంభ మెనుని ఉపయోగించి ఒక అనువర్తనాన్ని ప్రారంభించుట



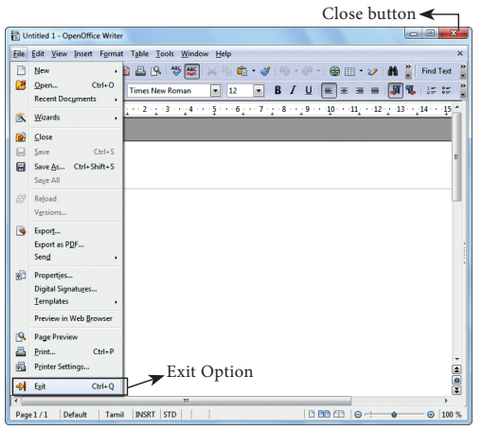
పటం 5.15 Run ఎంపికను ఉపయోగించి ఒక ప్రోగ్రామ్‌లు ప్రారంభించుట Starting a program using Run option

4. ఒక అనువర్తనము నుండి నిష్క్రమించడానికి అనువర్తన విండో యొక్క కుడి ఎగువ మూలలో వున్న మూసివేయు (Close) బటన్‌ని నొక్కండి. (పటం 5.16)

శిక్షణ శిబిరం (Workshop)

1. ◆ ప్రారంభ మెను మరియు Run ఎంపికను ఉపయోగించి అనువర్తన వర్డ్‌పాడ్ (application Wordpad) ను ప్రారంభించండి.
   ◆ ఫైల్ మెనుని ఉపయోగించి Wordpad అనువర్తనాన్ని మూసివేయండి.

11th Computer Science_Chapter-5 Telugu.indd 68 30-03-2023 18:28:39



పటం 5.16 మూసివేయు బటన్ మరియు నిష్క్రమణ (Exit) ఎంపికను ఉపయోగించి ఒక అనువర్తనాన్ని మూసివేయుట (Closing the application using Close button and Exit option)

5. (Windows 7) విండో 7 లో గల File → Exit మరియు File → Close అను ఎంపికపై నొక్కుట ద్వారా కూడా ఒక అనువర్తనము నుండి నిష్క్రమించవచ్చును.

5.11 ఫైల్స్ మరియు ఫోల్డర్లను నిర్వహించుట Managing Files and Folders

విండోస్ 7 లో మీ పత్రాలు మరియు ప్రోగ్రామ్‌లను ఫైల్స్ మరియు ఫోల్డర్ల రూపములో నిర్వహించవచ్చును. ఇందులో తరలించవచ్చు, కాపీ చేయవచ్చు, పేరు మార్చవచ్చు, తొలగించవచ్చు మరియు ఫైల్స్ మరియు ఫోల్డర్లను వెతకవచ్చు / శోధించవచ్చు.

5.11.1 ఫైల్స్ మరియు ఫోల్డర్లను సృష్టించుట (Creating files and Folders)

5.11.1.1 ఫోల్డర్లను సృష్టించుట (Creating Folders)

మీరు మీ ఫైళ్ళను చాలా స్థానాలలో నిల్వచేయవచ్చు అవి హార్డ్‌డిస్క్ లేదా ఇతర పరికరాలు. మీ ఫైళ్ళను బాగా నిర్వహించడానికి వాటిని ఫోల్డర్లలో నిల్వ చేయవచ్చు.

కొత్త ఫోల్డర్లను సృష్టించడానికి రెండు మార్గాలు కలవు.

విధానం 1:

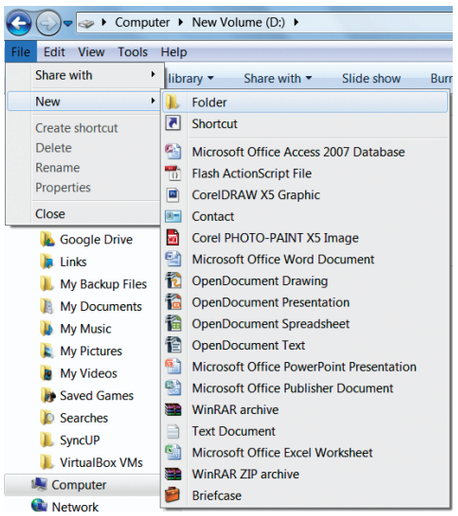
దశ 1 : **Computer Icon** తెరవండి

దశ 2 : కొత్త ఫోల్డర్‌ను సృష్టించడానికి మీకు కావలసిన చోట ఏదైనా డ్రైవ్ తెరవండి. (ఉదా॥ D ఎంచుకోండి). పటము: 5.17

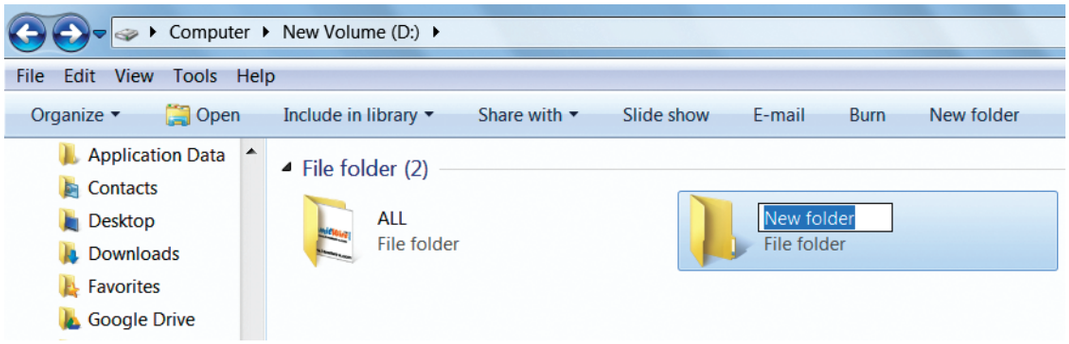
దశ 3 : File → New → Folder పై నొక్కండి.

దశ 4 : పటము: 5.18 New folder పేరుతో అని సృష్టించబడుతుంది.

దశ 5 : ఫోల్డర్ పేరును టైప్ చేసి ఎంటర్ (Enter) కీని నొక్కండి (కొత్తగా సృష్టించిన ఫోల్డర్‌ని "Test Folder" అని పటం 5.19 చూపిస్తుంది)

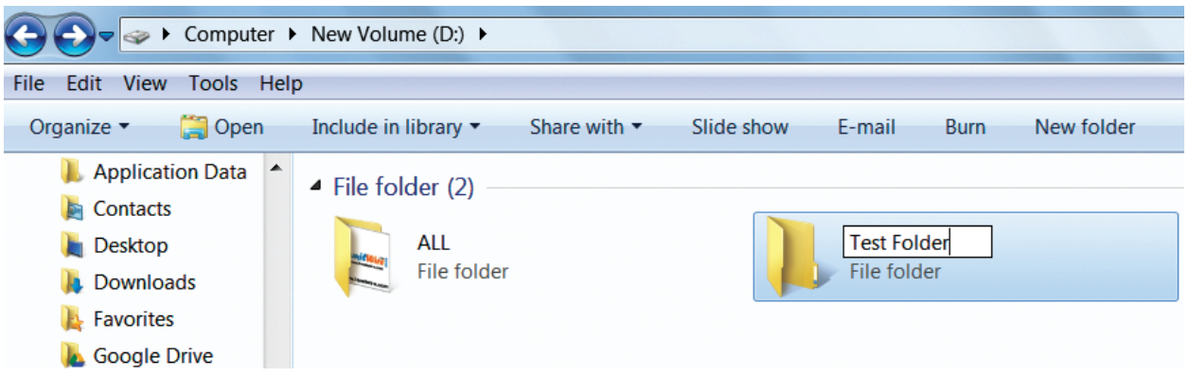


పటం 5.17 ఫైల్ మెనుని ఉపయోగించి ఒక ఫోల్డర్‌ను సృష్టించుట (Creating a Folder using File menu)



పటం 5.18 డిఫాల్ట్ పేరుతో కొత్త ఫోల్డర్‌ను సృష్టించబడును (New Folder created with the default name)

11th Computer Science_Chapter-5 Telugu.indd 69 30-03-2023 18:28:39



పటం 5.19 కొత్త ఫోల్డర్ యొక్క పేరు మార్చుట (Renaming the new Folder)

విధానం 2 :

డెస్క్‌టాప్‌లో ఒక ఫోల్డర్‌ని సృష్టించడానికి

దశ 1 : డెస్క్‌టాప్‌లో మౌస్ యొక్క కుడి బటన్‌ను నొక్కి New Folder ని నొక్కండి (పటం 5.20 లో చూపబడిన విధంగా)

దశ 2 : పేరు పెట్టబడని ఒక New folder కనిపిస్తుంది మరియు అది పటం 5.21 లో చూపబడిన విధంగా హైలైట్ చేయబడి వుంటుంది.

దశ 3 : మీకు కావలసిన పేరును టైప్ చేసి Enter Key ని నొక్కండి.

దశ 4 : ఫోల్డర్ పేరు మారుతుంది.

శిక్షణ శిబిరం

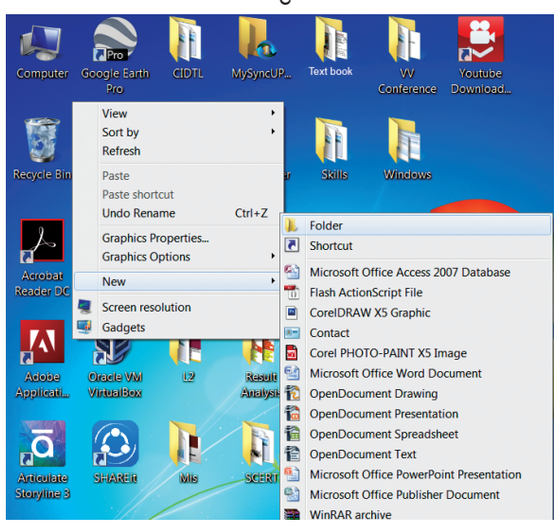
2. My Documents లో మీరు నేర్చుకున్న ఏదైనా ఒక పద్ధతిలో మీ పేరుతో ఒక ఫోల్డర్‌ని సృష్టించండి.

5.11.1.2 ఫైళ్ళను సృష్టించుట (Creating Files) (Wordpad)

Wordpad అనేది విండోస్ OS లో అంతర్నిర్మితమైన వర్డ్ ప్రాసెసర్ అనువర్తనం. ఇది వచన పత్రాలను సృష్టించుటకు మరియు మార్చుటకు ఉపయోగపడుతుంది.

wordpad లో ఫైళ్ళను సృష్టించుటకు ఈ క్రింది దశలను అనుసరించవలయును.

1. Start → All Programs → Accessories → Wordpad లేదా Run → type Wordpad, అని టైప్ చేసి OK ను నొక్కండి. పటం 5.22 లో చూపిన విధంగా Wordpad విండో తెరవబడుతుంది.
2. కార్యస్థలములో workspace విషయాలను టైప్ చేసి File → Save లేదా Ctrl + S ను ఉపయోగించి ఫైళ్ళను భద్రపరచుము.
3. Save As అనే డైలాగ్ బాక్స్ తెరవబడుతుంది.
4. డైలాగ్ బాక్స్‌లో డ్రాప్ డౌన్ జాబితా పెట్టెలో **look in** ని ఉపయోగించి మీ ఫైల్‌ని ఎక్కడ భద్రపరచాలనుకుంటున్నారో ఆ స్థానాన్ని ఎంచుకోండి.
5. **File name** అనే టెక్స్ట్ బాక్స్‌లో పేరును టైప్ చేయండి.
6. save బటన్‌ని నొక్కండి.

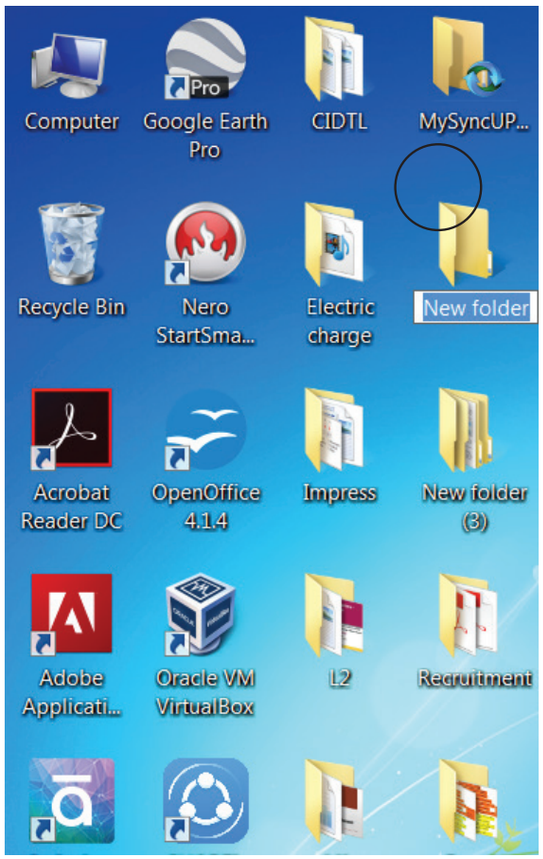


పటం 5.20 డెస్క్‌టాప్‌లో ఒక ఫోల్డర్‌ని సృష్టించుట

శిక్షణ శిబిరం

3. Wordpad ను ఉపయోగించి ఒక ఫైల్ సృష్టించి దానిని My Documents లో మీ పేరుతో సృష్టించబడిన ఫోల్డర్‌లో భద్రపరచండి.

11th Computer Science_Chapter-5 Telugu.indd 70 30-03-2023 18:28:39



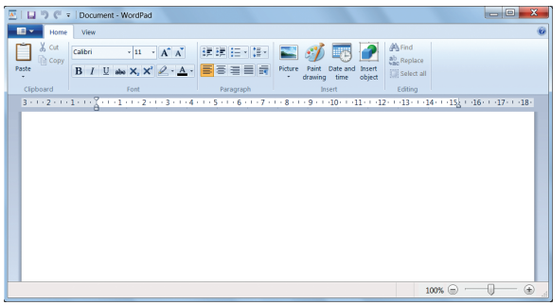
పటం 5.21 డెస్క్‌టాప్ మీద కొత్త ఫోల్డర్ చిహ్నం
(New folder icon on the dektop)

5.11.2. ఫైళ్ళు మరియు ఫోల్డర్లను కనుగొనడం (Finding Files and Folders)

కంప్యూటర్ లేదా నిర్ధిష్ట డ్రైవ్‌లో ఒక నిర్ధిష్ట ఫోల్డర్ లేదా ఫైల్‌ను త్వరగా శోధించడానికి **స్టార్ మెను**లోని సెర్చ్ బాక్స్‌ని ఉపయోగించవచ్చు.

ఫైల్ లేదా ఫోల్డర్‌ను కనుగొనడానికి

1. **Start** బటన్ని నొక్కండి. ప్రారంభ మెను యొక్క క్రింది భాగంలో **search** బాక్స్ కనిపిస్తుంది.
2. శోధించవలసిన ఫైల్ పేరు లేదా ఫోల్డర్‌ను టైప్ చేయండి ఫైల్ లేదా ఫోల్డర్ పేరు యొక్క కొంత భాగాన్ని టైప్ చేసినా కూడా నిర్ధిష్ట పేరుతో ప్రారంభమయ్యే అన్ని ఫైళ్ళు లేదా ఫోల్డర్ల జాబితాను ప్రదర్శిస్తుంది. (పటం 5.23)
3. నిర్ధిష్ట పేరుతో గల ఫైళ్ళు లేదా ఫోల్డర్లు కనిపిస్తాయి. ఆ ఫైల్ పైన నొక్కినచో ఆ ఫైల్ లేదా ఫోల్డర్ నేరుగా తెరవబడుతుంది.

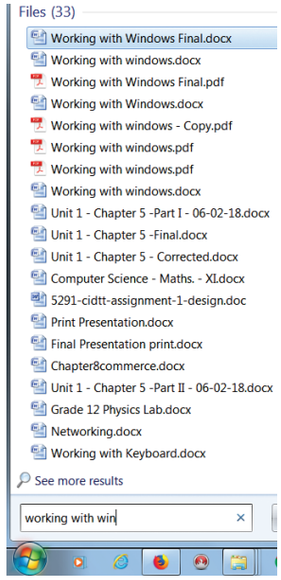


పటం 5.22 వర్డ్‌పాడ్ – వర్డ్ ప్రాసెసర్ అప్లికేషన్
(Wordpad - Word Processor application)

4. **search** బాక్స్‌కి పైన "**See more results**" అనే మరొక ఎంపిక కనిపిస్తుంది.
5. దానిని నొక్కిన **Search Results** అను డైలాగ్ బాక్స్ కనిపించును. దీని ద్వారా ఫైల్ లేదా ఫోల్డర్‌ని వెతికి తెరవవచ్చును.

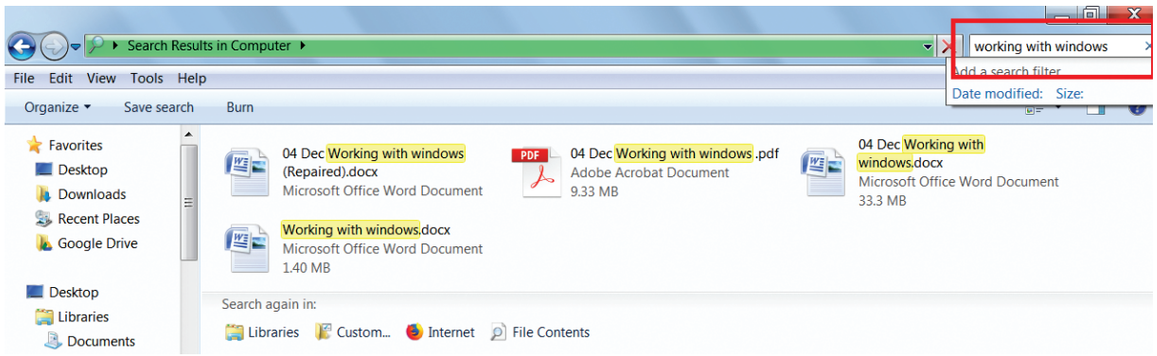
ఫైళ్ళు లేదా ఫోల్డర్లను కంప్యూటర్ ఐకాన్స్‌ని ఉపయోగించి శోధించుట (Searching Files or folders using Computer icon)

1. డెస్క్‌టాప్ నుండి లేదా **Start menu** నుండి **Computer Icon** ని నొక్కండి.
2. కంప్యూటర్ డిస్క్ డ్రైవ్ తెర కనిపించును మరియు ఆ తెర యొక్క కుడి ఎగువ మూలలో **search** బాక్స్ ఎంపిక వుండును. (పటం 5.24)
3. నీకు కావలసిన ఫైల్ లేదా ఫోల్డర్ పేరును టైప్ చేయండి. ఫైల్ లేదా ఫోల్డర్ పేరు యొక్క భాగాన్ని టైప్ చేసినా కూడా నిర్ధిష్ట పేరుతో ప్రారంభమయ్యే అన్ని ఫైళ్ళు లేదా ఫోల్డర్ల జాబితాను ప్రదర్శిస్తుంది.
4. ఆ ఫైల్ లేదా ఫోల్డర్‌ను క్లిక్ చేసి తెరవండి.



పటం 5.23 Start బటన్ ఉపయోగించి ఒక ఫైల్ / ఫోల్డర్‌ని కనుగొనుట. (Finding a File/Folder using Start button)

11th Computer Science_Chapter-5 Telugu.indd 71 30-03-2023 18:28:39



పటం 5.24 కంప్యూటర్ ఐకాన్ తెరలో ఫైల్ / ఫోల్డర్‌ని కనుగొనుము
(Finding a File/Folder in the Computer icon screen)

శిక్షణ శిబిరం

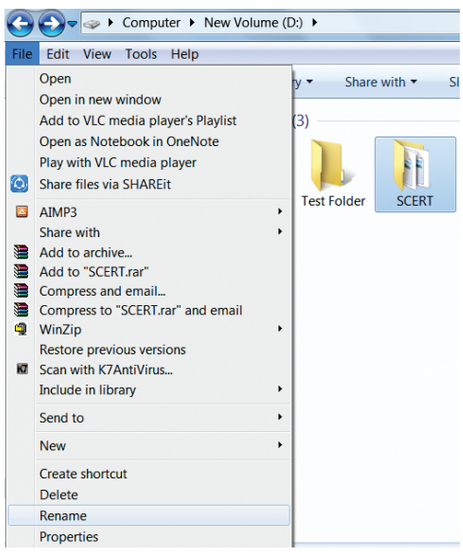
4. పై విధానాన్ని ఉపయోగించి Workshop-3 లో సృష్టించిన ఫైల్‌ను కనుగొనండి.

5.11.3 ఇప్పటికే ఉన్న ఫైళ్ళు లేదా ఫోల్డర్లను తెరవడం (Opening existing Files or Folders)

ఒక ఫైల్ లేదా ఫోల్డర్‌ను తెరవడానికి అత్యంత సాధారణ మార్గం దానిపై డబుల్ క్లిక్ చేయండి.

5.11.4 ఫైళ్ళు లేదా ఫోల్డర్ల పేరు మార్చడం (Renaming Files or Folders)

ఫోల్డర్ లేదా ఫైళ్ళ పేర్లను మార్చడానికి అనేక మార్గాలున్నాయి. ఫైల్ (File) మెను లేదా ఎడమ మౌస్ బటన్ లేదా కుడి మౌస్ బటన్ ఉపయోగించి పేరు మార్చవచ్చును.

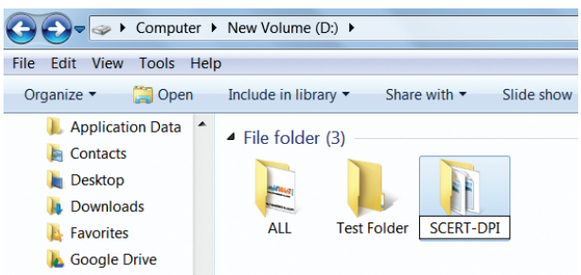


పటం 5.25 ఫైల్ మెనుని ఉపయోగించి ఫోల్డర్ / పేరు మార్చడం

విధానం 1

FILE ని ఉపయోగించి

1. పేరు మార్చవలసిన ఫైల్ లేదా ఫోల్డర్‌ను ఎంచుకొనుము.
2. క్లిక్ File→ Rename.
3. క్రొత్త పేరును టైప్ చేయండి.
4. పటం 5.25 లో చూపిన విధంగా పేరు మార్చు ఆపరేషన్ను ఖరారు చేయడానికి Enter బటన్ నొక్కండి.



పటం 5.26 పేరు మార్చబడిన ఫోల్డర్

పటం 5.26 *SCERT* నుండి *SCERT-DPI* గా పేరు మార్చిన ఫోల్డర్‌ను చూడవచ్చు.(*you can see that the folder is renamed as SCERT-DPI from SCERT.)*

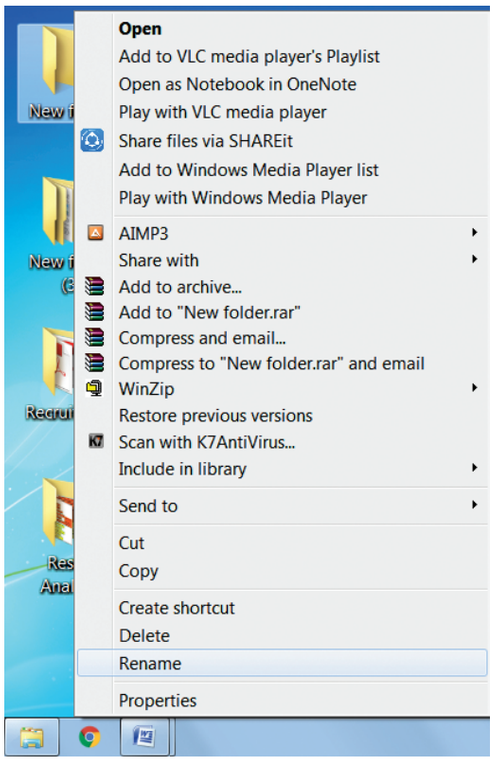
విధానం 2

కుడి మౌస్ బటన్ ఉపయోగించి

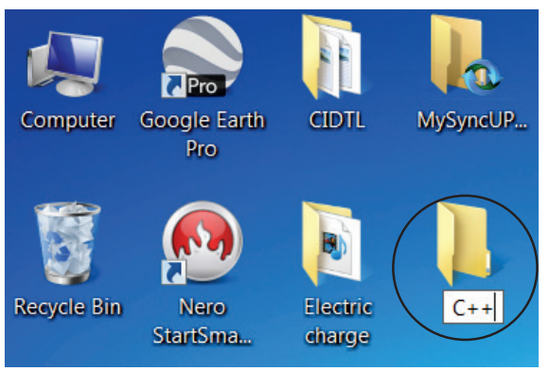
1. పేరు మార్చవలసిన ఫైల్ లేదా ఫోల్డర్‌ను ఎంచుకొనుము.
2. ఫైల్ లేదా ఫోల్డర్ పైన కుడి మౌస్ బటన్ నొక్కండి (పటం 5.27).
3. pop-up మెను నుండి Rename ను ఎంచుకొనుము.

11th Computer Science_Chapter-5 Telugu.indd 72 30-03-2023 18:28:40

4. కొత్త పేరును టైప్ చేయండి.
5. పేరు మార్చు ఆపరేషన్ను ఖరారు చేయడానికి Enter బటన్ నొక్కండి.
6. పటం 5.29 లో "New Folder" అనే ఫోల్డర్ C++ గా పేరు మార్చబడినదని చూడగలము.



పటం 5.27 కుడి మౌస్ బటను ఉపయోగించి ఫైల్ / ఫోల్డర్ పేరు మార్చుట (Renaming File/ Folders using the Right Mouse Button)



పటం 5.28 C++ గా పేరు మార్చబడిన New Folder (New Folder is renamed as C++)

విధానం 3

ఎడమ మౌస్ బటన్ ఉపయోగించి

1. పేరు మార్చవలసిన ఫైల్ లేదా ఫోల్డర్ను ఎంచుకొనుము.
2. F2 లేదా ఫైల్ లేక ఫోల్డర్పై నొక్కండి. పేరు చుట్టూ ఒక దీర్ఘచతురస్రాకారపు అమరిక కనిపించును.
3. క్రొత్త పేరును టైప్ చేయండి.
4. మార్చబడిన పేరును ఖరారు చేయడానికి Enter బటన్ నొక్కండి

శిక్షణ శిబిరం

5. మీరు సృష్టించిన ఫైల్ని File మెను ఎడమ మౌస్ బటన్ లేక కుడి మౌస్ బటన్ ఉపయోగించి పేరు మార్చండి.

5.11.5 ఫైళ్ళు మరియు ఫోల్డర్లను తరలించుట / కాపీ చేయుట (Moving/Copying Files and Folders)

వివిధ పద్ధతులను ఉపయోగించి ఫైళ్ళు లేదా ఫోల్డర్లను మరొక ప్రాంతానికి తరలించవచ్చు.

ఫైళ్ళు మరియు ఫోల్డర్లను తరలించడం (Moving Files and Folders)

విధానం 1 CUT మరియు PASTE

ఒక ఫైల్ లేదా ఫోల్డర్ని తరలించడానికి ఫైల్ లేదా ఫోల్డర్ని ఎంచుకొని తర్వాత ఈ క్రింది పద్ధతులలో ఒకదాన్ని ఎంచుకోండి.

- **Edit → Cut** లేదా **Ctrl + X** క్లిక్ చేయండి లేదా కుడి మౌస్ బటన్ని నొక్కిన pop-up మెను నుండి **click → cut** ఎంచుకొనుము.
- ఒక క్రొత్త స్థానంలోకి ఫైల్ **Edit → Paste** లేదా ఫోల్డర్ ను తరలించడానికి క్రొత్త స్థానానికి వెళ్ళి మౌస్ని ఉపయోగించి **Ctrl + V** అను ఎంపికను క్లిక్ చేసి పేస్ట్ చేయండి.
- లేదా మౌస్ యొక్క కుడి బటన్ నొక్కిన పాప్ అప్ మెను కనిపించును. అందులో **click → Paste** ఎంపికను క్లిక్ చేయండి మీరు ఎంచుకొనిన ఫైల్ క్రొత్త స్థానంలో పేస్ట్ చేయబడుతుంది.

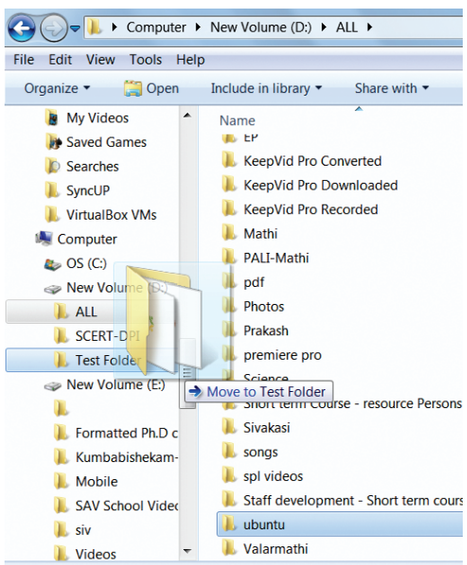
విధానం 2 లాగివదులుట (Method II – Drag and Drop)

డిస్క్ డ్రైవ్ విండోలో మనకు ఎడమ మరియు కుడి ఘన చదురమైన తలములు (panes) అని పిలువబడే రెండు ఘన చదురమైన తలములు (panes) ఉన్నాయి. ఎడమ పేన్లో ఫైళ్ళు లేదా ఫోల్డర్లు ఒక చెట్టు యొక్క శాఖలు వలే

11th Computer Science_Chapter-5 Telugu.indd 73 30-03-2023 18:28:40

ప్రదర్శించబడును. కుడి పేన్‌లో ఎడమ పేన్‌లో గల ఒక నిర్దిష్ట ఫోల్డర్‌లోని ఫైళ్ళు వివిధ ఎంపికలతో ప్రదర్శించబడతాయి.

- డిస్క్ డ్రైవ్ విండో యొక్క కుడి పేన్‌లో తరలించదలచిన ఫైల్ లేదా ఫోల్డర్‌ను ఎంచుకొనుము.
- కుడి పేన్ నుండి ఎంచుకొన్న ఫైల్ లేదా ఫోల్డర్‌ని క్లిక్ చేసి లాగి ఎడమ పేన్‌లో గల ఫోల్డర్ జాబితాలోకి చేర్చండి.
- చేర్చిన తరువాత మౌస్ బటన్‌ని వదలివేయండి.
- ఇప్పుడు మీ ఫైల్ లేదా ఫోల్డర్ క్రొత్త ప్రాంతంలో కనిపిస్తుంది. పటం 5.29 లో లాగి వదులు పద్ధతిని ఉపయోగించి ఫైళ్ళు లేదా ఫోల్డర్లను ఎలా తరలించాలో చూపబడింది.



పటం 5.29 లాగి వదులు పద్ధతిని ఉపయోగించి ఫైల్ / ఫోల్డర్‌ని తరలించుట (Moving the File/Folder using drag and drop)

ఫైళ్ళు మరియు ఫోల్డర్లను కాపీ చేయుట (Copying Files and Folders)

ఫైళ్ళు మరియు ఫోల్డర్లను కాపీ చేయడానికి వివిధ మార్గాలు ఉన్నాయి.

విధానం 1 - **COPY and PASTE**

ఒక ఫైల్ లేదా ఫోల్డర్‌ని కాపీ చేయడానికి ఫైల్ లేదా ఫోల్డర్‌ని ఎంచుకొని ఆపై ఈ క్రింది పద్ధతులలో ఒకదాన్ని ఎంచుకోండి.

- **Edit → Copy** లేదా **Ctrl + C** క్లిక్ చేయండి లేదా కుడి మౌస్ బటన్ నొక్కిన పాప్ అప్ మెను కనిపించును. అందులో pop-up అను ఎంపికను ఎంచుకొనుము.
- క్రొత్త ప్రాంతంలో పేస్ట్ చేయడానికి ఫైల్‌లు లేక ఫోల్డర్లను పేస్ట్ చేయడానికి లక్ష్య స్థానానికి వెళ్ళి తర్వాత ఈ క్రింది వానిలో ఒక దానిని చేయండి.
- **Edit → Paste** లేదా **Ctrl + V** ని క్లిక్ చేయండి.
- లేదా మౌస్ యొక్క కుడి బటన్ నొక్కిన పాప్ అప్ మెను కనిపించును. అందులో **Right click → Paste** ఎంపికను ఎంచుకొనుము.

విధానం 2 లాగి వదులుట **(Drag and Drop)**

- కుడి పేన్‌లో కాపీ చేయవలసిన ఫైల్ లేదా ఫోల్డర్‌ను ఎంచుకొనుము.
- ఎంచుకొన్న ఫైల్ లేదా ఫోల్డర్‌ని ఎడమ వైపు ఉన్న ఫోల్డర్ జాబితాకు క్లిక్ చేసి లాగి కాపీ చేయాలనుకున్న చోట వదలండి.
- ఇప్పుడు క్రొత్త ప్రాంతంలో మీ ఫైల్‌లు మరియు ఫోల్డర్‌లు కనిపించును.

గమనిక

మీరు బహుళ ఫైళ్ళు లేదా ఫోల్డర్లను ఎంచుకోవాలనుకుంటే **Ctrl** + క్లిక్ **(Click)** ని ఉపయోగించవలయును.

5.11.6 తొలగించగల డిస్క్‌లోకి ఫైళ్ళు మరియు ఫోల్డర్లను కాపీ చేయుట (Copying Files and Folders to removable disk)

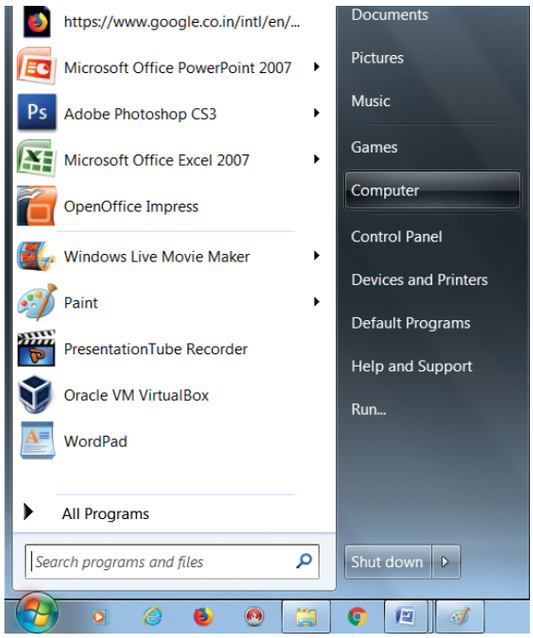
తొలగింపు గల డిస్క్ నుండి లేదా తొలగింపు గల డిస్క్‌లోనికి ఫైళ్ళును బదిలీ చేయుటకు అనేక పద్ధతులు కలవు.

- కాపీ మరియు పేస్ట్ (Copy and Paste)
- Send To

విధానం 1 కాపీ మరియు పేస్ట్ (Copy and Paste)

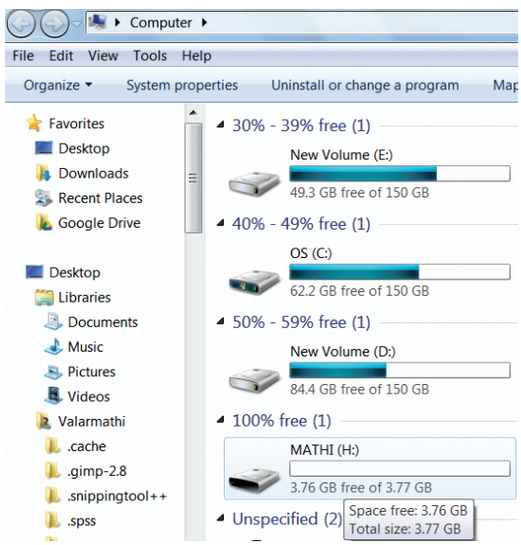
- USB ని అందుబాటులో ఉన్న USB పోర్ట్‌లోకి అమర్చండి.
- USB డ్రైవ్ లేదా బాహ్య డ్రైవ్ external drive ఫోల్టర్లు అయితే ఫోల్డర్ స్వయంచాలకంగా తెరవబడదు. కావున ఈ క్రింది దశలను అనుసరించండి.
- Start→Computer క్లిక్ చేయండి. (పటము 5.31).

11th Computer Science_Chapter-5 Telugu.indd 74 30-03-2023 18:28:40



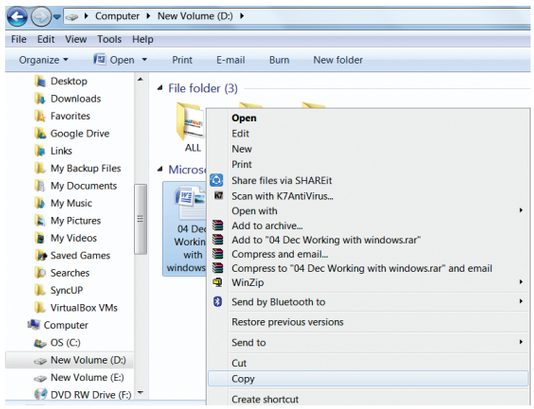
పటం 5.30 ప్రారంభ మెను నుండి కంప్యూటర్ (Computer) ఎంపికను ఎంచుకొనుట (Selecting Computer option from Start menu)

- USB flash డ్రైవ్‌తో అనుబంధించబడిన తొలగింపగల డిస్క్‌పై రెండు సార్లు క్లిక్ చేయుము. (పటం 5.31)



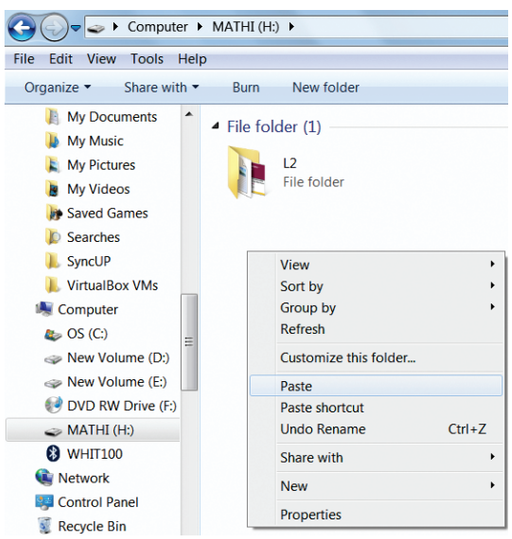
పటం 5.31 తీసివేయ గల డిస్క్‌పై రెండు సార్లు క్లిక్ చేయుట (Double Clicking Removable Disk)

- మీరు బదిలీ చేయదలచిన ఫైళ్ళను కలిగి వున్న మీ కంప్యూటర్‌లోని ఫోల్డర్‌లకు దారిచూపండి.
- కాపీ చేయదలచిన ఫైల్‌పై కుడి క్లిక్ చేయండి. ఆపై **Copy** అను ఎంపికను ఎంచుకోండి. (పటం 5.32)



పటం 5.32 కుడి క్లిక్‌ని ఉపయోగించి ఫైల్‌ని కాపీ చేయుట (Copying File using right click)

- తొలగించు డిస్క్ విండోకు తిరిగి వెళ్ళి, విండో లోపల కుడి క్లిక్ చేసి పేస్ట్ (**Paste**) ఎంచుకోండి (పటం 5.33).



పటం 5.33 కుడి క్లిక్‌ని ఉపయోగించి ఫైల్‌ని కాపీ చేయుట (Pasting File using right click)

విధానం II - Send To

- USB Flash డ్రైవ్‌ని అందుబాటులో ఉన్న USB ఫోర్ట్‌లోకి నేరుగా Plug చేయండి.
- మీరు బదిలీ చేయదలచిన ఫైళ్ళను కలిగి వున్న మీ కంప్యూటర్‌లోని ఫోల్డర్‌లను నావిగేట్ చేయండి.
- తొలగించగల డిస్క్‌లోకి బదిలీ చేయుటకు మీకు కావలసిన ఫైల్‌పై కుడి క్లిక్ చేయుము.
- **Send To** క్లిక్ చేసి మరియు USB flash డ్రైవ్‌తో అనుబంధించబడిన తీసివేయగల డిస్క్‌ని ఎంచుకోండి. (పటం 5.34)

11th Computer Science_Chapter-5 Telugu.indd 75 30-03-2023 18:28:40

శిక్షణ శిబిరం

6. ◆ My Documents లో మీరు సృష్టించిన ఫైల్‌ని డ్రైవ్ (Drive) D కు తరలించుము.

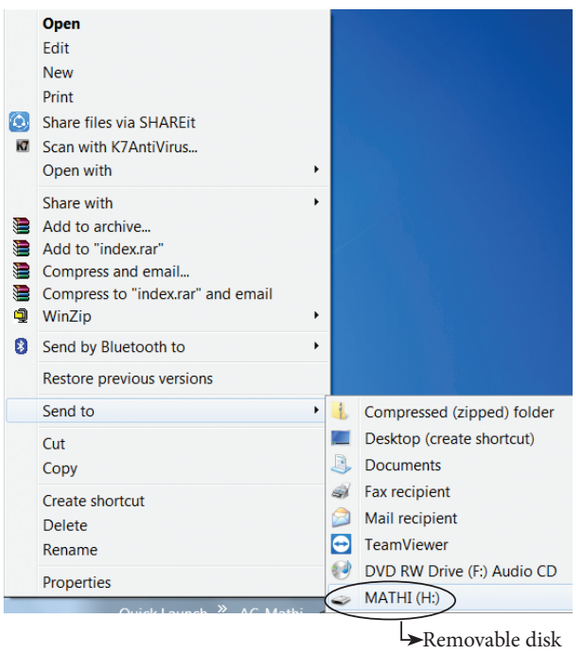
◆ మీరు సృష్టించిన ఫైల్‌ని డ్రైవ్ D: నుండి తొలగించగల డిస్క్‌లోకి కాపీ చేయుము.

5.11.7 ఫైళ్ళు మరియు ఫోల్డర్లను తొలగించుట (Deleting Files and Folders)

- ఒక ఫైల్ లేదా ఫోల్డర్‌ని తొలగించినప్పుడు అది Recycle Bin లోకి తరలించబడుతుంది.

ఒక ఫైల్ లేదా ఫోల్డర్‌ని తొలగించడానికి

నీవు తొలగించాలనుకుంటున్న ఫైల్ లేదా ఫోల్డర్‌ని ఎంచుకొనుము.



పటం 5.34 Send to ఎంపికను ఉపయోగించి ఫైల్‌ని కాపీచేయడం (Copying File using Send to option)

1. ఫైల్ లేదా ఫోల్డర్‌ని కుడి క్లిక్ చేసి పాప్ అప్ మెను నుండి **Delete** ఎంపికను ఎంచుకొనుము (లేదా) **File → Delete** క్లిక్ చేయండి (లేదా) కీబోర్డ్ లో **Delete** కీని నొక్కండి.
2. ఫైల్ తొలగించబడి Recycle bin లోనికి తరలించబడును.

శిక్షణ శిబిరం

7. మీరు సృష్టించిన ఫైల్‌ని My Documents క్రింద నకిలీ చేసిన duplicating తర్వాత ఆ ఫైల్‌ని తొలగించండి.

గమనిక

ఫైల్ లేదా ఫోల్డర్‌ను శాశ్వతంగా తొలగించడానికి (అనగా ఫైల్ లేదా ఫోల్డర్‌ని Recycle Bin లోకి పంపకుండా ఉండటానికి) కీ బోర్డ్‌లోని SHIFT ను నొక్కి **delete** ని నొక్కండి.

రీసైకిల్ బిన్ (Recycle Bin)

రీసైకిల్ బిన్ అనేది వినియోగదారుడు తొలగించిన ఫైళ్ళను లేదా ఫోల్డర్లను ఉంచడానికి ఒక ప్రత్యేక ఫోల్డర్ అంటే వాటిని తిరిగి పొందడానికి మీకు ఇంకా అవకాశం ఉంది. రీసైకిల్ బిన్‌లో అందుబాటులో ఉన్న ఫైళ్ళను లేదా ఫోల్డర్‌లను పునరుద్ధరించకుండా వినియోగదారు యాక్సెస్ చేయలేరు. రీసైకిల్ బిన్ నుండి ఫైల్ లేదా ఫోల్డర్‌ని పునరుద్ధరించడానికి

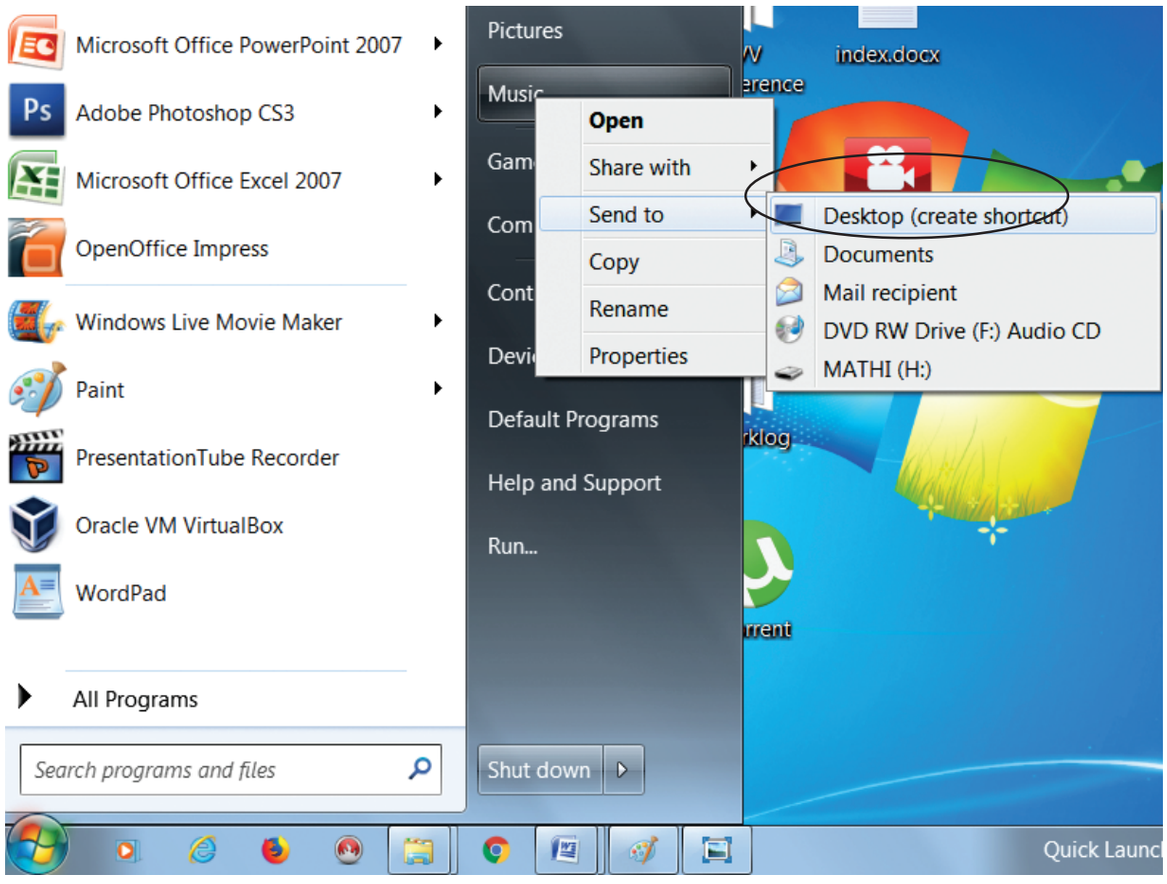
- రీసైకిల్ బిన్ తెరువుము
- పునరుద్ధరించడానికి ఫైల్ లేదా ఫోల్డర్‌పై కుడి క్లిక్ చేసి పాప్ అప్ మెనూ నుండి పునరుద్ధరించు ఎంపికను ఎంచుకోండి.
- బహుళ ఫైళ్ళు లేదా ఫోల్డర్లను పునరుద్ధరించడానికి Restore all items ని ఎంచుకోండి.
- రీసైకిల్ బిన్‌లోని అన్ని ఫైళ్ళను తొలగించడానికి **Empty the Recycle Bin** ని ఎంచుకోండి.

5.12 డెస్క్‌టాప్ పై షార్ట్‌కట్స్‌ను రూపొందించుట (Creating Shortcuts on the Desktop)

మీరు ఎక్కువగా ఉపయోగించే ఫైళ్ళకు Shortcuts సృష్టించబడతాయి మరియు మీ పనిని స్వయంగా automate చేయడంలో సహాయపడటానికి డెస్క్‌టాప్‌లో ఉంచవచ్చు.

- మీరు డెస్క్‌టాప్‌లో shortcutగా ఉండాలనుకుంటున్న ఫైల్ లేదా ఫోల్డర్‌ను ఎంచుకోండి.
- ఫైల్ లేదా ఫోల్డర్‌పై కుడి క్లిక్ చేయండి
- shortcut మెను నుండి **Send to** ఎంచుకొని ఆపై sub-menu నుండి డెస్క్‌టాప్ ఎంచుకొనుము (create shortcut).
- మీ డెస్క్‌టాప్‌పై ఫైల్ లేదా ఫోల్డర్ కొరకు shortcut కనిపించును. డెస్క్‌టాప్ నుండి ఏ ఇతర icon వలె మీ ఫైల్ లేదా ఫోల్డర్‌ను తెరవవచ్చును. (పటం 5.36).

11th Computer Science_Chapter-5 Telugu.indd 76 30-03-2023 18:28:40



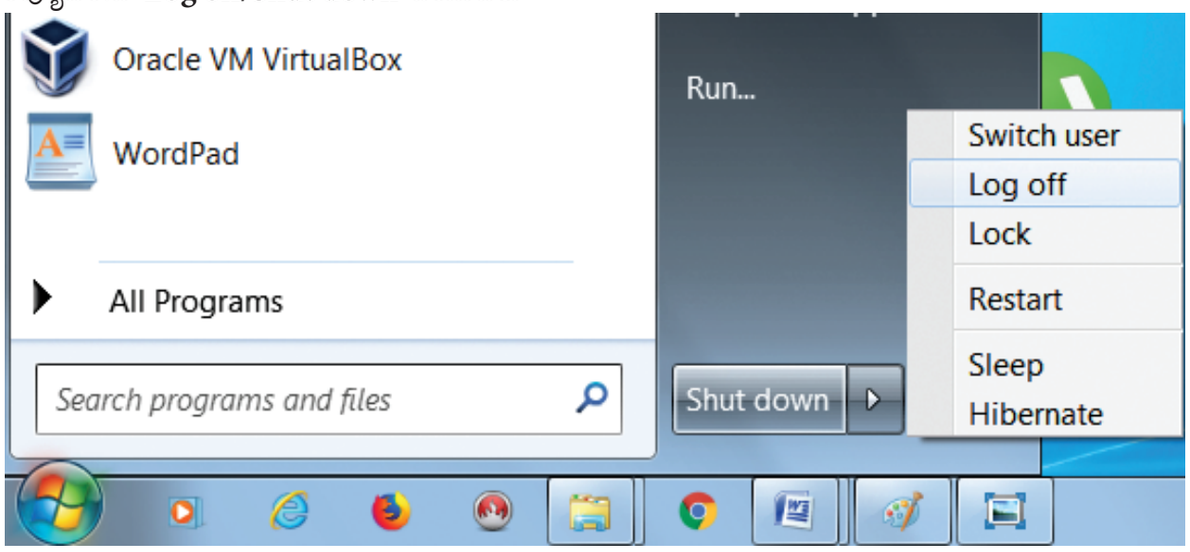
పటం 5.35 సత్వర మార్గాని సృష్టించుట (Creating Shortcut)

5.13 కంప్యూటర్‌ని మూసివేయడం లేదా లాగింగ్ ఆఫ్ చేయడం (Shutting down or Logging off a Computer)

తెరవబడిన అన్ని అనువర్తనాలను మూసివేసిన తర్వాత మీరు మీ కంప్యూటర్‌ను లాగ్ ఆఫ్ చేయవచ్చు లేదా కంప్యూటర్‌ను మూసివేయవచ్చు (shut down)

లాగ్ ఆఫ్ **(Log Off)**

కంప్యూటర్‌ను **Log off/Shut down** చేయుటకు



పటం 5.36 లాగ్ ఆఫ్ ఎంపిక (Log off option)

11th Computer Science_Chapter-5 Telugu.indd 77 30-03-2023 18:28:40

- **start → log off** లేదా **Start → Shutdown** ని క్లిక్ చేయండి. లేదా **Start → Shutdown** (పటం 5.36)
- ఏదైనా open programs లు ఉంటే వాటిని మూసివేయమని అడుగుతుంది అలా మూయకపోతే విండోస్ మూసివేయమని బలవంతం చేస్తుంది. (Force shut down) మీరు ఇలా చేస్తే ఏదైనా భద్రపరచని సమాచారం ఉంటే దానిని కోల్పోతారు.
- **Switch User:** మీ open programs లో మరియు Window processes లను మూసివేయకుండా కంప్యూటర్‌లోని మరొక వినియోగదారు (user) ఖాతాకు మారండి.
- **Log Off:** మీ open program లు మరియు Window processes లను మూసివేసిన తరువాత కంప్యూటర్‌లోని మరొక వినియోగదారు (user) ఖాతాకు మారండి.
- **Lock:** కంప్యూటర్‌కు దూరంగా ఉన్నప్పుడు దాన్ని లాక్ చేయండి.
- **Restart:** కంప్యూటర్‌ని రీబూట్ (Reboot) చేయండి. (ఇది క్రొత్త సాఫ్ట్‌వేర్ ఇన్‌స్టాల్ చేయునప్పుడు లేదా విండోస్‌ను నవీకరణ చేయునపుడు ఈ ఎంపిక తరచుగా అవసరమవుతుంది).
- **Sleep:** కంప్యూటర్‌ను తక్కువ పవర్ మోడ్ (low-power mode) లోకి ఉంచుతుంది. ఇది అన్ని రన్నింగ్ ప్రోగ్రామ్‌లను కలిగి ఉంటుంది. మరియు శీఘ్రు పున:ప్రారంభం (super-quick restart) కోసం కంప్యూటర్ మెమరీలో Windows ని తెరవండి.
- **Hibernate** (ఇది ల్యాప్‌టాప్‌లో మాత్రమే కనుగొనబడుతుంది.) అన్ని రన్నింగ్ ప్రోగ్రామ్‌లను save చేసిన తర్వాత కంప్యూటర్‌ను తక్కువ పవర్ శక్తి మోడ్ (low-power mode) లోకి ఉంచండి మరియు మెషిన్ యొక్క హార్డ్ డ్రైవ్‌లో విండోస్‌ను త్వరగా రీస్టార్ట్ చేయడానికి వాటిని తెరవండి.

కృత్యము

విద్యార్థి కృత్యము (Student Activity)

1. విండోస్ మరియు ఉబున్టు (Ubuntu) లను ఉపయోగించి ఫైళ్ళు మరియు ఫోల్డర్లను రూపొందించండి మరియు వాటిని పోల్చండి.
2. విండోస్ 7, విండోస్ 8 మరియు విండోస్ 10 లో ఒక ఫైల్ లేదా ఫోల్డర్‌ను సృష్టించండి. మీరు ఫైల్ లేదా ఫోల్డర్‌ను సృష్టించేటప్పుడు ఎదుర్కొంటున్న వ్యత్యాసాలపై నివేదికను తయారుచేయండి.

మూల్యాంకనం

విభాగము – అ

సరియైన సమాధానములిమ్ము.

1. క్రింద ఇవ్వబడిన వాటి నుండి ఆపరేటింగ్ సిస్టమ్ ద్వారా నిర్వహించే ఆపరేషన్లు ఎంచుకోండి.

   అ) మెమరీ (Memory)

   ఆ) ప్రక్రియలు (Processes)

   ఇ) డిస్కులు మరియు I/O పరికరాలు (Disks and I/O devices)

   ఈ) పైవన్నీ

2. విండోస్ అనువర్తనాలలో మీ ఫైల్‌ను ఏ డిఫాల్ట్ ఫోల్డర్‌లో save చేయబడును?

   అ) మై కంప్యూటర్ My Document

   ఆ) My Pictures

   ఇ) Documents and Settings

   ఈ) My Computer

3. క్రింది వాటిలో ఏ ఆపరేషన్ సిస్టమ్ OS లో Shift + Delete అనే ఎంపిక ఫైల్ లేదా ఫోల్డర్‌ను శాశ్వతంగా తొలగిస్తుంది?

   అ) Windows 7 ఆ) MS-DOS

   ఇ) Linux ఈ) Android OS

11th Computer Science_Chapter-5 Telugu.indd 78 30-03-2023 18:28:41

4. Windows XP/Windows 7 లో హైబర్నేట్ యొక్క అర్థమేమిటి?
   అ) safe mode లో కంప్యూటర్‌ని పున:ప్రారంభించడం
   ఆ) హైబర్నేట్ మోడ్ (hibernate mode) లో కంప్యూటర్‌ని పున:ప్రారంభించుట.
   ఇ) running applications ను మూసివేయుటకు కంప్యూటర్‌ను షట్డౌన్ (Shutdown) చేయుట.
   ఈ) running applications మూసివేయకుండా కంప్యూటర్‌ను షట్‌డౌన్ చేయుట.
5. విండోస్‌లో ఒక ఫైల్ యొక్క పేరును మార్చుటకు ఉపయోగించు సత్వర మార్గం (shortcut)
   అ) F2 ఆ) F4
   ఇ) F5 ఈ) F6

విభాగం-ఆ

సూక్ష్మముగా సమాధానములిమ్ము

1. మల్టీటాస్కింగ్ (Multitasking) అనగానేమి?
2. ప్రామాణిక చిహ్నాలు standard icons అనగానేమి?
3. ఫైళ్ళు మరియు ఫోల్డర్ల యొక్క వ్యత్యాసములను తెలపండి.
4. Save మరియు Save as ఎంపికల యొక్క వ్యత్యాసములను తెలపండి.
5. ఒక ఫైల్ యొక్క పేరును ఎలా మార్చవచ్చు?

విభాగము-ఇ

క్లుప్తముగా సమాధానములిమ్ము

1. విండోస్ ఆపరేటింగ్ సిస్టమ్ యొక్క విధులేవి?
2. రీసైకిల్ బిన్ గురించి వ్రాయండి.
3. విండో యొక్క అంశములను గురించి వ్రాయండి.
4. ఒక కొత్త ఫోల్డర్‌ను సృష్టించుటకు రెండు మార్గములను వ్రాయుము.
5. కాపీ మరియు తరలించుటను వ్యత్యాసపరచండి.

విభాగం - ఈ

విపులముగా సమాధానములిమ్ము

1. విండోస్ ఆపరేటింగ్ సిస్టమ్ యొక్క సంస్కరణలను వివరించండి.
2. ఒక ఫైల్ లేదా ఫోల్డర్‌ను కనుగొనడానికి వివిధ మార్గములను వివరించుము.
3. విండోస్ OS లో సత్వర మార్గమును సృష్టించు విధానమును వ్రాయుము.

RSVOM

పదకోశం

ఆపరేటింగ్ సిస్టమ్ (OS)	ఇతర సాఫ్ట్‌వేర్‌లతో కమ్యూనికేట్ చేయడానికి మరియు operate చేయడానికి హార్డ్‌వేర్‌ను అనుమతించే సిస్టమ్ సాఫ్ట్‌వేర్.
మౌస్ (**Mouse**)	GUI లో కర్సర్ (cursor) ను నియంత్రించే హ్యాండ్‌హెల్డ్ హార్డ్‌వేర్ ఇన్‌పుట్ పరికరము మరియు టెక్స్ట్ చిహ్నాలు ఫైళ్ళు మరియు ఫోల్డర్లను ఎంచుకొని తరలించవచ్చు.
విండోస్ (**Windows**)	మైక్రోసాఫ్ట్ కార్పొరేషన్ అభివృద్ధి చేసిన సుపరిచితమైన ఆపరేటింగ్ సిస్టమ్.
డెస్క్‌టాప్ (**Desktop**)	విండోస్ ఆపరేటింగ్ సిస్టమ్ యొక్క ప్రారంభ తెర.
Icon	ఒక ఆదేశాన్ని (command) సూచించే చిన్న చిత్రం.
ఫోల్డర్ (**Folder**)	ఫైళ్ళు కంటైనర్.
లైనక్స్ (**Linux**)	ఒక ఆపరేటింగ్ సిస్టమ్.

11th Computer Science_Chapter-5 Telugu.indd 79 30-03-2023 18:28:41

అధ్యాయము 6

స్పెసిఫికేషన్ మరియు సంగ్రహణ (Specification and Abstraction)

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును అభ్యసించిన తరువాత విద్యార్థులు తెలుసుకొను అంశములు.

- అల్గోరిథమిక్ సమస్యా పరిష్కారాలు అను భావనను అర్థం చేసుకోవటం.
- అల్గోరిథమిక్ సమస్యా పరిష్కారంలో టెక్నిక్స్‌ను (సాంకేతిక పద్ధతులను) వర్తింపజేయటం.

మనదైనందిన జీవితంలో చేసే అనేక ప్రక్రియ కార్యక్రమాలు దశల వారీ సూచనలను అమలు చేస్తాము. ఉదయం పాఠశాలకు సిద్ధం కావటం, ముగ్గులు వేయటం వంట చేయటం, రెండు సంఖ్యలను కూడటం మొదలైనవి ప్రక్రియలకు ఉదాహరణలు. అల్గోరిథంలను అమలు చేయడం వలన ప్రక్రియలు (Processes) ఉత్పత్తి చేయబడతాయి. ఈ అధ్యాయంలో అల్గోరిథంలు ఎలా పేర్కొనబడతాయి మరియు ప్రక్రియ యొక్క అంశాలు ఏ విధంగా పరిష్కరింపబడతాయి అనే విషయాలను గూర్చి చూస్తాము.

6.1 అల్గోరిథంలు (Algorithms)

అల్గోరిథం అనగా ఒక పనిని పూర్తి చేయడానికి లేదా సమస్యను పరిష్కరించడానికి మనము వరుస క్రమంలో తీసుకునే సూచనలు అని అర్థం. ఒక సూచన అనేది ఒక చర్యను సూచిస్తుంది. సూచనలను అమలు చేయునపుడు ఒక ప్రక్రియ ఉత్పన్నమవుతుంది. ఇది ఉద్దేశించిన పనిని నెరవేర్చుతుంది లేదా ఇచ్చిన సమస్యను పరిష్కరిస్తుంది. మనం ఒక అల్గోరిథంను ఒక వంట చేయటంతో పోల్చుకోవచ్చు. దాని ఫలితాన్ని వంటకంతో పోల్చ వచ్చు.

అల్గోరిథంను కంప్యూటర్‌లో అమలు చేయడమే మన ఉద్దేశం. కంప్యూటరు సూచనలను ప్రోగ్రామింగ్ భాషలో మాత్రమే అమలు చేయగలదు. అందువలన చివరికి అల్గోరిథంలను ప్రోగ్రామింగ్ భాష యొక్క స్టేట్‌మెంట్లుగా వ్యక్తపరచాలి.

ఇవ్వబడిన ఇన్పుట్ డేటా ఆశించిన అవుట్పుట్ డేటా మరియు ఇన్పుట్ అవుట్పుట్ డేటా మధ్య గల సంబంధం ద్వారా సమస్య నిర్దేశించబడుతుంది. ఒక అల్గోరిథం అనేది ఇన్పుట్ డేటాతో పూర్తి చేస్తుంది. ఎప్పుడైతే ఇన్పుట్ డేటా మరియు అవుట్పుట్ డేటా మధ్య సంబంధం సంతృప్తి చెందుతుందో అప్పుడే అల్గోరిథం ఇచ్చిన సమస్యను పరిష్కరిస్తుంది.

మీకు తెలుసా?

జి.పాలియా అనునతను హంగేరియా దేశపు గణిత శాస్త్రవేత్త. అతని యొక్క ప్రాథమిక రచనలు సంఖ్యా సిద్ధాంతము, సంఖ్యా విశ్లేషణ మరియు సంభావ్యత సిద్ధాంతం. ఇవి కాకుండా ఇతను సూక్ష్మ పరిశోధనకు సమస్యా పరిష్కారాన్ని కనుక్కోవడంలో విద్యార్థులు మరియు ఉపాధ్యాయుల కోసం గణిత శాస్త్రంలో చాలా ఆవిష్కరణలు చేశారు. "How to Solve It" లో అతను గణిత సమస్యలను పరిష్కరించేటప్పుడు ఈ క్రింది దశలను సూచనలను సూచించారు: 1. సమస్యను అర్థం చేసుకోవటం, 2, ఒక ప్రణాళికను రూపొందించుట, 3. ప్రణాళికలను అమలు చేయటం 4. మన పనిని సమీక్షించటం.



అల్గోరిథం అనేది సమస్యను పరిష్కరించడానికి ఉద్దేశించిన స్టేట్‌మెంట్ల యొక్క దశల వారీ క్రమం ఇన్పుట్ డేటాతో అమలు చేసేటప్పుడు ఇది గణన ప్రక్రియను ఉత్పత్తి చేస్తుంది (ప్రారంభిస్తుంది) మరియు అవుట్పుట్ డేటాతో ముగుస్తుంది. ఇన్పుట్ డేటా మరియు అవుట్పుట్ డేటా మధ్య పేర్కొన్న సంబంధాన్ని తృప్తి పరుస్తుంది.

11th Computer Science_Chapter-6 Telugu.indd 80 30-03-2023 18:34:16

ఉదాహరణ 6.1 రెండు సంఖ్యలను కూడటం: రెండు సంఖ్యలను కూడటానికి ముందుగా మనం కుడిచేతి వైపుగల చివరి అంకెలను కూడవలెను. తరువాత అదే విధంగా కుడిచేతివైపు చివరి అంకెకు ముందుగా గల అంకెలను కూడి ముందు కూడిన ఫలితముతో కలిపి కూడవలెను. ఇదే విధంగా మిగిలిన అంకెలను కూడవలెను. కంప్యూటర్‌లో 2586 మరియు 9237లను కూడుట గురించి పట్టిక 6.1లో వివరించియున్నది.

మెట్టు	5	4	3	2	1
ఫలిత సంఖ్య	1	0	1	1	-
సంఖ్య	1	2	5	8	6
సంఖ్య	2	9	2	3	7
మొత్తము	1	1	8	2	3

పట్టిక 6.1 రెండు సంఖ్యలను కూడే పద్ధతి

6.2 అల్గోరిథమిక్ సమస్యలు (Algorithmic Problems)

అల్గోరిథంలను నిర్మించటానికి (తయారుచేయటానికి) కొన్ని సూత్రాలు మరియు సాంకేతికతలు ఉన్నాయి. మనము సాధారణంగా ఒక సమస్యకు అల్గోరథమిక్ స్వభావం ఉందని చెబుతాము. దాని పరిష్కారం అల్గోరిథం నిర్మాణాన్నే కలిగి ఉంటుంది. కొన్ని సమస్యలు తక్షణమే ఆల్గరితమిక్ గా గుర్తించబడతాయి.

ఉదాహరణ 6.2 మనకు పరిచయమైన ప్రసిద్ధమైన మేక, గడ్డి, తోడేలు కథను సమస్యగా తీసుకుందాం: ఒక రైతు ఒక మేకను, ఒక గడ్డిమోపును మరియు ఒక తోడేలును పడవలో నది అవతల వైపుకు తీసుకువెళ్ళాలనుకుంటాడు. అయితే అతని వద్ద ఉన్న పడవలో ఒకసారికి ఒకటిని మాత్రమే ఎక్కించగలడు. అతను వాటిని అవతల ఒడ్డుకు చేర్చటానికి చాలా సార్లు ప్రయాణం చేయవలసి ఉంటుంది. అంతేకాకుండా అతను మేక మరియు గడ్డి మోపును లేక మేక మరియు తోడేలును ఒకటిగా వదిలి వెళ్ళటానికి వీలుకాదు. (కారణం తోడేలు మేకను తినేస్తుంది, లేకపోతే మేక గడ్డిని తినేస్తుంది) కాని రైతు తనతో ఉన్న మూడింటిని అవతలి ఒడ్డుకు చేర్చాలి. ఈ సమస్యను రైతు ఏ విధంగా పరిష్కరించగలడు? ప్రారంభ దశలో నలుగురు నదికి ఇవతల వైపు ఉన్నట్లుగా అనుకుందాం. చివరి దశలో నలుగురు తప్పని సరిగ్గా నదికి అవతల వైపు ఉండాలి. రైతు నదిని దాటేటప్పుడు పడవలో ఉండాలి. అతనితో పాటు ఎవరు లేక దేనిని తీసుకెళ్ళాలో నిర్దేశించుకోవటం ఒక పరిష్కారం. కాబట్టి ఇది ఒక అల్గోరిథమిక్ సమస్యకు ఉదాహరణ. ఇటువంటి అల్గోరిథమిక్ సమస్యకు పరిష్కారం ఈ క్రింది విధంగా ఉండవచ్చు.

రైతు తోడేలుతో నదిని దాటవలెను.
లేక
రైతు మాత్రం ఒంటరిగా నదిని దాటవలెను.

అయినప్పటికీ కొన్ని అల్గోరిథమిక్ సమస్యలను తీర్చటానికి అల్గోరిథమిక్‌ను రూపొందించాల్సిన అవసరం లేదు. దానికి బదులుగా మరొక అల్గోరిథంను ఇవ్వబడిన మనము వాటి ధర్మాలను నిరూపించవలెను.

ఉదాహరణ 6.3 క్రోంల్యాండ్ ఊసరవెల్లి సమస్యను పరిగణలోనికి తీసుకొందాం: క్రోంలాండ్ దీవిలో మూడు వివిధ రకాల (రంగుల) ఊసరవెల్లిలు కలవు. ఎర్రటి ఊసరవెళ్ళి, పచ్చటి ఊసరవెల్లి మరియు నీలం రంగు ఊసరవెల్లి, రెండు వేర్వేరు రంగుల ఊసరవెల్లిలు ఒక దగ్గర కలిసినపుడు అవి రెండు కూడా మూడోరంగులోకి మారిపోతాయి. ఎరుపు, పచ్చ, నీలం రంగు ఊసరవెల్లిలు ఎన్నిసార్లు ఒకటిగా కలిసినా అవి అన్నీ కూడా ఒకే రంగులోకి మారిపోతాయి. ఇది ఒక అల్గోరిథమిక్ సమస్య ఎందుకనగా ఊసరవెల్లిలు ఒకటిగా కలవటంలో ఒక అల్గోరిథం ఉంది. అల్గోరిథంలో గల కొన్ని లక్షణాలను ఉపయోగించి ఎన్ని ఊసరవెల్లిలు లక్ష్యాన్ని సాధించాయో కనుగొనవచ్చు.

6.3 అల్గోరిథమిక్ యొక్క బిల్డింగ్ బ్లాక్స్ (Building Blocks of Algorithms)

మనము ప్రాథమిక బిల్డింగ్ బ్లాక్స్‌ను ఉపయోగించి అల్గోరిథంలను నిర్మించగలము. అవి

- దత్తాంశాలు (Data)
- వేరియబుల్స్ (Variables)
- నియంత్రణ ప్రవాహం (Control flow)
- విధులు (Functions)

6.3.1 దత్తాంశాలు (డేటా) (Data)

అల్గోరిథంలు ఇన్పుట్ డేటాను తీసుకుంటాయి. డేటాను ప్రాసెస్ చేస్తాయి, మరియు

11th Computer Science_Chapter-6 Telugu.indd 81 30-03-2023 18:34:16

అవుట్‌పుట్ డేటాను ఉత్పత్తి చేస్తాయి. డేటాపైన కార్యకలాపాలను నిర్వహించడానికి కంప్యూటర్లు సూచనలను అందిస్తాయి. ఉదాహరణకు గణితశాస్త్రంలో ఉన్న కూడిక, తీసివేత, గుణకారం, భాగాహారం మరియు అంకగణిత కార్యకలాపాలను చేయడానికి ఉపయోగపడుతుంది.

6.3.2 వేరియబుల్స్ (Variables)

డేటాను నిల్వచేయటానికి వేరియబుల్స్‌కు boxes అని పేరు పెట్టారు. మనం డేటాపై ఆపరేషన్ చేసినపుడు ఫలితాలను వేరియబుల్స్‌లో నిల్వచేయాలి. ఒక వేరియబుల్‌లో నిల్వచేసిన డేటాను ఆ వేరియబుల్ విలువ అని కూడా అంటారు. అసైన్‌మెంట్ స్టేట్‌మెంట్స్‌ను ఉపయోగించి వేరియబుల్స్‌లో విలువలను నిల్వచేయవచ్చు లేదా విలువలను మార్చనూ వచ్చు.

వాస్తవ ప్రపంచంలో గణన ప్రక్రియ (విధానమునకు) ఒక పద్ధతి ఉంది. ఈ ప్రక్రియ అభివృద్ధి చెందినపుడు స్థితిలో మార్పు జరుగును. అల్గోరిథంలో ప్రక్రియ ద్వారా వేరియబుల్స్ యొక్క స్థితిలో మార్పును మనం ఏ విధంగా గుర్తించగలము? అల్గోరిథంలో ఒక ప్రక్రియ యొక్క స్థితి అనేది నిర్దేశించిన సమయంలో ఉన్న వేరియబుల్స్ యొక్క విలువ అని అర్థం. వేరియబుల్స్ యొక్క విలువ మారినపుడు దాని యొక్క స్థితి కూడా మారుతుంది.

ఉదాహరణ 6.4: రోడ్డు సిగ్నల్స్ (నియంత్రణ గుర్తులు) ఎరుపు, పసుపు, పచ్చ రంగులలో ఏదేని ఒకటి కావచ్చు రవాణా సక్రమంగా జరగడానికి సిగ్నల్స్ స్థితి మార్చబడుతుంది. ఆ యొక్క సిగ్నల్ స్థితి అనేది ఏదైనా ఒక వేరియబుల్ సిగ్నల్‌తో చూపించబడుతుంది. అవి ఈ మూడింటిలో ఎరుపు, పసుపు, లేదా పచ్చ అనే ఏదైనా ఒకటిగా ఉండాలి.

6.3.3 నియంత్రణ ప్రవాహం (Control flow)

అల్గోరిథం అనునది ఒక ప్రకటనల క్రమము. అయినప్పటికీ ఒక ప్రతిపాదన నిర్వహించిన తరువాతనే తరువాతి ప్రతిపాదనను నిర్వహించవలెను అన్న నిబంధన అల్గోరిథంలో లేదు. అది తదుపరి అమలు పరచవలసిన ప్రతిపాదన ప్రక్రియ యొక్క స్థితిపై ఆధారపడి ఉంటుంది. కాబట్టి అమలు చేయవలసిన ప్రతిపాదన క్రమం భిన్నంగా కూడా ఉండవచ్చు. ఈ క్రమంలో నిర్వహించే ప్రతిపాదనల క్రమాన్ని నియంత్రణ ప్రవాహం (control flow) అంటారు.

స్థితిని అనుసరించి statements ప్రవాహాలను మార్చటానికి మూడు ముఖ్యమైన statements ప్రవాహ ప్రకటన (నివేదికలు) ఒకదాని తరువాత ఒకటి అమలు చేయబడతాయి.

- ప్రత్యామ్నాయ statement ప్రవాహంలో statement అమలు చేయబడిన తరువాత అది వాస్తవమైతే ఆ statement అమలుపరచబడుతుంది.
- Statement అవాస్తమైతే ప్రత్యామ్నాయ నివేదిక అమలు చేయబడుతుంది. పునరుత్పత్తి statement ప్రవాహంలో statement పరీక్షించబడి statement వాస్తవమయితే అమలు చేయబడుతుంది.
- ఒక వేళ అవాస్తవం అయితే statement షరతులు తప్పు అయ్యేవరకు పరీక్షించటం, అమలు చేయటం అనే రెండు దశలు పునరావృతం అవుతాయి.

6.3.4 విధులు (Functions)

అల్గోరితంలు చాలా క్లిష్టమైనవిగా ఉంటాయి. ఒక అల్గోరిథం యొక్క వేరియబుల్స్ మరియు దానికి సంబంధించిన డిపెండెన్సీలు (ఆధారపడినవి) చాలా ఎక్కువ కావచ్చు. అప్పుడు అల్గోరిథంను నిర్మించటం చాలా కష్టం. అటువంటి పరిస్థితులలో మనము అల్గోరిథములను భాగాలుగా విడదీసి ప్రతి భాగాన్ని విడిగా నిర్మించి ఆపై అన్ని భాగాలను పూర్తి అల్గోరిథంగా అనుసందానిస్తాము.

అల్గోరిథంలోని భాగాలను దాని యొక్క functions అని కూడా అంటారు. ఒక function ఒక ఉప అల్గోరిథం లాంటిది. ఇది ఇన్పుట్‌ను తీసుకుని అవుట్పుట్‌ను ఇచ్చి ఇన్పుట్ అవుట్పుట్ మధ్య ఉన్న సంబంధాన్ని తృప్తి పరుస్తుంది.

ఉదాహరణ 6.5: r వ్యాసార్ధము h ఎత్తుగా గల ఒక స్తూపము యొక్క ఉపరితల వైశాల్యాన్ని లెక్కించాలనుకుందాం.

$$A = 2\pi r^2 + 2\pi rh$$

దీన్ని మనము రెండు విధాలుగా గుర్తించవచ్చు. ఒకటి వృత్తం యొక్క వైశాల్యాన్ని కనుగొనటం, రెండవది వృత్తం యొక్క చుట్టుకొలతను

11th Computer Science_Chapter-6 Telugu.indd 82 30-03-2023 18:34:16

కనుగొనటం, మనము ఈ రెండింటిని కనుగొన్న తర్వాత స్తూపము యొక్క ఉపరితల వైశాల్యాన్ని ఈ విధంగా లెక్కించవచ్చును.

స్థూపము యొక్క వైశాల్యము $(r,h) = 2X$ వృత్త వైశాల్యం (r) + వృత్తము యొక్క చుట్టుకొలత $(r) X h$

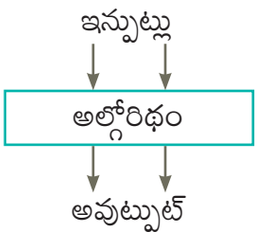
6.4 అల్గోరిథంను రూపొందించడంలో, టెక్నిక్స్ (Algorithm Design Techniques)

అల్గోరిథంను నిర్మించటానికి కొని ప్రాథమిక సిద్ధాంతములు మరియు సూక్ష్మ నైపుణ్యములు కలవు.

1. **వివరణము (Specification):** సమస్యను పరిష్కరించడంలో మొదటి సోపానము సమస్యను ఖచ్చితంగా గుర్తించటం, సమస్యను ముందుగా ఇవ్వబడిన ఇన్పుట్‌ను మరియు ఆశించిన అవుట్పుట్‌గా పేర్కొనాలి. స్పెసిఫికేషన్స్ అనేవి ఇవ్వబడినటువంటి ఇన్పుట్ యొక్క లక్షణాలు మరియు ఇన్పుట్ అవుట్పుట్ మధ్యగల సంబంధాన్ని కూడా తెలియజేయాలి.
2. **సంగ్రహణ (Abstraction):** ఒక సమస్య అనేది అనేక అంశాలను కలిగి ఉంటుంది. కాని ఆ సమస్యను పరిష్కరించడానికి ఈ అంశాలన్నీ అవసరం లేదు. కొన్ని వివరాలు మాత్రమే చాలు. అనవసరమైన వివరాలను వదిలిపెట్టడం, ముఖ్యమైన వివరాలను మాత్రమే ఉపయోగించుకోవటాన్ని సంగ్రహణ అంటారు. ఉదాహరణకు మనము ఒక ప్రక్రియ యొక్క స్థితిని సూచించినపుడు మనము ఆ సమస్యకు అవసరమైన వేరియబుల్స్‌ను (వివరాలను) మాత్రమే ఎంచుకుంటాం. అనవసరమైన వివరాలను విస్మరిస్తాం (వదిలిపెడతాం).
3. **కూర్పు లేక కూర్చటం (Composition):** అల్గోరిథం అనునది అసైన్‌మెంట్ మరియు నియంత్రణ ప్రవాహాల సమూహము. ఈ నియంత్రణ ప్రవాహ స్థితిని పరీక్షించి విలువను బట్టి తదుపరి స్టేట్మెంట్‌ను అమలు చేయాలో వద్దో అని నిర్ణయిస్తుంది.
4. **విభజించటం (Decomposition):** మనము ప్రధాన అల్గోరిథంను కొన్ని భాగాలుగా విభజిస్తాము. ప్రధాన అల్గోరిథం యొక్క ప్రతి విధిని మనమే స్వతంత్ర్యంగా నిర్మిస్తాము. చివరిగా ఫంక్షన్లులను ఉపయోగించి ప్రధాన అల్గోరిథంను నిర్మిస్తాము. మనము ఈ యొక్క విధులను ఉపయోగించినపుడు ఈ functions యొక్క ప్రాముఖ్యతను తెలుసుకుంటే చాలు functions ఎలా అమలు చేయబడునో తెలుసుకోవలసిన అవసరం లేదు.

6.5 స్పెసిఫికేషన్స్ (Specification)

ఒక సమస్యను పరిష్కరించాలంటే ముందుగా మనం ఆ సమస్యను గూర్చి స్పష్టంగా పూర్తిగా తెలుసుకోవాలి. ఇవ్వబడిన ఇన్పుట్ మరియు పొందవలసిన అవుట్పుట్ ద్వారా సమస్య పేర్కొనబడుతుంది. సమస్యను పరిష్కరించడానికి అల్గోరిథంని రూపొందించాలి ఇవ్వబడిన ఇన్పుట్ మరియు నిర్దేశించిన అవుట్పుట్ యొక్క లక్షణాలను తెలుసుకొని ఉండాలి. అల్గోరిథం యొక్క లక్ష్యం ఏమనగా ఇన్పుట్ మరియు నిర్దేశించిన అవుట్పుట్ మధ్య కావలసిన సంబంధాన్ని ఏర్పరచటం.



పటము 6.1 ఇన్పుట్ – అవుట్పుట్ మధ్య సంబంధం

అల్గోరిథం అనునది ఇవ్వబడిన ఇన్పుట్ యొక్క లక్షణాలు, ఇన్పుట్ మరియు అవుట్పుట్ మధ్య సంభంధం ద్వారా పేర్కొనబడుతుంది. అల్గోరిథం యొక్క స్పెసిఫికేషన్స్‌ను సాధారణ భాషలో (సరళ) నిర్దేశించిన ఇన్పుట్ మరియు అవుట్పుట్ మధ్య గల సంబంధం అని పేర్కొనవచ్చు.

ఇన్‌పుట్‌లు మరియు అవుట్‌పుట్‌లు అల్గోరిథమ్ మరియు వినియోగదారుల మధ్య పంపబడుతాయి. అల్గోరిథమ్ ప్రారంభించినపుడు Q యొక్క విలువలను ప్రారంభ స్థితి అని పిలుస్తారు. మరియు అల్గోరిథమ్ ముగిసినపుడు గల విలువలను తుది స్థితి అంటారు

ఇన్పుట్‌కు అవసరమైన ధర్మాలను P అని నిర్దేశించిన అవుట్పుట్ ధర్మాలను Q అని అనుకున్నచో అప్పుడు S అనే అల్గోరిథంను ఎలా చూపవచ్చు.

11th Computer Science_Chapter-6 Telugu.indd 83 30-03-2023 18:34:16

1. అల్గోరిథం పేరు (ఇన్పుట్లు)
2. -- ఇన్పుట్లు : P
3. -- అవుట్పుట్లు: Q

ఈ స్పెసిఫికేషన్ ప్రారంభ దశలో అల్గోరిథం P ని సంతృప్తిపరిచే ఇన్పుట్‌గా తీసుకుని సంతృప్తిపరిచే Q ని అవుట్పుట్‌గా ఇస్తుంది. మిగిలిన వ్యాక్యాన్ని '--' తో సూచిస్తుంది మిగిలిన వ్యాక్యము యొక్క గుర్తుగా సూచించబడుతుంది. అల్గోరిథంలో వ్యాఖ్యాలు అనగా మానవ పాఠకుల కోసం కంప్యూటర్‌లో ఒక ప్రోగ్రామింగ్ చేయటానికి అమలు చేయబడేవి. అల్గోరిథంను అర్థం చేసుకోవటానికి వ్యాఖ్యాలు ముఖ్యమైనవిగాను ఉపయోగకరమైనవిగాను ఉంటాయి. అల్గోరిథమిక్ సంకేతాలను గుర్తించటానికి డబుల్ డ్యాష్ (--) ను ఉపయోగిస్తాము. C++ లో డబుల్ స్లాష్ మిగిలిన పంక్తి యొక్క వ్యాఖ్యను సూచిస్తుంది.

ఉదాహరణ 6.6: A అను పూర్ణాంకమును మరొక B అను పూర్ణాంకముతో భాగించిన తరువాత గుణకం యొక్క శేషము (మిగిలిన) వాటిని లెక్కించటానికి అల్గోరిథం యొక్క వివరణ వ్రాయుము. ఉదాహరణకు,

విభజన (22, 5) = 4, 2
విభజన (15, 3) = 5 , 0

A మరియు B ని ఇన్పుట్ వేరియబుల్‌గా అనుకుందాం. గుణకాన్ని q అనే వేరియబుల్‌లో మరియు మిగిలిన దానిని r అనే వేరియబుల్‌లో నిల్వచేస్తాము.

కాబట్టి q మరియు r అనేవి అవుట్పుట్ వేరియబుల్స్. A మరియు B ల ఇన్పుట్ల లక్షణాలు ఏమిటి?

1. A ఒక పూర్ణాంకమై ఉండాలి. పూర్ణాంకము విభజనకు మాత్రమే శేషము అర్థవంతంగా ఉండాలి.
2. B అనునది 0 కాకూడదు ఎందుకనగా 0 తో భాగించటం వీలుకాదు.

మనము ఇన్పుట్ యొక్క లక్షణాలను గూర్చి పేర్కొందాం.

— ఇన్పుట్స్: A ఒక పూర్ణాంకము మరియు B ≠ 0

A మరియు B ఇన్పుట్లు, q మరియు r అవుట్పుట్ల మధ్య గల సంబంధం ఏమిటి?

1. Q మరియు R అను రెండు అవుట్పుట్లు ఈ ధర్మాన్ని తృప్తిపరచాలి.

   A = Q X B + r, మరియు

2. శేషము r అనునది భాజకము B కన్న తక్కువగా ఉండాలి,

   0 ≤ r < B

ఈ అవసరాలను కలపడం ద్వారా మనం కోరుకున్న ఇన్పుట్ అవుట్పుట్ సంబంధాన్ని పేర్కొంటాము.

— అవుట్పుట్లు: A = q X B + r మరియు **0 < r < B.**

'—' ఇన్పుట్లతో ప్రారంభమయ్యే వ్యాక్య వాస్తవానికి ఇచ్చిన ఇన్పుట్లుల property '—' అవుట్పుట్‌తో ప్రారంభమయ్యే వ్యాక్య ఇన్పుట్లు మరియు అవుట్పుట్ మధ్య గల సంబంధము. అల్గోరిథమ్ యొక్క వివరణ అనునది

1. divide (A , B)

2. -- inputs: A is an integer and B ≠ 0

3. -- outputs : A = q X B + r and 0 ≤ r < B

స్పెసిఫికేషన్ ఫార్మేట్ : మనము స్పెసిఫికేషన్స్‌ను మూడు ప్రామాణిక ఫార్మాట్‌లలో (రూపాలలో) వ్రాయవచ్చు.

- అల్గోరిథం యొక్క పేరు మరియు ఇన్పుట్లు
- ఇన్పుట్: ఇన్పుట్స్ యొక్క ధర్మములు (లక్షణములు)
- అవుట్పుట్: ఇన్పుట్ అవుట్పుట్ యొక్క సంబంధం.

మొదటి భాగంలో అల్గోరిథం మరియు ఇన్పుట్ పేరు రెండవ భాగంలో ఇన్పుట్ యొక్క ... మూడవ భాగంలో ఇన్పుట్ అవుట్పుట్ యొక్క సంబంధము. దీనినే మనము అవుట్పుట్‌తో ప్రారంభమయ్యే వ్యాఖ్య అని చెప్పుచున్నాము. ఇన్పుట్ మరియు అవుట్పుట్లను ఆంగ్లము మరియు గణిత సంజ్ఞామానాన్ని ఉపయోగించి వ్రాయవచ్చు.

ఉదాహరణ 6.7: ఒక సంఖ్య యొక్క వర్ణమాలమును కనుగొనటానికి అల్గోరిథం యొక్క స్పెసిఫికేషన్స్‌ను వ్రాయండి.

11th Computer Science_Chapter-6 Telugu.indd 84 30-03-2023 18:34:17

1. Square_root వర్గమూలం అనే అల్గోరిథం పేరు పెడదాం.
2. ఇది సంఖ్యను ఇన్పుట్‌గా తీసుకుంటుంది. కాబట్టి ఇన్పుట్‌కు 'n' అనే పేరు పెడదాం. 'n' అనేది ఋణాత్మక (negative) సంఖ్యగా ఉండకూడదు.
3. ఇది 'n' యొక్క వర్గమూలాన్ని అవుట్పుట్‌గా ఇస్తుంది. అవుట్పుట్‌కు 'y' అని పేరు పెడదాం. అప్పుడు 'n' అనేది 'y' యొక్క వర్గముగా ఉంటుంది.

ఇప్పుడు అల్గోరిథం యొక్క వివరణ

square_root (వర్గమూలం) (n)

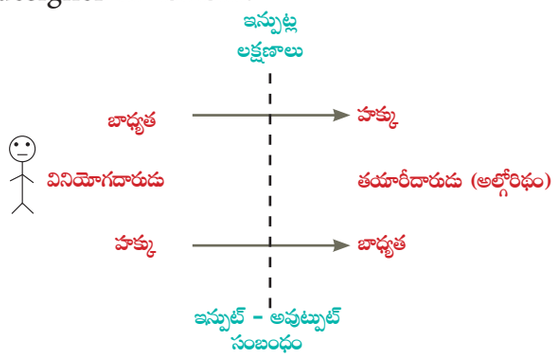
ఇన్పుట్లు : 'n' అనేది వాస్తవ సంఖ్య $n \geq 0$.

అవుట్పుట్లు : y అనేది వాస్తవ సంఖ్య అయితే $y^2 = n$.

6.5.1 స్పెసిఫికేషన్ యొక్క ఒప్పదం (Specification as contract)

అల్గోరిథం యొక్క స్పెసిఫికేషన్ అనేది అల్గోరిథం తయారీదారులకు అల్గోరిథం వినియోగదారులకు మధ్య ఒక ఒప్పందం లాగా పనిచేస్తుంది. ఎందుకనగా ఇది తయారీ దారునికి వినియోగదారునికి మధ్య గల హక్కులు మరియు బాధ్యతలను నిర్వచించును.

ఇన్పుట్లకు అవసరమైన లక్షణాలను (ధర్మాలను) నిర్వహించటం వినియోగదారుని బాధ్యత. కాని అది ఉత్పత్తిదారుని హక్కు ఆశించిన ఇన్పుట్ అవుట్పుట్ సంబంధం అనేది తయారీదారుని బాధ్యత మరియు వినియోగదారుని హక్కు. ముఖ్యంగా వినియోగదారుడు ఇన్పుట్ లక్షణాలను సంతృప్తిపరచటంలో విఫలమైతే ఆశించిన ఇన్పుట్ అవుట్పుట్ సంబంధాన్ని తృప్తిపరిచే బాధ్యత designer కు ఉండదు.



పటము 6.2 ఇన్పుట్ లక్షణాలు మరియు ఇన్పుట్ అవుట్పుట్ సంబంధ హక్కులు మరియు బాధ్యతలు.

ఉదాహరణ 6.8 అల్గోరిథం వర్గమూలం వివరణను (specification) తీసుకుందాం.

వర్గమూలం (n)

-- ఇన్పుట్ n అనేది ఒక వాస్తవ సంఖ్య $n \geq 0$.

-- అవుట్పుట్ : y అనేది ఒక వాస్తవ సంఖ్య అయినపుడు $y^2 = n$.

అల్గోరిథం తయారీదారుడు ఇవ్వబడిన సంఖ్య ఋణాత్మక సంఖ్య కాదు అని నిర్ధారించుకున్నప్పుడు మాత్రమే అల్గోరిథంను నిర్వహించడానికి వీలవుతుంది. వినియోగదారుడు ఆశించిన అవుట్పుట్ అనేది ఇన్పుట్‌గా ఇవ్వబడిన సంఖ్య యొక్క వర్గమూలంగా ఉంటుంది.

ఇవ్వబడిన సంఖ్య యొక్క అవుట్పుట్ అనేది ఋణాత్మక వర్గమూలం కావచ్చు. అవుట్పుట్ ధనాత్మక వర్గమూల సంఖ్యగా ఉండాలని వివరించబడలేదు. ఒకవేళ వినియోగదారుడు ఋణాత్మక సంఖ్యను ఇన్పుట్‌గా ఇచ్చినపుడు అవుట్పుట్ అనేది దాని యొక్క వర్గమూలంగా కానవసరం లేదు.

6.6 సంగ్రహణ (Abstraction)

సైకిల్ తొక్కే వ్యక్తి (నడిపే వ్యక్తి) పెడల్, హ్యాండిల్‌బార్, బ్రేక్ మరియు బెల్ యొక్క పనితీరును తెలుసుకుంటే సరిపోతుంది. రైడర్‌గా (నడిపే వ్యక్తిగా) ఈ నాలుగు విషయాలను దృష్టిలో ఉంచుకుని సైకిలును తయారుచేశారు. సైకిలు యందు ఇంకా చాలా విషయాలు కలవు. కాని సైకిలు త్రొక్కు వ్యక్తికి ఈ విషయాలన్నీ అనవసరం.

ఒక సమస్య చాలా వివరాలను కలిగి ఉంటుంది. సమస్యను పరిష్కరించటానికి ఈ వివరాలన్నీ అనవసరం. కొన్ని వివరాలు మాత్రమే సరిపోతాయి. సమస్యా పరిష్కారంలో అవసరమైన వివరాలను తీసుకుని అనవసర వివరాలను (విషయాలను) విస్మరించటం ముఖ్యం మన దైనందిన జీవితంలో కష్టమైన పనులు చేయటానికి మనకు తెలియకుండానే మనము సంగ్రహణలను (Abstraction) ను ఉపయోగిస్తాము. సంగ్రహణ అనేది క్లిష్టమైన క్రియలను నిర్వహించడానికి అత్యంత ప్రభావంతమైన మానసిక సాధనం. మనము సంగ్రహణలను తగినంతగా సంగ్రహించకపోతే

11th Computer Science_Chapter-6 Telugu.indd 85 30-03-2023 18:34:17

అనవసరమైన విషయాలతో సమస్యా పరిష్కారం చాలా క్లిష్టమవుతుంది.

ఉదాహరణ 6.9: ఒక మ్యాప్ అనునది భూమి మీద మనం చూస్తున్న అన్ని వస్తువుల సంగ్రహణం. మైదానం మీద ఉన్న ప్రతి వివరాన్నీ సూచించడానికి వీలుకాదు. map-maker మనం తెలుసుకోవలసిన విషయాన్ని మాత్రమే ఎంచుకుంటారు. వేర్వేరు ప్రయోజనాల కోసం వేర్వేరు మ్యాపులు గీస్తారు. కాబట్టి వారికి అవసరమైన వివరాలను మాత్రమే అందులో పొందుపరచటం జరుగుతుంది. రోడ్ మ్యాప్ డ్రైవర్ల కొరకు (వాహన చోదకులు) తయారు చేయబడినది కావున వారు తయారు చేసిన మ్యాపులో కొండలు గురించి చింతించరు కావున విస్మరించడం జరిగినది. రహదారి one-way street కాదా అనేదానిపై walker's map ఆశక్తి చూపదు. కాబట్టి అలాంటి విషయాలను విస్మరించబడుతాయి.

ఉదాహరణ 6.10: వైద్యరంగంలో వివిధ నిపుణులు మానవ శరీరం యొక్క విభిన్న సంగ్రహణలతో (abstractions) తో పనిచేస్తారు. ఎముకలకు సంబంధించిన నిపుణుడు అస్తిపంజర వ్యవస్థ యొక్క సంగ్రహణలతో పనిచేయగా ప్రేగులకు సంబంధించిన నిపుణుడు జీర్ణవ్యవస్థకు సంబంధించిన సంగ్రహణులతో పనిచేస్తాడు. ఫిజియోథెరపికి సంబంధించిన నిపుణుడు కండరాల వ్యవస్థకు సంబంధించిన సంగ్రహణులతో పనిచేస్తాడు.

అల్గోరిథంలను నిర్మించునపుడు మనము వివిధ మార్గాలలో సంగ్రహణలను ఉపయోగిస్తాము. సమస్యల నిర్దేశంలో వేరియబుల్స్ ద్వారా దాని స్థితిని సూచిస్తాము. సమస్యను పరిష్కరించటానికి ఏ వివరాలు అవసరం. ఏఏ వివరాలు అనవసరం అనే విషయాల్లో అల్గోరిథంను తయారుచేయు వ్యక్తి శిక్షితుడై ఉండాలి. ఒకవేళ అనవసర విషయములు చేర్చినట్లయితే సమస్యను పరిష్కరించటం చాలా క్లిష్టమవుతుంది.

స్పెసిఫికేషన్ ఇన్పుట్ యొక్క లక్షణములు మరియు ఇన్పుట్ అవుట్పుట్ (రిలేషన్) సంబంధం ద్వారా సమస్యను సంగ్రహిస్తుంది. మనము సమస్యను పరిష్కరించడానికి కావలసిన వివరాలను గుర్తించి అనవసరమైన విషయాలను విస్మరించవలెను.

స్థితి (State) దశ: అల్గోరిథంలో వేరియబుల్స్‌ను స్పష్టమైన మదింపు (గణన) చేసి నిర్మించబడుతుంది.
విధులు (Functions): ఒక అల్గోరిథం చాలా క్లిష్టంగా ఉన్నపుడు మనము దానిని చిన్నచిన్న విధులుగా (functions) గా విడగొట్టి ప్రతి యొక్క విధిని దాని యొక్క స్పెసిఫికేషన్స్ ద్వారా మదింపు చేస్తాము.

6.6.1 స్థితి (State) దశ

స్థితి అనునది ఒక ప్రాథమిక మరియు ముఖ్యమైన సంగ్రహణ. గణన ప్రక్రియ అనేది ప్రాథమిక స్థితిని కలిగి ఉంటుంది. గణన ప్రక్రియ ప్రారంభ స్థితితో ప్రారంభం అవుతుంది. గణన ప్రక్రియ చేసినపుడు దాని యొక్క స్థితి మారుతుంది. ఇది తుది స్థితితో ముగుస్తుంది. అల్గోరిథంలో ఒక గణన ప్రక్రియ యొక్క స్థితి వేరియబుల్స్ ద్వారా సంగ్రహించబడుతుంది. అల్గోరిథంలో ఒక వేరియబుల్ యొక్క స్థితి అనేది ఆ దశలో గల వేరియబుల్ యొక్క విలువ అగును.

ఉదాహరణ 6.11 చాక్లెట్ బార్‌లు (Chocolate Bars): ఒక దీర్ఘ చతురస్రాకారంలో ఉన్న చాక్లెట్ బార్‌ను అడ్డంగా మరియు నిలువుగా కత్తిరించి చతురస్రాకార చాక్లెట్ బార్‌లుగా విభజించాలని అనుకుందాం.

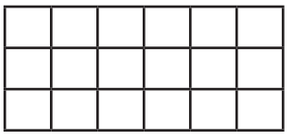
ప్రారంభించడానికి ముందు మనము చాక్లెట్ బార్ మొత్తాన్ని ఒక ముక్కగా కలిగి ఉన్నాము. మనము దానిని ఒకసారి కత్తిరించటం ద్వారా రెండు ముక్కలుగా చేయగలం. ఆ విధంగా ఎన్నిసార్లు కత్తరించటం వల్ల ఆ చాక్లెట్ బార్‌ను చతురస్రాకార ముక్కలుగా చేయగలం.

ఈ యొక్క ఉదాహరణలో సమస్య యొక్క స్పష్టమైన వేరియబుల్స్‌ను గుర్తించగలం. మనము ఈ సమస్యను ఉదాహరణ 8.6 లో పరిష్కరించగలము.
ప్రధాన వేరియబుల్స్ (Essential variables): ముక్కల సంఖ్య మరియు కత్తిరింపులు (కోతల) సంఖ్య అనేది సమస్య యొక్క ముఖ్యమైన వేరియబుల్స్. మనము వాటని వరుసగాp మరియు c అనే రెండు వేరియబుల్స్ ద్వారా సూచించగలము. అందువలన ఈ ప్రక్రియ యొక్క స్థితి p మరియు c అనే రెండు వేరియబుల్స్ ద్వారా సంగ్రహించబడుతుంది.

11th Computer Science_Chapter-6 Telugu.indd 86 30-03-2023 18:34:17

అసంబధమైన వివరాలు (Irrelevant details):

1. సమస్య చాక్లెట్ బార్‌ను చిన్న చిన్న ముక్కలుగా కత్తరించటం లేదా చిన్న చిన్న ముక్క వలే (కాగితాన్ని) కత్తిరించటం ఇది అసంబంధమైనది సమస్య కేవలం చతురస్రాకార గ్రిడ్‌ను చిన్న చిన్న ప్రత్యేక చతురస్రాకారంలో కత్తరించటం.



2. చేసినటువంటి కత్తరింపుల సంఖ్య ముక్కల ఆకారాలు మరియు పరిమాణములు కూడా ఇక్కడ అసంబంధం p మరియు c నుండి మనం ప్రత్యేకమైన చిన్న చిన్న ముక్కల పరిమాణాలను పునర్నిమించలేము. అయితే అది సమస్యను పరిష్కరించడానికి అసంబంధమైన విషయం.

ఉదా: 6.12 ఉదాహరణ 6.2 మేక గడ్డిమోపు మరియు తోడేలు సమస్యను పరిశీలించినట్లయితే ఈ ఉదాహరణలో మనం సమస్య యొక్క లక్షణాలను వ్రాయగలం. ఉదాహరణ 7.1 లో దానిని పరిష్కరించాము. ఈ సమస్యలో నలుగురు వ్యక్తులు ఉంటారు. ఇందులో ఇద్దరు నదికి ఏదేని ఒకవైపు ఉంటారు. దీని అర్థం మనం నాలుగు వేరియబుల్స్ ద్వారా స్థితిని సూచించగలము. వాటిలో ప్రతి రెండు విలువలను ఒకటిగా గుర్తించగలము. వేరియబుల్స్‌ను రైతు మేక గడ్డిమోపు మరియు తోడేలుగా అనుకుందాం.వాటి విలువలు L మరియు R. L యొక్క విలువ అంటే ఎడమవైపుగా R యొక్క విలువ అంటే కుడివైపు గలది అని అర్థం. ఓడ ఎల్లప్పుడూ రైతు వద్దే ఉన్నందువలన దాని స్థానాన్ని సూచించటానికి వేరియబుల్స్‌ని గుర్తుగా సూచించవలసిన అవసరం లేదు.

ప్రారంభ స్థితిలో నాలుగు వేరియబుల్స్ రైతు, మేక, గడ్డిమోపు మరియు తోడేలు యొక్క విలువలు

రైతు, మేక, గడ్డిమోపు మరియు తోడేలు = L, L, L, L

చివరి స్థితిలో నాలుగు వేరియబుల్స్ R విలువను కలిగి ఉండాలి.

రైతు మేక గడ్డిమోపు తోడేలు = R, R, R, R

సమస్య యొక్క స్పెసిఫికేషన్స్

నదిని దాటే క్రమంలో

-- ఇన్పుట్లు : రైతు మేక గడ్డిమోపు తోడేలు = L, L, L, L

-- అవుట్పుట్లు : రైతు మేక గడ్డిమోపు తోడేలు = R, R, R, R

సమస్య రెండు పరిమితులకు లోబడి ఉంటుంది

1. నిబంధనలను అనుసరించి మేకను గడ్డిమోపుతో ఒంటరిగా వదిలిపెట్టలేము.

   if goat = grass then farmer = goat

2. అదే విధముగా మేకను కూడా తోడేలుతో వదిలిపెట్టలేము:

   if goat = wolf then farmer = goat

6.6.2 అసైన్మెంట్ స్టేట్మెంట్ (Assignment statement)

విలువలను నిల్వ చేయటానికి వేరియబుల్స్‌ను boxes అని పేర్లు పెట్టాలి. విలువలను వేరియబుల్స్‌లో నిల్వచేయటానికి అసైన్మెంట్ స్టేట్మెంట్ ఉపయోగించబడుతుంది. ఇది అసైన్మెంట్ operator యొక్క ఎడమ వైపున ఉన్న వేరియబుల్స్‌తో కుడివైపున ఉన్న విలువలతో వ్రాయబడుతుంది.

వేరియబుల్స్ = విలువ

ఈ అసైన్మెంట్ అమలు చేసినపుడు కుడివైపున ఉన్న విలువ ఎడమ వైపున ఉన్న వేరియబుల్స్‌లో నిల్వచేయబడుతుంది.

m := 2

m అనే వేరియబుల్‌లో 2 విలువ నిల్వచేయబడుతుంది.

m

2

ఒకవేళ వేరియబుల్‌లో అప్పటికే ఏదేని విలువను నిల్వచేసిఉన్నట్లయితే అసైన్మెంట్ దాని విలువను కుడివైపున గల విలువకు మార్చుతుంది. వేరియబుల్‌లోని పాత విలువ పోతుంది.

11th Computer Science_Chapter-6 Telugu.indd 87 30-03-2023 18:34:17

అప్పగించిన వేరియబుల్స్‌ను కుడివైపు వ్యక్తీకరించవచ్చు.

వేరియబుల్ = వ్యక్తీకరణ

ఈ సందర్భంలో వ్యక్తీకరణ మూల్యాంకనం చేయబడి వ్యక్తీకరణ యొక్క విలువ వేరియబుల్స్‌లో నిల్వచేయబడుతుంది. వ్యక్తీకరణలో వేరియబుల్ ఉంటే వేరియబుల్ యొక్క ప్రస్తుత విలువ వ్యక్తీకరణను అంచనా వేయడంలో ఉపయోగింపబడి నవీకరించబడుతుంది.

m := m + 3

m యొక్క ప్రస్తుత విలువను ఉపయోగించి m + 3 యొక్క విలువను అంచనా వేస్తుంది

m + 3

= 2 + 3

= 5

వేరియబుల్ m లో విలువ 5 ను నిల్వచేస్తుంది.

m

5

అసైన్మెంట్ స్టేట్మెంట్ ఇరువైపులో := గుర్తుతో వేరుచేయబడిన దీనిని అసైన్మెంట్ ఆపరేటర్ అని పిలుస్తారు.

అసైన్మెంట్ స్టేట్మెంట్ **v := e**

ఇందులో v అనేది e గా చదువబడుతుంది. ఇందులో అసైన్మెంట్ ఆపరేటర్ ఈ క్వాలిటీ ఆపరేటర్ కాదని గుర్తించాలి v := e మరియు v = e యొక్క విలువలు యొక్క గణిత సమానత్వాన్ని పేర్కొనలేదు కాని వేరియబుల్ యొక్క విలువను మార్చుతుంది. m := m + 3 అను విలువలో m + 3 కు సమానమని పేర్కొనలేదు. దానికి బదులుగా ఇది వేరియబుల్ m యొక్క విలువను m + 3 గా మార్చును.

అసైన్మెంట్ స్టేట్మెంట్ వేరియబుల్స్ యొక్క విలువలను ఏకకాలంలో మార్చగలదు. అలాంటప్పుడు ఎడమవైపు ఉన్న వేరియబుల్స్ కుడివైపు ఉన్న వ్యక్తీకరణల సంఖ్యతో సరిపోవాలి. ఉదాహరణకు v1, v2 మరియు v3 అనే వేరియబుల్స్‌ను కేటాయించాలనుకున్నపుడు మనము వ్యక్తీకరణలను el, e2, e3 గా చెప్పాలి.

vl, v2, v3 := el, e2 , e3

ఎడమవైపు ఉన్నవి కామాలతో వేరుచేయబడిన వేరియబుల్స్ యొక్క జాబితా. కుడివైపు ఉన్నవి కమాతో వేరుచేయబడిన వ్యక్తీకరణల జాబితా దీనిని అమలు చేయటానికి దురదృష్టవశాత్తు C ++ తో సహా అనేక ప్రోగ్రామింగ్ భాషలు ఈ గుర్తును (చిహ్నాన్ని) అసైన్మెంట్ ఆపరేటర్‌గా మరొక గుర్తును ఈక్వివలిటీ ఆపరేటర్‌గా ఉపయోగిస్తాయి. అసైన్మెంట్ స్టేట్మెంట్ మొదట వేరియబుల్స్ యొక్క ప్రస్తుత విలువలను ఉపయోగించి కుడివైపున ఉన్న అన్ని వ్యక్తీకరణలను అంచనా వేయాలి. తరువాత ఎడమవైపున ఉన్న సంబంధిత వేరియబుల్స్‌లో నిల్వచేయాలి.

ఉదాహరణ 6.13 లో ఇవ్వబడిన గణన ప్రక్రియ (assignment) తరువాత m మరియు n అనే వేరియబుల్స్ యొక్క 1 వ వరుస మరియు (3) వరుసలో వాటి విలువ ఏమిటి?

1. **m, n := 2 , 5**
2. **-- m, n = ? , ?**
3. **m,n:=m+3,n-1**
4. **-- m, n = ? , ?**

1 వ వరుసలోని అసైన్మెంట్ వేరియబుల్ m లో విలువ 2 అనియు n అనే వేరియబుల్‌లో విలువ 5 అనియు నిల్వచేస్తుంది.

m	n
2	5

3 వ వరుసలోని విలువలను m + 3 మరియు n - 1 లను గణన ప్రక్రియ చేయుటకు m మరియు n యొక్క ప్రస్తుత విలువలను ఉపయోగించవలెను.

m + 3 , n - 1

=2+3,5-1

= 5,4

5 మరియు 4 యొక్క విలువలను వరుసగా m మరియు n అనే వేరియబుల్స్‌లో స్టోర్‌చేస్తుంది.

m	n
5	4

ఒక వేళ C++ సహా పలు ప్రోగ్రామింగ్ భాషలు ఉండినచో అసైన్‌మెంట్ ఆపరేటర్‌గా '=' చిహ్నాన్ని ఈక్వలిటీ ఆపరేటర్‌గా '==' చిహ్నాన్ని వినియోగించాలి.

11th Computer Science_Chapter-6 Telugu.indd 88 30-03-2023 18:34:17

1. **m, n := 2,5**
2. **-- m, n = 2 , 5**
3. **m, n := m + 3, n - 1**
4. **-- m, n = 2 + 3, 5-1 = 5, 4**

రెండు గణన ప్రక్రియలు ముగిసిన తర్వాత వేరియబుల్స్ యొక్క విలువలను 2 మరియు 4 వరుసలలో చూపించబడును.

ఉదాహరణ 6.14 ఉదాహరణ 6.11 లో మనము p మరియు c వేరియబుల్స్ యొక్క గణన ప్రక్రియ స్థితిని రెండుగా విభజించాము. దాని తదుపరి మెట్టు చాక్లెట్ బార్‌ను కత్తిరించే విధానాన్ని రూపొందించటం. ముక్కల సంఖ్య (p) మరియు కోతల (కత్తరింపుల) సంఖ్య (c) రెండూనూ పెరుగుతాయి. మనము దానిని అసైన్మెంట్ స్టేట్మెంట్ ద్వారా మాదిరి model చేయవచ్చును.

p, c := p + 1, c+1

p లను p + 1 మరియు c + 1 అని చదవవలెను.

గుర్తుంచుకోవలసిన విషయాలు

- ప్రోగ్రామింగ్ భాష అనునది ప్రాథమిక భాషా ప్రకటనలను మరియు సమ్మేళన భాషా ప్రకటనలను కూర్పు (composing) చేయటానికి సంకేతములను ఇవ్వబడుతుంది.
- అల్గోరిథం అనునది సమస్యను పరిష్కరించడానికి దశల వారీ ప్రకటనల క్రమం.
- అల్గోరిథం అమలు చేయబడినపుడు ఒక ప్రక్రియ (పద్ధతి) అమలు చేయబడుతుంది. ఈ ప్రక్రియ సమస్య పరిష్కారానికి దారి చూపుతుంది.
- అల్గోరిథం సమస్యా పరిష్కారంలో అల్గోరిథంల నిర్మాణంతో పాటు అల్గోరిథంల లక్షణాలను రుజువు చేస్తుంది.
- అల్గోరిథం స్పెసిఫికేషన్ అనేది (దాని యొక్క ఇన్పుట్‌తో కలిపి) అల్గోరిథం యొక్క పేరు, ఇన్పుట్ స్వభావము మరియు ఆశించిన ఇన్పుట్ అవుట్పుట్ మధ్య సంబంధం అవుతుంది.
- అల్గోరిథం యొక్క వివరణము (specification) అనేది అల్గోరిథంను తయారు చేసిన వ్యక్తికి మరియు వినియోగదారునికి మధ్య ఒప్పదం లాంటిది.
- సంగ్రహణ అనేది పనికి సంబంధంలేని వివరాలను విడిచిపెట్టడం లేదా విస్మరించటం. దీని ద్వారా దాని యొక్క ముఖ్యమైన లక్షణాలను మాత్రమే తీసుకోవటం దీని పని.
- స్పెసిఫికేషన్ సమస్య యొక్క ముఖ్యమైన వేరియబల్స్‌ను తీసుకుని సమస్యను సంగ్రహిస్తుంది.
- వేరియబుల్స్ యొక్క విలువ అనేది అల్గోరిథం యొక్క స్థితిని సూచిస్తుంది.
- అసైన్మెంట్ స్టేట్మెంట్ నందు వేరియబుల్స్ యొక్క విలువ మరియు వేరియబుల్స్ యొక్క స్థితి మారుతుంది.

మూల్యాంకనం

IFEDO

విభాగము –అ

i. సరియైన సమాధానమును ఎన్నుకొనుము

1. క్రింది వానిలో ఏ కార్యకలాపాలలు అల్గోరిథమిక్ స్వభావం కలిగి ఉన్నది?
   (అ) సైకిలుకు భాగాలను కూర్చటం. (ఆ) సైకిలు గూర్చి వివరించటం.
   (ఇ) సైకిలు యొక్క భాగాలకు పేర్లు పెట్టడం. (ఈ) సైకిలు ఏ విధంగా పనిచేస్తుందో వివరించటం.
2. క్రింది వాటిలో ఏ కార్యకలాపాలు అల్గోరిథమిక్ స్వభావం కలిగి ఉండదు?
   (అ) రెండు సంఖ్యలను గుణించటం. (ఆ) ముగ్గు గీయడం.
   (ఇ) ఉద్యానవనంలో నడవటం. (ఈ) రెండు సంఖ్యల మార్పిడి.

11th Computer Science_Chapter-6 Telugu.indd 89 30-03-2023 18:34:17

3. గణన ప్రక్రియలో అవసరంలేని (అనివార్యమైన) వివరాలను వదిలిపెట్టడం (విస్మరించటం) అవసరమైన విషయాలను మాత్రమే సూచించడాన్ని ఏమంటారు?

(అ) స్పెసిఫికేషన్ (ఆ) సంగ్రహణ (ఇ) కూర్పు (ఈ) విడగొట్టడం

4. ఇన్పుట్ యొక్క ధర్మము property మరియు ఇన్పుట్ – అవుట్పుట్ మధ్య సంబంధాన్ని ఏ విధంగా పేర్కొనవచ్చు.

(అ) స్పెసిఫికేషన్ (ఆ) వివరణము (స్టేట్మెంట్) (ఇ) అల్గోరిథం (ఈ) నిర్వచనం (definition)

5. ఇన్పుట్ అవుట్పుట్ సంబంధాన్ని నిర్వచించడం

(అ) అల్గోరిథం యొక్క బాధ్యత మరియు వినియోగదారుని యొక్క హక్కు.

(ఆ) వినియోగదారుని యొక్క బాధ్యత మరియు అల్గోరిథం యొక్క హక్కు.

(ఇ) అల్గోరిథం యొక్క బాధ్యత కాని వినియోగదారుని బాధ్యత కాదు.

(ఈ) వినియోగదారుడు మరియు అల్గోరిథం రెండింటి యొక్క బాధ్యత.

6. అసైన్మెంట్కు (అప్పగించిన పనికి) ముందు i = 5 ఉండి అసైన్మెంట్ తర్వాత i := i-1 అని ఉంటే i యొక్క విలువ

(అ) 5 (ఆ) 4 (ఇ) 3 (ఈ) 2

7. అసైన్మెంట్ ముందు 0 < i ఉండి అసైన్మెంట్ తర్వాత i := i-1 అని ఉంటే మనము దానిని ఏ విధంగా ముగించవచ్చు.

(అ) 0 < i (ఆ) 0 ≤ i (ఇ) i = 0 (ఈ) 0 ≥i

విభాగము – ఆ

ii. అతి సూక్ష్మ సమాధాన ప్రశ్నలు

1. అల్గోరిథం నిర్వచించండి.
2. అల్గోరిథం మరియు ప్రక్రియ (process) కు మధ్య గల భేదాలు ఏవి?
3. ప్రారంభ దశలో,

   రైతు, మేక, గడ్డిమోపు, తోడేలు = L, L, L, L

   రైతు మేకతో నదిని దాటుతాడు. అసైన్మెంట్ స్టేట్మెంటును ఉపయోగించి గణన ప్రక్రియను తయారు చేయండి.
4. కనీసం రెండు సంఖ్యలను కనుగొనడానికి ఒక ఫంక్షన్ను తెలియజేయండి.
5. ఒక వేళ $\sqrt{2}$ = 1.414 మరియు square_root() ను -1.414, అందిస్తే అది క్రింద వాటిని ఉల్లంఘిస్తుందా?

   -- వర్గమూలం (x)

   -- ఇన్పుట్లు: x అనేది వాస్తవ సంఖ్య , $x \geq 0$

   -- అవుట్పుట్లు: y అనేది వాస్తవ సంఖ్య అటువంటి సమయంలో $y^2=x$

విభాగము – ఇ

iii. సూక్ష్మ సమాధాన ప్రశ్నలు

1. ఒక సమస్య సరియైన అల్గోరిథంను అనుసరించి ఉన్నదని మనము ఎలా చెప్పగలం?
2. అల్గోరిథం స్పెసిఫికేషన్ యొక్క నిర్మాణము (format) ఏ విధంగా ఉంటుంది.?

11th Computer Science_Chapter-6 Telugu.indd 90 30-03-2023 18:34:17

3. సంగ్రహణ అనగానేమి?
4. అల్గోరిథంలో స్థితి (state) ఏ విధంగా సూచించబడుతుంది?
5. అసైన్మెంట్ స్టేట్మెంట్ యొక్క రూపము మరియు అర్థం ఏమిటి?
6. అసైన్మెంట్ ఆపరేటర్ మరియు ఈక్వాలిటీ ఆపరేటర్ మధ్య గల తేడా ఏమిటి?

విభాగము – ఈ

వివరంగా వ్రాయండి

1. లంబకోణ త్రిభుజం యొక్క రెండు చిన్న భుజాల పొడవు input మరియు output మూడవ వైపు పొడువుగా ఉండే Hypo tenuse యొక్క అల్గోరిథం యొక్క వివరణ వ్రాయండి.
2. మీరు quadratic equation $ax2 + bx + c = 0$ ని అల్గోరిథం ద్వారా పరిష్కరించాలనుకుంటే.

   quadratic_solve (a, b, c)

   -- ఇన్పుట్ : ?

   -- అవుట్పుట్: ?

   మీరు సూత్రాన్ని ఉపయోగించి కావలసిన (రావలసిన) స్పెసిఫికేషన్ వ్రాయండి.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

3. విషయ మార్పిడి చేయండి. A మరియు B గా గుర్తించబడిన రెండు గ్లాసులు ఇవ్వబడినవి. A అను గ్లాసు పూర్తిగా ఆపిల్ రసంతో B గ్లాసు పూర్తిగా ద్రాక్ష రసంతో నింపబడినది. A మరియు B గ్లాసుకు యొక్క రసమార్పిడి చేయడానికి తగిన వేరియబుల్స్ యొక్క స్థితిని సూచించి అల్గోరిథం యొక్క స్పెసిఫికేషన్ వ్రాయండి.

సూచన పుస్తకాలు (Reference):

1. Roland Backhouse, Algorithmic Problem Solving, John Wiley & Sons Ltd, 2011.
2. Krysia Broda, Susan Eisenbach, Hessam Khoshnevisan, Steve Vickers, Reasoned Programming, Prentice Hall, 1994

11th Computer Science_Chapter-6 Telugu.indd 91 30-03-2023 18:34:17

కూర్పు మరియు విభజన (Composition and Decomposition)

అభ్యసన లక్ష్యాలు

ఈ అధ్యాయములోని అంశమును అభ్యసించిన తరువాత విద్యార్థులు తెలుసుకోగలిగినవి.

- అల్గోరిథమిక్ టెక్నిక్స్‌లో సంజ్ఞామానం (notations) ఉపయోగించడం గురించి తెలుసుకొనుట.
- అల్గోరిథమిక్ టెక్నిక్స్‌లో కూర్పు మరియు విభజన విధానాలను అర్థము చేసుకొనుట.

అధ్యాయము – 1 లో అల్గోరిథమ్స్ అనునది statements తో ఏకీకృతం చేయబడినదని చూడడమైనది. statements లను సమ్మేళన statements లుగా కూడా వర్గీకరించవచ్చును. ఇది అల్గోరిథమ్‌ల యొక్క క్రమానగత నిర్మాణానికి దారితీస్తుంది. ప్రాథమిక సమస్య పరిష్కార పద్ధతులలో విభజన పద్ధతి ఒకటి. ఒక అల్గోరిథమ్‌ను కొన్ని భాగాలుగా విభజించవచ్చును. ఇది ఉన్నత స్థాయి వివరాలను మాత్రమే తెలియజేస్తుంది. అప్పుడు ప్రతి భాగాన్ని చక్కటి వివరాలకు వ్యక్తీకరించే చిన్న భాగాలుగా సులభతరము చేయవచ్చు. లేదా ప్రతి భాగాన్ని ఒక function గా సంగ్రహించవచ్చు.

7.1 అల్గోరిథమ్ కోసం సంకేతాలు (Notations for Algorithms)

అల్గోరిథమ్స్ ప్రాతినిధ్యపరచడానికి కొన్ని సంకేతాలు అవసరమగుతుంది. ఒక అల్గోరిథమ్‌ను ప్రాతినిధ్యపరచుటకు ప్రధానంగా మూడు సంజ్ఞానములు కలవు. అవి

- ప్రోగ్రామింగ్ లాంగ్వేజ్ అనునది కంప్యూటర్‌చే అమలుచేయవలసిన అల్గోరిథంను వ్యక్తీకరించడానికి ఉపయోగపడు ఒక సంజ్ఞామానం (notation) విధము అగును.
- సూడో కోడ్ అనునది ప్రోగ్రామింగ్ లాంగ్వేజ్‌లకు సమానమైన సంజ్ఞామానం. సూడో కోడ్‌లో వ్యక్తీకరించబడిన అల్గోరిథమ్‌లు కంప్యూటర్‌లో అమలు చేయుటకు వీలుకాదు. కాని వ్యక్తుల మధ్య కమ్యూనికేషన్ కేసం ఉద్దేశించబడ్డాయి.
- ఫ్లోచార్ట్ అనునది అల్గోరిథమ్‌ను సూచించడానికి ఒక రేఖా చిత్ర సంజ్ఞామానం అగును. ఇవి అమలు చేయునప్పుడు నియంత్ర ప్రవాహం యొక్క దృశ్యమాన అంతరదృష్టిని మీకు అందిస్తాయి.

7.1.1 ప్రోగ్రామింగ్ లాంగ్వేజ్ (Programming language)

ప్రోగ్రామింగ్ లాంగ్వేజ్ అనునది అల్గోరిథమ్‌ను వ్యక్తీకరించుటకు ఉపయోగించు ఒక సంజ్ఞామానము అగును. తద్వారా ఇది కంప్యూటర్‌లో అల్గోరిథమ్‌ను నిర్వహించును. ప్రోగ్రామింగ్ లాంగ్వేజ్‌లో వ్యక్తీకరించబడిన అల్గోరిథమ్స్‌ను (program) అని అంటారు. C, C++ మరియు పైథాన్ (Python) లాంటివి ప్రోగ్రామింగ్ లాంగ్వేజ్‌కి కొన్ని ఉదాహరణలు. ప్రోగ్రామింగ్ లాంగ్వేజ్ సరియైన విధానములో అధికారికశైలిని కలిగినది అగును. ప్రోగ్రామ్స్ ప్రోగ్రామింగ్ లాంగ్వేజ్ యొక్క grammar ని ఖచ్చితముగా పాటించాలి. విరామ చిహ్నాలు కూడా (punctuation symbols) అలాగే ఉండాలి. తమిళము లేదా ఆంగ్లము లాంటి భాషల అనధికారిక (informal style) శైలిని ఇవి అనుమతించవు. కంప్యూటర్ ద్వారా అమలు చేయగల సూచనలుగా ప్రోగ్రామ్‌ను అనువదించే అనువాదకుడు (translator) ఉండును. ఈ ప్రోగ్రామ్‌లో వ్యాకరణ లోపాలు ఉన్నట్లయితే ఈ (translator) అనువాదము చేయలేరు.

11th Computer Science_Chapter-7 Telugu.indd 92 3/15/2023 12:09:04 PM

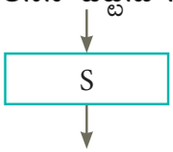
7.1.2 సూడో కోడ్ (Pseudo-code)

Pseudo code అనునది నిర్మాణాలుఉ మరిఉ సాదా ఆంగ్ల మిశ్రమం ఒక మిళిత ఆంగ్ల ప్రోగ్రామింగ్ – లాంగ్వేజ్ అగును. ఈ సాంకేతిక పద్ధతి ఒక అధికారకమైనది లేదా ఖచ్చితమైనది కాదు. ఇది వేరియబుల్స్ మరియు కంట్రోల్ ఫ్లో (control flow) వంటి ప్రోగ్రామ్‌ల వలె అదే బిల్డింగ్ బ్లాక్‌లను ఉపయోగిస్తుంది. కాని ఇఇ వాక్యాంశములు మరియు షరతుల కోసం సహజ ఆంగ్ల వాడకాన్ని అనుమతిస్తుంది. (pseudo) కోడ్ వలె వ్యక్తీకరించబడిన అల్గోరిథం కంప్యూటర్లు నేరుగా అమలు చేయడానికి వీలుకాదు. అయితే మానవ పాఠకులు అర్థం చేసుకోవడానికి అందువలన ప్రోగ్రామింగ్ లాంగ్వేజ్ యొక్క వ్యాకరణ నియమాలను అనుసరించవలసిన అవసరం లేదు. అయినను (pseudo code) లను కఠినమైన మరియు సరైనది. అల్గోరిథమ్‌ను సూచించడానికి సూడో కోడ్ అత్యంత విస్తృతంగా ఉపయోగించే సంజ్ఞామానం.

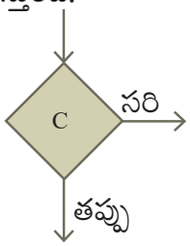
7.1.3 ఫ్లోచార్ట్ (Flowcharts)

ఫ్లోచార్ట్ అనునది అల్గోరిథమ్‌ను సూచించే రేఖా చిత్ర సంజ్ఞామానము. రేఖాచిత్రాలను దృశ్యమాన (visual manner) పద్ధతిలో ఉపయోగించి నియంత్రణ ప్రవాహాన్ని అవి చూపుతాయి. ఫ్లోచార్ట్‌లో దీర్ఘ చతురస్రాకార పెట్టెలు సాధారణ వాక్యాలను సూచిస్తాయి. డైమండ్ ఆకారపు పెట్టెలు షరతులను సూచిస్తాయి. మరియు అల్గోరిథమ్ అమలు సమయంలో నియంత్రణ ఎలా బాక్స్‌లను అమలు చేయాలను క్రమాన్ని చూపించే (arrows) వివరిస్తాయి. ఫ్లోచార్ట్ అంటే భాగాలతో అనుసందానించబడిన స్టేట్మెంట్లు మరియు షరతులను కలిగి ఉన్న బాక్స్‌ల సమూహారం.

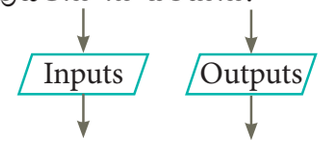
1. ఒకే (outgoing) గుర్తుతో దీర్ఘచతురస్రాకార పెట్టెలో ఒక స్టేట్‌మెంట్ ఉంటుంది. ఇది తదుపరి అమలుచేయవలసిన పెట్టెను సూచిస్తుంది.



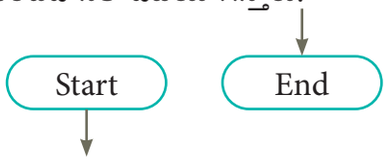
2. డైమండ్ ఆకారపు పెట్టెలో రెండు అవుట్‌గోయింగ్ (arrows) గుర్తులతో సరి (true) మరియు తప్పు (false) అని గుర్తించబడిన ఒక షరతు ఉంటుంది. షరతు సరియైనచో తదువరి అమలు చేయవలసిన పెట్టెకు సరి అనే arrows ను సూచిస్తుంది. షరతు తప్పుగా ఉంటే తప్పు అనే arrows ను తదుపరి అమలు చేయవలసిన పెట్టెను సూచిస్తుంది.



3. Parallelogram పెట్టెలు ఇన్‌పుట్ మరియు అవుట్‌పుట్‌లను సూచించును.



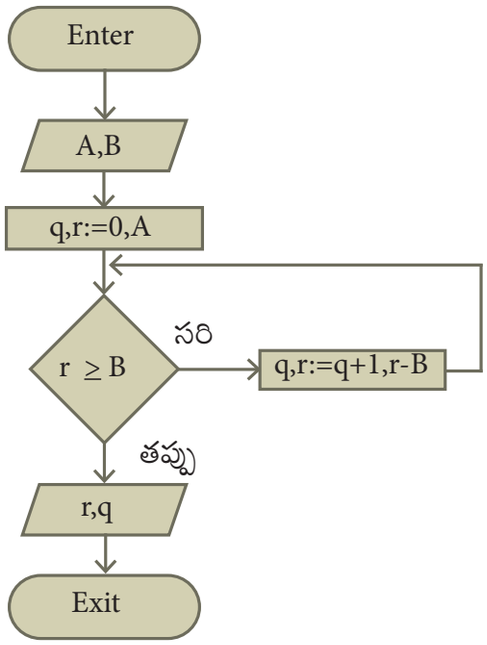
4. Start మరియు End అనే గుర్తు పెట్టబడిన ప్రత్యేక (Special) పెట్టెలు (execution) అమలు యొక్క ప్రారంభాన్ని మరియు ముగింపును సూచించడానికి ఉపయోగిస్తారు.



ఒక పూర్ణాంకము A ను మరొక పూర్ణాంకము B ద్వారా విభజించిన తరువాత భాగఫలం మరియు శేషంను లెక్కించడానికి ఒక అల్గోరిథమ్ యొక్క ఫ్లోచార్ట్ పటం 7.1 లో చూపబడినది. ఇన్‌పుట్ అవుట్‌పుట్ షరతు మరియు అసైన్‌మెంట్ మరియు బాక్స్‌ల మధ్య నియంత్రణ వంటి విభిన్న పెట్టెలను వివరిస్తుంది.అల్గోరిథమ్ ఉదాహరణ 7.4 లో వివరించబడినది.

ఫ్లోచార్ట్‌లోని ప్రతికూలతలు: (1) ప్రోగ్రామింగ్ లాంగ్వేజ్ లేదా సూడోకోడ్‌లో అల్గోరిథమ్‌ల ప్రాతినిధ్యం కంటే ఫ్లోచార్ట్ తక్కువ సంద్రమైనదిగా ఉండును. (2) అల్గోరిథమ్ యొక్కప్రాథమిక క్రమానుగత నిర్మాణమును స్పష్టం చేస్తారు. (3) ప్రత్యామ్నాయ వాక్యాలు మరయు లూప్స్ క్రమబద్ధమైన కూడిన నియంత్రణ నిర్మాణాలు ఉంటాయి. ఫ్లోచార్ట్‌లు క్రమబద్ధమైన నియంత్రణ ప్రవాహ నిర్మాణాలకు మనలను పరిమితం చేయదు.

11th Computer Science_Chapter-7 Telugu.indd 93 3/15/2023 12:09:04 PM



పటం 7.1 Flowchart ఫ్లోచార్ట్ for integer division

7.2 కూర్పు (Composition)

స్టేట్మెంట్ అనేది కంప్యూటర్‌ను పనిచేయమని ఆదేశించే పదబంధం. మనము ఇప్పటికే అసైన్‌మెంట్ స్టేట్‌మెంట్‌ను చూసాము. ఇది వేరియబుల్స్ యొక్క విలువలను మార్చడానికి ఉపయోగించే ఒక సాధారణ స్టేట్‌మెంట్. స్టేట్‌మెంట్‌లో ఇతర స్టేట్‌మెంట్‌లతో కూడి ఉండవచ్చు. ఇది అల్గోరిథమ్స్ యొక్క క్రమానుగత నిర్మాణమునకు దారితీస్తుంది.

ఇతర స్టేట్‌మెంట్‌లతో కూడిన స్టేట్‌మెంట్‌లను సమ్మేళనం స్టేట్‌మెంట్ అంటారు. కంట్రోప్ ఫ్లో స్టేట్‌మెంట్‌లు సమ్మేళనం స్టేట్‌మెంట్స్. ప్రక్రియ యొక్క స్థితిని బట్టి ప్రక్రియ యొక్క నియంత్రణ ప్రవాహాన్ని మార్చడానికి అవి ఉపయోగించబడుతాయి.

మూడు ముఖ్యమైన నియంత్రణ ప్రవాహాలు

- క్రమానుగత
- ప్రత్యామ్నాయం
- పునరావృత

నియంత్రణ ప్రవాహ వాక్యము అమలు చేయబడినపుడు ప్రక్రియ యొక్క స్థితి పరీక్షించబడుతుంది. మరియు ఫలితమును బట్టి అమలుకోసం ఒక స్టేట్‌మెంట్ ఎంపిక చేయబడుతుంది.

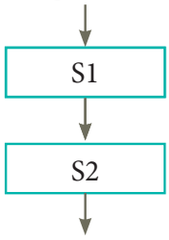
7.2.1 క్రమానుగత స్టేట్‌మెంట్స్ (Sequential statement)

క్రమానుగత స్టేట్‌మెంట్స్ స్టేట్‌మెంట్ యొక్క క్రమమును కలిగి ఉంటుంది. క్రమంలోని స్టేట్‌మెంట్‌ల ఒకదాని తరువాత ఒకటి. అవి అల్గోరిథమ్‌లో వ్రాసిన అదే క్రమంలో అమలు చేయబడుతాయి. మరియు నియంత్రణ ప్రవాహం క్రమానుగతంగా చెప్పబడుతుంది. S1 మరియు S2 స్టేట్‌మెంట్‌లుగా ఉండనివ్వండి. S1 మరియు S2 లతో కూడిన క్రమానుగత స్టేట్‌మెంట్‌లను S1 మరియు S2 గా వ్రాయబడుతుంది.

S1
S2

క్రమానుగత స్టేట్‌మెంట్‌ను అమలు చేయడానికి మొదట S1 చేయండి తరువాత S2 చేయండి

క్రమానుగత స్టేట్‌మెంట్‌లను అమలుచేయడానికి పటం 7.2 లో ఇచ్చిన ఫ్లోచార్ట్‌లో చూడవచ్చును. S1 నుండి S2 వరకు ఉన్న (arrow) గుర్తు S1 అమలు చేయబడినది అని సూచిస్తుంది. మరియు ఆ తరువాత S2 అమలు అగును.



పటం 7.2 క్రమానుగత నియంత్రణ ప్రసారము

ఒక సమస్యను పరిష్కరించుటకు P అను ఇన్‌పుట్‌ను మరియు అనే ఇన్‌పుట్ – అవుట్‌పుట్ సంబంధంగా Q ను ఉండనివ్వండి statement S సమస్యను పరిష్కరిస్తే అది ఇలా వ్రాయబడింది.

1. -- P
2. S
3. -- Q

ఒక సమస్యను రెండు భాగాలుగా విభజించబడిన మొదట సమస్య పరిష్కారానికి S1 మరియు S2 అను రెండు స్టేట్‌మెంట్‌లను క్రమంగా మేము S ని కంపోజ్ చేయాలి, అంటే S1 యొక్క

11th Computer Science_Chapter-7 Telugu.indd 94 3/15/2023 12:09:04 PM

ఇన్‌పుట్ అవుట్‌పుట్ సంబంధం R అని చెప్పండి. R అనునది S2 యొక్క ఇన్‌పుట్ ప్రాపర్టీ

1. -- P
2. S1
3. -- R
4. S2
5. -- Q

ఉదాహరణ 7.1 ఉదాహరణ 6.12 యొక్క రైతు మేక గడ్డి మరియు తోడేలు సమస్యను పరిష్కరిద్దాం. రైతు మేక గడ్డి మరియు తోడేలు అను నాలుగు వేరియబుల్స్ ద్వారా ఈ ప్రక్రియ యొక్క స్థితిని సూచించాలని మేము నిర్ణయించుకున్నాము. ప్రారంభ స్థితిలో నాలుగు వేరియబుల్స్ విలువ L (ఎడమవైపు) కలిగి ఉంటాయి. చివరి స్థితిలో నాలుగు వేరియబుల్స్ విలువ R (కుడివైపు) కలిగి ఉండాలి. ప్రారంభ స్థితి నుండి చివరి స్థితికి వెళ్లడానికి S అను స్టేట్‌మెంట్‌ను నిర్మించడమే లక్ష్యం.

1. **-- farmer, goat, grass, wolf = L, L, L, L**
2. **S**
3. **-- farmer , goat , grass , wolf = R, R, R, R**

మనము మద్యాంతర స్థితులు లేని అసైన్‌మెంట్ స్టేట్‌మెంట్ క్రమం వలె S ను కంపోజ్ చేయాలి.

1. మేక మరియు తోడేలు ఒకే విలువను కలిగి ఉంటాయి. కాని రైతుకు వ్యతిరేఖ విలువ ఉంది లేదా
2. మేక మరియు గడ్డి ఒకే విలువను కలిగి ఉంటాయి రైతుకు వ్యతిరేఖ విలువ ఉంది. ఈ షరత్తులకు లోబడి అసైన్‌మెంట్‌ల క్రమం మరియు ప్రతి అసైన్‌మెంట్‌ల తరువాత నియమాలు పటం 7.3 లో చూపబడింది

క్రమం 1 మరియు 15 పంక్తులు కాకుండా 7 వ పంక్తిలో మేక మరియు గడ్డి ఒకే విలువను కలిగి ఉంటాయి. అయితే రైతుకు కూడా వాటి విలువ అదే వరుస 9 లో మేక మరియు తోడేలు ఒకే విలువను కలిగి ఉన్నాయి. అయితే రైతుకు కూడా వాటి విలువ అదే. ఈ అడ్డంకులను ఉల్లంగించకుండా క్రమం లక్ష్య స్థితిని సాధించింది.

1. -- farmer, goat, grass, wolf = L, L, L, L
2. **farmer, goat := R, R**
3. -- farmer , goat , grass , wolf = R, R, L, L
4. **farmer := L**
5. farmer, goat, grass, wolf = L, R, L, L
6. **farmer, grass := R, R**
7. -- farmer , goat , grass , wolf = R, R, R, L
8. **farmer, goat := L, L**
9. -- farmer, goat, grass, wolf = L, L, R, L
10. **farmer, wolf := R, R**
11. -- farmer , goat , grass , wolf = R, L, R, R
12. **farmer : = L**
13. -- farmer , goat , grass , wolf = L, L, R, R
14. **farmer , goat : = R, R**
15. -- farmer , goat , grass , wolf = R, R, R, R

పటం 7.3 మేక గడ్డి మరియు తోడేలు సమస్య క్రమానుగత అసైన్‌మెంట్లు

7.2.2 ప్రత్యామ్నాయ స్టేట్‌మెంట్ (Alternative statement)

ఒక స్థితిని పరిక్షించడానికి ఉపయోగించే పదబంధం షరతు అగును.

C ఒక షతు మరియు S1 మరియు S2 అనునది స్టేట్‌మెంట్స్ అయితే

```
if  C
    S1
else
    S2
```

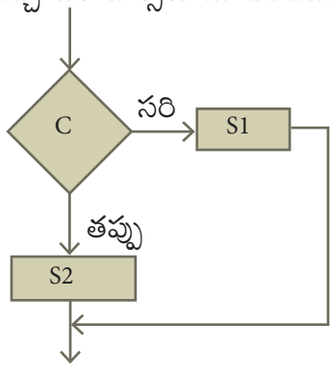
ఇది ప్రత్యామ్నాయ స్టేట్‌మెంట్ అని పిలువబడే స్టేట్‌మెంట్ అయితే ఈ క్రింది వాటిని వివరిస్తుంది.

1. C సరియా కాదా అని పరీక్షించండి.
2. C సరి అయితే S1 చేయండి : లేనిచో S2 చేయండి

సూడో కోడ్ లో కీలక S1 మరియు S2 అను ప్రత్యామ్నాయాలు వరుసగా (if) మరియు (else) అను పదాలకు సమకూర్చడం పదాల నుండి ద్వారా సూచించబడ్డాయి.ప్రత్యామ్నాయ నియంత్రణ ప్రవాహం పటం 7.4 యొక్క ఫ్లోచార్ట్‌లో

11th Computer Science_Chapter-7 Telugu.indd 95 3/15/2023 12:09:04 PM

వర్ణించబడినది. షరతు C లో రెండు అవుట్‌గోయింగ్ (arrow) కలిగి ఉంది. అవి సరి (true) మరియు తప్పు (false) అని లేబుల్ చేయబడింది. సరి arrow S1 పెట్టెను సూచిస్తుంది. తప్పు false arrow S2 పెట్టెఉ సూచిస్తుంది. S1 మరియు S2 యొక్క Out going arrow రెండును ప్రత్యామ్నాయ స్టేట్‌మెంట్ తరువాత వచ్చే ఒకే బాక్స్‌కు సూచించబడతాయి.



పటం 7.4 ప్రత్యామ్నాయ ప్రవాహ నియంత్రణ

కండీషినల్ స్టేట్‌మెంట్ (Conditional statement):

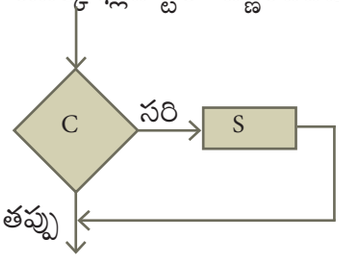
కొన్ని సార్లు మనం ఒక షరతు condition నిజమైతే మాత్రమే స్టేట్‌మెంట్‌ను అమలు చేయాలి మరియు షరతు తప్పుగా ఉంటే ఏమి చేయకూడదు. ఇది ప్రత్యామ్నాయ స్టేట్‌మెంట్‌కు సమానం దీనిలో else-clause ఖాళీగా ఉంటుంది. ప్రత్యామ్నాయ స్టేట్‌మెంట్ యొక్క ఈ రూపాన్ని conditional స్టేట్‌మెంట్ అంటారు. C ఒక షరతు మరియు S ఒక స్టేట్‌మెంట్ అయితే

అప్పుడు C
S

కండీషన్ స్టేట్‌మెంట్ అని పిలువబడే స్టేట్‌మెంట్ క్రింది వాటిని వివరిస్తుంది.

1. C సరి లేదా తప్పు అని పరీక్షించుట
2. C నిజమైతే S చేయండి లేకపోతే ఏమి చేయకండి

conditional control flow నియంత్రణ ప్రవాహం పటం 7.5 యొక్క ఫ్లోచార్ట్‌లో వర్ణించబడినది.



పటం *7.5:* షరతులతో కూడిన నియంత్రణ ప్రవాహం

ఉదాహరణ 7.2 రెండు సంఖ్యల కనిష్టము : a మరియు b అనే రెండు సంఖ్యలను ఇచ్చినట్లు అయితే ప్రత్యామ్నాయ స్టేట్‌మెంట్‌ను ఉపయోగించి రెండింటిలో కనిష్టమును కనుగొనాలనుకుంటున్నాము. ఫలితం result అను వేరియబుల్‌లో కనిష్టమును నిల్వచేద్దాం. a మరియు b యొక్క కనిష్టాన్ని a ↓ b లో సూచిద్దాం. (for instance, 4 ↓ 2 = 2, —5 ↓ 6 = -5). తరువాత అల్గోరిథమ్ కనిష్టం యొక్క స్పెసిఫికేషన్ అనునది.

minimum(a, b)
-- input s : a , b
-- outputs: result = a ↓ b

అల్గోరిథమ్ minimum కనిష్టాన్ని క్రింది విధంగా సూచించవచ్చును.

```
1.  minimum(a, b)
2.      -- a, b
3.      if a < b
4.          result : = a
5.      else
6.          result = b
7.      --  result = a ↓ b
```

7.2.3 విశ్లేషణ (Case analysis)

ప్రత్యామ్నాయ స్టేట్‌మెంట్ సమస్యను రెండు సందర్భాలలో విశ్లేషిస్తుంది. Case analysis స్టేట్‌మెంట్ దీనిని బహుళ Case లకు సాధారణీకరిస్తుంది (generalizes). Case analysis సమస్యను సమగ్రమైన విభజన Case లుగా విభజిస్తుంది. ప్రతి సందర్భంలో సమస్య స్వతంత్రంగా పరిష్కరించబడుతుంది. C1, C2, మరియు C3 నిబంధనలు మరియుS1,S2,S3 మరియు S4 స్టేట్‌మెంట్‌లు అయితే 4-case ల విశ్లేషణ స్టేట్‌మెంట్ క్రింది రూపాన్ని కలిగి ఉంటుంది.

```
1.  case  C1
2.      S1
3.  case  C2
4.      S2
5.  case  C3
6.      S3
7.  else
8.      S4
```


11th Computer Science_Chapter-7 Telugu.indd 96 3/15/2023 12:09:04 PM

C1, C2, మరియు C3 పరిస్థితులను క్రమంగా అంచనా వేస్తారు. ఒప్పు అని అంచనా వేసే మొదటి నిబంధన కోసం సంబంధిత స్టేట్‌మెంట్ అమలు చేయబడుతుంది మరియు Case విశ్లేషణ స్టేట్‌మెంట్ ముగుస్తుంది. ఏవైనా షరతులు ఏవి ఒప్పుకోక పోతే డీఫాల్ట్ Case S4 అమలు చేయబడుతుంది.

1. Case సమగ్రమైనవి: కనీసం ఒక Case అయిన సరి (True) నిబంధనలు తప్పు (False) అయితే డీఫాల్ట్ Case సరి (True) అగును.
2. Case లు వేరుగా (విభేదిస్తే) ఉన్నాయి: Case లలో ఒకటి మాత్రమే సరి ఇది సాధ్యమే అయినప్పటికి ఒకటి కంటే ఎక్కువ షరతులు సరి (True) Case విశ్లేషణ ఎల్లప్పుడు ఒక Case ను మాత్రమే అమలు చేస్తుంది. అది మొదటిది సరి (True) అగును. మూడు షరతులు విబేధిస్తే నాలుగు Case లు (1) C1, (2) C2, (3) C3, (4) C1 కాదు C2 కాదు C3 కాదు

ఉదాహరణ 7.3. మనకు రెండు సంఖ్యలను పోల్చి ఫలితమును ఒకటిగా ఉత్పత్తి చేసే అల్గోరిథమ్ కావాలి.

$$\text{compare}(a, b) = \begin{cases} 1- & \text{if } a < b \\ 0 & \text{if } a = b \\ 1 & \text{if } a > b \end{cases}$$

మనము ఈ స్థితిని ముందే సమగ్రంగా అసమ్మతి కేసుల సమితిగా విభజించవచ్చు. a < b, a = b మరియు a> b Case విశ్లేషణ ఉపయోగించి compare () ను మనం నిర్వచించవచ్చు.

```
1. compare(a, b)
2.     case a < b
3.         result := -1
4.     case a = b
5.         result := 0
6.     else -- a > b
7.         result : = 1
```

7.2.4 పునరావృత స్టేట్‌మెంట్ (Iterative statement)

ఒక పునరావృత ప్రక్రియ C షరతుకు లోబడి అదే చర్య పదే పదే అమలు చేయబడుతుంది. C ఒక షరతు మరియు S ఒక స్టేట్‌మెంట్ అయితే అప్పుడు

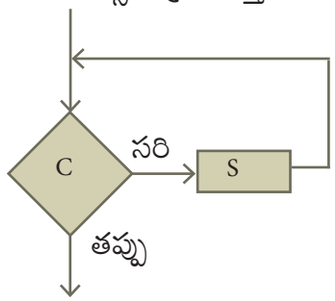
```
while C
    S
```

స్టేట్‌మెంట్‌లను పునరావృత స్టేట్‌మెంట్ అందురు.ఇది క్రింది చర్యను వివరిస్తుంది.

1. C సరియా (True) కాదా అని పరీక్షించండి
2. C నిజమైతే S చేయండి, 1 వ దశకు తిరిగి వెళ్ళండి లేకపోతే ఏమి చేయవద్దు.

పునరావృత స్టేట్‌మెంట్‌ను సాధారణంగా శ్రీశీశీజూ అని అంటారు. ఈ రెండు దశలు అనగా జ ను పరీక్షించడం 3 ను అమలు చేయడం అనేది జ తప్పు అయ్యే వరకు పునరావృతమవుతుంది. జ తప్పుగా మారినప్పుడు శ్రీశీశీజూ ముగుస్తుంది మరియు నియంత్రణ పునరావృత స్టేట్‌మెంట్ ప్రక్కన ఉన్న స్టేట్‌మెంట్‌కు ప్రసారమవుతు. నిబంధన (జశీఅఱఅశీఅ) జ మరియు స్టేట్‌మెంట్ ూ ను వరుసగా లూప్ షరతు మరియు లూప్ బాడీ అంటారు. శ్రీశీశీజూ స్థితిని పరీక్షించడం మరియు శ్రీశీశీజూ బాడీని ఒకసారి అమలు చేయడం పునరావృతం (ఱవతీఱఅశీఅ) అంటారు. జ కాని దానిని ముగింపు నిబంధన అని అంటారు.

పటం 7.6 యొక్క ఫ్లోచార్ట్‌లో పునరుత్పాదక నియంత్రణ ప్రవాహం వర్ణించబడింది. షరతు C లో రెండు అవుట్‌గోయింగ్ Arrows ఉన్నాయి. అవి సరి (True) మరియు తప్పు (False) సరియైన Arrow S పెట్టెను సూచిస్తుంది.C నిజం అయితే S బాక్స్ అమలు అవుతుంది. మరియు నియంత్రణ తిరిగి C బాక్స్‌కు ప్రవహిస్తుంది. పునరావృత స్టేట్‌మెంట్ తరువాత తప్పుడు Arrow స్టేట్‌మెంట్ బాక్స్‌ను చూపుతుంది. C తప్పు అయతే లూప్ ముగుస్తుంది మరియు నియంత్రణలు తరువాత తదుపరి బాక్స్‌కు ప్రవహిస్తుంది.



పటం 7.6 పునరుత్పాదక నియంత్రణ ప్రవాహం

ఉదాహరణ 7.4. ఒక పూర్ణాంక A ను మరొక పూర్ణాంకం B ద్వారా భాగించిన తరువాత

11th Computer Science_Chapter-7 Telugu.indd 97 3/15/2023 12:09:05 PM

కోషియంట్ (భాగఫలం మరియు రిమైండర్‌ను శేషం) లెక్కించడానికి ఒక పునరావృత అల్గోరిథమ్‌ను నిర్మించండి.

ఉదాహరణ 6.6 లోన అల్గోరిథమ్ యొక్క స్పెసిఫికేషన్ (A,B) గా విభజించాము

-- inputs: A ఒక పూర్ణాంకము మరియు B ≠ 0

-- outputs : q మరియు r అదేవిధంగా A = q X B + r మరియు

-- 0 ≤ r < B

ఇప్పుడు మనం వివరణను (స్పెసిఫికేషన్) సంతృప్తిపరిచే పునరావృత అల్గోరిథమ్ నిర్మించవచ్చు.

```
divide (A , B)
-- inputs: A ఒక పూర్ణాంకము మరియు B ≠ 0
-- outputs : q మరియు r అదేవిధంగా A = q X B + r మరియు
--          0 < r < B
    q, r : = 0, A
    while r ≥ B
        q, r := q + 1, r - B
```

పటం 7.1 లో అల్గోరిథమ్‌ను ఫ్లోచార్ట్‌లో ప్రదర్శించడమైనది

Test input కోసం దశల వారిగా అల్గోరిథమ్‌ను అమలు చేయగలము (A, B) = (22, 5) పట్టిక 7.1 యొక్క ప్రతి అడ్డు వరుస పునరావృత వ్యక్తీకరణల యొక్క మూల్యాంకనము మరియు వేరియబుల్స్ యొక్క విలువను చూపిస్తుంది. వ్యక్తీకరణ యొక్క మూల్యాంకనం మునుపటి వరుస నుండి వేరియబుల్స్ యొక్క విలువలను ఉపయోగిస్తుందని గమనించండి. అవుట్‌పుట్ వేరియబుల్స్ q మరియు R ప్రతి పునరావృతంలో వాటి విలువలను మారుస్తాయి. ఇన్‌పుట్ వేరియబుల్స్ A మరియు B వాటి విలువలను మార్చవు. లూప్ ప్రారంభమయ్యే ముందు పునరావృతం 0 విలువలను చూపుతుంది. పునరుక్తి 4 వ చివరిలో షరతు (r ≥ B) = (2 ≥ 5) అనునది తప్పు అందువలన లూప్ (q, r) = (4, 2) తో ముగుస్తుంది.

పునరుక్తి	q	q+1	r	r-B	A	B
0	0		22		22	5
1	1	0+1	17	22-5		
2	2	1 + 1	12	17-5		
3	3	2+1	7	12-5		
4	4	3+1	2	7-5		

పట్టిక *7.1* దశల వారి భాగాహారపు అమలు *(22, 5)*

ఉదాహరణ 7.5. ఉదాహరణ 7.5 ఉదాహరణ 1.3 యొక్క chameleons యొక్క క్రోమ్‌ల్యాండ్ సమస్యలో రెండు రకాల chameleons సమానంగా ఉన్నాయని అనుకుందాం. ఈ రెండు రకాల మధ్య సమావేశాలను ఏర్పాటు చేసే అల్గోరిథమ్‌ను నిర్మించండి. తద్వారా తమ రంగును మూడవ రకానికి మారుస్తారు. చివరకి అందురు ఒకే రంగును ప్రదర్శించాలి.

a,b,c వేరియబుల్స్ ద్వారా ప్రతి రకం chameleons సంఖ్యను మరియు వాటి విలువను వరుసగా A,B మరియు C ద్వారా సూచిద్దాం. a = b input ఆస్థిగా ఉండనివ్వండి. input - output సంబంధం a = b = 0 మరియు c = A+B+C. అల్గోరిథమ్‌కు మోనోక్రోమటైజ్ అని పేరుపెడదాం. ఈ అల్గోరిథమ్‌ను క్రింది విధంగా పేర్కొనవచ్చును.

```
monochromatize(a, b, c)
-- inputs: a=A, b=B, c=C, a=b
-- outputs : a = b = 0 , c = A+B+C
```

ప్రతి పునరుత్పాదక దశలో రెండు రకాలులోని రెండు chameleons కలుస్తాయి. మరియు వాటి రంగులను మూడవదానికి మారుస్తాయి. ఉదాహరణకు A, B, C = 4, 4, 6 అయితే సమావేశముల శ్రేణి పునరావృత్తానికి దారితీస్తుంది.

పునరుక్తి	a	b	c
0	4	4	6
1	3	3	8
2	2	2	10

11th Computer Science_Chapter-7 Telugu.indd 98 3/15/2023 12:09:05 PM

3	1	1	12
4	0	0	14

పట్టిక *7.2* రిండు సమాన సంఖ్యల గల ఊసరవెల్లిల మధ్య సమావేశాల శ్రేణి

ప్రతి సమావేశములో a మరియు b ఒక్కొక్కటి 1 తగ్గుతుంది. మరియు c 2 పెరుగుతుంది పరిష్కారము పునరావృత అల్గోరిథమ్ వలె వ్యక్తీకరించబడుతుంది.

monochromatize(a, b, c)

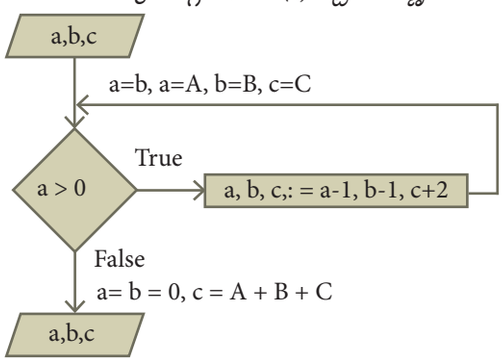
-- inputs: a=A, b=B, c=C, a=b

-- outputs: a = b = 0, c = A+B+C

while a > 0

a, b, c := a-1, b-1, c+2

పటం 7. 7 యొక్క అల్గోరిథమ్ ఫ్లోచార్ట్‌లో వర్ణించబడింది



పటం 7. 7 మోనక్రోమటైజ్ అల్గోరిథమ్

7.3 విభజన (Decomposition)

సమస్య విభాజన ప్రాథమిక సమస్య పరిష్కారాల పద్ధతులలో ఒకటి. ఇది ఒక సమస్యను చిన్న మరియు మరింత నిర్వహించతగిన సమస్యలుగా విభజించడం మరియు అసలు సమస్య పరిష్కారానికి చిన్న సమస్యల పరిష్కారాలను కలపడం. తరచుగా సమస్యలకు నిర్మాణము ఉంటుంది. మనము సమస్య యొక్క నిర్మాణమునఉ ఉపయోగించుకోవచ్చు. మరియు దానిని చిన్న సమస్యలుగా విడగొట్టవచ్చు. అప్పుడు చిన్న సమస్యలను ఇతర సరళమైన మార్గాల ద్వారా పరిష్కరించడానికి తగినంతగా మారే వరకు వాటిని మరింత విచ్ఛిన్నం చేయవచ్చు. అసలు సమస్యకు పరిష్కారం నిర్మించడానికి వాటి పరిష్కారాలు కలిసి ఉంటాయి.

7.3.1 మెరుగుపరచుట (Refinement)

ఒక సమస్యను చిన్న ఉప సమస్యలుగా విభజించిన తరువాత తదుపరి దశ ఉప సమస్యను పరిష్కరణ మెరుగుపరచడం లేదా ఉప సమస్యను సంగ్రహించడం

1. ప్రతి ఉప సమస్యను మరింత వివరణాత్మక దశలుగా విస్తరించవచ్చు. ప్రతి దశను ఇంకా మెరుగైన దశలకు విస్తరించవచ్చు మరియు మొదలైన వాటిని పరిష్కరణ అంటారు.
2. మనం ఉప సమస్యలను కూడా సంగ్రహించవచ్చు. మేము ప్రతి సమస్యను దాని ఇన్‌పుట్ property మరియు input - output సంబంధము ద్వారా పేర్కొంటాము. ప్రధాన సమస్యను పరిష్కరించేటపుడు మేము ఉప సమస్య యొక్క వివరణ స్పెసిఫికేషన్ మాత్రమే తెలుసుకోవాలి. మాకు ఉప సమస్యలను ఎలా పరిష్కరించుట అనేదాని గురించి తెలుసుకొనవలసిన అవసరం లేదు.

ఉదాహరణ 7.6. ఉదయం పాఠశాల వెళ్ళే దారి చర్యను పరిగణించండి. ఈ చర్యను పాఠశాలకు సిద్ధం చేసుకోండి అని వ్రాయవచ్చు.

1 Get ready for school

మనము ఈ చర్యను క్రమం తప్పకుండా తీసుకునే చిన్న మరింత నిర్వహించదగిన చర్య దశలుగా విభజించవచ్చు.

1 అల్పాహారం తీసుకోండి

2 బట్టలు ధరించండి

3 ఇంటిని వదిలి వెళ్ళండి

మేము ఒక చర్యను వివరణాత్మక క్రమంలోకి మెరుగుపరిచాయి. ఏదైన ఒక ప్రతి చర్యను మరింత వివరణాత్మక స్థాయిలో చర్యల శ్రేణిగా విస్తరించవచ్చు మరియు ఈ విస్తరణను పునరావృతం చేయవచ్చు. అల్పాహారం తినండి అనే చర్యను ఇలా విస్తరించవచ్చు.

1 -- అల్పాహారం తీసుకోండి

2 ఇడ్లి తినండి

3 గుడ్లు తినండి

11th Computer Science_Chapter-7 Telugu.indd 99 3/15/2023 12:09:05 PM

4 అరటిపండు తినండి

బట్టలు ధరించండి అనే చర్యను ఇలా విస్తరించవచ్చు.

1 -- బట్టలు ధరించండి
2 బ్లూ బట్టలు ధరించండి
3 సాక్స్ మరియు షూలను ధరించండి
4 ID కార్డ్‌ను వేసుకోండి

మరియు ఇంటిని వదిలి వెళ్ళండి అనే చర్యను ఇలా విస్తరించవచ్చు.

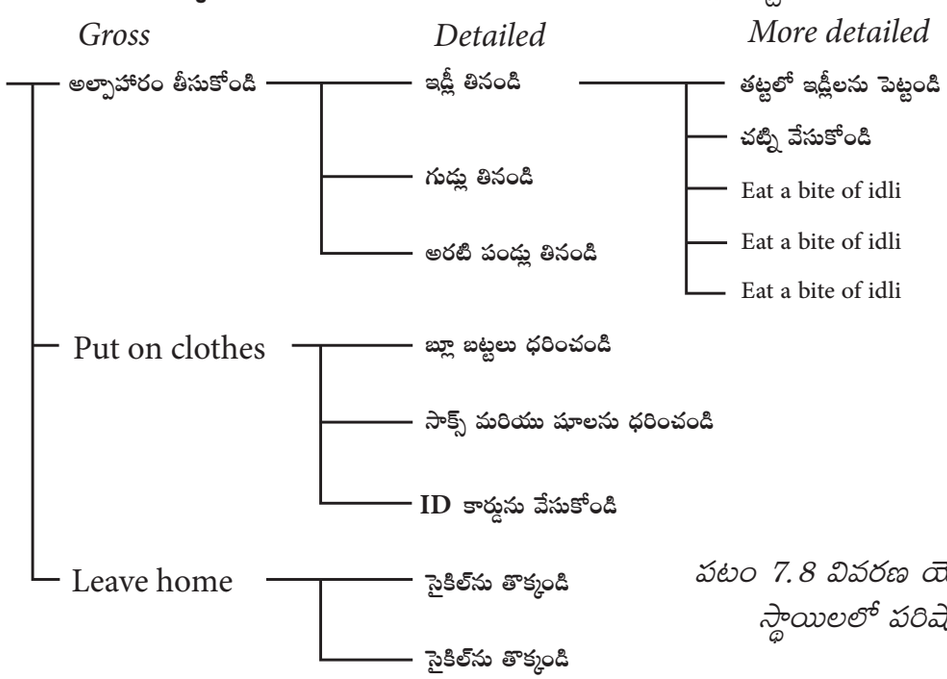
1 -- ఇంటిని వదిలి వెళ్ళండి
2 సైకిల్‌ను భయట పెట్టండి
3 సైకిల్ తొక్కండి

మొత్తంగా స్కూల్‌కు బయలుదేరు అను చర్యను క్రింది విధంగా విస్తరించవచ్చును.

1 -- అల్పాహారం తీసుకోండి
2 ఇడ్లీ తినండి
3 గుడ్లు తినండి
4 అరటి పండ్లు తినండి
5
6 -- బట్టలు ధరించండి
7 బ్లూ బట్టలు ధరించండి
8 సాక్స్ మరియు షూలను ధరించండి
9 ID కార్డును వేసుకోండి
10
11 -- ఇంటిని వదిలి వెళ్ళండి
12 సైకిల్‌ను బయట పెట్టండి
13 సైకిల్‌ను తొక్కండి

పరిష్కారం ఎల్లప్పుడు చర్యల క్రమం కాదు. విద్యార్థి చేసేది చర్య చుట్టుప్రక్కల పై ఆధారపడి ఉంటుంది. ఆమె అల్పాహారం ఎలా తీసుకుంటుందో ఆమె ఎంత ఆకలితో ఉందో మరియు టేబుల్ మీద ఉన్న దానిపై ఆధారపడి ఉంటుంది. ఆమె ఏ బట్టలు వేస్తుందో వారపు రోజులపై ఆధారపడి ఉంటుంది. షరతులతో కూడిన మరియు పునరుక్తి స్టేట్‌మెంట్‌లను ఉపయోగించి పర్యావరణలపై ఆధారపడే ప్రవర్తనను మనము మెరుగుపరచవచ్చు.

1 -- అల్పాహారం తినండి
2 ఆకలిగా ఉంటే మరియు టేబుల్‌పై ఇడ్లీలు ఉంటే
3 ఇడ్లీలు తినండి
4 ఆకలిగా ఉంటే మరియు టేబుల్ పై గుడ్లు ఉంటే
5 గుడ్లు తినండి
6 ఆకలిగా ఉంటే మరియు టేబుల్‌పై అరటిపండ్లు ఉంటే
7 అరటి పండు తినండి
8
8 -- బట్టలు ధరించండి
10 బుధవారం అయితే
11 బ్లూ బట్టలు ధరించండి
12 లేకపోతే
13 తెలుపు బట్టలు ధరించండి



పటం 7.8 వివరణ యొక్క వివిధ స్థాయిలలో పరిష్కారం

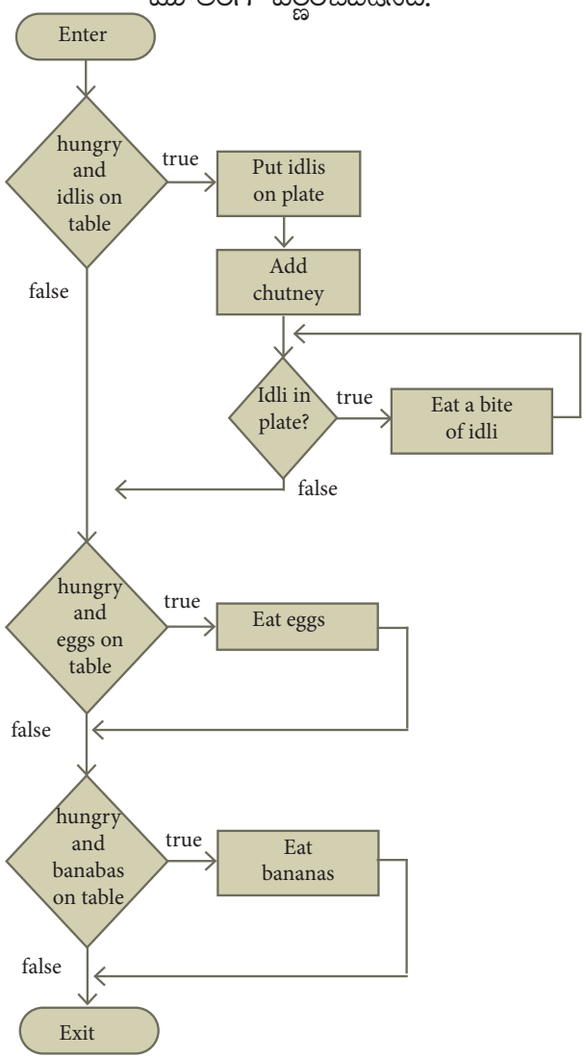
11th Computer Science_Chapter-7 Telugu.indd 100 3/15/2023 12:09:05 PM

14 సాక్స్ మరియు షూలను ధరించండి
15 ID కార్డు వేసుకోండి
16
17 -- ఇంటిని వదిలి వెళ్ళండి
18 సైకిల్ బయట పెట్టండి
19 సైకిల్‌ను తొక్కండి

ఇడ్లి తినడం అనే చర్యను మరొక చర్యగా పరిష్కరణ చేస్తే

1 -- ఇడ్లి తినండి
2 తట్టలో ఇడ్లీలను పెట్టండి
3 చట్నీ వేసుకోండి
4 తట్టలో ఇడ్లీలు ఉన్నంతవరకు
5 ఇడ్లీలు తినండి

పటం 7.8 లో పాఠశాలకు ఎలా సిద్ధం కావాలో అనే చర్య ఏ విధముగా పరిష్కరించబడుతుందో చూద్దాం. పటం 7.9 లో అల్పాహారం తీసుకొనే చర్య ఫ్లోచార్ట్ మూలంగా వర్ణించబడినది.

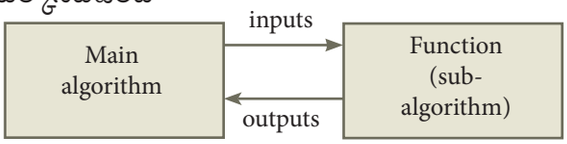


పటం 7.9 అల్పాహారం తీసుకొనుట – ఫ్లోచార్ట్

గమనిక: పై ఫ్లోచార్ట్ పరిష్కారము యొక్క క్రమానగత నిర్మాణమును చూపించదు.

7.3.2 ఫంక్షన్స్ (Functions)

ఒక అల్గోరిథమిక్ సమస్య ఉప సమస్యలుగా విభజించిన తరువాత మేము ఉప సమస్యలను ఫంక్షన్‌లుగా సంగ్రహించవచ్చు. ఒక ఫంక్షన్ ఉప అల్గోరిథమ్ లాంటిది. అల్గోరిథమ్ మాదిరిగానే ఇన్‌పుట్ ప్రాపర్టీ మరియు కావలసిన input-output సంబంధము (relation) ద్వారా ఒక ఫంక్షన్ పేర్కొనబడింది



పటం 7.10 ఫంక్షన్ నిర్వచనం

ప్రధాన అల్గారిథమ్‌లో ఒక ఫంక్షన్‌ను ఉపయోగించడానికి user ఫంక్షన్ యొక్క వివరణ మాత్రమే తెలుసుకోవాలి. ఫంక్షన్ పేరు ఇన్‌పుట్ property మరియు ఇన్‌పుట్ అవుట్‌పుట్ రిలేషన్ ఫంక్షన్‌కు పంపిన ఇన్‌పుట్‌లు పేర్కొన్న ఆస్థితిని సంతృప్తి పరుస్తాయని వినియోగదారుడు (user) నిర్ధారించాలిఇ. మరియు ఫంక్షన్ నుండి అవుట్‌పుట్‌లు ఇన్‌పుట్ – అవుట్‌పుట్ సంబంధాన్ని సంతృప్తిపరుస్తాయని అనుకోవచ్చు. అందువల్ల ఫంక్షన్ యొక్క వినియోగదారులు ఫంక్షన్ ఏమి చేస్తుందో తెలుసుకోవడం మాత్రం అవసరం మరియు ఇది ఫంక్షన్ ద్వారా ఎలా చేయబడుతుందో కాదు. ఫంక్షన్ ఇతర సమస్యలను పరిష్కరించడంలో బ్లాక్ బాక్స్‌ను ఉపయోగించవచ్చు.

అంతిమంగా ఎవరైన అల్గోరిథమ్‌ను ఉపయోగించి ఫంక్షన్‌ని అమలు చేస్తారు. అయినప్పటికి ఫంక్షన్ యొక్క యూజర్‌కు ఫంక్షన్‌ను అమలు చేయడానికి ఉపయోగించే అల్గోరిథమ్ గురించి తెలియదు. ఇది యూజర్ నుండి దాచబడినది. ఫంక్షన్ ఎలా ఉపయోగించాలో యూజర్ తెలుసుకోవలసిన అవసరం లేదు. ఒక ఫంక్షన్‌ను అమలు చేయడానికి ఉపయోగించే అల్గోరిథమ్ దాని స్వంత వేరియబుల్స్‌ను నిర్వచించవచ్చును. ఈ వేరియబుల్స్ ఫంక్షన్ యొక్క స్థానికంగా ఉంటాయి. ఫంక్షన్ యొక్క

11th Computer Science_Chapter-7 Telugu.indd 101 3/15/2023 12:09:05 PM

యూజర్‌కు కనిపించదు. పర్యావసనంగా యూజర్ ప్రధాన అల్గోరిథమ్‌లో నిర్వహించడానికి తక్కువ వేరియబుల్స్ కలిగి ఉంటారు. ప్రధాన అల్గోరిథమ్ యొక్క అయోమయంను తగ్గిస్తుంది.

ఉదాహరణ 7.7 త్రిభుజం లంకోణంలో ఉందో లేదో పరీక్షించే సమస్యను పరిగణించండి. దాని మూడు వైపుల a, b, c, ఇవ్వబడుతుంది.

ఈ లంకోణ త్రిభుజంలో

ఇక్కడ $c^2 = a^2 + b^2$

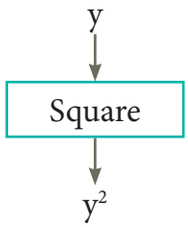
మేము ఒక సంఖ్యను వర్గీకరించే ఉప సమస్యను గుర్తించగలము మనకు function square() ఉందని అనుకొందాం.

square(y)

-- inputs : y

-- outputs : y^2

త్రిభుజం లంబకోణంలో ఉందో లేదో పరీక్షించడానికి మేము ఈ ఫంక్షన్‌ను మూడుసార్లు ఉపయోగించవచ్చు. square() అనేది ఒక "బ్లాక్ బాక్స్" — మేము ఫంక్షన్ ఎలా Square చేస్తుందో దాని గురించి తెలుసుకోము. మేము దాని వివరణను మాత్రం తెలుసుకుంటాం.



పటం 7.11 స్క్వయర్ ఫంక్షన్

```
1  right_angled(a, b, c)
2  -- inputs: c ≥ a, c ≥ b
3  -- outputs: result = true if c² = a² + b²;
4  --          result = false , otherwise
5  if square (c) = square (a) + square (b)
6     result := true
7  else
8     result := false
```

గుర్తుంచుకోవలసిన విషయాలు

- కాంపౌండ్ స్టేట్‌మెంట్‌లు సీక్వెన్షియల్ ప్రత్యామ్నాయ మరియు పునరుక్తి నియంత్రణ ప్రవాహ స్టేట్‌మెంట్‌లతో కలిగి ఉండును.
- వేరియబుల్ యొక్క విలువను బట్టి ఒక పరిస్థితి యొక్క విలువ సరి లేదా తప్పు
- ప్రత్యామ్నాయ స్టేట్‌మెంట్ షరతు విలువలను బట్టి రెండు స్టేట్‌మెంటట్‌లలో ఒకదాని ఖచ్చితంగా ఎంచుకుంటుంది మరియు అమలుచేస్తుంది.
- షరతు నిజమైతేనే షరతులతో కూడిన ప్రకటన అమలు అవుతుంది లేకపోతే ఏమి చేయబడదు.
- పునరావృత స్టేట్‌మెంట్ ఒక షరతును పదే పదే అంచనా వేస్తుంది. మరియు షరతు నిజమైనంత వరకు ఒక ప్రకటన అమలు చేస్తుంది.
- ప్రోగ్రామింగ్ లాంగ్వేజ్ సూడోకోడ్ మరియు ఫ్లో చార్ట్‌లు అనునది వ్యక్తీకరణ అల్గోరిథమ్‌కు సంకేతాలు
- విభజన సమస్యను చిన్న ఉప సమస్యలుగా విభజిస్తుంది మరియు వాటి పరిష్కారాలను మిలితం చేసి అసలు సమస్యను పరిష్కరిస్తుంది.
- ఒక ఫంక్షన్ అనునది ఉప సమస్య యొక్క సంగ్రహణ మరియు దాని ఇన్‌పుట్ ప్రాపర్టీ మరియు దాని ఇన్‌పుట్ అవుట్‌పుట్ రిలేషన్ ద్వారా పేర్కొనబడినది.
- ఫంక్షన్ యొక్క యూజర్ ఫంక్షన్ ఏమి చేస్తుందో మాత్రమే తెలుసుకోవాలి మరియు అది ఎలా చేయబడుతుందో కాదు.
- పరిష్కరించడంలో ఉన్నత స్థాయి ప్రారంభించి ప్రతి స్టేట్‌మెంట్ తదుపరి స్థాయిలో మరింత వివరంగా స్టేట్‌మెంట్‌ను వివరిస్తుంది.

11th Computer Science_Chapter-7 Telugu.indd 102 3/15/2023 12:09:05 PM

మూల్యాంకనము

విభాగము – అ

క్రింది వానిలో సరియైన సమాధానమును ఎనుకొన్నుము

1. ఒక వేల విలువ కట్టడానికి ముందు u, v = 10,5 అయిన, విలువ కట్టిన తరువాత u మరియు v విలువలు ఏవి?

```
1  u := v
2  v := u
```

(అ) u, v = 5 ,5 (ఇ) u, v = 10 ,5

(ఆ) u, v = 5 ,10 (ఈ) u, v = 10 ,10

2. విలువ కట్టిన తరువాత వరుస 3 కు సరిపోయే ధర్మం ఏది?

```
1  --i, j = 0, 0
2  i, j := i+1, j-1
3  -- ?
```

(అ) i+j >0 (ఆ) i+j < 0 (ఇ) i+j =0 (ఈ) i = j

3. C1 అనునది తప్పు మరియు C2 అనునది సరి అయితే కాంఫౌండ్ స్టేట్‌మెంట్

```
1  if C1
2     S1
3  else
4  if  C2
5      S2
6  else
7     S3
```

executes

(అ) S1 (ఆ) S2 (ఇ) S3 (ఈ) none

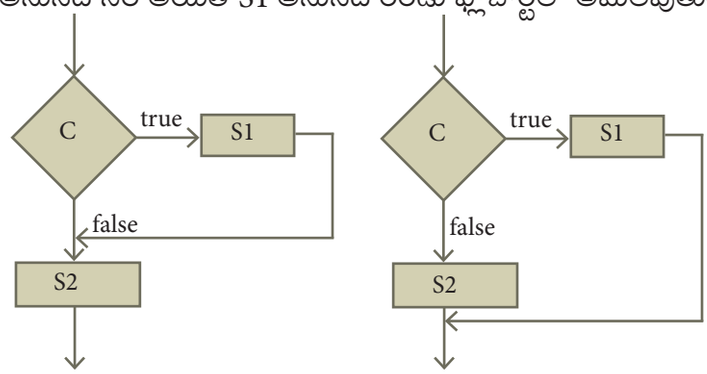
4. లూప్‌కు ముందు C అనునది తప్పు అయితే నియంత్రణ ప్రవాహం ద్వారా

```
1  S1
2  while C
3  S2
4  S3
```

(అ) S1 ; S3 (ఆ) S1 ; S2 ; S3

(ఇ) S1 ; S2 ; S2 ; S3 (ఈ) S1 ; S2 ; S2 ; S2 ; S3

5. C అనునది సరి అయితే S1 అనునది రెండు ఫ్లోచార్ట్‌లో అమలవుతుంది. అయితే S2 దేనిలో అమలవుతుంది.



(1) (2)

11th Computer Science_Chapter-7 Telugu.indd 103 3/15/2023 12:09:05 PM

(అ) (1) మాత్రమే (ఆ) (2) మాత్రమే

(ఇ) (1) మరియు (2) రెండును (ఈ) (1)మరియు(2)ను కాదు

6. లూప్ ఎన్ని సార్లు పునరావృతం అవుతుంది

```
i := 0
while i ≠ 5
  i := i + 1
```

(అ) 4 (ఆ) 5 (ఇ) 6 (ఈ) 0

విభాగము – ఆ

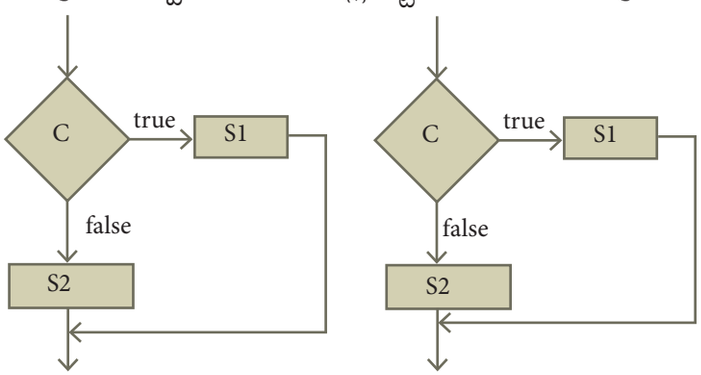
క్రింది లఘు ప్రశ్నలకు సమాధానములిమ్ము

1. షరతు మరియు స్టేట్‌మెంట్‌కు మధ్య గల భేదాలను వ్రాయుము
2. షరతు స్టేట్‌మెంట్ కోసం ఫ్లోచార్ట్ గీయండి
3. షరతులతో కూడిన స్టేట్‌మెంట్ మరియు పునరావృత స్టేట్‌మెంట్ అవి ఎలా భిన్నంగా ఉంటాయి.
4. అల్గోరిథమ్ మరియు ప్రోగ్రామ్ మధ్య గల భేదాలు ఏమి?
5. ఫంక్షన్ ఎందుకు సంగ్రహణ అగును.
6. ఒక స్టేట్‌మెంట్ ఏ విధంగా పరిష్కరించబడును?

విభాగము – ఇ

లఘు ప్రశ్నలకు సమాధానములిమ్ము

1. క్రింద ఇవ్వబడిన రెండు ఫ్లోచార్ట్‌కు సూడోకోడ్‌ను వ్రాయండి



2. ఇవ్వబడిన అల్గోరిథమ్ 2 వ లైన్‌లో C అనునది తప్పు అయితే నిబంధన ప్రవాహాన్ని కనుగొనుము

```
1  S1
2  -- C is false
3  if C
4  S2
5  else
```


11th Computer Science_Chapter-7 Telugu.indd 104 3/15/2023 12:09:05 PM

6 S3

7 S4

3. కేస్ విశ్లేషణ అనగానేమి

4. ప్రత్యామ్నాయ స్టేట్‌మెంట్‌ను ఉపయోగించి కేస్ -3 విశ్లేషణ కోసం ఫ్లోచార్ట్ గీయండి.

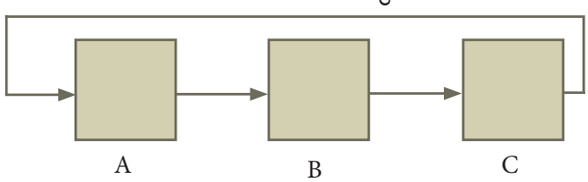
5. ఒక సంఖ్యని రెండు వేరు వేరు మార్గాలలో రెండింతలు చేయడానికి ఫంక్షన్‌ని నిర్వచించండి. (1) n + n, (2) 2 x n

విభాగము – ఈ

క్రింది వాటిని విపులీకరించుము

1. విషయాలను మార్పిడి చేయండి: ఇవ్వబడిన A మరియు B గా గుర్తించబడిన రెండు గ్లాసులలో, గ్లాస్ A నిండ ఆపిల్ పానియం మరియు గ్లాస్ B నిండుగా ద్రాక్ష పానియం నిండి ఉంది. గ్లాస్ A మరియు B యొక్క విషయాలను మార్పిడి చేయడానికి వివరణను వ్రాయండి మరియు వివరణను సంతృప్తి పరచడానికి అసైన్‌మెంట్‌ల క్రమాన్ని రాయండి.

2. విషయాలను ప్రసారం చేయండి: క్రింది చూపిన విధంగా A, B మరియు C వేరియబుల్స్ యొక్క విషయాలను ప్రసారం చేయడానికి వివరణను వ్రాసి ఒక అల్గోరిథమ్‌ను నిర్మించండి: Arrows A యొక్క విలువను B పొందుతాయని B యొక్క విలవని C మరియు C యొక్క విలువని A



3. డేకాంటింగ్ సమస్య మీకు 5,8 మరియు 3 లీటర్ల సామర్థ్యంతో మూడు సీసాలు ఇవ్వబడ్డాయి. 8 లీటర్ల బాటిల్ నూనెతో నిండి ఉండగా మిగత రెండు ఖాలీగా ఉన్నాయి. 8 లీటర్ల బాటిల్‌లోని నూనెను రెండు సమాన పరిమాణాలుగా విభజించండి. తగిన వేరియబుల్ ద్వారా ప్రక్రియ యొక్క స్థితిని సూచించండి. ప్రక్రియ యొక్క ప్రారంభ మరియు చివరి స్థితులు ఏమిటి? అసైన్‌మెంట్ ద్వారా ఒక సీసా నుండి మరొక బాటిల్‌కు చమురు వంచడం నమూనాను రూపొందించండి. తుది స్థితిని సాధించడానికి అసైన్‌మెంట్‌ల క్రమాన్ని వ్రాయండి.

4. కారకాలు 4 కోసం అల్గోరిథమ్ యొక్క దశాల వారి అమలును కనుగొనండి.

```
factorial(n)
-- inputs : n is an integer , n ≥ 0
-- outputs : f = n!
    f, i := 1 ,1
    while i ≤ n
      f, i := f × i, i+1
```


11th Computer Science_Chapter-7 Telugu.indd 105 3/15/2023 12:09:05 PM

అధ్యాయము 8

పునరుక్తి మరియు పునరావృతం (Iteration and recursion)

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును అభ్యసించిన తరువాత విద్యార్థులు గ్రహించునది.

- అల్గారిథమ్ టెక్నిక్స్‌ను ఉపయోగించి వేరియంట్స్ మరియు ఇన్ వేరియంట్స్ అంశాల గురించి తెలుసుకొనుట.
- పునరుక్తి మరియు పునరావృత ప్రక్రియలో అల్గారిథమిక్ పద్ధతిని వర్తించుట.

ఒకే చర్యను పదే పదే చేయడం ద్వారా పరిష్కరించగల సమస్యలు చాలా ఉన్నాయి. పునరుక్తి మరియు పునరావృతం రెండు ఒకే చర్యను పదే పదే అమలు చేయడానికి అల్గారిథమ్ డిజైన్ పద్ధతులు అమలగును. ఒకే చర్యను మళ్ళీ మళ్ళీ చేయడం వల్ల ఉపయోగం ఏమిటి? చర్య ఒకటి అయినప్పటికి చర్య అమలు చేయబడిన స్థితి ఒకేలా ఉండదు. ప్రతిసారి మనము చర్యను అమలుచేస్తున్నప్పుడు స్థానం మారుతుంది. అందువల్ల ఒకే చర్య పదే పదే అమలు చేయబడుతుంది. కాని వివిధ స్థానాలలో కావలసిన ఇన్‌పుట్ అవుట్‌పుట్ సంబంధాన్ని సాధించడానికి ప్రక్రియ పురోగమిస్తున్న విధంగా స్థానం మారుతుంది.

పునరుక్తి (Iteration) : పునరుక్తిలో లూప్ స్థితి నిజమైనంత వరకు లూప్ బాడీ పదే పదే అమలు అవుతుంది. లూప్ బాడీని అమలు చేస్తున్న ప్రతిసారి వేరియబుల్స్ నవీకరించబడుతాయి. ఏదేమైన వేరియబుల్స్ యొక్క ప్రాపర్టీ కూడా ఉంది. ఇది లూప్ బాడీ యొక్క అమలు ద్వారా మారదు. ఈ మార్పులేని ప్రాపర్టీని లూప్ ఇన్‌వేరియంట్ అని అంటారు. పునరుక్తి అల్గారిథమ్‌ల గురించి నిర్మించడానికి మరియు వాదించడానికి లూప్ ఇన్‌వేరియంట్ కీలకం.

మీకు తెలుసా?

కంప్యూటర్ సైన్స్ యొక్క ప్రభావంతమైన మార్గదర్శకాలలో E W Dijkstra ఒకరు. ప్రోగ్రామింగ్ లాంగ్వేజ్ డిజైన్ ఆపరేటింగ్ సిస్టమ్స్, ప్రోగ్రామింగ్ డిజైన్ వంటి విభిన్న రంగాలలో ఆయన ప్రాథమిక సూచనలు చేసారు. అతను సాఫ్ట్‌వేర్ ఇంజనీరింగ్ యొక్క క్రమశిక్షణకు పునాదులు వేయడానికి సహాయపడే structured programming అనే పదాన్ని ఉపయోగించినాడు. 1972 అతనికి ACM ట్యూరింగ్ అవార్డు లభించింది. ఇది కంప్యూటర్ సైన్స్ అత్యున్నత పురస్కారంగా పరిగణించబడింది. E W Dijkstra సారాంశం అని చెప్పబడింది. "కంప్యూటర్ సైన్స్ కంప్యూటర్‌ల గురించి ఖగోళశాస్త్రం టెలిస్కోప్‌ల కంటే ఎక్కువ కాదు".



పునరావృతం (Recursion) : పునరావృతం యొక్క మరొక అల్గారిథమ్ డిజైన్ టెక్నిక్. ఇది పునరుక్తితో దగ్గరి సంబంధం కలిగి ఉంటుంది. కాని మరింత శక్తివంతమైనది. పునరావృతంను ఉపయోగించి మేము ఇచ్చిన ఇన్‌పుట్‌తో సమస్యను సాధిస్తాము. ఇన్‌పుట్‌లోని ఒక భాగంతో అదే సమస్యను పరిష్కరించడం ద్వారా మరియు పరిష్కారము నుండి పాక్షిక ఇన్‌పుట్‌కు అసలు సమస్యకు పరిష్కారాన్ని నిర్మించడం.

8.1 ఇన్‌వేరియంట్ (Invariants)

ఉదాహరణ 8.1 : క్రింది అసైన్‌మెంట్ (u, v) = (20,15) తో అమలు చేయబడింది అని అనుకుందాం అప్పగించిన ముందు తరువాత మనం ఉల్లేఖనం చేయవచ్చు.

11th Computer Science_Chapter-8 Telugu.indd 106 30-03-2023 18:42:11

-- before: u, v = 20, 15

u, v :=u+5,v-5

-- after: u, v = 25, 10

అప్పగించిన తరువాత (u, v) = (25, 10) కాని u + v? ఫంక్షన్ విలువ గురించి మీరు ఏమీ గమనిస్తారు.

before: u + v = 20 + 15 = 35

after: u + v = 25 + 10 = 35

అసైన్‌మెంట్ u + v విలువ మార్చలేదు. u + v అసైన్‌మెంట్ యొక్క ఇన్‌వేరియంట్ అని మేము చెబుతాము. ఇన్‌వేరియంట్ వ్యక్తీకరణతో అప్పగించిన ముందు మరియు తరువాత మేము ఉల్లేఖించవచ్చు.

-- before: u + v = 35

u, v : = u + 5, v - 5

-- after : u + v = 35

u + v అనునది ఒక ఇన్‌వేరియంట్ అని చెప్పగలము ఇది 35 కి ముందు మరియు తరువాత లేదా u + v =35 ఒక ఇన్‌వేరియంట్ అని చెప్పవచ్చు. ఇది ముందు మరియు తరువాత నిజం.

ఉదాహరణ 8.2: మేము ఈ క్రింది అసైన్‌మెంట్‌ను (p, c = 10, 9), తో అసైన్‌మెంట్ తరువాత (p, c) = (11, 10) తో అమలుచేస్తే

-- before : p, c = 10 , 9

p, c := p + 1, c+1

-- after: p, c = 11 , 10

మీరు ఇన్‌వేరియంట్‌ని కనుగొనగలరా? ముందు మరియు తరువాత p - c విలువ ఏమిటి?

before: p — c = 10 — 9 = 1

after: p — c = 11 — 10 = 1

p - c = 1 అనునది ఒక ఇన్‌వేరియంట్ అని కనుగొన్నాము. సాధారణంగా వేరియబుల్స్ యొక్క వ్యక్తీకరణ ఒక అసైన్‌మెంట్‌కు ముందు మరియు తరువాత ఒకే విలువను కలిగి ఉంటే అది అసైన్‌మెంట్ యొక్క ఇన్‌వేరియంట్ P(u, v) వేరియబుల్స్ u మరియు v లతో కూడిన వ్యక్తీకరణగా ఉండనివ్వండి.

P(u, v)[u, v:= el, e2] ను P(u, v) నుండి పొందవచ్చు. ఏకకాలంలో e2 అనేది అసైన్‌మెంట్ P(u, v) అయితే u, v := el, e2 అగును.

P(u,v) [u,v := el, e2] = P(u,v)

ఉదాహరణ 8.3: p - c అనునది అసైన్‌మెంట్ యొక్క ఇన్‌వేరియంట్ అని చూపుము

p, c := p + 1, c + 1

Let P(p, c) = p - c. అయితే

P (p, c) [p, c := p + 1, c + l]

= p — c [p, c := p + 1, c + l]

= (p + 1) — (c + 1)

= p — c

= P(P , c)

కాబట్టి (p - c)[p, c := p+l, c+l] = p - c, p - c అనేది అసైన్‌మెంట్ ... యొక్క ఇన్‌వేరియంట్ p, c := p + 1, c + 1.

ఉదాహరణ 8.4 : m, n := m + 3, n - 1 క్రింద m మరియు n అనే రెండు వేరియబుల్స్‌ను పరిగణించండి. m + 3n వ్యక్తీకరణ ఒక ఇన్‌వేరియంట్ ఇక్కడ P(m, n) = m + 3n అయిన

P(m, n) [m, n := m + 3, n — l]

= m + 3n [m, n := m + 3, n — l]

= (m + 3) + 3(n — l)

= m + 3 + 3n — 3

= m + 3n

= P(m, n)

కావున (m + 3n) [m, n : = m + 3, n - 1] = m + 3n, m + 3n అనునది అసైన్‌మెంట్ m, n := m + 3, n - l యొక్క ఇన్‌వేరియంట్

8.2 లూప్ ఇన్‌వేరియంట్ (Loop invariant)

ఒక లూప్‌లో L అనునది లూప్ బాడీ B యొక్క ఇన్‌వేరియంట్ అయితే L ను లూప్ ఇన్‌వేరియంట్ అంటారు.

11th Computer Science_Chapter-8 Telugu.indd 107 30-03-2023 18:42:11

while C

-- L

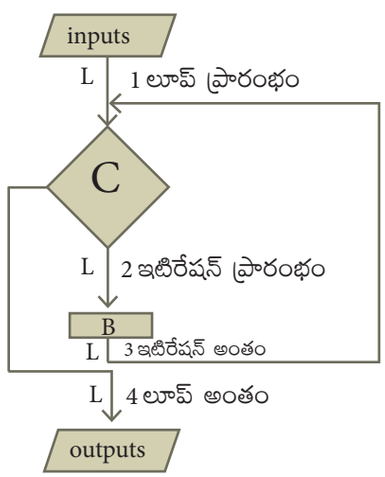
B

-- L

లూప్ బాడీ ముందు మరియు లూప్ బాడీ తరువాత ప్రతిసారి లూప్ ఇన్‌వేరియంట్ నిజం మొదటి పునరావృతం ప్రారంభంలో L నిజం కనుక లూప్ ప్రారంభంలో L నిజం. చివరి పునరావృతం చివరిలో L నిజం కనుక లూప్ కూడ ముగిసినపుడు L నిజం. ఈ విధంగా L ఒక లూప్ వేరియంట్ అయితే అల్గారిథమ్‌లో నాలుగు ముఖ్యమైన పాయింట్‌ల వద్ద ఇది నిజం. అల్గారిథమ్‌లో ఉల్లేఖించినట్లు మరియు పటం 3.1 లో చూపబడింది.

1. లూప్ ప్రారంభంలో (లూప్‌కు ముందు)
2. ప్రతి పునరావృత ప్రారంభానికి ముందు (లూప్ బాడీకి ముందు)
3. ప్రతి పునరావృతానికి చివర (లూప్ బాడీ తరువాత)
4. లూప్ చివరిలో (లూప్‌కు తరువాత)

1. -- L, start of loop
while
C
2. -- L, start of iteration
B
3. -- L, end of iteration
4. -- L, end of loop



పటము 8.1 లూప్‌ఇన్ వేరియంట్ వాస్తవ స్థాన బిందువులు

లూప్‌ను నిర్మించడానికి,

1. లూప్ ప్రారంభంలో లూప్ ఇన్‌వేరియంట్ ఏర్పాటు చేయండి
2. లూప్ బాడీ వేరియబుల్స్‌ను అప్‌డేట్ చేయాలి. తద్వారా చివరికి పురోగమిస్తుంది మరియు అదే సమయంలో లూప్ అస్థిరతను కొనసాగించాలి.
3. లూప్ ముగిసినపుడు ముగింపు పరిస్థితి లూప్ వేరియంట్ ఇన్‌పుట్ అవుట్‌పుట్ సంబంధాన్ని ఏర్పాటు చేయాలి.

8.3 ఇన్‌వేరియంట్ ఉదాహరణలు (Invariants — Examples)

లూప్‌లోని నాలుగు కీలకమైన పాయింట్‌లలో లూప్ వేరియంట్ నిజం. లూప్‌ను ఉపయోగించడం వల్ల లూప్ వేరియంట్‌ను నిర్మించవచ్చు. ఈ పాయింట్స్‌ల వద్ద వేరియబుల్స్ యొక్క లక్షణాలు మరియు కారణాలు గురించి వివరించవచ్చు.

ఉదాహరణ 8.5: a^n ను లెక్కించటానికి ఒక పునరుక్తి అల్గారిథమ్‌ను రూపకల్పన చేయండి. అల్గారిథమ్ (a, n) పేరు పెడుదాం. ఉదాహరణకు

power(10, 4) = 10000
power (5, 3) = 125
power (2, 5) = 32

అల్గారిథమ్ (a, n) ఒక సంచిత n సార్లు గుణించడం ద్వారా a^n ను లెక్కిస్తుంది.

$$a^n = \underbrace{a \times a \times \ldots \times a}_{n \text{ times}}$$

స్పెసిఫికేషన్ మరియు లూప్ వేరియంట్‌లు (comments) గా చూపబడుతాయి.

power (a, n)
-- inputs: n is a positive integer
-- outputs: p = a^n
p, i := 1, 0
while i ≠ n

11th Computer Science_Chapter-8 Telugu.indd 108 30-03-2023 18:42:12

-- loop invariant: $p = a^i$

p, i :=p X a, i+1

power (2, 5) యొక్క దశల వారి అమలు పట్టిక 8.1 లో చూపబడినది. ప్రతి అడ్డు వరుస పునరుక్తి చివరిలో p మరియు i అనే రెండు వేరియబుల్స్ యొక్క విలువలను మరియు అవి లెక్కించబడు విధానమును సూచిస్తుంది. లూప్ ప్రారంభం వద్ద $p = a^i$ నిజం అని మేము చూస్తాము మరియు ప్రతి వరుసలో నిజం కాబట్టి ఇది లూప్ ఇన్‌వేరియంట్.

iteration	p	p x a	i	i+1	a^i
0	1		0		2^0
1	2	1x2	1	0 + 1	2^1
2	4	2x2	2	1 + 1	2^2
3	8	4x2	3	2 + 1	2^3
4	16	8x2	4	3 + 1	2^4
5	32	16x2	5	4+1	2^5

పట్టిక 8.1 ట్రేస్ ఆఫ్ పవర్ (2, 5)

ఇప్పుడు లూప్ ముగిసినపుడు $p = a^1$ అనునది ఇది నిజం అయితే i = 5 సాధారణంగా లూప్ ముగిసినపుడు $p = a^5$ ఈ విధంగా (a, n) దాని స్పెసిఫికేషన్‌ను సంతృప్తి పరుస్తుందని మేము దృవీకరించాము.

ఉదాహరణ 8.6: ఉదాహరణ 6.11 యొక్క చాక్లెట్ బార్ సమస్యను గుర్తుచేసుకోండి. ఎన్ని కోతలు (cuts) బార్‌లు దాని వ్యక్తిగత చతురస్రాలలోక విచ్ఛిన్న చేయడానికి అవసరమా?

ముక్కల సంఖ్యను మరియు కోతల సంఖ్య వరుసగా p మరియు c వేరియబుల్స్ ద్వారా సూచించాలని నిర్ణయించుకున్నారు. మేము ఒక (cuts) చేసినపుడు కోతల సంఖ్య ఒకటి పెరుగుతుంది మరియు ముక్కల సంఖ్య కూడా ఒకటి పెరుగుతుంది. మేము దానిని అసైన్‌మెంట్ ద్వారా మోడల్ చేయాలని నిర్ణయించుకున్నాము.

p, c := p + 1, c+1

బార్‌ను కత్తిరించే విధానంను లూప్ ద్వారా రూపొందించవచ్చు. మేము ఒక ముక్క మరియు సున్న కోతలతో ప్రారంభిస్తాము.

p = 1 మరియు c = 0

n వ్యక్తిగత చతురస్రాల సంఖ్యగా ఉండనివ్వండి. ముక్కల సంఖ్య p వ్యక్తిగత చతురస్రాల సంఖ్యకు సమానమైనపుడు ప్రక్రియ ముగుస్తుంది.

p, c : = 1 , 0

while p ≠ n

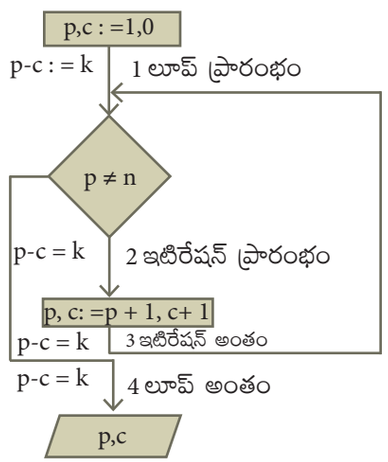
p, c := p + 1, c+1 అనేది

అసైన్‌మెంట్ p యొక్క ఇన్‌వేరియంట్ అని మేము గమనించాము. (ఉదాహరణ 8.2 లో) p, c := p + 1, c + 1 ఇక్కడ p - c = k అనునది ఒక స్థిరాంకము ... నిజం అయితే అల్గారిథమ్‌లోని పాయింట్‌లు దిగువ అల్గారిథమ్‌లో మరియు పటం 8.2 యొక్క ఫ్లోచార్ట్ చూపించబడ్డాయి.

p, c : = 1 , 0

1. -- p - c = k
while p ≠ n
2. -- p - c = k
p, c := p+1, c+1
3. -- p - c = k
4. --p-c=k,p=n

లూప్ ఇన్‌వేరియంట్ p- c = k లూప్ ప్రారంభంలో నిజం (line 1) అంతేకాక లూప్ ప్రారంభంలో p- c = 1 కాబట్టి k = 1 మరియు లూప్ ఇన్‌వేరియంట్ p - c = 1.



పటము 8.2 లూప్ ఇన్ వేరియంట్ల వాస్తవ స్థాన బిందువులు

11th Computer Science_Chapter-8 Telugu.indd 109 30-03-2023 18:42:12

లూప్ ముగిసినపుడు (line 4) లూప్ ఇన్‌వేరియంట్ ఇప్పటికి నిజం (p - c = 1) అంతేకాక లూప్ షరతు తప్పు అయిన (p = n) p - c = 1 మరియు p = n నుండి

1. **p — c = 1** లూప్ ఇన్‌వేరియంట్
2. **p = n** లూప్ చివర
3. **n — c = 1** నుండి **(1), (2)** వరకు
4. **c = n — 1** అనునది **(3)** నుండి

ప్రక్రియ ముగుస్తున్నప్పుడు కోతల (cuts) సంఖ్య చతురస్రాల సంఖ్య కంటే ఒకటి తక్కువ.

ఉదాహరణ 8.7 : 6 చెట్లు సమాన దూరంలో ఉన్నది ఒక్కొక్క చెట్టుపై ఒక పక్షి ఉండే విధంగా 6 చెట్లపై 6 పక్షులు కుర్చున్నాయి. ఒక పక్షి ఒక చెట్టు నుండి మరొక చెట్టుకు ఎగురుతుంటే అదే సమయంలో మరొక పక్షి దాని చెట్టు నుండి మరొక చెట్టుకు అదూ దూరంలో ఎగుతుంది. పక్షులన్ని ఒకే చెట్టు పైకి చేరడం సాధ్యమేనా?

1 నుండి 6 వరకు చెట్లను సూచిక చేద్దాం. పక్షి యొక్క సూచిక అది ప్రస్తుతం కుర్చున్న చెట్టు యొక్క సూచిక. ఒక జత పక్షులు లూప్ యొక్క పునరుక్తి దశగా రూపొందించబడతాయి. చెట్టు వద్ద ఒక పక్షి i చెట్టు i + d కి ఎగిరినపుడు చెట్టు j వద్ద మరొక పక్షి చెట్టుకు ఎగురుతుంది. j — d అందువల్ల ప్రతి పునరుక్తి S తరువాత పక్షుల సూచికల మొత్తం S మారుతూ ఉంటుంది. అంతేకాక లూప్ ఇన్‌వేరియంట్ ప్రారంభంలో మరియు లూప్ చివరిలో నిజం. లూప్ ప్రారంభంలో ఇన్‌వేరియంట్ యొక్క విలువ

S = 1 + 2 + 3 + 4 + 5 + 6 = 21

లూప్ ముగుస్తున్నప్పుడు లూప్ ఇన్‌వేరియంట్ అదే విలువను కలిగి ఉంటుంది. అయినప్పటికి లూప్ ముగిసినపుడు అన్ని పక్షులు ఒకే చెట్టుపై ఉంటే k అని చెప్పండి.
అప్పుడు S = 6k.

S = 21,	లూప్ ప్రారంభంలో లూప్ ఇన్‌వేరియంట్
S = 6k,	లూప్ చివరన లూప్ ఇన్‌వేరియంట్
6k= 21,	లూప్ ఇన్‌వేరియంట్ ప్రారంభ మరియు చివరిలో ఒకే విలువ

21 ను 6 చే గుణించాలి. ఇది 6 కు సాధ్యం కాదు పక్షుల సూచికల యొక్క కావలసిన తుది విలువలు లూప్ మార్పుతో సాధ్యం కాదు. అందువల్ల అన్ని పక్షులు ఒక చెట్టుమీదకి చేరలేవు.

ఉదాహరణ 8.8: ఉదాహరణ 6.3 యొక్క క్రోమ్ ల్యాండ్ యొక్క (Chameleons) ను పరిగణించండి. క్రోమ్ ల్యాండ్‌లో 13 ఎరుపు 15 ఆకుపచ్చ మరియు 17 నీలం (Chameleons) ఉన్నాయి. వేరువేరు రంగులు రెండు (Chameleons) కలిసినపుడు అవి రెండు వాటి రంగును మూడవ దానికి మారుస్తాయి. ఉదాహరణకు (ఎరుపు మరియు ఆకుపచ్చ కలిస్తే రెండు నీలం రంగులోకి మారుతాయి) అన్ని (Chameleons) నీలం రంగును ప్రదర్శించే సమావేశాలను ఏర్పాటు చేయడం సాధ్యమేనా?

r, g, మరియు b ఎరుపు ఆకుపచ్చ నీలం (r, g, b) సంఖ్యలుగా ఉండనివ్వండి. మనము రెండు రకాల సమావేశాలను పునరావృత ప్రక్రియగా మోడల్ చేయవచ్చు. సమావేశం (r-1, g-1, b+2) లేదా (r-1, g+2, b-1) లేదా (r+2, g-1, b-1) ఉదాహరణకు ఎరుపు మరియు ఆకుపచ్చ ఊసరవెల్లి యొక్క సమావేశమును పరిగణించండి.

r, g, b := r-1, g-1, b+2

ఏదైనా రెండు రకాల సంఖ్యల వ్యత్యాసం 3 ద్వారా మారదు లేదా ఇది మార్పు లేనిది.

r - 1 - (g - 1) = r - g

r - 1 - (b + 2) = (r - b) - 3

g - 1 - (b + 2) = (g - b) - 3

ఈ మూడు కేసులకు ఇది వర్తిస్తుంది. పునరుత్పాదక ప్రక్రియ ప్రారంభంలో ఏదైన రెండు రకాలు 3 గుణకారంతో విభిన్నంగా ఉంటే పునరుక్తి ప్రక్రియ ముగిసినపుడు వ్యత్యాసమును 3 నుండి 0 దశలో తగ్గించవచ్చు. అయితే ప్రారంభంలో

r - g = 13 - 15 = -2

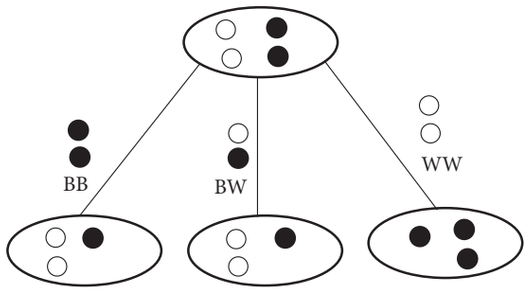
11th Computer Science_Chapter-8 Telugu.indd 110 30-03-2023 18:42:12

g - b = 15 - 17 = -2

b - r = 17 - 13 = 4

రెండు రంగులు 3 యొక్క గుణకారంతో సంఖ్యతో విభిన్నంగా లేవు. అందువల్ల అన్ని chameleons ఒకే రంగుకు మార్చలేము.

ఉదాహరణ 8.9. గోళీల కూజ: మీకు తెలుపు మరియు నలుపు అనే రెండు రకాల గోళీలతో నిండిన కూజ ఇవ్వబడుతుంది. మరియు ఈ ఆట ఆడమని కోరడమైనది. యాదృశ్చికముగా కూజా నుండి రెండు గోళీలను ఎంచుకోండి. అవి ఒకే రంగులో ఉంటే వాటిని విసిరేయండి. మరొక నల్ల గోళీలని ఉంచండి. అవి వేరువేరు రంగులు అయితే తెలుపు రంగును తిరిగి కూజలో ఉంచి నలుపు రంగును విసిరేయండి తెలుపు మరియు నలుపు గోళీల యొక్క అసలు సంఖ్యలు మీకు తెలిస్తే కూజాలోని చివరి గోళీ యొక్క రంగు ఏమిటి?



పటము 8.3 గోళీల కూజలోని స్థాన మార్పిడి

కూజాలోని తెలుపు మరియు నలుపు గోళీల సంఖ్యను w మరియు b అనే రెండు వేరియబుల్ ద్వారా సూచించవచ్చు. ప్రతి పునరుత్పాదక దశలో తీసిన రెండు గోళీల రంగులను బట్టి b మరియు w మారుతుంది. బ్లాక్ బ్లాక్, బ్లాక్ వైట్ లేదా వైట్ వైట్. ఇది పటం 8.3 లో వివరించబడింది మరియు దిగువ అల్గారిథమ్‌లో ఉల్లేఖించబడింది.

1 కూజాలో కనీసం రెండు గోళీల అయితే
2 -- b , w
3 ఏదైనా రెండు గోళీలను తీయండి
4 రెండు case లు నల్లగా ఉంటాయి BB
5 రెండు గోళీలను విసిరివేయండి
6 ఒక నల్ల గోళీను పాలరాయి కూజాలో వేయండి
7 -- b = b '-1, w = w', b+w = b'+w' -1
8 రెండు case లు తెలుపుగా ఉంటే WW
9 రెండు గోళీలను విసిరివేయండి
10 ఒక నల్ల గోళీను పాలరాయి కూజాలో వేయండి
11 --b = b'+1, w = w'-2, --b+w = b'+w'-1
12 else --BW
13 నల్లరంగు గోళీను విసిరివేయండి
14 తెలుపు రంగు గోళీను పాలరాయి కూజాలో వేయండి
15 -- b = b'-1, w = w', b+w = b'+w'-1

ప్రతి సందర్భంలోనూ b, w, b+w ఎలా మారునో అని అల్గారిథమ్‌లో చూపబడుతుంది. ఇక్కడ b' మరియు w' రెండు గోళీలను తీసే ముందు వేరియబుల్స్ యొక్క విలువలు అగును. w' ఎలా మారుతుంది అని గమనించండి. ఒకటి అది మారదు లేదా 2 తగ్గుతుంది. దీని అర్థం w' యొక్క సమానత్వం బేసి అయిన లేదా సరియైనది అయిన మారదు w' యొక్క సమానత్వం ఇన్‌వేరియంట్

ఆట ప్రారంభంలో w సమానంగా అనకుందాం ఆట ముగిసినపుడు w ఇంకా సమానంగా ఉంటుంది. అంతేకాక ఒక గోళీ మాత్రమే మిగిలి ఉంది. w+b = 1.

1	w + b = 1	లూప్ చివర
2	w = 0 లేదా w = 1	1 నుండి
3	w అనునది సరి సంఖ్య	లూప్ ఇన్‌వేరియంట్
4	w = 0	2,3 నుండి
5	b = 1	1,4 నుండి

చివరి గోళీ నలుపుగా ఉండాలి అదే విధంగా ఆట ప్రారంభంలో బేసి సంఖ్యలో తెలుపు ఉంటే చివరి గోళీ తెలుపుగా ఉండాలి.

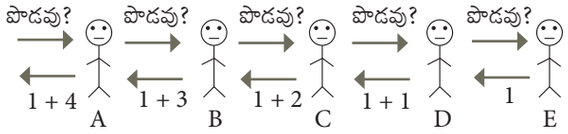
చివరి ప్రశ్న : మనం ఎప్పుడైన ఒకే గోళీ ఉన్న స్థితికి చేరుకుంటామా? అవును ఎందుకంటే ప్రతి దశలో b+w యొక్క మొత్తం గోళీల సంఖ్య ఒక్కొక్కటిగా తగ్గుతుంది. చివరికి అది ఒకటి అవుతుంది.

11th Computer Science_Chapter-8 Telugu.indd 111 30-03-2023 18:42:12

8.4 పునరావృతం (Recursion)

పునరావృతం అనేది అల్గోరిథం డిజైన్ టెక్నిక్ ఇది ప్రేరణకు దగ్గరి సంబంధం కలిగి ఉంటుంది. ఇది పునరావృతంతో సమానంగా ఉంటుంది. కానీ మరింత శక్తివంతమైనది. పునరావృత్తిని ఉపయోగించి ఇన్‌పుట్ యొక్క ఒక భాగంతో సమస్య యొక్క సందర్భాలను పరిష్కరించడం ద్వారా ఇచ్చిన ఇన్‌పుట్‌తో సమస్యను పరిష్కరించవచ్చు.

ఉదాహరణ 8.10: కస్టమర్‌ల కౌంటర్ వద్ద ఒక వరుసలో వేచియున్నారు. కౌంటర్‌లో ఉన్న వ్యక్తి ఎంతమంది కస్టమర్‌లు లైన్‌లో వేచి ఉన్నారు అని తెలుసుకోవాలి అనుకుంటున్నారు.



పటం 8.4 ఒక రేఖ యొక్క పొడవు

పొడవును లెక్కించడానికి బదులుగా అతను కస్టమర్ A ని తన వద్ద ఉన్న రేఖ యొక్క పొడవును అడుగుతాడు, కస్టమర్ A కస్టమర్ B ని పొడవు ఎంత అని అడుగుతుంది. కస్టమ్ B కస్టమర్ E ని అడగుతుంది ఈ విధంగా ప్రతి కస్టమర్ D అని వారి వారి వెనుక ఉన్నవారిని అడుగుతారు. వరుసలో చివరిది అవుతుంది అనుకుందాం. ఈ ప్రశ్న అడిగినపుడు అతను తనను తాను అడిగిన D కి ఒకటి అని సమాధానము ఇస్తాడు. 1+1 = 2 అని C కి చెప్పడం ద్వారా 1+2 = 3 అని D కి చెప్పి స్పందిస్తుంది. 1+3 = 4 అని A కి చెబుతుంది మరియు 1+4=5 అని అడిగిన కస్టమర్‌కు ప్రతిస్పందిస్తారు.

8.4.1 పునరావృత ప్రక్రియ (Recursive process)

ఉదాహరణ 8.11: పునరావృత ప్రక్రియను వివరిస్తుంది. A, B, C, D మరియు E వంటి ఐదు కస్టమర్‌ల క్రమమును సూచిద్దాం.

[A,B,C,D,E]

క్రమం [A,B,C,D,E] యొక్క పొడవును లెక్కించడం ఒక సమస్య మన పరిష్కార పొడవుకు పేరు పెట్టండి. మేము ఒక క్రమమును ఇన్‌పుట్‌గా పాస్‌చేస్తే పరిష్కార పొడవు క్రమం యొక్క పొడవును అవుట్‌పుట్ చేయాలి.

పొడవు **[A,B,C,D,E] = 5**

పరిష్కార పొడవు వరుసను [A,B,C,D,E] దాని మొదటి కస్టమర్ మరియు మిగిలిన క్రమంలోకి విచ్ఛిన్నం చేస్తుంది.

మొదట **[A ,B,C,D,E] = A**

మిగిలిన **[A ,B,C,D,E] = [B ,C,D,E]**

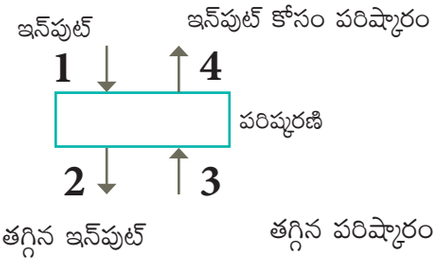
సమస్యను పునరావృతంగా పరిష్కరించడానికి పరిష్కార పొడవు తగ్గిన క్రమంను [B,C,D,E] ను ఉప పరిష్కారించే ఇన్‌పుట్‌గా పంపుతుంది. ఇది పొడవు యొక్క మరొక ఉదాహరణ ఉప పరిష్కారం [B,C,D,E] యొక్క పొడవును ఉత్పత్తి చేస్తుంది 1 ను కూడుతుంది మరియు దానిని [A,B,C,D,E] యొక్క పొడవుగా ఉత్పత్తి చేస్తుంది పరిష్కారం వహిస్తుంది.

పొడవు **[A,B,C,D,E] = 1 +** పొడవు **[B,C,D,E]**

ప్రతి పరిష్కరణి

1. ఇన్‌పుట్‌ను అందుకుంటుంది,
2. ఉప పరిష్కారానికి తగ్గిన పరిమాణం యొక్క ఇన్‌పుట్‌ను పంపుతుంది,
3. ఉప పరిష్కారం నుండి తగ్గిన ఇన్‌పుట్ పరిష్కారాన్ని అందుకుంటుంది. మరియు ఇచ్చిన ఇన్‌పుట్ కోసం పరిష్కారమును ఉత్పత్తి చేస్తుంది

పటం 8.5 లో వివరించిట్లుగా

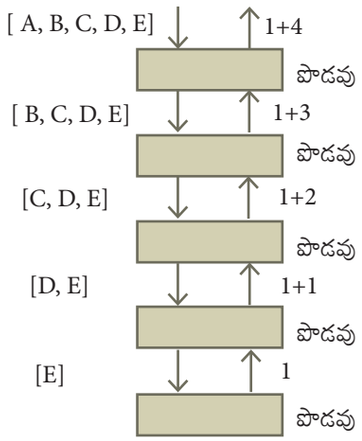


పటం 8.5 ఒక పరిష్కరణి యొక్క ఒక ఉదాహరణ పునరావృత ప్రక్రియ

పటం 8.6 అందుకున్న ఇన్‌పుట్ ఉత్పత్తి చేసిన ప్రతి పరిష్కారమును సూచిస్తుంది. ఉదాహరణ 8.10 కోసం పరిష్కరణి ప్రతి పరిష్కరణి ఇన్‌పుట్ యొక్క పరిమాణాన్ని ఒకదానితో తగ్గిస్తుంది. మరియు

11th Computer Science_Chapter-8 Telugu.indd 112 30-03-2023 18:42:12

దానిని ఉప పరిష్కరణికి పంపుతుంది. దీని ఫలితంగా ఐదు పరిష్కారాలు ఉంటాయి. పరిష్కర్త అందుకున్న ఇన్‌పుట్ నేరుగా పరిష్కారాన్ని అవుట్‌పుట్ చేయడానికి సరిపోయేంత వరకు కొనసాగుతుంది. చివరి పరిష్కార ఇన్‌పుట్‌గా [E]ను అందుకున్నారు. [E] తగినంత చిన్నది కనుక పరిష్కరణి [E] యొక్క పొడవు 1 వెంటనే మరియు పునరావృతం అవుతుంది.



పటము 8.6 పరిష్కర్త, ఉప–పరిష్కర్తలు కల్గిన రికర్సివ్ ప్రాసెస్.

పటం 8.6 పరిష్కర్తలు మరియు ఉప పరిష్కారాలతో పునరావృత ప్రక్రియ పునరావృత పొడవు కోసం ప్రక్రియ [A,B,C,D,E] పటం 8.7 లో చూపబడింది.

1 పొడవు [A,B,C,D,E]

2 = 1 + పొడవు [B,C,D,E]

3 = 1+ 1 + పొడవు [C,D,E]

4 = 1 +1 + 1 + పొడవు [D,E]

5 = 1 + 1 +1 + 1 + పొడవు [E]

6 = 1 + 1 + 1 + 1 +1

7 = 1 +1 +1 + 2

8 = 1 + 1 + 3

9 = 1 + 4

10 = 5

పటము 8.7 ఒక వరుస సంఖ్యా నిడివిని గణించుటకు రికర్సివ్ ప్రాసెస్

8.4.2 పునరావృత సమస్య పరిష్కారం (Recursive problem solving)

ప్రతి పరిష్కరణి ఇన్‌పుట్ పరిమాణాన్ని పరిక్షించాలి. పరిమాణం తగినంతగా ఉంటే పరిష్కర్త ... నేరుగా సమస్యకు పరిష్కారాన్ని అవుట్‌పుట్ చేయాలి. పరిమాణం తగినంతగా లేకపోతే పరిష్కర్త ఇన్‌పుట్ యొక్క పరిమాణాన్ని తగ్గించాలి. మరియు తగిన ఇన్‌పుట్ సమస్యను పరిష్కరించడానికి ఉప పరిష్కారాన్ని పిలవాలి. ఉదాహరణ 8.10 కోసం పరిష్కారం యొక్క అల్గారిథమ్ ఇలా వ్యక్తీకరించబడింది.

$$\text{క్రమం యొక్క పొడవు} = \begin{cases} 1 & \text{క్రమంలో ఒకే ఒక కస్టమర్ ఉంటే} \\ 1 + \text{తోక పొడవు, లేకపోతే} \end{cases}$$

సమస్యను పునరావృత్తంగా పరిష్కరించడానికి ఈ పరిష్కర్త సమస్యను ఉప సమస్యలుగా తగ్గిస్తుంది. మరియు ఉప సమస్యలను పరిష్కరించడానికి ఉప పరిష్కారి అని పిలువబడే పరిష్కర్త యొక్క మరొక ఉదాహరణను పిలుస్తుంది. ఉప సమస్యకు ఇన్‌పుట్ పరిమాణం అసలు సమ్యసకు ఇన్‌పుట్ పరిమాణం కంటే చిన్నది. పరిష్కర్త ఉప పరిష్కారాన్ని పిలిచినపుడు దీనిని పునరావృత కాల్ అంటారు. పునరావృత మాయాజలం (magic) ఉప పరిష్కరణిని ఉప సమస్యకు పరిష్కారమును అందిస్తుందని ఊహించుకోవడానికి అనుమతిస్తుంది. అప్పుడు పరిష్కారం నుండి ఉప సమస్య వరకు ఇచ్చిన సమస్యను పరిష్కర్త పరాష్కారాన్ని నిర్మిస్తారు.

ఉప పరిష్కారాలు సమస్యను చిన్న ఉప సమస్యలుగా పరిమాణాన్ని తగ్గిస్తాయి. చివరికి ఉప సమస్య పునరావృతం లేకుండా నేరుగా పరిష్కరించడానికి సరిపోతుంది. అందువల్ల పునరావృత పరిష్కారానికి రెండు కేసులు (case) ఉన్నాయి.

1. **బేస్ కేస్ (Base case)** : సమస్య పరిమాణం నేరుగా పరిష్కరించడానికి పరిష్కారమును అవుట్‌పుట్ చేయంగా కనీసం ఒక బేస్ కేస్ ఉండాలి.

11th Computer Science_Chapter-8 Telugu.indd 113 30-03-2023 18:42:12

2. **పునరావృత దశ (Recursion step)** సమస్య పరిమాణం తగినంత చిన్నది కాదు ఇచ్చిన సమస్య కంటే పరిమాణంలో ఖచ్చితంగా చిన్నదిగా సమస్యను ఉప సమస్యలుగా మార్చండి. ఉప సమస్యలను పరిష్కరించడానికి ఉప పరిష్కారికి కాల్ చేయండి. ఉప పరిష్కర్త ఉప సమస్యకు పరిష్కారాన్ని ఇస్తుందని అనుకుందాం. ఇవ్వబడిన సమస్యకు పరిష్కారాన్ని నిర్మించండి.

పునరావృత సమస్య పరిష్కార సాంకేతికత యొక్క ఈ రూపురేఖలు (outline) క్రింద చూపించబడ్డాయి.

```
solver (input)
  if input is small enough
  construct solution
else
  find sub_problems of reduced
  input
  solutions to sub_problems =
  solver  for each sub_problem
  construct solution to the
  problem from
  solutions to the sub_problems
```

ఉప సమస్యలకు పరిష్కారాలు మేము పునరావృతాన్ని ఉపయోగించి సమస్యను పరిష్కరించినపుడల్లా మేము ఈ రెండు సందర్భాలను నిర్ధారించాలి. పునరావృత దశలో పునరావృత కాల్ ఇన్‌పుట్ పరిమాణం ఖచ్చితంగా ఉంటుంది. ఇచ్చిన ఇన్‌పుట్ పరిమాణం కంటే చిన్నది మరియు కనీసం ఒక బేస్ కేస్ ఉంటుంది.

8.4.3 పునరావృతం - ఉదాహరణలు (Recursion — Examples)

ఉదాహరణ 8.11: పునరావృత అల్గారిథమ్ ఒక క్రమం యొక్క పొడవు కోసం ఇలా వ్రాయవచ్చు.

```
length (s)
-- inputs : s
-- outputs : length of s
  if s has one customer -- base case
    1
  else
    1 + length(tail(s)) -- recursion step
```

ఉదాహరణ 8.12 పునరావృత రూపకల్పన ఉదాహరణ 8.5 లో గణించడానికి ఒక పునరుక్తి అల్గారిథమ్‌ను నిర్మించాము. a^n పునరావృతంగా నిర్వచించవచ్చు.

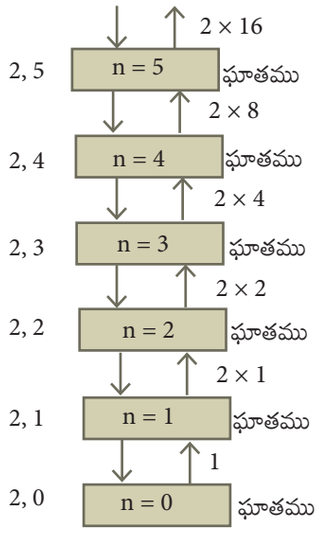
ఉదాహరణ 8.5. a^n can be defined recursively as

$$a^n = \begin{cases} 1 & \text{if } n = 0 \\ a \times a^{n-1} & \text{otherwise} \end{cases}$$

ఈ పునరావృత నిర్వచనంను పునరావృత పరిష్కరణిగా కంప్యూటింగ్ చేయడానికి (a, n) గా వ్యక్తీకరించబడుతుంది.

```
power (a, n)
-- inputs: n is an integer , n ≥ 0
-- outputs : a^n
if n = 0 -- base case
  1
else    --recursion step
        a × power (a, n-1)
```

(2, 5) ను లెక్కించడానికి పునరావృత ప్రక్రియ పరిష్కారముతో పటము 8.8 లో చూపబడుతుంది.



పటం 8.8 ను లెక్కించడానికి పరిష్కారంతో పునరావృత ప్రక్రియ power(2, 5) నుండి వచ్చే పునరావృత ప్రక్రియ

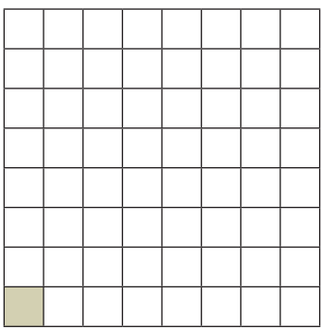
11th Computer Science_Chapter-8 Telugu.indd 114 30-03-2023 18:42:12

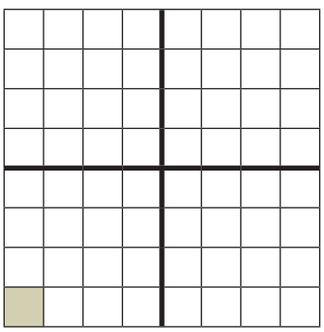
ఘాతం (2,5) నుండి రూపొందెడి రికర్సివ్ ప్రాసెస్. పటం 8.9లో చూపబడింది.

ఘాతం (2,5)
= 2 × ఘాతం (2,4)
= 2 × 2 × ఘాతం(2,3)
= 2 × 2 × 2 × ఘాతం(2, 2)
= 2 × 2 × 2 × 2 × ఘాతం (2,1)
= 2 × 2 × 2 × 2 × 2 × ఘాతం (2,0)
= 2 × 2 × 2 × 2 × 2 × 1
= 2 × 2 × 2 × 2 × 2
= 2 × 2 × 2 × 4
= 2 × 2 × 8
= 2 × 16
= 32

పటం 8.9 ఘాతం*(2, 5)* కోసం పునరావృత ప్రక్రియ

ఉదాహరణ 8.13 మూలలో కప్పబడిన బోర్డ్ ($2^n \times 2^n$) చతురస్రాల బోర్డ్. దీనిలో ఒక మూలలోని చదరము ఒకే చదరపు పలకచే కప్పబడి ఉంటుంది. triominoe అనేది L ఆకారపు జైల్. ఇది మూడు ప్రక్కనే ఉన్న చతురస్రాలతో ఏర్పడుతుంది. (పటం 8.10 లో చూడండి) మూలలో కప్పబడిన బోర్డ్‌ను అతి వ్యాప్తి లేకుండా L ఆకారపు triominoe లతో కప్పండి. triominoe ను కావలసిన విధముగా త్రిప్పుకోవచ్చు.

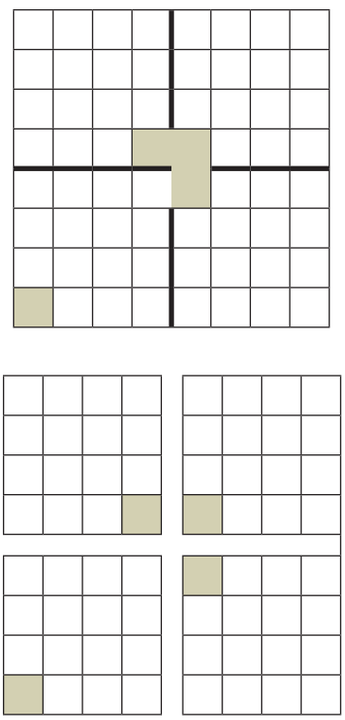




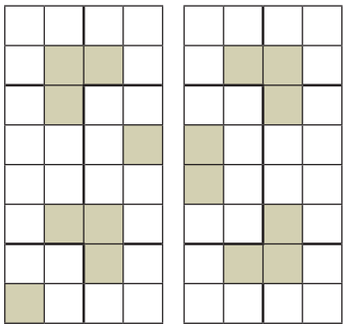


పటము 8.10 మూలలు కప్పబడిన బోర్డ్ మరియు triominoe

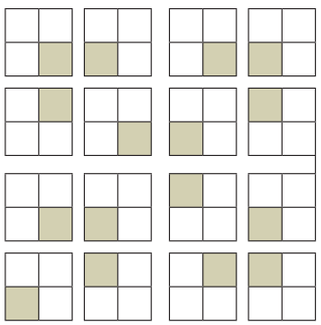
సమస్య యొక్క పరిమాణం n (బోర్డ్ యొక్క పరిమాణం $2^n \times 2^n$) మేము పునరావృతం ద్వారా సమస్యను పరిష్కరించగలము. ఈ బేస్ కేస్ n = 1 ఇది 2×2 మూలలో కప్పబడిన బోర్డ్. మనము దానిని ఒక triominoe తో కప్పవచ్చు మరియు సమస్యను పరిష్కరించవచ్చు. పునరావృత దశలో మూలలో కప్పబడిన $2^n \times 2^n$ పరిమాణమును 4 ఉప బోర్డ్‌లుగా విభజించండి. ప్రతి పరిమాణము $2^{n-1} \times 2^{n-1}$ మధ్యలో సమాంతర మరియు నిలువ గీతలు బోర్డ్ మధ్యలో గీయడం ద్వారా పటం 8.11 లో చూపిన విధంగా మూలలో కప్పబడిన సబ్ బోర్డ్‌ను కవర్ చేయకుండా ఉండడానికి మొత్తం బోర్డ్ మధ్యలో ఒక triominoe ను ఉంచండి. ఇప్పుడు మన దగ్గర నాలుగు మూలలు కప్పబడిన బోర్డ్స్ పరిమాణం $2^{n-1} \times 2^{n-1}$ కలదు.

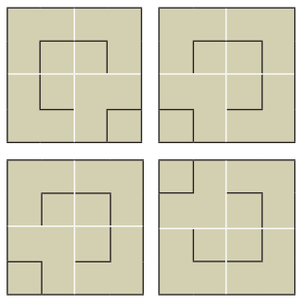


11th Computer Science_Chapter-8 Telugu.indd 115 30-03-2023 18:42:12



పటం 8.11 పరిమాణం 2 x 2^3 యొక్క మూలలతో కప్పబడిన బోర్డ్‌ను cover చేసే పునరావృత ప్రక్రియ





మన వద్ద 4 ఉప సమస్యలు ఉన్నాయి. దీని పరిమాణం ఇచ్చిన సమస్య యొక్క పరిమాణం కంటే ఖచ్చితంగా తక్కువగా ఉంటుంది. మన ప్రతి ఉప సమస్యలను పునరావృతంగా పరిష్కరించగలము.

```
tile corner_covered board of size n
  if n = 1 -- base case
     cover the 3 squares with one trominoe
else    -- recursion step
     divide board into 4 sub_boards of size n-1
     place a trominoe at centre of board ,
     leaving out the corner_covered sub-board
     tile each sub_board of size n-1
```

2^3 x 2^3 మూలలో కప్పబడిన బోర్డ్‌ను కవర్ చేయడానికి వచ్చే పునరావృత ప్రక్రియ పటం 8.11 లో వివరించబడినది.

గుర్తుంచుకోవలసిన అంశాలు

- షరతు నిజమైనంతవరకు ఒక పరిస్థితిని అంచనవేయడానికి మరియు ఒక స్టేట్‌మెంట్‌ను అమలు చేయడానికి రెండు దశలను పునరావృతం చేస్తుంది.
- వేరియబుల్స్‌తో కూడిన వ్యక్తీకరణ ఈ వేరియబుల్స్‌తో ఒకదానికి అప్పగించడం ద్వారా మారదు. దీనిని అసైన్‌మెంట్ యొక్క ఇన్‌వేరియంట్ అని పిలుస్తారు.
- లూప్ బాడి యొక్క ఇన్‌వేరియంట్‌ను లూప్ ఇన్‌వేరియంట్ అంటారు.
- లూప్ ఇన్‌వేరియంట్ నిజం అగును.
- (a) లూప్ ప్రారంభంలో (లూప్‌కు ముందు)
- (b) ప్రతి పునరావృత ప్రారంభంలో (లూప్ బాడీ)
- (c) ప్రతి పునరావృతం చివరిలో (లూప్ బాడీ తరువాత)
- (d) లూప్ చివరిలో (లూప్ తరువాత)
- లూప్ ముగిసినపుడు లూప్ ఇన్‌వేరియంట్ నిజం అదనంగా ముగింపు షరతు కూడా నిజం
- పునరావృతానికి కనీసం ఒక బేస్ కేస్ ఉండాలి.
- పునరావృత దశ సమస్యను చిన్న పరిమాణంని ఉప సమస్యలుగా విభజిస్తుంది. ఉప సమస్యలకు పరిష్కారాలు పునరావృత కాల్ ద్వారా ఇవ్వబడతాయి. మరియు ఇచ్చిన సమస్యకు పరిష్కారమును నిర్మిస్తుంది.
- పునరావృతంలో ఉప సమస్యకు ఇన్‌పుట్ పరిమాణం ఇచ్చిన ఇన్‌పుట్ పరిమాణం కంటే ఖచ్చితంగా తక్కువగా ఉండాలి.

11th Computer Science_Chapter-8 Telugu.indd 116 30-03-2023 18:42:12

మూల్యాంకనము

WG0U8

విభాగం - అ

సరియైన సమాధానమును ఎన్నుకొనుము

1. లూప్ ఇన్‌వేరియంట్ నిజం కాదు

   (అ) లూప్ ప్రారంభంలో (ఆ) ప్రతి పునరావృత ప్రారంభంలో

   (ఇ) ప్రతి పునరావృత చివరిలో (ఈ) అల్గారిథమ్ ప్రారంభంలో

2. మేము చెస్ బోర్డ్‌ను డామినోతో కప్పాలనుకొంటున్నాము □□ ఈ నల్ల చతురస్రాల సంఖ్య మరియు డామినీల్స్ కప్పబడిన తెల్ల చతురస్రాల సంఖ్య వరుసగా డామినల్ డబ్బాను ఉంచడం రూపొందించబడుతుంది.

   (అ) b := b + 2 (ఆ) w := w + 2 (ఇ) b, w := b+1, w+1 (ఈ) b := w

3. m x a + n x b అనునది అసైన్‌మెంట్ యొక్క ఇన్‌వేరియంట్ అయితే a అయిన b : = a + 8, b + 7 మరియు n విలువ

   (అ) m = 8, n = 7 (ఆ) m = 7, n = -8 (ఇ) m = 7, n = 8 (ఈ) m = 8, n = -7

4. అయిన క్రింది వాటిలో ఏది అసైన్‌మెంట్ యొక్క ఇన్‌వేరియంట్ కాదు.

   m, n := m+2, n+3

   (అ) m mod 2 (ఆ) n mod 3 (ఇ) 3 X m - 2 X n (ఈ) 2 X m - 3 X n

5. ఫిబినాకి సంఖ్యను పునరావృతంగా నిర్వచించబడితే

$$\mathbf{F(n)} = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F(n-1) + F(n-2) & \text{otherwise} \end{cases}$$

   ను అంచనావేయండి F(4), F() ఎన్ని సార్లు వర్తించబడుతుంది?

   (అ) 3 (ఆ) 4 (ఇ) 8 (ఈ) 9

6. ఈ పునరావృత నిర్వచనాన్ని ఉపయోగించడం ద్వారా

$$\mathbf{a}^{\mathbf{n}} = \begin{cases} 1 & \text{if } n = 0 \\ a \times a^{n-1} & \text{otherwise} \end{cases}$$

   a^{10}? ను లెక్కించడానికి ఎన్ని గుణాంకాలు అవసరం?

   (అ) 11 (ఆ) 10 (ఇ) 9 (ఈ) 8

విభాగం - ఆ

లఘు ప్రశ్నలకు సమాధానములిమ్ము

1. ఇన్‌వేరియంట్ అంటే ఏమిటి?
2. లూప్ ఇన్‌వేరియంట్‌ను నిర్వచించండి.
3. లూప్ పరిస్థితిని పరీక్షించడం ద్వారా లూప్ ఇన్‌వేరియంట్ ప్రభావితం అగునా? ఎందుకు?

11th Computer Science_Chapter-8 Telugu.indd 117 30-03-2023 18:42:12

4. లూప్ ఇన్‌వేరియంట్ లూప్ కండీషన్ మరియు ఇన్‌పుట్ అవుట్‌పుట్ మధ్య గల సంబంధము ఏమిటి?

5. పునరావృత సమస్య పరిష్కారం అంటే ఏమిటి?

6. పునరావృతంగా సహజ సంఖ్య యొక్క ఫాక్టోరియల్‌ను నిర్వచించండి

విభాగం - ఇ

క్రింది సంక్షిప్త ప్రశ్నలకు సమాధానములిమ్ము

1. ఒక టేబుల్ పైన 7 టంబ్లర్‌లు ఉన్నాయి. అన్ని తల క్రిందులుగా బోర్లించబడి ఉన్నాయి. ఒక కదలికలో ఏదైన రెండు టంబ్లర్‌లను ఒకేసారి త్రిప్పడానికి మీకు అనుమతి ఉంది. టంబ్లర్‌లు అన్ని కుడి వైపు ఉన్నప్పుడు పరిస్థితిని చేరుకోవడం సాధ్యమేనా? (సూచన: తలక్రిందులుగా దొర్లే టంబ్లర్‌ల సంఖ్య యొక్క ఇన్‌వేరియంట్ సమానత్వము)

2. నాకౌట్ టోర్నమెంట్ అనేది ఆటల శ్రేణి ప్రతి ఆటలో ఇద్దరు ఆటగాళ్ళు పోటీపడుతారు. ఓడిపోయిన వ్యక్తి పాల్గొనడు (అనగా ఇకపై ఆడడు) విజేత కొనసాగుతాడు మిగిలిన ఆటగాళ్ళు అందరిని నాకౌట్ చేసిన తరువాత మిగిలిపోయిన ఆటగాడు టోర్నమెంట్ విజేత. ఒక టోర్నమెంట్‌లో 1234 ఆటగాళ్ళు ఉన్నారని అనుకుందాం. టోర్నమెంట్ విజేత నిర్ణయించడానికి ముందు ఎన్ని ఆటలు ఆడతారు.

3. రాజు విక్రమాదిత్యకు రెండు మాయాజాల (magic) కత్తులు ఉన్నాయి. ఒకదానితో అతను ఒక డ్రాగన్ యొక్క 19 తలలను కత్తిరించగలడు. కానీ ఆ తరువాత డ్రాగన్‌కు 13 తలలు పెరుగుతంది ఇతర కత్తితో అతను 7 తలలను కత్తిరించగలడం కాని 22 తలలు పెరుగుతాయి. అన్ని తలలు కత్తిరించిన యెడల డ్రాగన్ చనిపోతుంది. డ్రాగన్ వాస్తవానికి 1000 తలలను కలిగి ఉంటే అది ఎప్పుడైన చనిపోగలదా? (సూచన : హెడ్ mod సంఖ్య 3 ఒక ఇన్‌వేరియంట్)

విభాగం - ఈ

క్రింది ప్రశ్నలను విపులీకరించుము

1. సాధారణ రంగుతో 8×8 చెస్ బోర్డ్‌ను ఊహించుకోండి. రీకలరింగ్ ఆపరేషన్ వరుస లేదా కాలం యొక్క అన్ని చతురస్రాల రంగును మారుస్తుంది. మీరు తిరిగి రంగువేయవచ్చు. కేవలం ఒక నల్ల చతురస్రాన్ని సాధించడమే లక్ష్యం మీరు లక్ష్యాన్ని సాధించలేరని చూపించండి. (సూచన: అడ్డువరుస లేదా నిలువ వరుస ఉంటే Seen బ్లాక్ b లను కలిగి ఉంది ఇది (|8 - b) - b|) ద్వారా మారుతుంది.

2. Power ని కూడా పునరావృతంగా నిర్వచించవచ్చు

$$a^n = \begin{cases} 1 & \text{if } n = 0 \\ a \times a^{n-1} & \text{if } n \text{ is odd} \\ a^{n/2} \times a^{n/2} & \text{if } n \text{ is even} \end{cases}$$

ఈ నిర్వచనాన్ని ఉపయోగించి పునరావృత అల్గారిథమ్‌ను నిర్మించండి. a^{10} ను లెక్కించడానికి ఎన్ని గణాంకాలు అవసరం?

3. ఒకే చదరపుతో కప్పబడిన బోర్డ్ $2^n \times 2^n$ చతురస్రాల బోర్డ్ దీనిలో ఒక చదరము ఒకే చదరపు కొలతలతో కప్పబడి ఉంటుంది. ఈ బోర్డ్‌ను అతి వ్యాప్తి లేకుండా ట్రిమినోస్‌లతో కప్పడం సాధ్యమని చూపండి.

11th Computer Science_Chapter-8 Telugu.indd 118 30-03-2023 18:42:12

C++ పరిచయం

9.1 పరిచయం (Introduction)

C++ అనునది అత్యంత ప్రజాదరణ పొందిన ప్రోగ్రామింగ్ భాషలలో ఒకటి, ఇది విధానపరమైన మరియు లక్ష్యాపరమైన నమూనాలకు మద్ధతును ఇస్తుంది. కావున C++ ను సంకరభాష (హైబ్రిడ్ లాంగ్వేజ్) అందురు. C++ అనునది దాని పూర్వపు C భాష యొక్క విస్తరణ అగును. జర్నెస్ట్రాస్టప్ తన ఈ నవీన భాషకు "C విత్ క్లాసెస్" అని పేరు పెట్టెను. తర్వాత రిక్ మస్కిట్టి భాషకు C++ అనే పేరు పెట్టెను. ఇక్కడ ++ అనునది C భాష యొక్క వృద్ధిని సూచించును.

అభ్యసన లక్ష్యాలు

ఈ అధ్యాయం చదివిన తర్వాత విద్యార్థులు

- C++ ప్రోగ్రామింగ్ లాంగ్వేజ్ (C++ కార్యక్రమ రచక భాష) నందలి ఆధార నిర్మాణ అంశములను అర్థం చేసుకుంటారు.
- సరళ C++ ప్రోగ్రామ్ (నిర్దేశము) లను రూపొందించగలరు.
- డీబగ్ (దోషాలు లేని) C++ ప్రోగ్రామ్‌లను నిర్వహించగలరు.

జర్నెస్ట్రాస్టప్

C++ ప్రోగ్రామింగ్ లాంగ్వేజ్ ఆవిష్కర్త.



డెన్మార్కుకు చెందిన కంప్యూటర్ శాస్త్రవేత్త జర్నెస్ట్రాస్టప్1950, డిసెంబరు 30న జన్మించెను. ఇతడు డెన్మార్క్‌నందలి ఆరస్ యూనివర్సిటీలో 1975వ సంవత్సరం గణితశాస్త్రం మరియు కంప్యూటర్ సైన్స్‌లో మాస్టర్ డిగ్రీ పొందెను. ఇంగ్లాండులోని కేంబ్రిడ్జి యూనివర్సిటీలో 1979వ సం॥ కంప్యూటరు సైన్సులో పిహెచ్.డి డిగ్రీ పొందెను.

C++ యొక్క చరిత్ర (History of C++)

1979 వ సంవత్సరములో AT మరియు T బెల్ ప్రయోగశాలలో జర్నెస్ట్రోస్టప్ C++ ను అభివృద్ధి చేసిరి. C++ అనునది C భాష నుండి ఉత్పాదించబడినది మరియు ఇది సిములా Simula, BCPL, Ada, ML, CLU మరియు ALGOL 68 వంటి అనేక భాషలతో ప్రభావితం కాబడినది. 1983 వరకు దీనిని కొత్త "New C" మరియు "C విత్ క్లాసెస్ "C with Classes" గా చెప్పబడినది. తర్వాత ఈ పేరును C++ గా రిక్ మస్కిట్టి మార్చినారు.

C++ నేర్చుకొనుట వలన కలుగు ప్రయోజనాలు

- బహుళ పరికరము మరియు బహుళ అనువర్తన వేదిక ల అభివృద్ధిలో తరచుగా ఎంచుకో గల భాషలో C++ అతి సురక్షితమైనది.
- C++ అనునది ఒక లక్ష్య పరమైన నిర్దేశక భాష (ఆబ్జెక్ట్ ఓరియంటడ్ ప్రోగ్రామింగ్ లాంగ్వేజ్). ఇది విభాగాలు (క్లాసెస్) సూత్రాలు (ఇన్‌హెరిటెన్స్) బహు రూపాంతరత (పాలీ మార్ఫిసమ్) దత్తాంశ సంగ్రహణ (డాటా అబ్ స్ట్రాక్ష మరియు నిబంధన (ఎన్‌కాప్యులేషన్) లను పొందుపరచును.
- C++ తక్షణ సౌలభ్యత కొరకు సేకరించబడిన నిర్దేశాలను పుష్కలంగా కలిగి ఉంటుంది.

11th Computer Science_Chapter-9 Telugu.indd 119 16-03-2023 17:44:17

- C++ నందు సాధ్యము కాని మినహాయింపు నిర్వహణ, సూత్రాలు మరియు అదనపు సూచనలను అనుమతిస్తుంది. కాని, ఇది, C లో ఉండదు.
- C++ శక్తివంతమైన సమర్థవంతమైన మరియు వేగవంతమైన భాష. ఇది విస్తారమయిన అనువర్తనాలను కలిగి ఉన్నది. ఇది GUI అనువర్తనాల నుండి ఆటల కొరకు 3D గ్రాఫిక్స్‌ను ఖచ్చితమైన గణిత ప్రేరణలుగా మార్చును.

9.2 అక్షర సమితి (క్యారక్టర్ సెట్) (Character set)

అక్షర సమితి అనునది C++ నిర్దేశములను వ్రాయుటకు ఉపయోగించు అక్షరాల సముదాయం. ఇక్కడ అక్షరం అనునది కీబోర్డులో లభ్యమగు వర్ణమాలగానో సంఖ్యగానో లేదా ఏదైనా గుర్తు (ప్రత్యేక అక్షరం) గానో ఉండవచ్చు. C++ ఈ క్రింది అక్షరాలను అనుమతిస్తుంది.

Alphabets	A …. Z, a ….. z
Numeric	0 …. 9
Special Characters	+ - * / ~ ! @ # $ % ^& [ ] ( ) { } = >< _ \ | ? . , : ‘ “ ;
White space	Blank space, Horizontal tab (→), Carriage return (↲), Newline, Form feed
Other characters	C++ can process any of the 256 ASCII characters as data.

9.3 లెక్సికల్ యూనిట్స్ (టోకెన్స్) (Lexical Units (Tokens)):

C++ నిర్దేశక వాక్యములు పలు వేర్వేరు చిన్న మూలాకాలగు ఆదేశాలు, చరరాశులు స్థిరరాశులు మరియు ప్రత్యేక క్రియా చిహ్నాలు (ఆపరేటర్లు) మరియు (విరామ చిహ్నాలు) పంక్షువేటర్లు అనబడు మరెన్నో గుర్తుల కూర్పు అగును. ఈ వ్యక్తిగత మూలకములను సంగ్రహంగా లెక్సికల్ యూనిట్‌లు లేదా లెక్సికల్ మూలకములు (ఎలిమెంట్స్) లేదా టోకెన్లు అంటారు. C++ ఈ క్రింది టోకెన్లను కల్గి ఉంటుంది.

- మూల పదములు (కీవర్డ్స్)
- అక్షరాలు (లిటరల్స్)
- విరామ చిహ్నాలు (పంక్చ్యువేటర్స్)
- గుర్తించగల చిహ్నాలు (ఐడెంటిఫైయర్స్)
- ప్రత్యేక క్రియా చిహ్నాలు (ఆపరేటర్స్)

టోకెన్:

నిర్దేశకములోని అతి చిన్న వ్యక్తిగత యూనిట్‌ను టోకెన్ లేదా లెక్సికల్ యూనిట్ అని పిలుస్తారు.

9.3.1 మూల పదములు (కీవర్డ్స్) (Keywords)

మూలపదములు అనునది C++ ఏకీకరణ (కంపైలర్) లలో నిర్దిష్టమైన అర్థమును ఇచ్చు ప్రత్యేకమైన పదములు. అవి: C++ నిర్ధేశ రచనకు అత్యవసర మూలకాలు అగును. C, C++ మరియు జావాలలో వాడు కీలక పదాలు ఒకే విధంగా యుండును.

C++ అనునది అక్షర సున్నిత నిర్దేశక భాష అందువలన అన్నీ మూలపదములు చిన్న అక్షరాలై lowercase ఉండవలెను.

పట్టిక 9.1 C++ మూల పదములు (కీవర్డ్స్)

asm	auto	break	case	catch
char	class	const	continue	default
delete	do	double	else	enum
extern	float	for	friend	goto
if	inline	int	long	new
operator	private	protected	public	register
return	short	signed	sizeof	static
struct	switch	template	this	throw
try	typedef	union	unsigned	virtual
void	volatile	while		

- మార్పులు, చేర్పుల తర్వాత ప్రస్తుతము వాడబడు మూలపదాల జాబితా.

using, namespace, bal, static_cast, const_cast, dynamic_cast, true, false

- డబుల్ అండర్‌స్కోర్ గల ఐడెండిఫైయర్‌లు C++ అమలు చేయు మరియు నిర్ధిష్ట లైబ్రరీలను వినియోగానికి నిశ్చయించబడినది మరియు వీనిని వినియోగదారులు వాడుట తిస్కరించవలెను.

11th Computer Science_Chapter-9 Telugu.indd 120 16-03-2023 17:44:18

9.3.2 ఐడెంటిఫైయర్ (Identifiers)

C++ ప్రోగ్రామ్ యొక్క వివిధ భాగాలకు వినియోగదారు నిర్వహించిన పేర్లు ఐడెంటిఫైయర్‌లు ఇవి చరరాశులు, ఫంక్షన్లు, శ్రేణులు (అరేస్) మరియు తరగతులు మొదలగునవి. ఇవి ప్రోగ్రామ్ యొక్క నిర్మాణ ఆధారాలు. ఐడెంటిఫైయర్‌లకు పేర్లను ఇచ్చుటకు ప్రతి భాషలోనూ, నిర్దిష్ట నియమాలు కలవు.

ఐడెంటిఫైయర్‌లకు పేరు పెట్టుటకు గల నియమాలు:

- ఐడెంటిఫైయర్ యొక్క మొదటి క్యారక్టర్ వర్ణమాల (ఆల్ఫాబెట్) లేదా అండర్‌స్కోర్ (_) అయివుండాలి.
- ఐడెంటిఫైయర్ యొక్క భాగాలుగా వర్ణమాల, సంఖ్యలు మరియు అండర్‌స్కోర్ మాత్రమే అనుమతించబడును. ఇతర క్యారక్టర్‌లు అనుమతించబడవు.
- C++ అక్షర సున్నితం (కేస్ సెన్సిటివ్) కావున ఇది అప్పర్ కేస్ మరియు లోవర్ కేస్ అక్షరాలను వేరువేరుగా పరిగణించును.
- నిశ్చిత పదాలు లేదా మూలపదాలను ఐడెంటిఫైయర్ పేర్లుగా ఉపయోగించలేము.

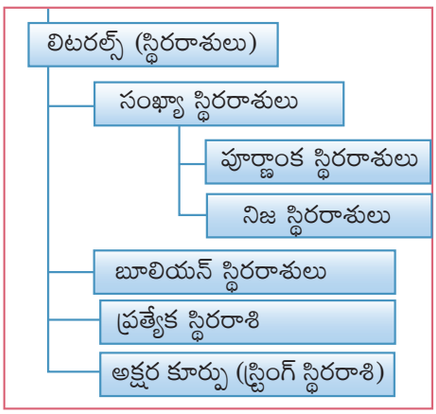
ANSI ప్రమాణాల ప్రకారం C++ దీర్ఘతకు పరిమితి లేదు మరియు ఇందులోగల అన్నీ అక్షరాలు ముఖ్యమైనవే.

ఐడెంటిఫైయర్	వాలిడ్ / ఇన్‌వాలిడ్	ఇన్‌వాలిడ్‌కు కారణం
Num	వాలిడ్	
NUM	వాలిడ్	
_add	వాలిడ్	
total_sales	వాలిడ్	
tamilMark	వాలిడ్	
num-add	ఇన్‌వాలిడ్	ప్రత్యేక క్యారెక్టర్ (-) ఉండును
this	ఇన్‌వాలిడ్	ఇది ఒక "కీవర్డ్'8 కీవర్డ్‌ను ఐడెంటిఫైయర్ పేర్లుగా ఉపయోగించుట లేదు.
2myfile	ఇన్‌వాలిడ్	పేరు ఒక అక్షరంతో లేదా అండర్‌స్కోర్‌తో ప్రారంభంకావాలి.

- చరరాశి పేర్లలో అండర్‌స్కోర్‌ను పేరు యొక్క భాగాలను వ్యత్యాసపరచుటకు వాడవచ్చు. ఉదాహరణ ***total_sales*** అనునది చెల్లుబాటు అగు చరరాశి మరియు ***total sales*** అనునది చెల్లని ఐడెంటిఫైయర్.
- ***tamilMark*** వంటి పెద్ద అక్షర శైలి సంజ్ఞామానము (కేపిటల్ స్టైల్ నొటేషన్) వాడవచ్చు. అనగా రెండవ పదములోని మొదటి అక్షరాన్ని పెద్ద అక్షరంగా (కేపిటల్) వ్రాయవచ్చు.

9.3.3 లిటరల్స్ (స్థిరరాశులు) (Literals (Constants))

నిర్దేశములను (ప్రోగ్రామ్) అమలు చేయునప్పుడు విలువలలో మార్పు కానట్టి సమాచార అంశములను లిటరల్స్ అంటారు. కావున లిటరల్స్‌ను స్థిరరాశులు అంటారు. C++ నందు వివిధ రకాల లిటరల్స్ వుంటాయి.



పటం 9.1 స్థిరరాశులు రకాలు

సంఖ్యా స్థిరరాశులు (Numeric Constants):

పేరులో ఉన్నట్టుగా సంఖ్యా స్థిరరాశులు అనునవి సంఖ్యలు. వీటిని స్థిరరాశులుగా ఉపయోగిస్తారు. సంఖ్యా స్థిరరాశులను మరలా ఈ క్రింది విధంగా వర్గీకరించవచ్చును.

1. పూర్ణాంక స్థిరరాశులు (లేదా) నిర్దిష్ట స్థిరరాశులు.
2. నిజ స్థిరరాశులు (లేదా) చల స్థిరరాశులు.

(1) పూర్ణాంక స్థిరరాశులు (లేదా) నిర్దిష్ట స్థిరరాశులు:

పూర్ణాంకములు అనునది భిన్నములు లేని పూర్ణ సంఖ్యలు అగును. ఒక పూర్ణాంక స్థిరరాశి అనునది కనీసం దశాంశ స్థానము లేని ఒక సంఖ్యగా ఉండవలెను. ఇది సంజ్ఞగాను

11th Computer Science_Chapter-9 Telugu.indd 121 16-03-2023 17:44:18

లేదా అసంజ్ఞగాను ఉండవచ్చు. సంజ్ఞగా గల పూర్ణాంకము ఋణ సంఖ్యలను కామా మరియు ఖాళీ స్థలములను వాని భాగములుగా అనుమతించదు. C++ నందు (i) దశాంశ (ii) అష్టక (iii) షడ్ దశాంశ పూర్ణాంక స్థిరరాశులు కలవు

(i) దశాంశ (Decimal)

ఒకటి లేదా అంతకన్నా ఎక్కువ సంఖ్యలు (0 .... 9) ఏదైనా ఒక శ్రేణి

వాలిడ్	ఇన్‌వాలిడ్
725	7,500 (కామా అనుమతింపబడదు).
-27	66 5 (ఖాళీస్థలం అనుమతింపబడదు).
4.56	9$ (స్పెషల్ క్యారెక్టర్ అనుమతింపబడదు)

ఒకవేళ నీవు 4.56 ను పూర్ణాంక దశాంశ స్థిరరాశిగా ఇచ్చినట్లయితే కంపైలర్ దీని పూర్ణాంక భాగమును మాత్రమే పరిగణలోనికి తీసుకొని దశాంశ భాగాన్ని పట్టించుకోకుండా వదలివేస్తుంది. అంటే 4 ను మాత్రమే పరిగణించి **56** ను వదలివేస్తుంది.

గమనిక

దశాంశ స్థిరరాశిని భిన్నముతో పాటు ఇచ్చినప్పటికీ కంపైలర్ ఈ భిన్నము నందు గల పూర్ణాంకమును మాత్రమే పరిగణలోనికి తీసుకొని భిన్న భాగమును పట్టించుకోకుండా వదలివేస్తుంది. దీనినే ఇంప్లిసిట్ కన్వర్షన్ అందురు. దీని గురించి తర్వాతి అధ్యాయంలో చర్చించుకుందాం.

(ii) అష్టక (Octal)

ఒకటి లేదా అంతకన్నా ఎక్కువ అష్టక విలువలు (0 .... 7) ను కలిగి ఉండి, 0 తో ప్రారంభమగు వరుసలను అష్టక స్థిరరాశులుగా పరిగణించవచ్చు.

వాలిడ్	ఇన్‌వాలిడ్
012	05,600 (కామా అనుమతింపబడదు)
-027	04.56 (దశాంశబిందువు అనుమతింపబడదు)**
+0231	0158 (అష్టక వ్యవస్థలో 8 ప్రవేశార్హత లేని సంఖ్య)

గమనిక

0 సున్నతో ప్రారంభమగు భిన్నమును ఉపయోగించునప్పుడు C++ దీనిని పూర్ణాంకముగా పరిగణిస్తుంది కాని అష్టకముగా కాదు

(iii) షడ్ దశాంశ (Hexadecimal)

ఒకటి లేదా అంతకన్నా ఎక్కువ షడ్ దశాంశ విలువలను (0 .... 9, A .... F) ను కలిగి 0x లేదా 0Xis లతో ప్రారంభమయ్యే వరుసు షడ్ దశాంశ స్థిరరాశిగా పరిగణించబడును.

వాలిడ్	ఇన్‌వాలిడ్
0x123	0x1,A5 (కామా అనుమతింపబడదు)
0X568	0x.14E (ఈ విధమైన దశాంశ బిందువు అనుమతింపబడదు)

ఏదైనా స్థిరరాశికి L లేదా l మరియు U లేదా u లను పరపదములుగా చేర్చినప్పుడు వరుసగా ఇవి దీర్ఘమయిన లేదా సంజ్ఞలేని స్థిరరాశులుగా వ్రాయబడును.

(2) నిజ స్థిరరాశులు లేదా చల స్థిరరాశులు

భిన్నము భాగముగా గల సంజ్ఞా స్థిరరాశియే నిజ లేదా చల స్థిరరాశి అగును. ఈ స్థిరరాశులను భిన్న రూపంలోనూ లేదా ఘాత రూపంలోనూ వ్రాయవచ్చును.

నిజ స్థిరరాశి యొక్క భిన్నరూపం అనునది సంజ్ఞ గల లేదా సంజ్ఞలేని సంఖ్యల మధ్య దశాంశ స్థానముతో కూడిన వనరులను పొందుపరచును. దశాంశ స్థానమునకు ముందు లేక వెనుకన కనీసం ఒక సంఖ్య అయినా ఉండవలెను. ఇది + లేదా - లను ఉపసర్గగా prefixed ను కల్గి ఉండును. ఏ సంకేతము అనగా + లేదా - జతకాని నిజ స్థిరరాశిని ధనాత్మకముగా పరిగణించబడుతుంది.

నిజ స్థిరరాశి యొక్క ఘాతరూపము రెండు భాగములను కలిగి ఉండును. **మాంటిసా** మరియు ఘాతము (ఎక్స్‌పోనెన్ట్) మాంటిసా అనునది పూర్ణాంకముగానో నిజ స్థిరరాశిగానో ఉండవలెను. మాంటిసా అక్షరం E లేదా e ముగిస్తుంది. మరియు ఘాతము ఖచ్చితంగా పూర్ణాంకముగానో ఉండవలెను.

ఉదాహరణకు, 58000000.00 ను 0.58×10^8 లేదా 0.58E8 గ్రా వ్రాయవచ్చును.

11th Computer Science_Chapter-9 Telugu.indd 122 16-03-2023 17:44:18

మాంటిసా E కు ముందు	ఘాతము E తర్వాత
0.58	8

ఉదాహరణ:

5.864 E1 ➔ 5.864×10^{1} ➔ 58.64
5864 E-2 ➔ 5864×10^{-2} ➔ 58.64
0.5864 E2 ➔ 0.5864×10^{2} ➔ 58.64

బూలియన్ లిటరల్స్ (Boolean Literals)

బూలియన్ విలువలు (సరి లేదా తప్పు) లలో ఏదో ఒకదానిని సూచించడానికి బూలియన్ లిటరల్స్‌ను ఉపయోగిస్తారు. అంత:గా సరి True విలువ 1 మరియు తప్పు false విలువ 0.

అక్షర స్థిరాంకం (క్యారక్టర్ కాన్‌స్టెంట్) (Character constant)

C++ నందు ఏదైనా చెల్లుబాటు కాగల ఏక అక్షరాన్ని అనుకరణ చిహ్నం ‘ ’ తో జతచేయబడి ఉంటుంది. చెల్లుబాటు అయ్యే అక్షర స్థిరరాశులు ‘A’, ‘2’, ‘$’ చెల్లుబాటుకాని అక్షర స్థిరరాశి “A”

ఏక అక్షర స్థిరరాశి యొక్క విలువకు సమానమగు ASCII విలువ కలదు. ఉదాహరణకు ‘A’ యొక్క విలువ 65.

ముద్రణ కాని వరుసలు లేదా నాన్ గ్రాఫిక్ క్యారక్టర్:

C++ అక్షర స్థిరరాశిగా ప్రకటించబడి ముద్రణ కాలేని కొన్ని అక్షరాలను అనుమతిస్తుంది. ఈ ముద్రణకాని అక్షరాలను నాన్ గ్రాఫిక్ క్యారక్టర్ అని కూడా అంటారు. C++ నిర్దేశాలను అమలు చేసేటప్పుడు ప్రత్యక్షంగా కీబోర్డ్ ద్వారా టైప్ (ముద్రించలేని) చేయలేని అక్షరాలే ముద్రణకాని అక్షరాలగును. **ఉదాహరణకు :** బ్యాక్ స్పేస్, ట్యాబ్స్ మొదలగునవి. ఇలాంటి ముద్రింపలేని అక్షరాలను ఎస్కేప్ వరుసలను ఉపయోగించి సూచించవచ్చును. ఒకటి లేదా రెండు అక్షరాలతో ముగియు బ్యాక్ స్లాష్‌తో ఈ ఎస్కేప్ వరుసలు సూచించబడతాయి.

పట్టిక 9.2 ఎస్కేప్ వరుసలు

ఎస్కేప్ వరుసలు	నాన్ గ్రాఫిక్ క్యారక్టర్
\a	Audible or alert bell
\b	Backspace
\f	Form feed
\n	Newline or linefeed
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
****	Backslash
\’	Single quote
\”	Double quote
\?	Question Mark
\On	Octal number
\xHn	Hexadecimal number
\0	Null

ఎస్కేప్ వరుస రెండు అక్షరాలను కలిగియున్నప్పటికీ single quotes ఏక అనుకరణ చిహ్నాలతోనే జోడించబడతాయి. ఎందుకనగా C++ ఈ ఎస్కేప్ వరుసలను అక్షర స్థిరరాశిగానే పరిగణిస్తుంది మరియు దీనికి ASCII నందు ఒక బైట్ byte మాత్రమే కేటాయించబడుతుంది.

మీకు తెలుసా?

ASCII (అమెరికన్ సాండర్డ్ కోడ్ ఫర్ ఇన్ఫర్మేషన్ ఇంటర్ చేంజ్) మొట్టమొదటగా 1963 లో అమెరికన్ స్టాండర్డ్స్ అసోసియేషన్ (ASA) యొక్క భాగమైన X3 కమిటిచే అభివృద్ధి చేయబడి ప్రచురించబడినది.

స్వయం మూల్యాంకనము:

1. లిటెరల్స్ అనగానేమి? C++ నందు లభ్యమగు పూర్ణాంక లిటెరల్స్ యొక్క రకములు ఏవి?
2. ఈ క్రింది స్థిరరాశులు ఏ రకానికి చెందినవి?
   i) 26 ii) 015 iii) 0xF iv) 014.9
3. C++ నందు అక్షర స్థిరరాశి అంటే ఏమిటి?
4. C++ నందు నాన్ గ్రాఫిక్ క్యారక్టర్‌లను ఏ విధంగా సూచింతురు?
5. ఈ క్రింది నిజ స్థిరరాశులను ఘాత రూపములోకి మార్చుము.
   i) 32.179 ii) 8.124 iii) 0.00007
6. ఈ క్రింది నిజ స్థిరరాశులను భిన్నరూపములోనికి మార్చుము:
   i) 0.23E4 ii) 0.517E-3 iii) 0.5E-5
7. స్ట్రింగ్ నందు శూన్య (Null) (\0) అక్షరము యొక్క ప్రాముఖ్యత ఏమిటి?

11th Computer Science_Chapter-9 Telugu.indd 123 16-03-2023 17:44:18

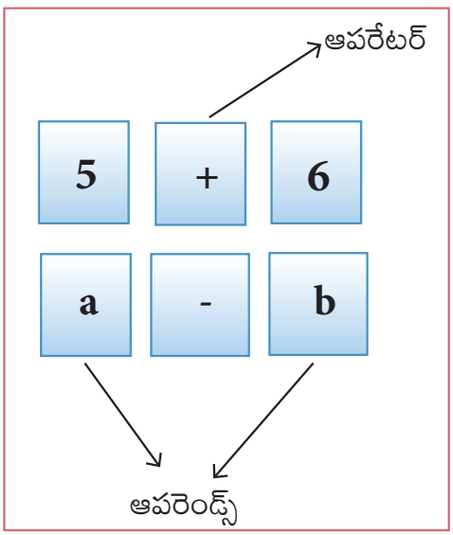
స్ట్రింగ్ లిటెరల్స్ (String Literals)

ద్వి అనుకరణ చిహ్నాలు (డబుల్ కోట్స్) తో జోడించబడిన అక్షరాల యొక్క వరుసను స్ట్రింగ్ లిటెరల్స్ అని అంటారు. విధిగా స్ట్రింగ్ లిటెరల్స్ తనంతటతానే ప్రత్యేక అక్షరమైన బ్లాక్ స్లాష్ జిరోతో ముగియును. అందువలన "welcome" అను స్ట్రింగ్ నిజంగా మెమరీలో "welcome\0" గానే సూచించబడుతుంది మరియు ఈ స్ట్రింగ్ యొక్క పరిమాణం 7 కాదు. కాని 8 అక్షరాలు (అనగా చివరన జోడించిన \0 అక్షరముతో కలిపి) చెల్లుబాటు అయ్యే స్ట్రింగ్ లిటెరల్స్ "A", "Welcome"

చెల్లుబాటు కాని స్ట్రింగ్ లిటెరల్స్ 'Welcome', "1234"

9.3.4 ఆపరేటర్స్ (Operators)

కొన్ని గణిత లేక తార్కిక ప్రక్రియలను ఉపయోగించు చిహ్నాలను "**ఆపరేటర్స్**" అందురు. ఆపరేటర్లుగా పనిచేయు సమాచార అంశాలను లేదా విలువలను "**ఆపరెండ్స్**" అంటారు.



C++ లో ఆపరెండ్స్ సంఖ్యను ఆధారంగా ఆపరేటర్లు ఈ క్రింది విధముగా వర్గీకరించబడినది.

(i) **యూనరీ ఆపరేటర్స్** - ఒకే ఒక ఆపరెండ్ కొరకు

(ii) **ద్వి ఆపరేటర్స్ (Binary)** - రెండు ఆపరెండ్ కొరకు

(iii) **త్రి (టెర్నరీ) ఆపరేటర్స్** - మూడు ఆపరెండ్ కొరకు

C++ బైనరీ ఆపరేటర్స్‌ను ఈ క్రింది విధముగా వర్గీకరించబడినది:

(1) అంకగణిత ఆపరేటర్స్

(2) సంబంధిత ఆపరేటర్స్

(3) తార్కిక ఆపరేటర్స్

(4) అసైన్‌మెంట్ ఆపరేటర్స్

(5) నిబంధనల ఆపరేటర్స్

(1) అంకగణిత ఆపరేటర్స్ (Arithmetic Operators)

కూడికలు, తీసివేతలు, గుణకారాలు, భాగాహారాలు మొదలగు సరళమైన అంకగణిత ప్రక్రియలను ఈ అంకగణిత ఆపరేటర్లు చేయును.

ఆపరేటర్	ఆపరేషన్	ఉదాహరణ
+	కూడికలు	10 + 5 = 15
-	తీసివేయుట	10 – 5 = 5
*	గుణించుట	10 * 5 = 50
/	భాగించుట	10 / 5 = 2 (Quotient of the division)
%	Modulus (To find the remind-er of a division)	10 % 3 = 1(Remainder of the division)

- పై పట్టికలో చూపబడిన అంకగణిత ఆపరేటర్లని బైనరీ ఆపరేటర్లు అంటారు. ఈ ప్రక్రియనందు కనీసం రెండు ఆపరెండ్స్ అవసరమగును.

ఇంక్రిమెంట్ మరియు డిక్రిమెంట్ ఆపరేటర్స్ (Increment and Decrement Operators)

++ (ప్లస్ ప్లస్) ఇంక్రిమెంట్ ఆపరేటర్

-- (మైనస్ మైనస్) డిక్రిమెంట్ ఆపరేటర్

ఇంక్రిమెంట్ లేదా డిక్రిమెంట్ ఆపరేటర్ ఒకే ఒక ఆంపరెండ్‌పై పనిచేసి కొత్త విలువను ఇచ్చును. కావున ఇవి యూనరీ ఆపరేటర్స్. ఇంక్రిమెంట్ ఆపరేటర్ దాని ఆపరెండ్‌కు ఒకటిని కలుపును. మరియు డిక్రిమెంట్ ఆపరేటర్ దాని ఆపరెండ్ నుండి ఒకటిని తీసివేయును. ఉదాహరణకు,

- **x++** లేదా **++ x** అగును **x = x+1;**

  ఇది ప్రస్తుత x యొక్క విలువకు 1 ని కలుపును

- **x --** లేదా **-- x** అగును **x = x–1;**

11th Computer Science_Chapter-9 Telugu.indd 124 16-03-2023 17:44:18

ఇది ప్రస్తుత x యొక్క విలువ నుండి 1 ని తీసివేయును.

ఈ ++ లేదా -- ఆపరేటర్స్‌ను చరరాశికి పూర్వపదముగానో (ప్రిఫిక్స్) లేదా పరపదముగానో (సఫిక్స్) ఉంచవచ్చు. ఈ పూర్వపద రూపం (వర్షన్)లో C++ ఆపరెండ్ ఉపయోగానికి మునుపే ఇంక్రిమెంట్ / డిక్రిమెంట్ చేస్తుంది.

(2) సంబంధిత ఆపరేటర్స్ (Relational Operators)

Operator	Operation	Example
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b
==	Equal to	a == b
!=	Not equal	a != b

ఆపరెండ్ల మధ్యగల సంబంధమును కనుగొనుటకు ఈ సంబంధిత ఆపరేటర్స్ ఉపయోగపడును. ఏదైనా రెండు ఆపరెండ్ల మధ్య సంబంధిత ఆపరేటర్స్‌ను ప్రయోగించినపుడు ఫలితంగా బూలియన్ విలువగా ఉండును. అంటే 1 లేదా 0 వరుసగా సరి లేదా తప్పును సూచించును. C++ ఆరు సంబంధిత ఆపరేటర్స్‌ను ఇచ్చును. అవి,

- పై ఉదాహరణలలో ఆపరెండ్ *a* ను ఆపరెండ్ *b* తో పోల్చినపుడు వాని మధ్య సంబంధం ఆధారంగా ఫలితం 1 లేదా 0 గా వచ్చును. అంటే 1 అనునది సరి 1 అనునది తప్పు.
- అన్నీ ఆరు సంబంధిత ఆపరేటర్స్ బైనరీ ఆపరేటర్స్ అగును.

(3) తార్కిక (లాజికల్) ఆపరేటర్స్ (Logical Operators)

తార్కిక లేదా సంబంధిత సమాసాలను మూల్యాంకనం చేయడానికి ఈ లాజికల్ ఆపరేటర్స్‌లను ఉపయోగిస్తారు. తమంతటతామే పనిచేయు లాజికల్ ఆపరేటర్స్‌ను తార్కిక సమాసాలు అని పిలుస్తారు. C++ మూడు లాజికల్ ఆపరేటర్స్‌ను ఇచ్చును.

పట్టిక 9.3 లాజికల్ ఆపరేటర్స్

ఆపరేటర్	ప్రక్రియ	వివరణ
&&	AND	రెండు వేరు వేరు సంబంధిత సమాసాలను ఒకదానికొకటి కలుపునది లాజికల్ అండ్. ఒక వేళ రెండు సమాసాలు సరి అయితే ఇది 1 సరిని ఇచ్చును, అలాకాకపోతే 0 తప్పు ఇచ్చును.
||	OR	రెండు వేరే వేరు సంబంధిత సమాసాలను ఒకదానికొకటి కలుపునది లాజికల్ అండ్. ఒక వేళ ఏదైనా ఒక సమాసము సరి అయితే ఇది 1 సరిని ఇచ్చును. ఒకవేళ రెండు సమాసాలు 0 అయితే ఇది తప్పును ఇచ్చును.
!	NOT	ఒకే ఒక సమాసము 1 ఆపరెండ్‌పై NOT పని చేయును. ఇది సాధారణముగా సరియైన విలువను వ్యతిరేకం అనగా ఒకవేళ ఒక ఆపరెండ్ 1 సమాసము 1 సరి అయితే ఈ ఆపరేటర్ దీనికి 0 తప్పు బదులుగా ఇచ్చును. ఒకవేళ ఒక ఆపరెండ్ 1 సమాసము 0 తప్పు అయితే ఈ ఆపరేటర్ దీనికి 1 సరి బదులుగా ఇచ్చును.

- AND, OR రెండు బైనరీ ఆపరేటర్స్ కానీ NOT ఒక యూనరీ ఆపరేటర్.

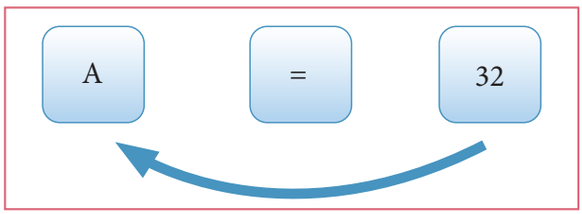
ఉదాహరణ: a = 5, b = 6, c = 7

11th Computer Science_Chapter-9 Telugu.indd 125 16-03-2023 17:44:19

Expression	Result
(a<b) && (b<c)	1 (True)
(a>b) && (b<c)	0 (False)
(a<b) || (b>c)	1 (True)
!(a>b)	1 (True)

(4) అసైన్‌మెంట్ ఆపరేటర్ (Assignment Operator)

ఒక అసైన్‌మెంట్ స్టేట్‌మెంట్ యొక్క ఎడమచేతి వైపు గల చరరాశికి విలువను ఇవ్వడానికి అసైన్‌మెంట్ ఆపరేటర్ ఉపయోగించబడుతుంది. అన్నీ కంప్యూటర్ నిర్దేశక (ప్రోగ్రామింగ్) భాషలో అసైన్‌మెట్ ఆపరేటర్ సాధారణంగా వాడబడును. ఈ ఆపరేటర్ కుడివైపు గల ఆపరేటర్ విలువను ఎడమవైపు గల చరరాశికి కాపీ చేయును. ఇది కూడా బైనరీ ఆపరేటర్.



C++ వివిధ రకాల అసైన్‌మెంట్ ఆపరేటర్‌ను ఉపయోగిస్తుంది. ఇవి సార్ట్‌హేండ్ అసైన్‌మెంట్ ఆపరేటర్లుగా పిలువబడును.

Operator	Name of Operator	Example

+=	Addition Assignment	a = 10;
		c = a += 5;
		(ie, a = a + 5)
		c = 15

-=	Subtraction Assignment	a = 10;
		c = a -= 5;
		(ie. a = a – 5)
		c = 5

*=	Multiplication Assignment	a = 10;
		c = a *= 5;
		(ie. a = a * 5)
		c = 50

/=	Division Assignment	a = 10;
		c = a /= 5;
		(ie. a = a / 5)
		c = 2

%=	Modulus Assignment	a = 10;
		c = a %= 5;
		(ie. a = a % 5)
		c = 0

= మరియు == ఆపరేటర్ల మధ్య గల భేదములను చర్చించుము.

(5) నిబంధిత ఆపరేటర్ (Conditional Operator)

C++ నందు ఒకే ఒక నిబంధిత ఆపరేటర్ కలదు. ?: ఒక నిబంధిత ఆపరేటర్. దీనిని టెర్నరీ ఆపరేటర్ అని కూడా చెప్పవచ్చు. if … else వంటి కంట్రోల్ స్టేట్‌మెంట్‌లను బదులుగా ఈ ఆపరేటర్‌ను ఉపయోగించవచ్చు. if … else గల ఆపరేటర్ గురించిన అధిక సమాచారాన్ని రాబోవు అధ్యయనాలలో నేర్చుకొందాం.

ఇతర ఆపరేటర్స్ (Other Operators):

కామా ఆపరేటర్	C++ నందు కామా (,) ఆపరేటర్‌ను వివిధ సమాసాలను కూర్చుటకు ఉపయోగిస్తారు. కామాతో వేరుచేయబడిన సమాసాల సమూహాన్ని ఎడమ నుండి కుడికి మూల్యాంకనం చేస్తారు.
సైజ్ ఆఫ్	ఇది కంపైల్ టైమ్ ఆపరేటర్ అని పిలువబడుతుంది. ఇది చరరాశి యొక్క పరిమాణాన్ని bytes లోకి మార్చును.
పాయింటర్	* Pointer to a variable & Address of
Component selection	. Direct component selector -> Indirect component selector
Class member operators	:: Scope access / resolution .* Dereference ->* Dereference pointer to class member

11th Computer Science_Chapter-9 Telugu.indd 126 16-03-2023 17:44:19

ఆపరేటర్స్ యొక్క ప్రాముఖ్యత:

ప్రెసిడెన్స్ అనుసరించిన క్రమములోనే ఆపరేటర్స్ అమలుచేయబడుతుంది. ఆపరెండ్స్ మరియు ఆపరేటర్లను మూల్యాంకనం చేయుట కొరకు నిర్ధిష్ట తార్కిక సమూహాలుగా చేయబడతాయి. ఈ తార్కిక సమూహాలను అసోసియేషన్ (సంయోగం) అందురు.

ప్రెసిడెన్స్ యొక్క క్రమము:

( ) [ ]	Operators within parenthesis are performed first	Higher
++, --	Postfix increment / decrement	
++, --	Prefix increment / decrement	
*, /, %	Multiplication, Division, Modulus	
+, -	Addition, Subtraction	
<, <=, >, >=	Less than, Less than or equal to, Greater than, Greater than or equal to	
==, !=	Equal to, Not equal to	
&&	Logical AND	
||	Logical OR	
?:	Conditional Operator	
=	Simple Assignment	
+=, -=, *=, /=	Shorthand operators	
,	Comma operator	Lower

9.3.5 పంక్షుయేటర్స్ (Punctuators)

C++ నిర్దేశకములను వ్రాయునప్పుడు పరిమితులు లేనిది delimiters గా వాడు చిహ్నాలే పంక్షుయేటర్స్. ఇవి "**సెపరేటర్స్**" గానూ పిలువబడును. C++ నందు ఈ క్రింది పంక్షుయేటర్స్‌ను ఉపయోగిస్తారు. ఈ చిహ్నాలు C మరియు జావా చిహ్నాలను పోలి ఉంటాయి.

సెపరేటర్	వివరణ	ఉదాహరణ

కర్లీ బ్రెసెస్ { }	కర్లీ బ్రెసెస్ యొక్క ప్రారంభము మరియు ముగింపు కోడ్‌ల యొక్క సమూహము ప్రారంభము మరియు ముగింపులను సూచించును. కోడ్‌ల యొక్కసమూహము ఒకటి కన్నా ఎక్కువ అమలు చేయగల స్టేట్‌మెంట్‌లను కలిగియుండును. ఈ స్టేట్‌మెంట్‌లన్నింటిని కలిపి కాంపౌండ్ స్టేట్‌మెంట్ అంటారు.	int main ( )
		{
		int x=10, y=20, sum;
		sum = x + y;
		cout << sum;
		}

పారెన్ థెసిస్ ( )	పారెన్‌థెసిస్ ( ) యొక్క ప్రారంభం మరియు ముగింపు పంక్షన్ పారామీటర్ ను సూచించును.	clrscr( ); int main ( )
స్క్వాయర్ బ్రాకెట్స్ [ ]	ఇది ఏక మరియు బహు మితియ (మల్టీ డైమెంక్షనల్) ఎర్రస్ లను సూచించును.	int num[5]; char name[50];
కామా ,	కామాను సమాసములో సెపరేటర్స్‌గా ఉపయోగిస్తారు.	int x=10, y=20, sum;

11th Computer Science_Chapter-9 Telugu.indd 127 16-03-2023 17:44:19

సెమికొలన్ ;	C++ నందు అమలు చేయుగల ప్రతి స్టేట్‌మెంట్‌ను తప్పసి సరి సెమికోలన్‌తో ముగించాలి.	int main ( )
		{
		int x=10, y=20, sum;
		sum = x + y;
		cout << sum;
		}

కొలన్ :	ఇది స్టేట్‌మెంట్‌ను గుర్తించుటకు ఉపయోగిస్తారు.	private:

కమంట్స్
//
/* */
	// తో ప్రారంభమగు ఏ స్టేట్‌మెంట్‌నైనా కమెంట్‌గా పరిగణించవచ్చును. సాధారణంగా ఈ కమెంట్‌లు కంపైలర్‌చే విస్మరించబడతాయి. అనగా // తో ప్రారంభమగు ఏ స్టేట్‌మెంట్‌నైనా కంపైలర్ అమలు కావుటను అనుమతించదు. **// Single line comment**
	/* ……. */ Multiline comment
		/* This is written by me to learn CPP */
		int main ( )
		{
		int x=10, y=20, sum;
		// to sum x and y
		sum = x + y;
		cout << sum;
		}

మీకు తెలుసా? C++ నందు ఒకటి లేదా రెండు ఆపరేటర్స్‌ను వేరువేరు ప్రదేశాలలో వేరు వేరు అర్థాలతో ఉపయోగించవచ్చు. ఉదాహరణకు (*) ను చరరాశి గుణకారమునకు మరియు పాయింటర్ (గుర్తించుట) కు ఉపయోగిస్తారు..

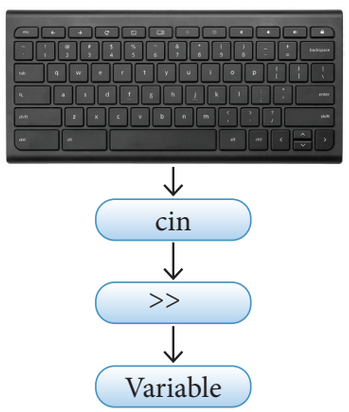
స్వయం మూల్యాంకనము

1. ఆపరేటర్స్ యొక్క ఉపయోగమేమి?
2. బైనరీ ఆపరేటర్స్ ఏవి? అంకగణిత బైనరీ ఆపరేటర్స్‌కు ఉదాహరణలిమ్ము.
3. మాడ్యులస్ ఆపరేటర్స్ % ఏమి చేయును?
4. 8.5% 2 యొక్క ఫలితమేమి?
5. i = 8, j = 10, k = 8, గా ఇవ్వబడినది. అయితే ఈ క్రింది సమాసాల ఫలితాలు ఏవి?
   (i) i < k (ii) i < j (iii) i > = k
   (iv) i = = j (v) j ! = k
6. ఈ క్రింది సమాసాల మూల్యాంకము యొక్క క్రమములు ఏ విధముగా ఉండును?
   (i) i + 3 >= j - 9 (ii) a +10 < p - 3 + 2 q
7. ఒక వేళ మార్కులు 75 మరియు గ్రిడ్ 'A' ను తార్కిక ఆపరేటర్‌తో పరీక్షించుటకు తగు సమయాన్ని వ్రాయము.

9.4 I/O ఆపరేటర్స్ I/O Operators

9.4.1 ఇన్పుట్ ఆపరేటర్ (Input operator)

C++ ఇన్పుట్ పొందుటకు ఆపరేటర్ >> ను అందిస్తుంది. ఇది కీబోర్డ్ ద్వారా విలువను సంగ్రహించి దాని కుడివైపున వున్న చరరాశికి కేటాయిస్తుంది. కావున దీనిని **"Stream extraction"** లేదా **"get from"** అని అందురు. ఇది ఒక బైనరీ ఆపరేటర్ అనగా దీనికి రెండు ఆపరెండ్‌లు అవసరం. మొదటి ఆపరెండ్ కీబోర్డ్‌ను ఇన్పుట్ సాధనంగా గుర్తించగల ప్రీ డిఫైన్డ్ ఐడెంటిఫైయర్ సిన్ అగును. రెండవ ఆపరెండ్ ఖచ్చితంగా ఒక చరరాశి అగును.



పటము 9.4 cin యొక్క ప్రక్రియ – పనితీరు

ఒకేసారి ఒకటికన్నా ఎక్కువ విలువలను ఒక చరరాశి ఇచ్చునప్పుడు >> ఆపరేటర్‌ను ఉపయోగించాలి దీనినే ఆపరేటర్ యొక్క క్యాస్‌క్యాడింగ్ అందురు.

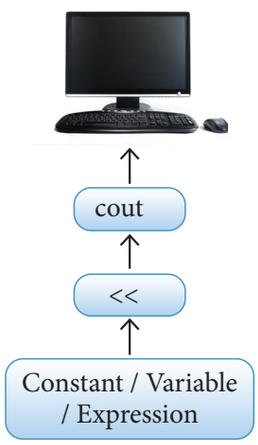
11th Computer Science_Chapter-9 Telugu.indd 128 16-03-2023 17:44:19

ఉదాహరణ:

సిన్ >> **n u m ;**	పూర్వ నిర్వచిత వస్తువు సిన్ ప్రీ డిపైన్డ్ ఆబ్జెక్ట్. కీబోర్డ్‌పై టైప్ చేసిన విలువలను పొందును మరియు దీనిని చరరాశిగా భద్రపరచును **num.**
cin >>x >> y;	రెండు విలువలను పొందుటకు దీనిని ఉపయోగిస్తారు. cin మొదటి విలువను స్వీకరించి తక్షణమే చరరాశి x; కు ఇచ్చును. తర్వాత ఇది ఖాళీ స్థలం (స్పేస్) తర్వాత టైప్ చేయబడిన రెండవ విలువను స్వీకరించి y కు అందించును. స్పేస్ ప్రతి ఇన్పుట్‌నకు మధ్య సెపరేటర్‌గా వాడబడును.

9.4.2 అవుట్‌పుట్ ఆపరేటర్ (Output Operator):

C++ అవుట్‌పుట్ ఆపరేషన్ (ప్రక్రియను సాధించుటకు) << ఆపరేటర్‌ను అందిస్తుంది. ఈ **"Stream insertion"** or **"put to"** ఆపరేర్‌ను "స్ట్రీమ్ ఇన్ సెర్సన్ లేదా << ఆపరేటర్ అని పిలుస్తారు. దీనిని చరరాశి యొక్క కూర్పులు లేదా విలువలను వస్తువు తన కుడివైపు నుండి ఎడమవైపు పంపుటకు ఉపయోగపడుతుంది. << అనునది ఒక బైనరీ ఆపరేటర్ పూర్వ నిర్వచిత ఐడెంటిఫైయర్ కౌట్ C-Out గా ఉచ్చరించబడుతుంది. మొదటి ఆపరెండ్ మానిటర్‌ను ప్రామాణికమైన అవుట్‌పుట్‌గా గుర్తిస్తు & ంది. రెండవ ఆపరెండ్ ఒక స్థిరరాశిగానూ చరరాశిగానూ లేదా సమాసముగానో ఉండవచ్చును.



పటము 9.5 కౌట్ ప్రక్రియ పనితీరు

ఒకే సమయంలో ఒకటి కన్నా ఎక్కువ విలువలను (సిర్థరాశి / చరరాశి / సమాసము) పంపించు ప్రతిసారి << ఆపరేటర్‌ను ఉపయోగించాలి. ఇది ఆపరేటర్ యొక్క క్యాస్‌క్యాడింగ్ అని పిలవబడుతుంది.

ఉదాహరణ:

cout << "Welcome";	ఈ పూర్వ నిర్వచిత ఆబ్జెక్ట్ కౌట్ "Welcome" ఇవ్వబడిన కూర్పును తెరపైకి పంపించును.
cout << "The sum = " << sum;	కౌట్ "The Sum = " ను కూర్పును మొదటగా తెరపైకి పంపును మరియు ఇది చరరాశి యొక్కవిలువను Sum తర్వాత నాన్ గ్రాఫిక్ క్యారక్టర్స్ తప్ప మిగిలిన కూర్పు లేదా క్యారక్టర్ స్థిరరాశులు వంటి వాటిని నిర్ధిష్టమైన ఏక లేదా ద్వి అనుకరణ చిహ్నాలతో పొందుపరచి పంపించును.
cout <<"\n The Area: " <<3.14*r*r;	మొదటిగా కౌట్ \n తప్ప తెరపైకి పంపించు ప్రతి దానిని నిర్ధిష్టమైన ద్వి అనుకరణ చిహ్నాలతోనే పంపిస్తుంది. తర్వాత ఇది సమాసాలన్నీ 3.14*r*r మూల్యాంకనం చేసి ఫలితాన్ని తెరపైకి పంపిస్తుంది. \n కొత్త పంక్తిని ప్రారంభించు ఒక నాన్ గ్రాఫిక్ కారక్టర్
cout << a + b ;	కౌట్ a మరియు b ల మొత్తాన్ని మానిటర్ తెరపై చూపించును.

9.4.3. I/O ఆపరేటర్స్ యొక్క క్యాస్‌క్యాడింగ్ (Cascading of I/O operators)

>> మరియు << వంటి ఇన్పుట్ మరియు అవుట్‌పుట్ ఆపరేటర్లను ఒకే స్టేట్‌మెంట్‌లో పలుమార్లు ఉపయోగించుటను I/O ఆపరేటర్ యొక్క క్యాస్ కాడింగ్ అంటారు.

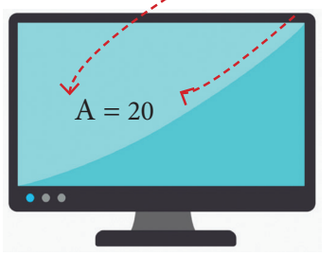
11th Computer Science_Chapter-9 Telugu.indd 129 16-03-2023 17:44:19

క్యాస్ కాడింగ్ కౌట్ **(Cascading cout):**

```
int Num=20;
cout << “A=” << Num;
```

పటము 9.6 నందు క్యాస్‌కాడింగ్ కౌట్ స్టేట్‌మెంట్ పనితీరును అర్థం చేసుకోవచ్చును.

```
cout << "A=" << Num;
```

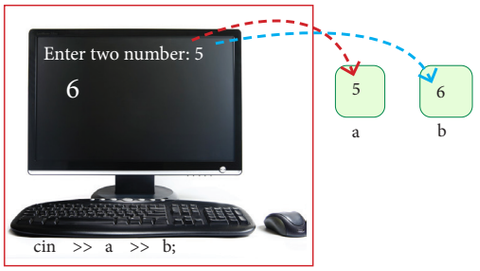


పటం 9.6 క్యాస్‌కాడింగ్ కౌట్

క్యాస్‌క్యాడింగ్ సిన్ – ఉదాహరణ:

```
cout >> “Enter two number: ”;
cin >> a >> b;
```

పటము 9.7 ద్వారా క్యాస్‌క్యాడింగ్ cin స్టేట్‌మెంట్ యొక్క పనితీరును అర్థం చేసుకోవచ్చును.



పటము 9.7 క్యాస్‌క్యాడింగ్ cin

9.5 C++ నిర్దేశకము – నిర్దేశక నమూనా
(Sample program – A first look at C++ program)

“Welcome అను స్ట్రింగ్ అను తెరపై ముద్రించుటకు కావలసిన C++” నిర్దేశాలతో C++ నిర్దేశకమును మొదలు పెడతాం.

Sample Prog 9_5.cpp

```
1  // C++ program to print a string
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      cout << "Welcome to Programming in C++";
7      return 0;
8  }
```

పై నిర్దేశము ఈ క్రింది అవుట్‌పుట్ ఇచ్చును.

C++ ప్రోగ్రామింగ్‌లోనికి స్వాగతం

ప్రతి C++ ప్రోగ్రామ్‌నందు గల అతి సరళ ప్రాథమిక మూలకాలను గురించి తెలుసుకొనుట.

1	// C++ ప్రోగ్రామ్‌నందు స్ట్రింగ్‌ను ముద్రించుట కొరకు

ఇది ఒక వివరణాత్మక (కమెంట్) స్టేట్‌మెంట్‌తో ప్రారంభమగు ఏ స్టేట్‌మెంట్‌నైనను కమెంట్‌గా పరిగణించును. కంపైలర్ ఏ కమెంట్‌ను ప్రోగ్రామ్ యొక్క భాగముగా నిర్వహించదు మరియు దీనిని సాధారణంగా వదలివేస్తుంది. ఒకవేళ కమెంట్స్‌తో పలు పంక్తులు వ్రాయవలసి వస్తే /* ……. */ మనము దీనిని ఉపయోగించవచ్చును.

2	# జతచేయు <iostream>

మామూలుగా అన్నీ C++ ప్రోగ్రామ్స్ ఇన్‌క్లూడ్ స్టేట్‌మెంట్ # (hash / pound) తో ప్రారంభమగును. # చిహ్నము పూర్వ పురోగమనమునకు నిర్దేశకము. దీనికి అర్థము ఈ స్టేట్‌మెంట్స్ కంపైలేషన్ ప్రక్రియకు మునుపే అమలుచేయబడును.

 # include <iostream> స్టేట్‌మెంట్ కంపైలర్ ప్రాసెసర్‌ను ప్రోగ్రామ్‌లో **“iostream”** ను హెడర్ ఫైల్‌గా జతచేయమని చెబుతుంది.

ప్రతి C++ ప్రోగ్రామ్‌లో ఇన్పుట్ / అవుట్‌పుట్ ప్రక్రియలను అమలుచేయడానికి తప్పకుండా హెడర్‌ఫైల్ **iostream** ను జతచేయాలి. సరళంగా చెప్పాలంటే **iostream** హెడర్‌ఫైల్ తన సభ్యులైన cin, cout యొక్క నిర్వచనాలను కలిగియుండును. ఒకవేళ మీ ప్రోగ్రామ్‌నందు **iostream** ని జతచేయక పోయినట్లయితే cin మరియు cout పై ఒక దోష సందేశము ఏర్పడును. కావున ఇది ఎలాంటి ఇన్పుట్‌ను ఇవ్వజాలదు లేదా ఎలాంటి అవుట్‌పుట్ పంపజాలదు.

11th Computer Science_Chapter-9 Telugu.indd 130 16-03-2023 17:44:20

3	నేమ్ స్పేస్ std; ఉపయోగించుట;

నేమ్ స్పేస్ std; పంక్తులలో వాడుట, కంపైలర్‌కు ప్రామాణిక నేమ్ స్పేస్ ఉపయోగించమని చెప్పుటయే. తరగతి, వస్తువు మరియు చరరాశులు కొరకు ఉపయోగించబడే ఐడెంటిఫైయర్‌ను నేమ్ స్పేస్ సేకరించును. భారీ ప్రాజెక్ట్‌లలో ఏర్పడు నామ వివాదాలను నివారించు పద్ధతులను నేమ్ స్పేస్ ఇచ్చును ANSI C++ ప్రామాణిక కమిటిచే ఈ క్రొత్త ఆలోచన పరిచయం చేయబడినది.

4	int main ( )

C++ ఒక ఫంక్షన్‌ల సేకరణ ప్రతి C++ ప్రోగ్రామ్ ఏదైనా ఒక ప్రధాన ఫంక్షన్‌ను కలిగియుండును. అన్నీ C++ ప్రోగ్రామ్‌ల అమలు ప్రధాన ( ) ఫంక్షన్‌తోనే ప్రారంభమగును. అందువలన అమలు కాబడే స్టేట్‌మెంట్ తప్పనిసరిగా ప్రధాన ( ) ఫంక్షన్ లోపలే ఉండును.

```
5  {
6      cout << "Welcome to Programming in C++";
7      return 0;
8  }
```

కర్లీ బ్రెసెస్ (Line number 5 to 8) మధ్య గల స్టేట్‌మెంట్లు అన్నీ అమలుకాగల స్టేట్‌మెంట్స్ నిజానికి దీనిని బ్లాక్ ఆప్ కోడ్ అందురు. 6 వ పంక్తిలో సాధారణంగా కౌట్ cout స్ట్రింగ్ స్థిరరాశి C++" ను తెరపైకి పంపును. మనము ముందుగానే చర్చించుకొన్నట్లు అమలుకాగల ప్రతి స్టేట్‌మెంట్‌ను సెమికొలన్‌తోనే ముగించాలి. 7 వ పంక్తిలో రిటర్న్ అనే మూలపదము ప్రత్యేక ఫంక్షన్‌కు కేటాయించిన విలువను తిరిగి ఇవ్వడానికి ఉపయోగపడుతుంది. ఇక్కడ ఇది 0 ను మెయిన్ ఫంక్షన్ ( ) కు తిరిగి ఇచ్చును.

9.6 C++ ప్రోగ్రామ్ అమలు (Execution of C++ program)

C++ ప్రోగ్రామ్‌ను వ్రాయడానికి మరియు అమలు చేయడానికి నాలుగు ముఖ్యమైన సోపానాలను అనుసరించాలి.

(1) మూలకోడ్‌ను సృష్టించుట:

C++ కంపైలర్ అనుసరించే నిబంధనల ప్రకారం చెల్లుబాటు అయ్యే C++ కోడ్‌ను సృష్టించుట అనునది టైప్ చేయడం మరియు ఎడిట్ చేయడం వంటి వాటిని పొందుపరచును.

(2) మూలకోడ్ విస్తరణతో .cpp భద్రపరచుట:

టైప్ చేసిన తరువాత మూలకోడ్ (సోర్స్ కోడ్) ను విస్తరణ .cpp లో తప్పనిసరిగా భద్రపరచాలి.

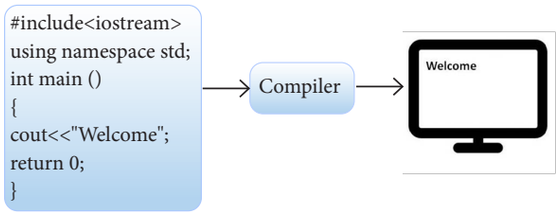
(3) సేకరించుట (కంపైలేషన్)

ప్రోగ్రామ్ రచనలో ఇది ఒక ముఖ్యమైన సోపానం. సేకరణలో కంపైలర్ లైబ్రరీ ఫైళ్ళను మూలకోడ్‌తో అనుసందానం చేస్తుంది. మరియు కోడ్ యొక్క ప్రతి పంక్తిని తనిఖీ చేస్తుంది. ఒకవేళ ఏదైనా తప్పు లేదా దోషము ఉన్నట్లయితే వీనిని దోష సందేశాలుగా బయటకు పంపబడును. ఒకవేళ దోషము లేనట్లయితే ఇది మూలకోడ్‌ను యంత్రం చదనగలిగే ఒక విస్తరణ. obj ఆబ్జెక్ట్ ఫైల్‌గా అనువదించును.

(4) అమలుచేయుట (ఎక్స్‌క్యూషన్)

ఇది C++ ప్రోగ్రామ్ నందు గల చివరి సోపానము ఇక్కడ ఆబ్జెక్ట్ ఫైల్ ఎక్స్‌టెంషన్ .exe తో అమలుకాగల ఫైల్‌గా మారును. ఒకసారిఇ ప్రోగ్రామ్ అమలుకాగల ఫైల్‌గా మారినప్పుడు ఆ ప్రోగ్రామ్ స్వతంత్ర ఉనికిని పొందును. దీనికి అర్థం మీరు మీ విన్నపాన్ని (అప్లికేషన్) ను కంపైలర్ లేదా IDE సహాయం లేకుండా చేయవచ్చును.

11th Computer Science_Chapter-9 Telugu.indd 131 16-03-2023 17:44:20



పటము 9.8 అమలుచేయుట

9.7 C++ అభివృద్ధి ఇన్విరాన్‌మెంట్

C++ నందు ఎన్నో IDE ప్రోగ్రామ్‌లు లభించును. C++ ప్రోగ్రామ్‌ను సృష్టించుటను, సేకరించుటను మరియు అమలు చేయుటను IDE సులభతరం చేయును. చాలా వరకు IDE లు ఓపెన్ సోర్స్ అప్లికేషన్ అగును. ఇవి ఉచితంగా లభించును.

9.7.1 C++ తో గల ప్రసిద్ధ IDE కంపైలర్స్

పట్టిక 9.4 ఓపెన్‌సోర్స్ కంపైలర్స్

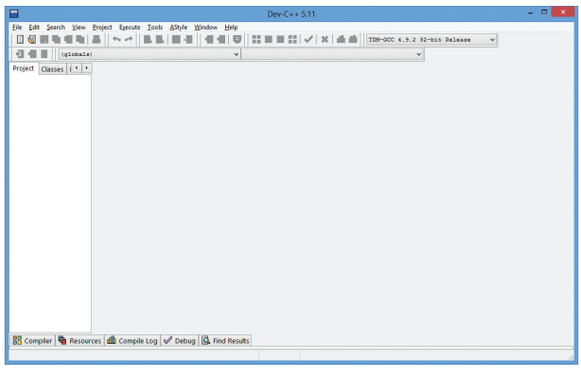
Compiler	Availability
Dev C++	Open source
Geany	Open source
Code::blocks	Open source
Code Lite	Open source
Net Beans	Open source
Digital Mars	Open source
Sky IDE	Open source
Eclipse	Open source

9.7.2 Dev C++ తో పనిచేయుట

దిజన్ల కొద్ది IDE లలో C++ ప్రోగ్రామ్స్ సృష్టించుటకు మనము "Dev C++" కంపైలర్‌ను తీసుకొంటాము. Dev C++ కంపైలర్ ఆధారంగానే ఈ పుస్తకమునందు గల ప్రోగ్రామింగ్ సాంకేతికత మరియు ఉదాహరణలుగా ఉండు ప్రోగ్రామ్స్ అన్నీ అమరియున్నాయి.

Dev C++ ఒక ఓపెన్ సోర్స్ క్రాస్ ప్లాట్ ఫాం (Linux కోసం ఆల్ఫా వర్షన్ అందుబాటులో కలదు) పూర్తి అంశాలగు ఇంటిగ్రేటెడ్ డెవలప్‌మెంట్ ఇన్విరాన్‌మెంట్ (IDE) జనరల్ పబ్లిక్ లైసెన్స్‌తో C మరియు C++ ప్రోగ్రామింగ్ విస్తరించింది. దీనిని డెల్ఫీ (delphi) నందు రాయబడుతుంది. దీనిని ***http://www.bloodshed.net/dev/devcpp.html*** నుండి డౌన్‌లోడ్ చేసుకోవచ్చు.

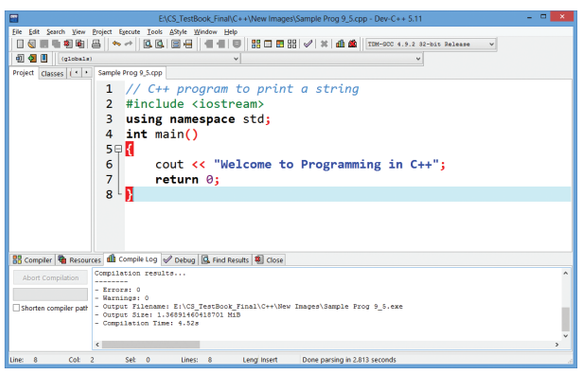
1. **Dev C++** ను ఇన్‌స్టాల్ చేసిన తర్వాత సంబంధిత ఐకాన్ డెస్క్‌టాప్ తెరపైకి వచ్చును. IDE ను ఓపెన్ చేయుటకు డబుల్ క్లిక్ చేయవలెను. తర్వాత Dev C++ IDE ఈ క్రింది విధంగా కనిపిస్తుంది.



పటము 9.9 Dev C++ ఓపెనింగ్ విండో

2. మూల ఫైల్ (సోర్స్ ఫైల్) **File** ను సృష్టించుటకు ను ఎంచుకొనుము. **(Select) → New → Source file** లేదా **Press Ctrl + N.**

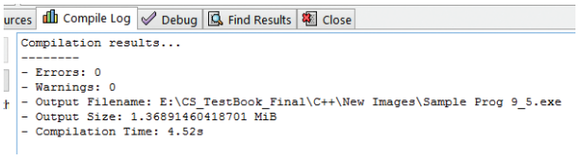
3. కనిపించు తెర (screen) పై మీ C++ ప్రోగ్రామ్‌ను టైప్ చేయండి మరియు ఈ ఫైల్‌ను భద్రపరచుటకు (సేవ్) Click **File → Save** or Pressing **Ctrl + S.** ఇది మీ మూల ఫైల్ యొక్క చివరన .cpp గా జతచేయబడుతుంది. ఫైల్ పేరుతో .cpp అని టైపు చేయనవసరం లేదు.



పటం 9.10 ప్రోగ్రామ్‌తో Dev C++ IDE

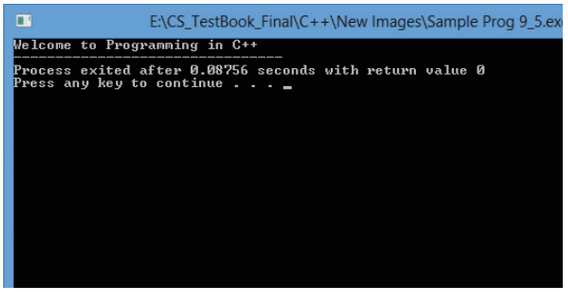
4. **భద్రపరచిన తర్వాత Click Execute → Compile and Run or press F11 key.**
మీ ప్రోగ్రామ్ నందు ఏదైనా దోషాలు (error) ఉన్నట్లయితే ఈ దోషాలు **compile log** క్రింద కనిపించును. మీ ప్రోగ్రామ్ నందు ఎలాంటి దోషములు లేకపోతే **Display** ఈ క్రింది విధముగా కనిపించును.

11th Computer Science_Chapter-9 Telugu.indd 132 16-03-2023 17:44:20



పటము 9.11 Dev C++ Compile Log

5. కంపైలేషన్ విజయవంతముగా పూర్తి అయిన తర్వాత అవుట్‌పుట్ కన్‌సోల్‌పై ఈ క్రింద కనబడే అవుట్‌పుట్ కన్పించును.



పటము 9.12 Dev C++ అవుట్‌పుట్ విండో

9.8 దోషాలలోని రకాలు

కొన్ని సాధారణ దోషములు – వాని రకములు ఈ క్రింద ఇవ్వబడినవి:

దోష రకము	వివరణ
వాక్య నిర్మాణ దోషం (Syntax Error)	• సింటాక్స్ అనునది ఒక ప్రోగ్రామ్ వ్రాయుటకు వాడే వ్యాకరణ నియమాల సమితి. ప్రతి ప్రోగ్రామింగ్ భాష సోర్స్ కోడ్ నిర్మాణము కోసం ప్రత్యేక నియమాలను కలిగియుంటుంది. • C++ యొక్క వ్యాకరణ నియమాలు ఉల్లంఘించబడినపుడు వాక్య నిర్మాణ దోషాలు ఏర్పడుతాయి. • ఉదాహరణ ఈ క్రింది విధముగా టైప్ చేసినప్పుడు C++ దీనిని దోషంగా చూపిస్తుంది. **cout << "Welcome to Programming in C++"** • C++ వ్యాకరణ నియమాల ప్రకారం అమలు కాగల ప్రతి ప్రోగ్రామ్‌ను తప్పనిసరిగా సెమికొలన్‌తో ముగించాలి కాని ఈ స్టేట్‌మెంట్ సెమికొలన్‌తో ముగియలేదు.
అర్థ ధోరణి దోషం (Semantic Error)	• ప్రోగ్రామ్ వ్యాకరణపరంగా సరియైనప్పటికి ఆశించిన ఫలితాన్ని ఇవ్వదు. చరరాశి, ఆపరేటర్, అమలుపరచే క్రమము మొదలగు వానిని సరిగ్గా ఉపయోగించకపోవడం వలన ఇది జరగవచ్చు. దీనికి అర్థం ప్రోగ్రామ్ వ్యాకరణ పరంగా సరియైనది కానీ ఇది కొన్ని తార్కిక దోషాలను కలిగియున్నాయి. కాబట్టి సిమాంటిక్ దోషమును తార్కిక దోషము **"Logic Error"** అని కూడా పిలుస్తారు.
రన్ టైమ్ దోషం (Run-time error)	• ప్రోగ్రామ్ అమలుచేసే సమయంలో ఈ రన్ టైమ్ దోషాలు ఏర్పడుతాయి. ఇది కొన్ని అక్రమ ఆపరేషన్లు వలన జరుగుతాయి. • ఉదాహరణకు, లభ్యతలోలేని ఫైల్‌ను ఒక ప్రోగ్రామ్ తెరచుటకు ప్రయత్నించుట వలన రన్ టైమ్ దోషాలు వస్తాయి.

గుర్తించుకోవాల్సిన అంశాలు

- C++ ను 1979 AT మరియు T బెల్ పరిశోధనా శాలలో జర్నె స్ట్రోస్ట్రప్ అభివృద్ధి చేసారు.
- అక్షర సమితి అనునది C++ ప్రోగ్రామ్స్ వ్రాయుటకు అనుమతించబడిన క్యారక్టర్ (అక్షరాల) సమితి.
- వ్యక్తిగత మూలకాల సమిష్టిని లాక్సికల్ యూనిట్స్ లేదా లాక్సికల్ ఎలిమెంట్స్ లేదా టొకెన్స్ అంటారు.
- మూలపదములు అనునవి C++ కంపైలర్‌కు ఇచ్చు నిర్దిష్టమైన అర్థమునిచ్చు ప్రత్యేక పదాలు.

11th Computer Science_Chapter-9 Telugu.indd 133 16-03-2023 17:44:20

- C++ ప్రోగ్రామ్‌నందలి వివిధ భాగాలైన వేరియబుల్స్ ఫంక్షన్ ఎర్రేస్, క్లాసస్ వంటి వాటికి యూసర్ డీఫైన్డ్ నేమ్స్ (వినియోగదారు నిర్వచిత నామము) ఐడెంటిఫైయర్స్ అగును.
- ప్రోగ్రామ్ అమలు అయ్యే సమయంలో ఏలాంటి మార్పులకు లోనవ్వని సమాచార అంశాలను (డేటా ఐటమ్స్) లిటరల్స్ అంటారు. అందువలన లిటరల్స్ స్థిరరాశులుగా పిలువబడతాయి.
- C++ నందు వివిధ రకాల లిటరల్స్‌ను ఉపయోగిస్తారు. (పూర్ణాంక, చల, క్యారక్టర్, స్ట్రింగ్)
- గణిత తార్కిక ప్రక్రియలను చేయుటకు ఉపయోగించు చిహ్నాలను ఆపరేటర్స్ అందురు.
- C++ ప్రోగ్రామ్ నిర్మాణములో వాడబడు డిలిమిటర్స్ పరిమిత చిహ్నాలను పంక్షువేటర్స్ అందురు. ఇవి సెపరేటర్స్ అని కూడా పిలుస్తారు.
- ఎక్స్‌ట్రాక్షన్ ఆపరేటర్ (>>) మరియు ఇన్‌సర్షన్ ఆపరేటర్ (<<) లను C++ నందు ఇన్పుట్ మరియు అవుట్‌పుట్‌లను పొందుటకు ఉపయోగిస్తారు.

అభ్యాసము (Hands on practice)



- ఈ క్రింది C++ ప్రోగ్రామ్‌ను Dev C++ నందు టైపు చేసి అమలుచేయుము. కంపైలర్ దోషమును సూచించినట్లయితే సరిచేయుటకు ప్రయత్నించుము. మరియు కావలసిన ఫలితం వచ్చు వరకు మరల మరలా ప్రోగ్రామ్‌ను ఎక్సిక్యూట్ చేయుము.

1. మూడు సబ్జెక్ట్‌లలో పొందిన మార్కుల మొత్తాన్ని కనుగొనుటకు C++ ప్రోగ్రామ్‌ను వ్రాయుము

```
#include <iostream>
using namespace std;
int main()
{
        int m1, m2, m3, sum;
        cout << "\n Enter Mark 1: ";
        cin >> m1;
        cout << "\n Enter Mark 2: ";
        cin >> m2;
        cout << "\n Enter Mark 3: ";
        cin >> m3;
        sum = m1 + m2 + m3;
        cout << "\n The sum = " << sum;
}
```

- ఇవ్వబడిన మార్కుల సరాసరిని పొందుటకు పై కోడ్ నందు మార్పు చేయవలెను.

2. వృత్తము యొక్క వైశాల్యమును కనుగొనుటకు C++ ప్రోగ్రామ్

```
#include <iostream>
using namespace std;
int main()
{
        int radius;
        float area;
```


11th Computer Science_Chapter-9 Telugu.indd 134 16-03-2023 17:44:20

```
        cout << "\n Enter Radius: ";
        cin >> radius;
        area = 3.14 * radius * radius;
        cout << "\n The area of circle = " << area;
}
```

3. ఈ క్రింది ప్రోగ్రామ్ నందు గల దోషాలను గుర్తించండి

```
Using namespace std;
int main( )
{
cout << “Enter a value ”;
cin << num1 >> num2
num+num2=sum;
cout >> “\n The Sum= ” >> sum;
```

4. ఈ క్రింది ప్రోగ్రామ్ నందు గల దోష రకమును గుర్తించండి

```
#include <iostream>
using namespace std;
int main()
{
        int h=10; w=12;
        cout << "Area of rectangle " << h+w;
}
```

మూల్యాంకనం





భాగము – అ

సరియైన సమాధానాన్ని ఎన్నుకొనుము:

1. C++ ను అభివృద్ధి చేసినది ఎవరు?

   (అ) చార్లెస్ బాబేజీ (ఆ) జర్నె స్ట్రోస్ట్రప్

   (ఇ) బిల్‌గేట్స్ (ఈ) సుందర్ పిచ్చై

2. C++ కు ఇవ్వబడిన మూల నామము పేరు ఏది?

   (అ) CPP (ఆ) అడ్వాన్స్‌డ్ C (ఇ) C విత్ క్లాసస్ (ఈ) క్లాసస్ విత్ C

3. C++ గా పేరు పెట్టినవారు ఎవరు?

   (అ) రిక్ మస్కిట్టి (ఆ) రిక్ జర్నె (ఇ) బిల్ గేట్స్ (ఈ) డెన్నిస్ రిట్చ్

4. ప్రోగ్రామ్‌లో గల అతి చిన్న వ్యక్తిగత యూనిట్:

   (అ) ప్రోగామ్ (ఆ) అల్గారిథం (ఇ) ఫ్లోచార్ట్ (ఈ) టోకెన్స్

5. ఈ క్రింది ఆపరేటర్లలో C++ నందు గల ఎక్స్‌ట్రాక్షన్ ఆపరేటర్ ఏది?

   (అ) >> (ఆ) << (ఇ) <> (ఈ) ^^

6. ఈ క్రింది వానిలో సరికాని వాక్యమేది?

   (అ) C++ కంపైలర్‌కు మూల పదములు అనునవి నిర్దిష్టమైన అర్థాన్ని తెలుపు నిశ్చిత పదాలు.

   (ఆ) నిశ్చిత పదాలు లేదా మూల పదాలను ఐడెంటిఫైయర్స్ పేరుగా ఉపయోగించవచ్చు.

11th Computer Science_Chapter-9 Telugu.indd 135 16-03-2023 17:44:20

(ఇ) పూర్ణాంక స్థిరరాశి తప్పనిసరిగా దశాంశ స్థానము లేని కనీసం ఒక అంకెనైనా కలిగియుండాలి.
(ఈ) నిజ స్థిరాంకము యొక్క ఘాత రూపము రెండు భాగాలను కలిగి ఉంటుంది.

7. ఈ క్రింది వానిలో చెల్లుబాటయ్యే స్ట్రింగ్ లిటరల్ ఏది?
(అ) 'A' (ఆ) 'Welcome' (ఇ) 1232 (ఈ) "1232"

8. ఉన్నత స్థాయి భాష (హై లెవల్ లాంగ్వేజ్) లో రాయబడిన ప్రోగ్రామ్‌ను ..... అంటారు.
(అ) ఆబ్జెక్ట్స్ కోడ్ (ఆ) సోర్స్ కోడ్ (సి) ఎక్సిగ్యూటబుల్ కోడ్ (ఈ) పైవన్నీ

9. a=5, b=6 అనుకొనుము; అయితే a&b విలువ ఎంత?
(అ) 4 (ఆ) 5 (ఇ) 1 (ఈ) 0

10. ఈ క్రింది వానిలో కంపైలర్ టైమ్ ఆపరేటర్ అని దేనిని అంటారు?
(అ) సైజ్ ఆఫ్ (ఆ) పాయింటర్ (ఇ) వర్చ్యువల్ (ఈ) దిస్

భాగము - ఆ

లఘు సమాధాన ప్రశ్నలు

1. టోకెన్ అంటే ఏమిటి? C++ నందు అందుబాటులో గల టోకెన్‌లను పేర్కొనుము.
2. మూల పదములు అనగానేమి? మూల పదములను ఐడెంటిఫైయర్స్‌గా ఉపయోగించవచ్చునా?
3. ఈ క్రింది స్థిరరాశులు ఏ రకానికి చెందినవి?
(i) 39 (ii) 032 (iii) 0XCAFE (iv) 04.14
4. ఈ క్రింది నిజ స్థిరరాశులను ఘాతరూపంలోనికి మార్చి రాయుము:
(i) 23.197 (ii) 7.214 (iii) 0.00005 (iv) 0.319
5. n=10 అనుకొనుము. అయితే n++ మరియు --n విలువ ఎంత?
6. ఈ క్రింది వానిని జతపరచుము

A	B
(a) Modulus	(1) Tokens
(b) Separators	(2) Remainder of a division
(c) Stream extraction	(3) Punctuators
(d) Lexical Units	(4) get from

విభాగము – ఇ

సంక్షిప్త సమాధాన ప్రశ్నలు

1. మూల పదములు మరియు ఐడెంటిఫైయర్స్ మధ్య గల భేదాలను వర్ణించండి.
2. C++ అక్షర సున్నితత్వమును గలదా? అక్షర సున్నితత్వం అను పదమునకు అర్థమేమి?
3. "=" మరియు "==" లను వ్యత్యాసపరచుము.
4. హెడర్ ఫైల్ ఉపయోగమేమి?
5. మెయిన్ ఫంక్షన్‌కు గల ప్రత్యేకత ఏమిటి?

11th Computer Science_Chapter-9 Telugu.indd 136 16-03-2023 17:44:20

విభాగము – ఈ

వివరముగా సమాధానమునిమ్ము

1. C++ నందు ఉపయోగించే బైనరీ ఆపరేటర్స్ గురించి వ్రాయుము
2. దోషములలో గల రకాలు ఏవి?

ఆధార గ్రంధాలు (References):

(1) Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.
(2) The Complete Reference C++ (Forth Edition), Herbert Schildt.Mc.Graw Hills.
(3) Computer Science with C++ (A text book of CBSE XI and XII), Sumita Arora, Dhanpat Rai & Co.

దత్తాంశ రకాలు, చరరాశులు మరియు సమాసాలు

9.10 పరిచయం

ప్రతి ప్రోగ్రామింగ్ భాషనందు రెండు ప్రాథమిక అంశాలు ఉంటాయి అవి దత్తాంశ రకాలు మరియు చరరాశులు. ఇవి అతి ప్రాథమిక ప్రోగ్రామ్‌లను రాయుటలోనూ తప్పనిసరి అంశాలగును. దత్తాంశ అంశాలను నిర్వహించుటక తగు పూర్వ నిర్వచిత దత్తాంశ రకములను C++ అందించును. ఇలాంటి దత్తాంశ రకములను ప్రాథమిక లేక నిర్మిత దత్తాంశ రకము కాక ప్రోగ్రామర్ కూడా తన సొంత దత్తాంశ రకము అనగా వినియోగదారు. నిర్వచిత దత్తాంశ రకమును సృష్టించుకోవచ్చు. ఈ అధ్యాయం నందు మనము, నిర్మిత దత్తాంశ రకమును గురించి నేర్చుకొందుము.

9.11 దత్తాంశ రకముల భావన

ఈ క్రింది ఉదాహరణను చూద్దాం,

Name = Ram
Age = 15
Average_Mark = 85.6

పై ఉదాహరణ యందు Name, Age, Average_mark వంటి అంశాలు fields వరుసగా Ram, 15 మరియు 85.6 విలువను సూచిస్తున్నది.

ప్రోగ్రామింగ్ భాష నందు **అంశం (field)** ను **చరరాశిగా** మరియు **విలువను దత్తాంశమును** సూచిస్తుంది. పై ఉదాహరణలో ప్రతి దత్తాంశ అంశము వేరువేరుగా ఉన్నది. అనగా "Ram" అక్షరాల వరుస మరియు మిగిలిన రెండు దత్తాంశ అంశాల సంఖ్యలు. మొదటి విలువ పూర్ణసంఖ్య మరియు రెండవది దశాంశ సంఖ్య. వాస్తవికతకు మనం నిత్యజీవితంలో వివిధ రకాల దత్తాంశమును నిర్వహిస్తున్నాయి. దత్తాంశ అంశములు వాని స్వభావము లేదా రకము మారుతూ ఉంటుంది. ఉదాహరణకు దూరం (మీ ఇంటి నుండి పాఠశాలకు) టికెట్టు ధర, పెన్ను ధర మార్కులు ఉష్ణోగ్రత మొదలగునవి.

C++ ప్రోగ్రామ్ నందు ఏదైనా సమాచారాన్ని నిర్వహించుటకు ముందు లాంగ్వేజ్ కంపైలర్‌కు దత్తాంశ రకమును నిర్దిష్టమైన స్పష్టతతో తప్పనిసరిగా అందించాలి. (కొన్ని పూర్వ నిర్వచిత దత్తాంశ రకముతో)

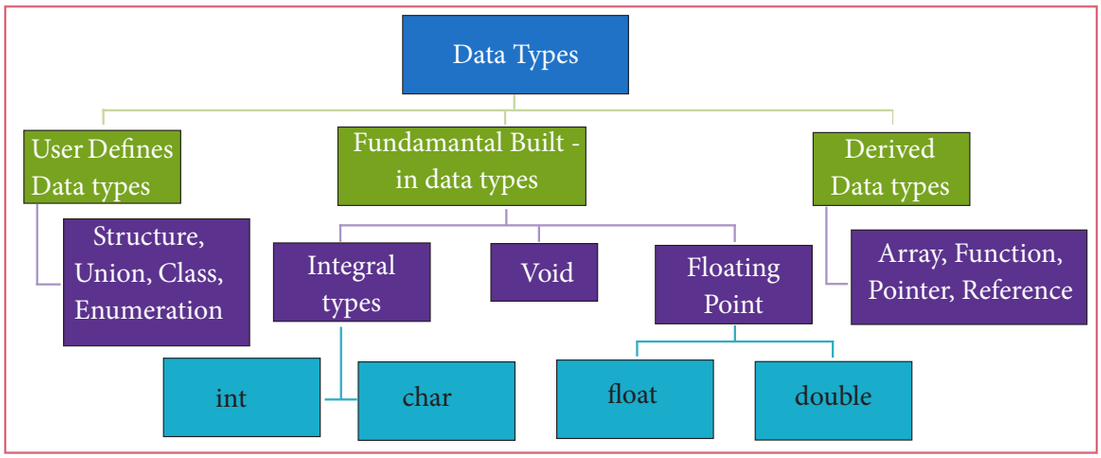
9.12 C++ దత్తాంశ రకములు

C++ నందు రకములను మూడు ప్రధాన వర్గములుగా విభజించబడినది.

(1) ప్రాథమిక దత్తాంశ రకములు (ఫండమెంటల్ డేటా టైప్స్)

11th Computer Science_Chapter-9 Telugu.indd 137 16-03-2023 17:44:20

(2) వినియోగదారు నిర్వచించిన దత్తాంశ రకములు (యుసర్ డిపైన్డ్ డేటా టైప్స్)
(3) ఉత్పాదిత దత్తాంశ రకములు (డిరైవ్డ్ డేటా టైప్స్)



పటము 9.13 C++ నందు దత్తాంశ రకములు

ఈ అధ్యాయములో ప్రాథమిక దత్తాంశ రకములు గురించి మాత్రము నేర్చుకోబోతున్నాము.

దత్తాంశ రకములు పనితీరును అర్థం చేసుకొనుటకు ముందుగా చరరాశులను గురించి తెలుసుకొనుట అవసరం. చరరాశులు అనునవి నిర్దిష్టమైన దత్తాంశ రకములను సేకరించి జ్ఞప్తి కేంద్రాలకు ప్రసిద్ధి. వాస్తవానికి C++ లో చరరాశులను ఉపయోగించుటకు ముందుగానే వాని దత్తాంశ రకములో స్పష్టముగా ప్రకటించబడి ఉండాలి.

ఒక చరరాశిని ప్రకటించుటకు సింటాక్స్:

<data type> <variable name>;

ఉదాహరణ:

int num1;

ఒకే ఒక స్టేట్‌మెంట్‌ను ఉపయోగించి ఒకే దత్తాంశ రకమునకు చెందిన ఒకటి కన్నా ఎక్కువ చరరాశులను ప్రకటించవలసివస్తే ఈ చరరాశులను కామాతో వేరుచేసి ప్రకటించవచ్చు.

ఉదాహరణ:

int num1, num2, sum;

ఉదాహరణకు మీరు కంప్యూటర్ సైన్స్ మార్కులను ఇవ్వదలచుకుంటే మొదటగా మీ మార్కులను సేకరించి ఉంచగల సరియైన దత్తాంశ రకమును చరరాశిగా ప్రకటించాలి. తగిన దత్తాంశ రకమును ఎంచుకొనుటకు అధిక జ్ఞానము మరియు అనుభవం అవసరం. సాధారణంగా మార్పులు పూర్ణ సంఖ్యలుగానే సూచించబడతాయి. అందువలన కంప్యూటర్ సైన్స్ మార్కులను నిల్వ చేయు చరరాశి పూర్ణాంక దత్తాంశ రకమై ఉండాలి.

ఉదాహరణ:

int comp_science_marks;

ఇప్పుడు **comp_science_marks** గా పేరు పెట్టబడిన చరరాశి ఈ మార్పులను నిల్వ చేయడానికి తయారు చరరాశులను గురించి మరిన్ని విషయాలను ఈ అధ్యాయంలో తర్వాత నేర్చుకుందాం.

9.12.1 ప్రాథమిక దత్తాంశ రకములు – పరిచయం

ప్రాథమిక దత్తాంశ రకము C++ లో లభ్యమగు పూర్వ నిర్వచిత దత్తాంశ రకమగును. C++ లో ఐదు రకముల ప్రాథమిక దత్తాంశ రకములు ఉన్నాయి. ఇవి **char, int, float, double** మరియు **void** నిజానికి ఇవి దత్తాంశ రకములను నిర్వచించు మూలపదములు.

11th Computer Science_Chapter-9 Telugu.indd 138 16-03-2023 17:44:20

(1) int దత్తాంశ రకము:

పూర్ణాంక దత్తాంశ రకమునునది పూర్ణాంకములను మాత్రమే స్వీకరించును మరియు వెనుకకు పంపును. ఒక చరరాశిని **int** అని ప్రకటించినట్లయితే C++ కంపైలర్ తనలో పూర్ణాంక విలువలను మాత్రమే నిల్వకావడాన్ని అనుమతిస్తుంది. ఒక వేళ నీవు దత్తాంశ సంఖ్యను నిల్వచేయడానికి ప్రయత్నిస్తే int రకపు చరరాశి పూర్ణాంక భాగమును మాత్రమే స్వీకరించి దశాంశ భాగమును విస్మరిస్తుంది.

ఉదాహరణకు
int num=12;

num1 చరరాశిని పూర్ణాంక రకపు చరరాశిగా ప్రకటించినట్లయితే ఇది పూర్ణాంక విలువను మాత్రమే నిల్వచేస్తుంది.

(2) char దత్తాంశ రకము:

క్యారక్టర్ దత్తాంశ రకమునునది చెల్లుబాటయ్యే ASCII క్యారక్టర్స్‌ను మాత్రమే స్వీకరించును మరియు పంపును. క్యారక్టర్స్ దత్తాంశ రకమును పూర్ణాంక దత్తాంశ రకముగాను చెప్పబడును. ఎందుకనగా అన్ని క్యారక్టర్లు వాని **ASCII కోడ్‌లతో** సమన్వయించి మెమరితో ఉంచబడుతుంది. ఒక చరరాశి char గా ప్రకటించబడినట్లయితే C++ కంపైలర్ ఏదైనా క్యారక్టర్ లేదా పూర్ణాంక విలువ నిల్వను అనుమతిస్తుంది.

ఉదాహరణ 1:-
char c='A';
cout<<ch ;

పై కోడ్ నందు, **ch** అనునది ఒక char రకపు చరరాశిగా ప్రకటించినట్లయితే ఇది క్యారక్టర్‌లను మాత్రమే స్వీకరిస్తుంది. ఇది క్యారక్టర్ A ను తెరపై చూపిస్తుంది.

ఉదాహరణ 2:-

char ch='A'
cout<<ch=ch+1;

పై స్టేట్‌మెంట్స్‌లో **ch** విలువ 1 గా పెంచబడినది మరియు ఈ కొత్త విలువ అదే చరరాశి **ch** లోనే నిల్వ చేయబడుతుంది (అంకగణిత ప్రక్రియలన్నీ సంఖ్యలపై మాత్రమే ప్రయోగించగలము కాని అక్షరాలపై (ఆల్ఫాబేట్‌లపై) కాదని గుర్తుపెట్టుకోండి) కావున ఇది B ను తెరపై చూపిస్తుంది.

int మరియు **char** రకపు దత్తాంశ రకములు ఒకదానితో ఒకటి కలిసి ఎలా పనిచేస్తుందో ఈ క్రింది ప్రోగ్రామ్ వివరిస్తుంది.

ఉదాహరణ 9.3: C++ program to get an ASCII value and display the corresponding character

```
#include <iostream>
using namespace std;
int main ()
{
        int n;
        char ch;
        cout << "\n Enter an ASCII code (0 to 255): ";
        cin >> n;
        ch = n;
        cout << "\n Equivalent Character: " << ch;
}
```

The output

```
Enter an ASCII code (0 to 255): 100
Equivalent Character: d
```


11th Computer Science_Chapter-9 Telugu.indd 139 16-03-2023 17:44:20

పై ప్రోగ్రామ్‌నందు చరరాశి **n** ఒక int రకముగానూ మరొక చరరాశి **ch** రకముగాను ప్రకటించబడినది. అమలు చేయునప్పుడు ప్రోఆమ్ వినియోగదారుని ASCII విలువను పొందుపరచమని సూచిస్తుంది. వినియోగదారుడు ASCII విలువగా ఒక పూర్ణాంకమును పొందుపరచినపుడు ఇది చరరాశి **n** లో నిల్వ అవుతుంది. స్టేట్‌మెంట్ **ch = n**; లో **n** విలువను **ch** కు ఇవ్వబడుతుంది. ఇక్కడ **ch** అనునది char రకపు చరరాశి అని గుర్తుపెట్టుకోండి.

ఉదాహరణకు వినియోగదారు 100 ను ఎంటర్ చేసినపుడు ప్రారంభంలో 100 చరరాశి **n** నందు నిల్వచేయబడుతుంది. తర్వాతి స్టేట్‌మెంట్ నందు **n** విలువ అనగా 100 ను **ch** కు ఇవ్వబడుతుంది. **ch** అనునది char రకము అయినప్పటికీ ఇది తగిన ASCII క్యారక్టర్‌ను అవుట్‌పుట్‌గా చూపిస్తుంది.

(100 కు సమానమైన ASCII క్యారక్టర్ d)

(3) చల దత్తాంశ రకము:

ఒక చరరాశిని చల (స్వేచ్ఛ / తేలియాడునది) గా ప్రకటించినట్లయితే అన్నీ విలువ చల స్థాన విలువగానే నిల్వచేయబడతాయి.

చల దత్తాంశ రకమును ఉపయోగించుట వలన కలుగు రెండు ప్రయోజనాలు.

(1) ఇవి పూర్ణాంకముల మధ్య గల విలువలకు ప్రాతినిధ్యం వహిస్తాయి.

(2) ఇవి అత్యధిక వ్యాప్తి గల విలువలకు ప్రాతినిథ్యం వహిస్తాయి.

అదే సమయంలో, చల స్థానవిలువల ప్రక్రియ, పూర్ణాంక రక విలువల ప్రక్రియ కన్నా చాల నెమ్మదిగా జరుగును. ఇది ఈ చల స్థాన రక ప్రక్రియనందు గల నష్టము.

ఉదాహరణకు

float num=13.4;

పై ఉదాహరణలో num variable చల రకముగా ప్రకటించబడినది.

(4) డబుల్ దత్తాంశ రకము:

ఇది ద్వి ఖచ్చితత్వ స్థాన సంఖ్యల కోసం (ఖచ్చితత్వం అనగా దశాంశ స్థానము తర్వాత ప్రాధాన్యత గల సంఖ్యలు ఉంటాయ). చల స్థాన సంఖ్యల ప్రక్రియల కొరకు ద్వి ను వాడుదురు. కాని ఇవి / ఈ రకం చల రకమును కన్నా రెండింతల స్పేస్ ఆక్రమిస్తాయి. దీనికి అర్థం, చలరకాని కన్నా అధిక ప్రక్రియలు, ఈ ద్వి చల రకములో చోటు చేసుకొంటాయి.

(5) void దత్తాంశ రకము:

void కు వ్యాకరణాత్మక అర్థం శూన్య / ఖాళీ ప్రదేశం ఇక్కడ C++ నందు void దత్తాంశ రకపు చరరాశులు ప్రత్యేకంగా శూన్య సమితి విలువను సూచిస్తాయి. ఎలాంటి విలువలను ఇవ్వని ప్రక్రియల కోసం ఈ రకపు చరరాశిని వెనుకకు పంపు / త్రిప్పి పంపు విధముగా (return type) గా ఉపయోగిస్తారు.

స్వయం మూల్యాంకనము

1. ప్రాథమిక దత్తాంశ రకము అంటే అర్థమేమిటి?
2. char దత్తాంశ రకమును క్యారక్టర్లను సూచించుటకు ఉపయోగిస్తారు అయినను వీటిని పూర్ణాంక రకము అని ఎందుకు అంటారు?
3. పూర్ణాంకములపై చల స్థాన సంఖ్యల ప్రయోజనాలేవి?
4. ద్వి దత్తాంశ రకము మరో చల స్థాన రకము అగును. అయితే దీనిని ప్రత్యేకమైన దత్తాంశ రకముగా ఎందుకు పరిగణిస్తారు?
5. void దత్తాంశ రకము యొక్క ఉపయోగమేమి?

11th Computer Science_Chapter-9 Telugu.indd 140 16-03-2023 17:44:20

9.12.2 ప్రాథమిక దత్తాంశ రకముల యొక్క జ్ఞప్తి ప్రాతినిధ్యం:

ఒక చరరాశిని ప్రత్యేక దత్తాంశ రకముగా ప్రకటించుటకు గల కారణాలలో ముఖ్య కారణం వానికి స్టోరేజ్ మెమరి తగు స్థలాన్ని కేటాయించుట. నిల్వచేయబడిన ప్రోగ్రామ్ అంశం ప్రకారం ప్రక్రియ అమలు కావుటకు ముందే ప్రధాన మెమరి యందు ప్రతి సమాచారానికి స్థలము కేటాయించబడాలి. కాబట్టి C++ కంపైలర్ ప్రతి ఒక సమాచారమునకు తగు స్థలమును కంపైలర్ ప్రామాణితను అనుసరించి నిర్ధిష్ట మెమరిలో కేటాయించును.

ఈ క్రింది 9.5 పట్టిక ప్రతి ప్రాథమిక దత్తాంశ రకమునకు ఎంత స్థలము కేటాయించబడినదో చూపును. ప్రతి సమాచారము కంప్యూటర్ మెమరీలో బైనరీ డిజిట్స్‌గానే నిల్వచేయబడతాయని గుర్తించుకోండి (యూనిట్ 1 అధ్యాయం 2 ను చూడుము).

పట్టిక 9.5 ప్రాథమిక దత్తాంశ రకమునకు కేటాయించబడిన మెమరీ

దత్తాంశ రకము	మెమరీ స్థలం		విలువ పరిధి
	in terms of bytes	in terms of bits	
char	1 బైటు	8 బిట్లు	-128 to 127
int	2 బైట్లు	16 బిట్లు	-32,768 to 32,767
float	4 బైట్లు	32 బిట్లు	3.4×10^{-38} to 3.4×10^{38} -1
double	8 బైట్లు	64 బిట్లు	1.7×10^{-308} to 1.7×10^{308}-1

9.12.3 దత్తాంశ రకము మాడిఫైయర్స్ (పరివర్తకులు):

void రకపు దత్తాంశ రకమును మినహాయించి మిగిలిన అన్నీ ప్రాథమిక దత్తాంశ రకమును నిల్వ సామర్థ్యమును పరివర్తన చేయుటకు ఉపయోగించునని మాడిఫైయర్ మాములుగా ప్రతి ప్రాథమిక దత్తాంశ రకములు వాని విలువల వ్యాప్తిని నిల్వచేయుటకు మెమరీనందు దత్తాంశ అంశాలను కలిగియుంటాయి. ఉదాహరణకు int దత్తాంశ రకము రెండు byte ల దత్తాంశాన్ని మాత్రమే నిల్వచేయును. వాస్తవానికి కొన్ని పూర్ణాంక దత్తాంశాలు దీర్ఘమైనవిగా ఉండవచ్చు మరియు వీటికి మెమరీ నందు ఎక్కువ స్థానం అవసరమవుతుంది. ఈ సందర్భంలో మనము భారీ పూర్ణాంక విలువలకు మెమరీ నందు తగు పరివర్తనలు చేయవలసి ఉంటుంది. **ఏదైనా ప్రాథమిక దత్తాంశ రకము యొక్క మెమరీ పరివర్తన (అనగా విస్తరించుట లేదా కుదించుట) చేసి మెమరీ నందు సరియైన స్థలమును కేటాయించుటకు మాడిఫైయర్స్ ఉపయోగపడుతాయి. వీటిని క్వాలిఫైయర్స్ అని కూడా పిలుస్తారు.**

C++ నందు నాలుగు మాడిఫైయర్స్ ఉపయోగించబడతాయి. అవి:

(1) సంజ్ఞాత్మక (2) సంజ్ఞలేనివి (3) దీర్ఘము (4) హ్రస్వం

ఈ నాలుగు మాడిఫైయర్స్ ఏ ప్రాథమిక దత్తాంశ రకముతోనైనా ఉపయోగించవచ్చు. ప్రతి దత్తాంశ రకము (అంటే మాడిఫైయర్ఽస లేదా మాడిఫైయర్స్ లేకుండా) యొక్క మెమరీ కేటాయింపు ఈ క్రింది పట్టికలో చూపబడినది.

11th Computer Science_Chapter-9 Telugu.indd 141 16-03-2023 17:44:21

పూర్ణాంక రకము

పట్టిక 9.6 దత్తాంశ రకముల మెమరీ కేటాయింపు

దత్తాంశ రకము		మెమరీ స్థలం		విలువ పరిధి
		in terms of bytes	in terms of bits	
short	**short int** కు సంక్షిప్తనామమే 'షార్ట్'	2 బైట్లు	16 బిట్లు	-32,768 to 32,767
unsigned short	తీసివేత (మైనస్) సైన్‌లేని ఒక పూర్ణ సంఖ్య	2 బైట్లు	16 బిట్లు	0 to 65535
signed short	మైనస్ సైన్ కలిగిన ఒక పూర్ణ సంఖ్య	2 బైట్లు	32 బిట్లు	-32,768 to 32,767
షార్ట్ మరియు సైన్ షార్ట్ రెండునూ ఒకేలాగ ఉండును				
int	సైన్ట్ అయిన లేదా సైన్డ్ కాని ఒక పూర్ణసంఖ్య	2 బైట్లు	16 బిట్లు	-32,768 to 32,767
unsigned int	ఎలాంటి సైన్ లేని పూర్ణసంఖ్య	2 బైట్లు	16 బిట్లు	0 to 65535
signed int	సైన్ కలిగిన పూర్ణంఖ్య	2 బైట్లు	16 బిట్లు	-32,768 to 32,767
షార్ట్ మరియు ఇంట్ ఇవి రెండునూ ఒకేలాంటివి				
long	లాంగ్ ఇంట్‌కు సంక్షిప్త నామమే లాంగ్	4 బైట్లు	32 బిట్లు	-2147483648 to 2147483647
unsigned long	ఎలాంటి సైన్ లేని రెండు ఖాళీస్థలాలు కల్గిన పూర్ణసంఖ్య	4 బైట్లు	32 బిట్లు	0 to 4,294,967,295
signed long	సైన్ కలిగిన రెండు ఖాళీస్థలాలు కల్గిన పూర్ణసంఖ్య	4 బైట్లు	32 బిట్లు	-2147483648 to 2147483647

పూర్ణాంక రకము రెండు byte డేటాను మాత్రమే అంగీకరిస్తుందని పై పట్టిక స్పష్టంగా సూచిస్తుంది. అయితే దీర్ఘమైన int (పూర్ణాంక రకము) రెట్టింపు డేటాను అంగీకరిస్తుంది. అంటే నాలుగు byte ల దత్తాంశము. కాబట్టి మనము ఎక్కువ అంకెలను ఒక దీర్ఘ int లో నిల్వచేయవచ్చు (దీర్ఘం అనే మాడిఫైయర్ ప్రాథమిక దత్తాంశ రకము)

char రకము

పట్టిక 9.7 క్యారక్టర్ రకముల మెమరీ కేటాయింపు

దత్తాంశ రకము		మెమరీ స్థలం		విలువ పరిధి
		in terms of bytes	in terms of bits	
char	Signed ASCII character	1 బైటు	8 బిట్లు	-128 to 127
unsigned char	ASCII character without sign	1 బైటు	8 బిట్లు	0 to 255
signed char	ASCII character with sign	1 బైటు	8 బిట్లు	-128 to 127

11th Computer Science_Chapter-9 Telugu.indd 142 16-03-2023 17:44:21

స్వేచ్ఛా స్థాన రకము (ఫ్లోటింట్ పాయింట్ టైప్)

పట్టిక 9.8 స్వేచ్ఛా స్థాన దత్తాంశ రకముల మెమరీ కేటాయింపు

దత్తాంశ రకము		మెమరీ స్థలం		విలువ పరిధి
		in terms of bytes	in terms of bits	
float	signed fractional value	4 బైట్లు	32 బిట్లు	3.4×10^{-38} to $3.4\times10^{38}-1$
double	signed more precision fractional value	8 బైట్లు	64 బిట్లు	1.7×10^{-308} to $1.7\times10^{308}-1$
long double	signed more precision fractional value	10 బైట్లు	80 బిట్లు	3.4×10^{-4932} to $1.1\times10^{4932}-1$

ఉపయోగించబడబోవు కంపైలర్ రకము ఆధారంగా మెమరీ కేటాయింపు మారుతూ ఉంటుంది. ఇక్కడ అన్నీ విలువలు **Turbo C++** కంపైలర్ ప్రకారం ఇవ్వబడి ఉన్నాయి. **Dev C++** పూర్ణాంక **int** మరియు దీర్ఘ **long** రకములకు ఎక్కువ స్థలాన్ని ఇస్తుంది. ఈ క్రింది పట్టిక 9.9 నందు Turbo C++ మరియు Dev C++ లలో మెమరీ కేటాయింపు మధ్యగల తేడాను చూపిస్తుంది.

పట్టిక 9.9 Turbo C++ మరియు Dev C++ ల యొక్క మెమరీ కేటాయింపు

దత్తాంశ రకము	Memory size in bytes	
	Turbo C++	Dev C++
short	2	2
unsigned short	2	2
signed short	2	2
int	2	**4**
unsigned int	2	**4**
signed int	2	**4**
long	4	4
unsigned long	4	4
signed long	4	4
char	1	1
unsigned char	1	1
signed char	1	1
float	4	4
double	8	8
long double	10	**12**

int మరియు దీర్ఘ రకములకు Dev C++ 4 బైట్లను కేటాయిస్తున్నది. కావున ఇందులో ఏ రకానైనా భారీ పూర్ణాంక విలువలతో Dev C++ లో వ్రాయు ప్రోగ్రామ్‌ల నిర్వహణకు ఉపయోగించవచ్చు.

గమనిక: sizeof( ) ఒక ఆపరేటర్ ఇది దత్తాంశ రకము యొక్క పరిమాణాన్ని తెలియజేస్తుంది.

11th Computer Science_Chapter-9 Telugu.indd 143 16-03-2023 17:44:21

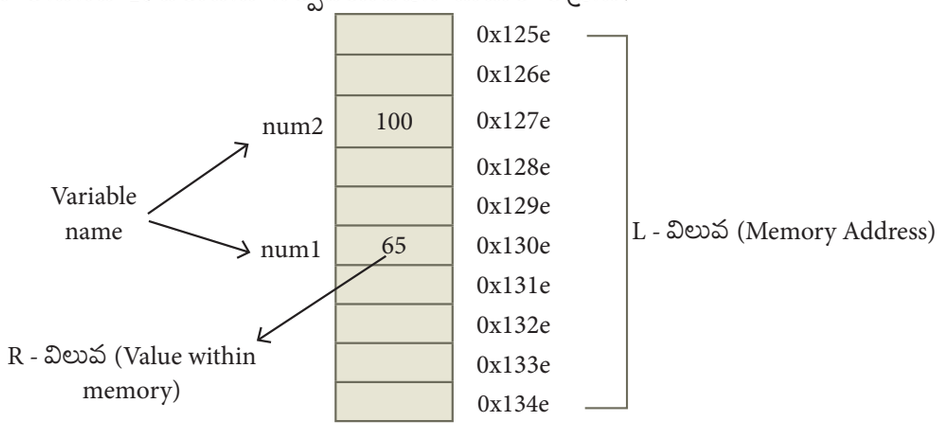
C++ లో సంఖ్య ప్రత్యాయాలు (పూర్వ పదము):

పూర్ణాంకములకు మరియు స్వేచ్ఛా స్థాన సంఖ్యలకు పలు విధములయ్యిన పూర్వ పదములు కలవు. ఈ పూర్వ పదమనునది ఒకే విలువకు వేర్వేరు దత్తాంశ రకములను కేటాయించునది. ఉదాహరణకు మీరు int, దీర్ఘ అసంజ్ఞాత్మక int మరియు అసంజ్ఞాత్మక దీర్ఘ int లలో 45 ను నిల్వచేయడానికి పూర్వపదం **L** లేదా **U** తో జతచేసి అనగా **45L** లేక **45U** లు ఉపయోగించవచ్చు. ఈ రకపు ప్రకటనలు ఇచ్చిన విలువలను దీర్ఘ మరియు అసంజ్ఞాత్మకముగా నిల్వచేయమని కంపైలర్‌కు సూచించుటయే స్వేచ్ఛా స్థాన విలువను 'F' తో సూచించవచ్చును. ఉదాహరణ 3.14F

9.13 చరరాశులు (Variables)

చరరాశులు అనునవి మెమరీ నందు విలువలను నిల్వచేయడానికి కేటాయించబడిన నిర్ధిష్ట స్థానాలకు వినియోగదారు నిర్వచించిన పేర్లు చరరాశులు, ఐడెంటిఫైయర్స్ కూడా అవుట వలన ఐడెంటిఫైయర్‌లకు పేరు పెట్టుటలో గల అవే నియమాలను ఒక చరరాశికి పేరు పెట్టుటలోనూ ఆచరించవలెను. ఇవి పేరు పెట్టబడిన స్థానాలు కావున దీనిని సింభాలిక్ వేరియబుల్స్ అని కూడా అంటారు.
సింబాలిక్ వేరియబుల్స్ రెండు విలువలతో అనుసంధానింపబడి ఉండును. అవి **R**- విలువ మరియు **L**- విలువ

- R-విలువ అనునది మెమరీ స్థానంలో నిల్వచేయబడిన దత్తాంశము
- L-విలువ అనునది R-విలువను నిల్వచేయబడిన మెమరీ అడ్రసు.



పటము 9.14 చరరాశి యొక్క మెమరీ కేటాయింపు

మెమరీ అడ్రసులు షడ్ దశాంశ విలువల యొక్క రూపంలోనే ఉంటుందని గుర్తించుకోండి.

9.13.1 చరరాశుల ప్రకటన

ప్రోగ్రామ్‌లో ఉపయోగించడానికి ముందుగానే ప్రతి చరరాశి ప్రకటించబడాలి. డిక్లరేషన్ అనునది వేరియబుల్ పేరుతో పాటు నిర్దేశిత రకమును అనుసరించి మెమరీను కేటాయించమని కంపైలర్‌కు సూచించే ప్రక్రియ. ఉదాహరణకు Dev C++ లో ఒక చరరాశిని int రకముగా ప్రకటించినట్లయితే దీనికి కంపైలర్ నాలుగు byte ల మెమరీను కేటాయిస్తుంది. అందువలన నిల్వ చేయబడాల్సిన ప్రతి యొక్క చరరాశి విలువ తన యొక్క రకముతో పాటు ప్రకటించబడాలి.
ఒకటి కన్నా ఎక్కువ చరరాశులు ప్రకటించుట:

ఒకే రకానికి చెందిన ఒకటి కన్నా ఎక్కువ చరరాశులను ఒకే ఒక స్టేట్‌మెంట్‌లో వ్యక్తిగత చరరాశులను కామాను ఉపయోగించి వ్రాయవచ్చు.

సింటాక్స్ :
<data type> <var1>, <var2>, <var3> …… <var_n>;

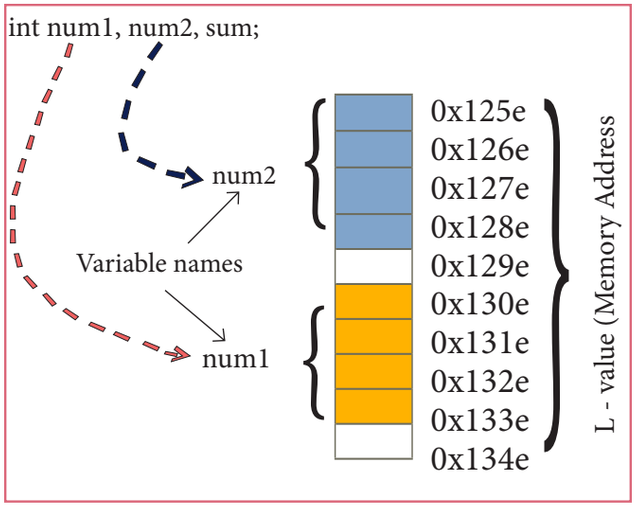
11th Computer Science_Chapter-9 Telugu.indd 144 16-03-2023 17:44:21

ఉదాహరణ:

int num1, num2, sum;

పై ఉదాహరణలో మూడు చరరాశులు (int) రకముగా ప్రకటించబడి ఉన్నాయి. దీనికి అర్థం (num1, num2) మరియు (sum) లందు పూర్ణాంక విలువలు మాత్రమే నిల్వచేయబడతాయి. పై ఉదాహరణలో ప్రకటించినట్లు ప్రతి చరరాశికి 4 బైట్ల మెమరీ కేటాయించబడుతుంది.

For the above declaration, the C++ compiler allocates 4 bytes of memory (i.e. 4 memory boxes) for each variable.



పటము 9.15 int రక చరరాశులు – మెమరీ కేటాయింపు

ఒక చరరాశిని ఏ ప్రారంభ విలువ లేకుండా ప్రకటించినట్లయితే ఈ చరరాశికి మెమరీలో కేటాయించబడిన స్థలాన్ని ఏదో ఒక తెలియని విలువ ఆక్రమిస్తుంది. ఇలాంటి తెలియని విలువను **"Junk"** లేదా **"Garbage"** విలువలు అంటారు.

```
#include <iostream>
using namespace std;
int main()
{
        int num1, num2, sum;
        cout << num1 << endl;
        cout << num2 << endl;
        cout << num1 + num2;
}
```

పై ప్రోగ్రామ్‌లో చరరాశులు **num1** మరియు **num2** లకు కేటాయించబడిన మెమరీను కొన్ని తెలియని విలువలు ఆక్రమించబడియున్నది మరియు స్టేట్‌మెంట్ **cout << num1 + num2;** ఆ తెలియని వ్యర్థ విలువల మొత్తాన్ని తెరపై చూపిస్తుంది.

9.13.2 చరరాశుల యొక్క ప్రారంభీకరణ (Initialization of variables)

ఒక చరరాశిని ప్రకటించునప్పుడు ప్రారంభ విలువను ఆ చరరాశికి కేటాయించడాన్ని ప్రారంభీకరణ అంటారు.

ఉదాహరణ:

int num = 100;

float pi = 3.14;

double price = 231.45;

11th Computer Science_Chapter-9 Telugu.indd 145 16-03-2023 17:44:21

ఇక్కడ చరరాశులైన (num, pi) మరియు (price) అన్నియూ ప్రకటించునప్పుడు ప్రారంభీకరణ చేయబడింది. ఈ ప్రారంభ విలువను ప్రోగ్రామ్‌ను అమలు చేసేటప్పుడు మార్చుకోవచ్చు.

ఉదాహరణ 9.6 స్థూపము యొక్క వక్రతల వైశాల్యము **(CSA) (CSA = 2 pi r * h)**

```
#include <iostream>
using namespace std;
int main()
{
   float pi = 3.14, radius, height, CSA;
   cout << "\n Curved Surface Area of a cylinder";
   cout << "\n Enter Radius (in cm): ";
   cin >> radius;
   cout << "\n Enter Height (in cm): ";
   cin >> height;
   CSA = (2*pi*radius)*height;
   system("cls");
   cout << "\n Radius: " << radius <<"cm";
   cout << "\n Height: " << height << "cm";
   cout << "\n Curved Surface Area of a Cylinder is " << CSA <<" sq. cm.";
}
```

Output:

```
Curved Surface Area of a cylinder
Enter Radius (in cm): 7
Enter Height (in cm): 20
Radius: 7cm
Height: 20cm
Curved Surface Area of a Cylinder is 879.2 sq. cm.
```

ఒక స్టేట్‌మెంట్‌లో ఒకే రకానికి చెందిన చరరాశులను ప్రారంభీకరణ చేయవలెను.

ఉదాహరణ:

int x1 = -1, x2 = 1, x3, n;

9.13.3 వ్యూహాత్మక ప్రారంభీకరణ (Dynamic Initialization)

ప్రోగ్రామ్‌ను అమలు చేసేటప్పుడు ఒక చరరాశిని ప్రారంభించడమే "**వ్యూహాత్మక ప్రారంభీకరణ**" అంటారు. ఉదాహరణకు,

```
int num1, num2, sum;
sum = num1 + num2;
```

పై రెండు స్టేట్‌మెంట్‌ను ఒకే స్టేట్‌మెంట్‌లో ఈ క్రింది విధంగా కలిపి చూపవచ్చును:

int sum = num1+num2;

ఇది అమలు సమయంలో తెలిసిన num1 మరియు num2 విలువల మొత్తాన్ని ప్రారంభీకరణం చేస్తుంది

ఉదాహరణ 9.7 C++ Program to illustrate dynamic initialization

```
#include <iostream>
using namespace std;
int main()
{
        int num1, num2;
        cout << "\n Enter number 1: ";
        cin >> num1;
        cout << "\n Enter number 2: ";
        cin >> num2;
        int sum = num1 + num2; // Dynamic initialization
        cout << "\n Average: " << sum /2;
}
```

Output:

```
Enter number 1: 78
Enter number 2: 65
Average: 71
```


11th Computer Science_Chapter-9 Telugu.indd 146 16-03-2023 17:44:21

పై ప్రోగ్రామ్‌నందు num1 మరియు num2 విలువలు వచ్చిన తరువాత, వాటి మొత్తం ప్రకటించబడుతుంది మరియు ఈ రెండు చరరాశుల కూడిక ప్రారంభీకరణం చేయబడింది. ఆ తర్వాత ఇది 2 చే భాగించబడుతుంది.

ఉదాహరణ 9.8: C++ program to find the perimeter and area of a semi circle

```
#include <iostream>
using namespace std;
int main()
{
        int radius;
        float pi = 3.14;
        cout << "\n Enter Radius (in cm): ";
        cin >> radius;
        float perimeter = (pi+2)*radius;   // dynamic initialization
        float area = (pi*radius*radius)/2; // dynamic initialization
        cout << "\n Perimeter of the semicircle is " << perimeter << " cm";
        cout << "\n Area of the semicircle is " << area << " sq.cm";
}
```

Output:
Enter Radius (in cm): 14
Perimeter of the semicircle is 71.96 cm
Area of the semicircle is 307.72 sq.cm

9.13.4 యాక్సెస్ మాడిఫైయర్ కాన్స్ట్ (The Access modifier const)

(const) అనునది ఒక స్థిరరాశిని ప్రకటించుటకు ఉపయోగపడే మూలపదము. స్థిరరాశులను గురించి ఈ అధ్యాయంలో మీరు ముందుగానే చదివియున్నారు. (const) మూలపదం ఒక స్థిరరాశి యొక్క ప్రాప్యతను పరివర్తనను పరిమితం చేస్తుంది. కాబట్టి ఇది యాక్సెస్ మాడిఫైయర్ పిలవబడుతుంది.

ఉదాహరణకు,

int num = 100;

పై స్టేట్‌మెంట్ నందు ఒక చరరాశి విలువ num ను ప్రారంభ విలువ 100 గా ప్రకటించబడినది. ఎలాగైనా num యొక్క విలువను ప్రోగ్రామ్ అమలు చేసేటప్పుడు మార్చుకోవచ్చు. పై నిర్వచనాన్ని const int గా పరివర్తన చేసినప్పుడు num = 100 అంటే num చరరాశి ఒక స్థిరరాశిగా ప్రోగ్రామ్ ముగిసేంతవరకు దాని విలువ 100 గానే మిగిలిపోతుంది. మరియు ఈ విలువను ప్రోగ్రామ్ అమలు చేసేటప్పుడు మార్చడానికి వీలుకాదు.

```
#include <iostream>
using namespace std;
int main()
{
        const int num=100;
        cout << "\n Value of num is = " << num;
        num = num + 1; // Trying to increment the constant
        cout << "\n Value of num after increment " << num;
}
```

పై కోడ్ నందు టర్బో కంపైలర్ అయితే **"Cannot modify the const object"** అనే దోష సందేశం మరియు Dev C++ కంపైలర్ అయితే **"assignment of read only memory num"** అనే దోష సందేశం తెరపై ప్రదర్శితమవుతుంది.

11th Computer Science_Chapter-9 Telugu.indd 147 16-03-2023 17:44:21

9.13.5 ఆధారాలు (References)

రెఫరెన్స్ అనునది ముందుగానే నిర్వచింపబడిన చరరాశికి మారుపేరు (alias) ను అందిస్తుంది. రెఫరెన్స్ యొక్క ప్రకటన ఆధార రకము మరియు ఒక చిహ్నం (గుర్తు (ampersand) లను కలిగి ఉంటుంది. రెఫరెన్స్ చరరాశి పేరు అనునది ముందుగానే నిర్వచింపబడిన చరరాశి విలువకు కేటాయించబడుతుంది.

సింటాక్స్ :

<type> <& reference_variable> = <original_variable>;

ఉదాహరణ 9.9: రిఫరెన్స్ వేరియబుల్‌ను ప్రకటించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main()
{
        int num;
        int &temp = num; //declaration of a reference variable temp
        num = 100;
cout << "\n The value of num = " << num;
cout << "\n The value of temp = " << temp;
}
```

పై ప్రోగ్రాం యొక్క ఔట్‌పుట్ విలువలు

num విలువ = 100

temp విలువ = 100

స్వయం మూల్యాంకనము

1. మాడిఫైయర్లు అనగానేమి? మాడిఫైయర్స్ ఉపయోగాలేవి?
2. ఈ క్రింది ... స్టేట్‌మెంట్ నందు గల తప్పు ఏది ?
3. చరరాశి అనగానేమి? చరరాశి చిహ్నాత్మక చరరాశిగా ఎందుకు పిలువబడుతుంది?
4. చరరాశి యొక్క వ్యూహాత్మక ప్రారంభీకరణం అర్థం ఏమి? ఉదాహరణలిమ్ము?
5. ఈ క్రింది స్టేట్‌మెంట్‌లో గల తప్పును కనుగొనుము const int x;

9.14 అవుట్‌పుట్ ఫార్మాటింగ్ (Formatting Output)

సులభంగా చదవడం మరియు అర్థం చేసుకోవడానికి అవుట్‌పుట్ తెరలను అభివృద్ధి చేయుట అవుట్‌పుట్‌ను రూపొందిచడం చాలా ముఖ్యం. నిర్వాహకాలు (మానిపులేటర్స్) ఏదైనా C++ ప్రోగ్రామ్‌లో అవుట్‌పుట్‌ను రూపొందించుటకు ఉపయోగపడతాయి. మానిపులేటర్స్ అనునవి ప్రత్యేకించిన ప్రక్రియలగు insertion (<<) మరియు extraction(>>) ఆపరేటర్స్‌తో కలిపి ఉపయోగించబడును.

C++ ఫార్మెటింగ్ చేయుటకు పలు రకాల ఇన్పుట్ మరియు అవుట్‌పుట్ మానిపులేటర్స్‌ను ఇచ్చుచున్నది. **endl, setw, setfill, setprecision** మరియు **setf** లు సాధారణంగా వాడబడే మానిపులేటర్స్ ఈ మానిపులేటర్స్‌ను ఉపయోగించేటప్పుడు మీరు వీనికి తగ్గ హెడర్ ఫైల్‌ను జతచేయవలసి ఉంటుంది. **endl** మానిపులేటర్ iostream హెడర్ ఫైల్‌లోని అంశము. **setw, setfill, setprecision** మరియు **setf మానిపులేటర్స్** iomanip హెడర్ ఫైల్‌లోని అంశాలు endl (పంక్తి ముగింపు)

11th Computer Science_Chapter-9 Telugu.indd 148 16-03-2023 17:44:21

endl (End the Line)

C++. లో endl అనునది లైన్ ఫీడర్‌గా ఉపయోగించబడుతుంది. దీనిని '\n'. కు మారుగా కూడా ఉపయోగించవచ్చు మరో మాటలతో చెప్పాలంటే endl అనునది ఒక కొత్త పంక్తిని inserts చేసి cursor ను మరో కొత్త పంక్తి ప్రారంభంలో ఉంచుతుంది. endl మరియు '\n', ల మధ్య భేదములు ఉన్నప్పటికీ అవి ఒకే విధమయిన పనులను చేస్తుంది.

- endl – కొత్త పంక్తిని Inserts చేసి buffer ను తొలగిస్తుంది
- '\n' – కొత్త పంక్తిని మాత్రమే Insert చేస్తుంది

ఉదాహరణ :

cout << "**\n** The value of num = " << num;

cout << "The value of num = " << num <<**endl;**

పై రెండూ స్టేట్‌మెంట్స్ ఒకే విధమయిన అవుట్‌పుట్‌ను ఇస్తాయి.

setw ( )

setw మానిపులేటర్ అవుట్‌పుట్ కోసం సరిపోడు స్థల వెడల్పును నిర్ణయిస్తుంది. **ఈ స్థల వెడల్పు** (field width) అవుట్‌పుట్ నందు వ్రాయబడాల్సిన క్యారక్టర్ కనిష్ట సంఖ్యను నిర్ణయిస్తుంది

సింటాక్స్:

setw (క్యారక్టర్ల సంఖ్య)

ఉదాహరణ 9.10: నికర జీతాన్ని లెక్కించెడి ప్రోగ్రాం

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
        float basic, da, hra, gpf, tax, gross, np;
        char name[30];
        cout << "\n Enter Basic Pay: ";
        cin >> basic;
        cout << "\n Enter D.A : ";
        cin >> da;
        cout << "\n Enter H.R.A: ";
        cin >> hra;
        gross = basic+da+hra; // sum of basic, da and hra
        gpf = (basic+da) * 0.10; // 10% 0f basic and da
        tax = gross * 0.10; //10% of gross pay
        np = gross - (gpf+tax); //netpay = earnings - deductions
        cout << setw(25) << "Basic Pay : " << setw(10)<< basic<< endl;
        cout << setw(25) << "Dearness Allowance : "<< setw(10)<<da<< endl;
        cout<<setw(25)<<"House Rent Allowance : "<<setw(10)<< hra<<endl;
        cout << setw(25) << "Gross Pay : " << setw(10) << gross << endl;
        cout << setw(25) << "G.P.F : " << setw(10) << gpf << endl;
        cout << setw(25) << "Income Tax : " << setw(10)<< tax << endl;
        cout << setw(25) << "Net Pay : " << setw(10) << np << endl;
}
```

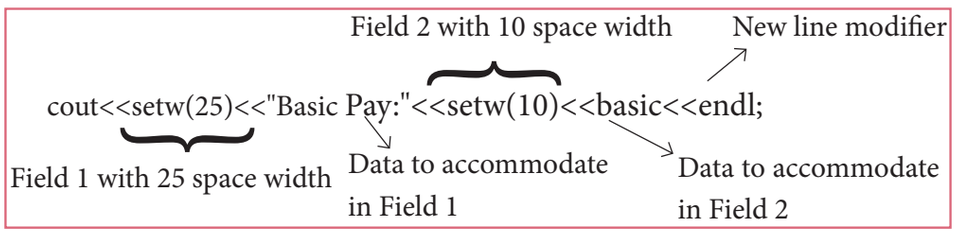

11th Computer Science_Chapter-9 Telugu.indd 149 16-03-2023 17:44:21

The output will be,

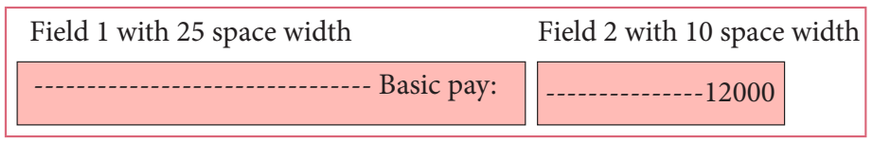
```
Enter Basic Pay: 12000
Enter D.A : 1250
Enter H.R.A : 1450
                    Basic Pay  :      12000
           Dearness Allowance  :       1250
       House Rent Allowance    :       1450
                    Gross Pay  :      14700
                        G.P.F  :       1325
                   Income Tax  :       1470
                      Net Pay  :      11905
```

(HOT: Try to make multiple output statements as a single cout statement)

పై ప్రోగ్రామ్‌నందు ప్రతి అవుట్‌పుట్ స్టేట్‌మెంట్ రెండు setw ( ) మానిపులేటర్స్‌ను కలిగి ఉన్నవి. మొదటి setw (25) స్పేసస్ గల స్థలమును (ఫీల్డ్‌ను) సృష్టిస్తుంది మరియు రెండవ setw(10) స్పేసస్ గల మరో స్థలమును సృష్టిస్తుంది. ఈ ఫీల్డ్ నందు ఏదైనా ఒక విలువకు ప్రాతినిధ్యం ఇచ్చినప్పుడు ఫీల్డ్‌లోపలే కుడి నుండి ఎడమకు కనిపిస్తుంది.



ఫీల్డ్ 1 మరియు 2 లలో Basic Pay కూర్పు అగును మరియు “Basic Pay” విలువను పటము 9.16 లో చూపబడినది.



పటము 9.16 setw( ) function

సెట్ ప్రెసిషన్ (setprecision) ( )

నిర్ధిష్టత సంఖ్య గల అంకెలను వాని భిన్నాలతో పాటు ప్రదర్శించడానికి ఇది ఉపయోగపడుతుంది.

సింటాక్స్:

సెట్ ప్రెసిషన్ (అంకెల యొక్క సంఖ్య)

ఉదాహరణ:

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{       float hra = 1200.123;
        cout << setprecision (5) << hra; }
```


11th Computer Science_Chapter-9 Telugu.indd 150 16-03-2023 17:44:21

పై కోడ్‌నందు ఇవ్వబడిన విలువ 1200.123 భిన్నముతో పాటు 5 అంకెలుగా చూపబడినది. కావున అవుట్‌పుట్ **1200.1** అగును.

సెట్ ప్రెసిషన్ ( ) ఈ విలువలను ఎడమ నుండి కుడికి ముద్రిస్తుంది. పై కోడ్ కోసం మొదటగా ఇది నాలుగు అంకెలను తీసుకొనును. మిగిలిన ఒక అంకెను భిన్న భాగము నుండి తీసుకొని ముద్రించును.

సెట్‌ప్రెసిషన్‌ను ప్రదర్శింపబడు సంఖ్య యొక్క దశాంశ స్థానమును నిర్ణయించుటకు కూడా ఉపయోగిస్తారు. ఈ కృత్యం చేయుట కొరకు ఒక ios ఫ్లాగ్ setf() లో పొందుపరచబడిన మానిపులేటర్స్ అందుబాటులో ఉంటుంది. దీనిని రెండు విధముగా ఉపయోగిస్తారు. అవి (i) **నిర్ణీత** (ii) **శాస్త్రీయ**

మూల పదములను నిర్ణయించినపుడు మాత్రమే ఈ రెండు విధములైన అంటే **నిర్ణీత** లేక **శాస్త్రీయ** సెట్ ప్రెసిషల్ మానిపులేటర్స్ ఉపయోగించబడుతాయి.

ఉదాహరణ:

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{       cout.setf(ios::fixed);
        cout << setprecision(2)<<0.1; }
```

పై ప్రోగ్రామ్‌నందు ios flag నిర్ణీత రకముగా ఉన్నది. ఇది స్వేచ్ఛా స్థాన సంఖ్యను **నిశ్చిత** సంకేతాలు (నొటేషన్) లతో ముద్రించబడుతుంది. కావున అవుట్‌పుట్ 0.10 గా ఉండును.

```
cout.setf(ios::scientific);
cout << setprecision(2) << 0.1;
```

పై స్టేట్‌మెంట్‌లలో ios flag శాస్త్రీయ రకముగా ఇవ్వబడినది. ఇది **స్వేచ్ఛా** స్థాన సంఖ్యలను శాస్త్రీయ సంకేతాలుగా ముద్రిస్తుంది. కావున అవుట్‌పుట్ 1.00e-001 గా ఉండును.

9.15 సమాసము (Expression)

సమాసము అనునది C++. నియమాలను అనుసరించి ఆపరేటర్స్ స్థిరరాశులు మరియు చరరాశు అమరికల సమ్మేళనము. ఇది విలువలను వెనుకకు తెచ్చు ఫంక్షన్ కాల్స్‌ను కూడా పొందుపరుస్తుంది. (ఫంక్షన్స్ గురించి రాబోవు అధ్యాయాలలో నేర్చుకొందాము).

సమాసము ఒకటి లేదా అంతకన్నా ఎక్కువ ఆపరెండ్స్‌ను కలిగి ఉంటుంది. మరియు శూన్య లేక మరిన్ని ఆపరేటర్స్ విలువలను ఇస్తాయి. C++. నందు 7 రకాల సమాసములు ఉన్నాయి.

అవి

(i) స్థిర సమాసము
(ii) పూర్ణాంక సమాసము
(iii) స్వేచ్ఛా సమాసము
(iv) సంబంధిత సమాసము
(v) తార్కిక సమాసము
(vi) బిట్ ప్రకార సమాసము
(vii) సూచిక సమాసము

11th Computer Science_Chapter-9 Telugu.indd 151 16-03-2023 17:44:21

వ.సం	సమాసము	వివరణ	ఉదాహరణ
1	స్థిర సమాసం	స్థిర సమాసాలు స్థిరరాశులను మాత్రమే కల్గి ఉంటాయి	int num=100;
2	పూర్ణాంక సమాసం	పూర్ణాంక విలువలను ఫలితముగా ఇవ్వగల పూర్ణాంక మరియు క్యారక్టర్ and/or చరరాశి విలువల సాధారణ అంకగణిత ఆపరేటర్‌ల సమ్మేళనం.	sum=num1+num2; avg=sum/5;
3	స్వేచ్ఛా సమాసం	స్వేచ్ఛా స్థాన విలువలను ఫలితముగా ఇవ్వగల and/or స్వేచ్ఛా స్థాన చరరాశి విలువల సాధారణ అంకగణిత ఆపరేటర్స్‌లో సమ్మేళనం.	Area=3.14*r*r;
4	సంబంధిత సమాసం	బూల్ విలువలను (సరి అంటే 1 మరియు తప్పు అంటే 0) ఫలితముగా ఇవ్వగల and/or చరరాశుల సంబంధిత ఆపరేటర్స్‌ల సమ్మేళనం	x>y; a+b==c+d;
5	తార్కిక సమాసం	బూల్ విలువలను ఫలితముగా ఇవ్వగల and/or చరరాశి విలువలు తార్కిక ఆపరేటర్స్‌ల సమ్మేళనం.	(a>b)&& (c==10);
6	బిట్ ప్రకార సమాసం	and/or చరరాశి విలువలు, బిట్ ప్రకార ఆపరేటర్స్‌ల సమ్మేళనం	x>>3; a<<2;
7	సూచిక సమాసం	సూచిక అనునది ఒక చరరాశి అడ్రసుని సూచిస్తుంది. ఈ సూచిక చరరాశులను (*) చిహ్నంతో ప్రకటిస్తారు.	int *ptr;

పట్టిక 9.10 Types of Expressions

9.16 రకం మార్పిడి (Type Conversion)

ఒక ప్రాథమిక దత్తాంశ రకమును మరో దత్తాంశ రకముగా మార్చు ప్రక్రియను రక మార్పిడి అని అంటారు. C++ రెండు రకముల మార్పిడులను అందిస్తున్నది..

(1) అవ్యక్త రకం మార్పిడి (Implicit type conversion)

(2) స్పష్టం రకం మార్పిడి (Explicit type conversion)

(1) అవ్యక్త రకం మార్పిడి (Implicit type conversion)

అవ్యక్త రకం మార్పిడి అను మార్పిడి కంపైలర్‌చే స్వయంచాలకంగా ప్రదర్శించబడునది. కావున ఈ అవ్యక్త రకం మార్పిడిని **ఆటోమేటిక్ కన్వర్షన్** అని కూడా పిలుస్తారు.

సాధారణంగా ఇలాంటి రకం మార్పిడులను సమాసము నందు వేర్వేరు దత్తాంశ రకాలు అంతర్లీనంగా కలిసిపోయినపుడు ప్రయోగిస్తారు. **Type Promotion** అనబడే స్వల్ప smaller రకమును విస్తృత wider రకముగా మార్చుచేయు నియమాన్ని అనుసరించి ఆపరెండ్లు వేర్వేరుగా ఉన్నట్లయితే కంపైలర్ ఒక దానిని ఇంకొకదానితో జత చేయడం కోసం మార్చుతుంది.

11th Computer Science_Chapter-9 Telugu.indd 152 16-03-2023 17:44:21

ఉదాహరణకు:

```
#include <iostream>
using namespace std;
int main()
{
        int a=6;
        float b=3.14;
        cout << a+b;
}
```

పై ప్రోగ్రామ్‌నందు ఆపరెండ్ **a** ఒక పూర్ణాంక రకము మరియు **b** ఒక స్వేచ్ఛా రకము. ప్రోగ్రామ్‌ను అమలు చేసేటప్పుడు int, float గా మార్చబడుతుంది. ఎందుకంటే float, int కన్నా విస్తృతమయినది. అందువలన పై ప్రోగ్రామ్ అవుట్‌పుట్ **9.14** అగును.

ఈ క్రింది పట్టిక 9.11 మార్పిడి క్రమమును చూపిస్తున్నది.

RHO / LHO	**char**	**short**	**int**	**long**	**float**	**double**	**long double**
char	int	int	int	long	float	double	long double
short	int	int	int	long	float	double	long double
int	int	int	int	long	float	double	long double
long	long	long	long	long	float	double	long double
float	float	float	float	float	float	double	long double
double	double	double	double	double	double	double	long double
long double	long double	long double	long double	long double	long double	long double	long double

RHO – కుడి చేతి ఆపరెండ్ LHO – ఎడమ చేతి ఆపరెండ్

పట్టిక 9.11 మిశ్రిత ఆపరెండ్ల అవ్యక్త మార్పిడి

(2) స్పష్ట రకం మార్పిడి (Explicit type conversion)

ప్రోగ్రామర్‌చే ఒక చరరాశి లేదా సమాసము ఒక దత్తాంశ రకము నుండి మరొక నిర్ధిష్ట దత్తాంశ రకముగా స్పష్ట మార్పిడి చేయుటకు C++ అనుమతిస్తుంది. దీనినే "**type casting**" అని పిలుస్తారు.

సింటాక్స్

(type-name) expression;

ఇక్కడ (type-name) అనునది చెల్లుబాటయ్యే దత్తాంశ రకము ఐడియో మార్పిడికి కావల్సినది

ఉదాహరణ:

```
#include <iostream>
using namespace std;
```


11th Computer Science_Chapter-9 Telugu.indd 153 16-03-2023 17:44:21

```
int main( )
{
        float varf=78.685;
        cout << (int) varf;
}
```

పై ప్రోగ్రామ్ నందు చరరాశి **varf** ప్రారంభ విలువ, 78.685 గా గల ఒక **స్వేచ్ఛా రకముగా** ప్రకటించబడినది. **varf** యొక్క విలువ cout స్టేట్‌మెంట్‌లో స్పష్టముగా **int** రకముగా మార్చబడియున్నది. కావున ఆఖరిలో అవుట్‌పుట్ 78 అగును.

స్పష్ట మార్పిడి జరిగేటప్పుడు అధిక వ్యాప్తి గల రకమునకు విలువను కేటాయించినప్పుడు ఇది ఎలాంటి ఇబ్బందిని కల్గించదు. కావున స్వల్ప రకమునకు భారీ విలువలను కేటాయించినపుడు ఫలితములో ఖచ్చితమైన విలువలను పొందలేము (నష్టపోతాము)

వ.సం	స్పష్ట రక మార్పిడి	ఇబ్బంది
1	ద్వి రకము నుండి స్వేచ్ఛా రకమునకు	ఖచ్చితత్వమును పొందలేము అసలు విలువ లక్ష్య రకపు పరిధిని అధిగమించి ఫలితము నిర్వచింపలేనిగా మారుతుంది.
2	స్వేచ్ఛా రకము నుండి పూర్ణాంక రకమునకు	భిన్న భాగమును కోల్పోతాము అసలు విలువ లక్ష్య రకపు పరిధిలో ఉండకపోవచ్చు. ఫలితం అనిర్వచితం.
3	దీర్ఘ రకము నుండి హ్రస్వ రకమునకు	దత్తాంశమును కోల్పోతాము.

పట్టిక 9.12 స్పష్ట రక మార్పిడినందు ఇబ్బందులు

ఉదాహరణ

```
#include <iostream>
using namespace std;
int main()
{
        double varf=178.25255685;
        cout << (float) varf << endl;
        cout << (int) varf << endl;
}
```

Output:

178.253
178

స్వయం మూల్యాంకనం

1. రకం మార్పిడి అంటే అర్థమేమిటి?
2. అవ్యక్త రక మార్పిడి స్పష్ట రక మార్పిడితో ఎలా భేదపడుతుంది?
3. endl మరియు \n ల మధ్య భేదమేమి?
4. ఆధారాలు (references) ల ఉపయోగమేమి?
5. సెట్ ప్రెసిషన్ ( ) యొక్క ఉపయోగమేమి? –

11th Computer Science_Chapter-9 Telugu.indd 154 16-03-2023 17:44:21

అభ్యాసము (Hands on practice)



1. రెండు చరరాశుల విలువలను బదిలీ చేయుటకు తగు C++ ప్రోగ్రామ్‌ను వ్రాయుము.
   a. మూడవ చరరాశిని ఉపయోగించి
   b. మూడవ చరరాశిని ఉపయోగించకుండా
2. ఈ క్రింది వానిని సాధించుటకు C++ ప్రోగ్రామ్‌లను రాయుము:
   a. చతుర్భుజం యొక్క చుట్టుకొలత మరియు వైశాల్యం కనుగొనుము
   b. త్రిభుజం యొక్క వైశాల్యమును కనుగొనుము.
   c. ఉష్ణోగ్రతను సెల్సియస్ నుండి పారన్ హీట్లలోనికి మార్చుము
3. నీవు పదవ తరగతి పబ్లిక్ పరీక్షలో పొందిన మొత్తం మార్కులు మరియు శాతమును కనుగొనుటకు తగు C++ ప్రోగ్రామ్ రాయుము. అన్నీ మార్కులు ఒకదాని తర్వాత మరొకటి వాటి మొత్తం మరియు శాతములను formatting functions లను ఉపయోగించి తెరపై చూపించుము.

గుర్తుంచుకోవాల్సిన అంశాలు

- ప్రతి ప్రోగ్రామింగ్ భాష మరియు రెండు ప్రాథమిక అంశాలు ఉంటాయి. అవి దత్తాంశ రకములు మరియు చరరాశులు.
- C++ నందు గల దత్తాంశ రకములు మూడు రకాలుగా వర్గీకరించబడినది. (1) స్వనిర్మిత దత్తాంశ రకము (2) వినియోగదారు – నిర్వచిత దత్తాంశ రకము (3) ఉత్పాదిత దత్తాంశ రకములు
- చరరాశులు అనునవి కొన్ని విలువలను సూచించు name space అవుతుంది.
- C++ నందు ఐదు ప్రాథమిక దత్తాంశ రకములు కలవు. char, int, float, double
- కంపైలర్ ప్రమాణాలను అనుసరించి ప్రతి ఒక అమలు చేయబడిన దత్తాంశానికి C++ కంపైలర్ మెమరీ నందు నిర్దిష్ట స్థలమును కేటాయిస్తుంది.
- చరరాశులు అనునవి విలువలను నిల్వచేయు మెమరీ స్థానాలకు వినియోగదారు నిర్వచించిన పేర్లగును.
- ప్రకటించబడుట అను ప్రక్రియ కంపైలర్‌కు నిర్దేశించబడిన రకము మరియు చరరాశి పేరుకు మెమరీ నందు తగు స్థలమును కేటాయించమని అదేశించడం అగును.
- ఏ C++ ప్రోగ్రామ్ యొక్క అవుట్‌పుట్‌ను format చేయుటకైనా మానిపులేటర్స్ ఉపయోగపడుతాయి. (<<) మరియు (>>) వంటి ఆపరేటర్స్‌ను ఉపయోగించుటకు ప్రత్యేకముగా రూపకల్పన చేయబడిన ఫంక్షన్స్ మానిపులేటర్స్ అగును.
- సమాసము అనునది C++ ప్రోగ్రామ్ నియమాలను అనుసరించి అమర్చబడిన ఆపరేటర్స్, స్థిరరాశులు మరియు చరరాశుల సమ్మేళనం.
- ఒక ప్రాథమిక దత్తాంశ రకమును మరో దత్తాంశ రకముగా మార్పిడి చేయు ప్రక్రియను రకం మార్పిడి అంటారు. C++ రెండు రకాల మార్పిడులను ఇస్తున్నది (1) అవ్యక్త మార్పిడి (2) స్పష్ట రక మార్పిడి.

11th Computer Science_Chapter-9 Telugu.indd 155 16-03-2023 17:44:21

మూల్యాంకనం

భాగము - అ

సరియైన సమాధానమును ఎన్నుకొనుము

1. C++ నందు ఎన్ని వర్గాల దత్తాంశ రకములు అందుబాటులో ఉన్నాయి?
   (అ) 5 (ఆ) 4 (ఇ) 3 (ఈ) 2
2. ఈ క్రింది దత్తాంశ రకాలలో ప్రాథమిక రకము కానిదేది?
   (అ) సంజ్ఞాత్మక (ఆ) పూర్ణాంక (ఇ) స్వేచ్ఛా (ఈ) క్యారక్టర్
3. ఈ క్రింది స్టేట్‌మెంట్ యొక్క ఫలితం ఏదైయుండును.
   char ch= 'B';
   cout << (int) ch;
   (అ) B (ఆ) b (ఇ) 65 (ఈ) 66
4. స్వేచ్ఛా స్థాన విలువను సూచించుటకు ఏ క్యారక్టర్‌ను పూర్వపదంగా ఉపయోగిస్తారు?
   (అ) F (ఆ) C (ఇ) L (ఈ) D
5. మీరు Dev C++ ను ఉపయోగించినట్లయితే ఈ క్రింది చరరాశి ప్రకటనకు మెమరీనందు ఎన్ని బైట్‌లు కేటాయించబడతాయి. Dev C++? short int x;
   (అ) 2 (ఆ) 4 (ఇ) 6 (ఈ) 8
6. ఈ క్రింది స్నిప్పెట్ (లఘు ప్రోగ్రామ్) యొక్క అవుట్‌పుట్ ఏది?
   char ch = 'A';
   ch = ch + 1;
   (అ) B (ఆ) A1 (ఇ) F (ఈ) 1A
7. ఈ క్రింది వానిలో దత్తాంశ రకపు మాడిఫైయర్ కానిదేది?
   (అ) సంజ్ఞాత్మక (ఆ) పూర్ణాంక (ఇ) దీర్ఘ (ఈ) హ్రస్వ
8. ఈ క్రింది ఆపరేటర్‌లో దత్తాంశ రకము యొక్క పరిమాణాన్ని వెనుకకు తెచ్చునది
   (అ) sizeof( ) (ఆ) int( ) (ఇ) long ( ) (ఈ) double ( )
9. చరరాశి యొక్క సూచికను ఆక్సెస్ చేయడానికి ఏ ఆపరేటర్ ఉపయోగపడుతుంది.
   (అ) $ (ఆ) # (ఇ) & (ఈ) !
10. endl కమాండ్‌కు మారుగా ఉపయోగించునది.
    (అ) \t (ఆ) \b (ఇ) \0 (ఈ) \n

భాగము - ఆ

సంక్షిప్తముగా సమాధానములను

1. const keyword (స్థిరరాశి – మూలపదం) గురించి తగు ఉదాహరణలతో లఘు వ్యాఖ్య వ్రాయండి.
2. setw( ) విధానపు మానిపులేటర్ యొక్క ఉపయోగమేమి?
3. char (క్యారక్టర్) దత్తాంశ రకమును పూర్ణాంక రకముగా పరిగణించుటకు గల కారణమేమి?
4. రెఫరెన్స్ చరరాశి అనగానేమి? దీని ఉపయోగమేమి?

11th Computer Science_Chapter-9 Telugu.indd 156 16-03-2023 17:44:21

5. ఈ క్రిందిది C++ స్టేట్‌మెంట్‌ను పరిగణించుము ఇవి సమానమైనవా?
char ch = 67; char ch = 'C';

6. 56L మరియు 56 ల మధ్య బేధమేమి?

7. ఈ క్రింది వానిలో చెల్లుబాటయ్యే (valid) స్థిరరాశిలను గుర్తించండి? మరియు వాని రకమును సూచించండి.

(i) 0.5 (ii) 'Name' (iii) '\t' (iv) 27,822

8. x మరియు y లు రెండు ద్వి రకపు (double type) చరరాశులు అయితే వీనికి పూర్ణాంకం 1 ని కూడి పూర్ణాంక చరరాశిగా మార్చుటకు తగు C++ స్టేట్‌మెంట్‌ను రాయుము.

9. ప్రారంభంలో num=6 గా ఉంటే ఈ క్రింది వాని ఫలితమేమి?

(అ) cout << num;

(ఆ) cout << (num==5);

10. ఈ క్రింది రెండు స్టేట్‌మెంట్‌లో చెల్లుబాటయ్యేవి ఏవి? ఎందుకు? మరియు వాని ఫలితాలను రాయుము.

(i) int a; a = 3,014; (ii) int a; a=(3,014);

భాగము - ఇ

క్లుప్తముగా జవాబులిమ్ము

1. C++ నందు గల అంకగణిత ఆపరేటర్స్ ఏవి? యూనరీ మరియు బైనరీ అంకగణిత ఉపరేటర్స్ మధ్య భేధములను వ్రాయుము. ప్రతి దానికి ఉదాహరణలిమ్ము.

2. సంబంధిత ఆపరేటర్స్ మరియు తార్కిక ఆపరేటర్స్ ఒకదానితో ఒకటి ఎలా సంబంధించబడి ఉన్నాయి?

3. ఈ క్రింది C++ సమాసాలను మూల్యాంకనం చేయండి? ఇక్కడ x, y, z లు పూర్ణాంకాలు మరియు m, n స్వేచ్ఛా స్థాన సంఖ్యలు x = 5, y = 4 మరియు m=2.5;

(i) n = x + y / x;

(ii) z = m * x + y;

(iii) z *= x * m + x;

ఆధార గ్రంధాలు (Reference)

(1) Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.

(2) The Complete Reference C++ (Forth Edition), Herbert Schildt. Mc.Graw Hills.

(3) Computer Science with C++ (A text book of CBSE XI and XII), Sumita Arora, Dhanpat Rai & Co.

11th Computer Science_Chapter-9 Telugu.indd 157 16-03-2023 17:44:21

ఫ్లో ఆఫ్ కంట్రోల్
(Flow of Control)

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును అభ్యసించిన తరువాత విద్యార్థులు నేర్చుకొను అంశములు.

- వివిధ రకములైన స్టేట్‌మెంట్‌ల గురించి అర్థము చేసుకొనుట.
- C++ లోని వివిధ రకాల నియంత్రణ ప్రవాహపు స్టేట్‌మెంట్‌ను నిర్మించుట

10.1 పరిచయం

C++ ప్రోగ్రామింగ్‌లోని చలరాశులు, స్థిరాంకాలు, ప్రక్రియలు, దత్తాంశ రకములు మొదలగు ప్రాథమిక అంశములు గూర్చి ముందు పాఠ్యాంశములో నేర్చుకున్నాము. సాధారణంగా ఒక ప్రోగ్రామ్ యొక్క స్టేట్‌మెంట్‌లు ప్రారంభ స్థితి నుండి చివరి వరకు ఒకటి తరువాత ఒకటి అమలు పరచబడును. అయినను ఈ రకమైన వరుస క్రమము అనునది నియంత్రించబడినది మరియు కొంత ఉపయోగకరముగా నుండును. అనేక పరిస్థితులలో నిర్దిష్ట కండిషనల్‌ల ఆధారముగా అమలు చేయబడిన కోడ్ బ్లాక్‌ను నిర్ణయించడం ఉపయోగకరంగా ఉండే సందర్భాలు చాలా ఉన్నాయి. ఇలాంటి పరిస్థితిలో నియంత్రణ ప్రవాహం కోడ్ యొక్క ఒక విభాగము నుండి మరొక విభాగమునకు జంప్ చేయబడుతుంది. అటువంటి జంప్‌లకు కారణమైన ప్రోగ్రామ్ స్టేట్‌మెంట్‌లను కంట్రోల్ ఫ్లో అందురు. ఈ అధ్యాయము "సెక్షన్" "ఇటిరేషన్" మరియు "జంప్" స్టేట్‌మెంట్ వంటి ప్రాథమిక అంశాలతో వ్యవహరిస్తుంది.

10.2 స్టేట్‌మెంట్స్ (Statements)

కంప్యూటర్ ప్రోగ్రామ్ అనునది ఒక నిర్దిష్ట పనిని నిర్వహించడానికి స్టేట్‌మెంట్ లేదా ఇన్‌స్ట్రక్షన్‌ల సమితి అగును. ఈ స్టేట్‌మెంట్‌లు ఒక నిర్దిష్ట చర్య జరుగుటను ఉద్దేశించబడ్డాయి. చర్య అనునది వేరియబుల్ డిక్లరేషన్, ఎక్స్‌ప్రషన్, ఎవాల్యుయేషన్, అసైన్‌మెంట్, ఆపరేషన్‌లు, నిర్ణయం తీసుకోవడం, లూపింగ్ మొదలైనవి కావచ్చు.

C++ ప్రోగ్రామ్‌లో రెండు రకములైన స్టేట్‌మెంట్‌లు ఉపయోగించబడుచున్నవి.

(i) నల్ స్టేట్‌మెంట్

(ii) కాంపౌండ్ స్టేట్‌మెంట్:

10.2.1 నల్ స్టేట్‌మెంట్(Null statement)

నల్ లేదా శూన్య స్టేట్‌మెంట్ అనేది సెమికోలన్‌ను మాత్రమే కలిగి ఉన్న స్టేట్‌మెంట్. ఇది క్రింది రూపమును తీసుకొంటుంది.

```
; // ఇది ఒక నల్ స్టేట్‌మెంట్
```

నల్ స్టేట్‌మెంట్‌లు సాధారణంగా పునరావృత స్టేట్‌మెంట్‌లలో ప్లేస్ హోల్డర్ (placeholder) గా లేదా కాంపౌండ్ స్టేట్‌మెంట్‌లు లేదా ఫంక్షన్‌ల చివరిలో లేబుల్‌లను ఉంచే స్టేట్‌మెంట్‌లుగా ఉపయోగించబడతాయి.

10.2.2 కాంపౌండ్ స్టేట్‌మెంట్ (Compound (Block) statement)

C++ pair of braces {} ద్వారా జత చేయబడిన స్టేట్‌మెంట్‌ల సమూహాన్ని అనుమతిస్తుంది. ఈ స్టేట్‌మెంట్‌ల సమూహాన్ని కాంపౌండ్ స్టేట్‌మెంట్ లేదా బ్లాక్ అందురు.

కాంపౌండ్ స్టేట్‌మెంట్ యొక్క సాధారణ రూపము:

```
{
    ప్రతిపాదము 1;
    ప్రతిపాదము 2;
    ప్రతిపాదము 3;
}
```


11th Computer Science_Chapter-10 Telugu.indd 158 30-03-2023 18:47:07

ఉదాహరణకు

```
{
        int x, y;
        x = 10;
        y = x + 10;
}
```

కాంపౌండ్ స్టేట్‌మెంట్ లేదా బ్లాక్ అనేది ఒకే యూనిట్‌గా పరిగణింపబడుచున్నది మరియు ఇది ప్రోగ్రామ్‌లో ఎచ్చటనైన కనిపించవచ్చు.

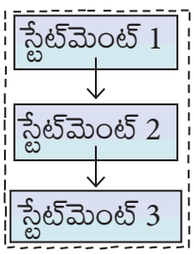
10.3 కంట్రోల్ స్టేట్‌మెంట్‌లు (Control Statements)

కంట్రోల్ స్టేట్‌మెంట్‌లు సూచనల ప్రవాహ క్రమాన్ని మార్చే స్టేట్‌మెంట్స్. ఒక ప్రోగ్రామ్‌లోని స్టేట్‌మెంట్ వరుసగా ఎంపిక లేదా పునరావృతంగా అమలుచేయబడుతాయి.

ప్రతి ప్రోగ్రామింగ్ భాషల క్రమము, ఎంపిక మరియు పునరావృతానికి మద్ధతు ఇవ్వడానికి స్టేట్‌మెంట్‌ను అందిస్తాయి.

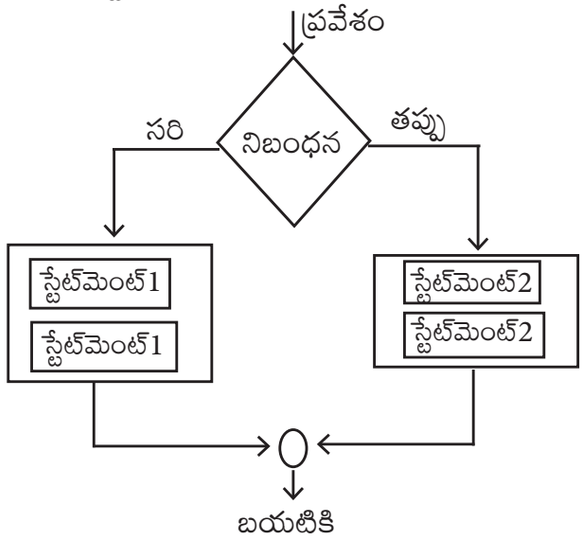
స్టేట్‌మెంట్‌లు వరుసగా అమలుచేయబడితే ఆ ప్రవాహాన్ని సీక్వెన్సీ ఫ్లో అంటారు. కొన్ని సందర్భాలలో స్టేట్‌మెంట్‌లు బ్రాంచింగ్ ఇటరేషన్ జంపింగ్ మరియు ఫంక్షన్ కాల్‌ల వంటి అమలు ప్రవాహాన్ని మార్చినట్లు అయితే ఈ ప్రవాహంను కంట్రోల్ ఫ్లో అంటారు.

సీక్వెన్స్ స్టేట్‌మెంట్



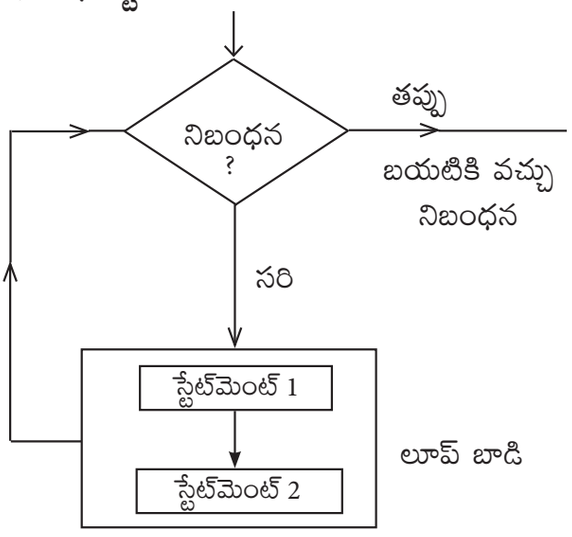
స్వీకెన్స్ స్టేట్‌మెంట్ అనునది స్టేట్‌మెంట్‌లు. అవి ఒకదాని తరువాత ఒకటి పై నుండి క్రిందికి ఒకసారి మాత్రమే అమలుచేయబడుతుంది. ఈ స్టేట్‌మెంట్ అమలు ప్రవాహంను మార్చదు. ఈ స్టేట్‌మెంట్‌లను సీక్వెన్షియల్ ఫ్లో స్టేట్‌మెంట్ అందురు. ఇది ఎల్లప్పుడు (;) సెమికోలన్ తో ముగుస్తుంది.

సెలక్షన్ స్టేట్‌మెంట్ (Selection statement)



ఎంపిక స్టేట్‌మెంట్ అంటే అమలుచేయబడిన స్టేట్‌మెంట్ షరతుపైన ఆధారపడి ఉంటుంది. కండీషన్ నిజం అయితే నిజమైన బ్లాక్ అమలుచేయబడుతుంది. లేకపోతే తప్పుడు బ్లాక్ అమలుచేయబడుతుంది. ఈ స్టేట్‌మెంట్‌ను డెషిషన్ స్టేట్‌మెంట్ లేదా సెలక్షన్ స్టేట్‌మెంట్ అని కూడా పిలుస్తారు. ఎందుకంటే ఇది ఏ స్టేట్‌మెంట్ సెట్‌ను ఎక్స్‌క్యూట్ చేయాలో నిర్ణయం తీసుకోవడంలో సహాయపడును.

ఇటిరేషన్ స్టేట్‌మెంట్



ఇటిరేషన్ స్టేట్‌మెంట్ అనునది ఒక ఆదేశాల సమితి కండీషన్‌ల ఆధారంగా మరల మరల నెరవేర్చబడును. ఇవ్వబడిన కండీషన్ సరి అని ఉండు వరకు, ఆదేశ సమితి మరలా మరలా నెరవేర్చబడును. కండిషన్ తప్పు అయితే కొనసాగుట

11th Computer Science_Chapter-10 Telugu.indd 159 30-03-2023 18:47:07

అగుతుంది. దీనిని కండీషన్ స్టేట్‌మెంట్ లేదా లూప్ స్టేట్‌మెంట్ అని అందురు. పైన ఇవ్వబడిన ప్రవాహపు పటము ఇటిరేషన్ స్టేట్‌మెంట్‌ను వివరించుచున్నది. మరలా మరలా నెరవేర్చబడు కండీషన్‌ల సమితిని లూప్ యొక్క శరీర భాగము అని అందురు. ఒక లూప్‌లోని ప్రవేశము లేదా బయటికి వచ్చుటకు గల కారణమగు కండిషన్‌ను బయటికి వచ్చు కండీషన్ లేదా టెస్ట్ కండీషన్ అని అందుము.

సాధారణముగా అన్ని ప్రోగ్రామింగ్ భాషలు సమస్యను బట్టి ప్రోగ్రామ్‌లను రాయడానికి ఈ రకమైన స్టేట్‌మెంట్‌లను మద్దతునిస్తుంది. C++ కూడా ఈ రకపు స్టేట్‌మెంట్‌లకు మద్దతిస్తుంది. ఈ స్టేట్‌మెంట్‌లు రాబోయే విభాగాలలో చర్చించబడతాయి.

గమనిక

C++ లో ప్రతికూల సంఖ్యలతో సహా సున్న కానిది ఏదైన నిజమైనదిగా పరిగణించబడుతుంది. మరియు సున్న తప్పుగా పరిగణింపబడుతుంది.

షరతులతో కూడిన వ్యక్తీకరణపై ఆధారపడి సెలక్షన్ స్టేట్‌మెంట్‌లు మరియు పునరావృత స్టేట్‌మెంట్‌లు అమలుచేయబడుతాయి. షరతులతో కూడిన వ్యక్తీకరణ నిజం లేదా తప్పుగా అంచనావేస్తుంది.

10.4 సెలక్షన్ స్టేట్‌మెంట్ (Selection statements)

ప్రోగ్రామ్‌లో ఒక నిర్ణయం ప్రోగ్రామ్‌లోని వేరే భాగానికి ఒకేసారి వేళ్ళేలా చేస్తుంది. C++ లో నిర్ణయాలు అనేక విధాలుగా తీసుకోబడుతాయి. ముఖ్యంగా if తో ఇది రెండు ప్రత్యామ్నాయల మధ్య ఎంచుకోబడుతుంది. మరొక నిర్ణయ స్టేట్‌మెంట్ స్విచ్ ఒకే వేరియబుల్ విలువను బట్టి కోడ్ యొక్క బహుళ ప్రత్యామ్నాయ విభాగాల కోసం శాఖలను సృష్టిస్తుంది.

10.4.1 'if' స్టేట్‌మెంట్

if స్టేట్‌మెంట్ అనునది ఒక తీర్మానించు ప్రతిపాదమగును. **if** ప్రతిపాద నిబంధనను పరిశోధించు నిబంధన సరి అయినచో సరి అగు భాగము స్టేట్‌మెంట్ లేదా ఆదేశాల సమితిని నెరవేర్చబడును. లేనిచో సరి అను భాగము వదలివేయబడును. **if** స్టేట్‌మెంట్ యొక్క సామాన్య వాక్య నిర్మాణము syntax

```
if (expression) నిబంధన సమాసము
        true-block అగు భాగము;
statement-x;
```

ఉదాహరణ 10.1 if స్టేట్‌మెంట్ ఉపయోగించి ఒక వ్యక్తి ఓటు వేయడానికి అర్హుడా అని పరిశోధించి C++ ప్రోగ్రామ్‌ను వ్రాయుము

```
#include <iostream>
using namespace std;
int main()
{
        int age;
        cout<< "\n Enter your age: ";
        cin>> age;
        if(age>=18)
                cout<< "\n You are eligible for voting ....";
        cout<< "This statement is always executed.";
        return 0;
}
```

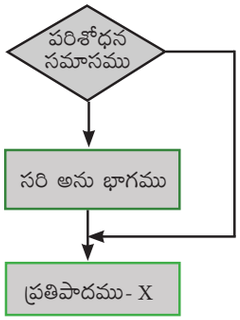
The pair of braces is not required because if condition followed by only one statement

Output

```
Enter your age: 23
You are eligible for voting….
This statement is always executed.
```


11th Computer Science_Chapter-10 Telugu.indd 160 30-03-2023 18:47:07

పై వాక్య నిర్మాణములో **if** అనునది కూడలీ కరణంలోని వ్యక్తీకరణ లేదా సమాసము లేదా కండీషన్ ఉండే కీలక పదం కప్పబడి ఉండును. ఇవ్వబడిన సమాసము యొక్క విలువ సరి అయినచో (శూన్యముకాని) సరి అను భాగము నెరవేర్చబడును. దానిని వెంబడించి x - స్టేట్‌మెంట్‌ను తీసుకొనిపోవును, ఒక సరిభాగము ఏక స్టేట్‌మెంట్, సమ్మేళన స్టేట్‌మెంట్ లేదా ఖాళీ స్టేట్‌మెంట్ కలిగియుండును.

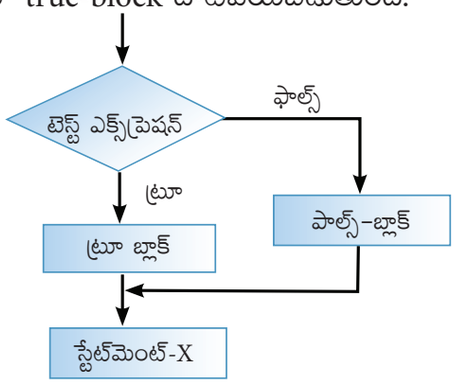


10.4.2 if-else స్టేట్‌మెంట్

if యొక్క పై ఉదాహరణలో మీరు చూసినట్లుగా condition నిజమని మూల్యాంకనం అయితే మాత్రం statement block అమలు చేయబడుతుంది. ఒకవేళ condition తప్పు అని మూల్యాంకనం అయితే అనుసరించాల్సిన చర్య మరొకటి ఉంటే ఏమి చేయాలి? వేరే it caluse condition ను అందించడం ద్వారా ఈ రకమైన లేదా condition ను అనుమతించినయితే మరొక రూపం ఉంటుంది. if-else syntax క్రింద ఇవ్వబడినది.

```
if ( expression)
{
        True-block;
}
else
{
        False-block;
}
Statement-x
```

if-else statement లో మొదట వ్యక్తీకరణ లేదా condition తప్పు లేదా నిజమైనదిగా అంచనా వేయబడుతుంది ఫలితం నిజమైతే true-block అమలు చేయబడుతాయి. మరియు false-block దాటివేయబడుతాయి. ఫలితం తప్పు అయితే false-block లోపల statement అమలు చేయబడుతుంది. అంటే true-block దాటివేయబడుతుంది.



ఉదాహరణ 10.2 : if-else స్టేట్‌మెంట్‌ను ఉపయోగించి ఇవ్వబడిన సంఖ్య బేసి సంఖ్య లేదా సరి సంఖ్య అని కనుగొనుము C++ ప్రోగ్రామ్‌లో వ్రాయుము

```
#include <iostream>
using namespace std;
int main()
{
        int num, rem;
        cout<< "\n Enter a number: ";
        cin>>num;
        rem = num % 2;
        if (rem==0)
                cout<< "\n The given number" <<num<< " is Even";
        else
                cout<< "\n The given number "<<num<< " is Odd";
        return 0;
}
```

Output

```
Enter number: 10
The given number 10 is Even
```


11th Computer Science_Chapter-10 Telugu.indd 161 30-03-2023 18:47:07

పైనున్న ప్రోగ్రామ్‌లో ఇవ్వబడిన సంఖ్య శేషము 'rem' అను చలరాశిని నిల్వచేసుకొని, 'rem' యొక్క విలువ శూన్యము అయిన ఇవ్వబడిన సంఖ్య సరి సంఖ్య అని విలువనిచ్చును; లేనిచో బేసి సంఖ్య అని విలువనిచ్చును.

10.4.3 Nested if

ఒక if statement మరొక if statement ను కలిగియుంటే దానిని nested if అని అంటారు. nested if క్రింది మూడు రూపములో ఏదైన ఒకదానిని కలిగియుండును. అవి:

1. If nested inside if part
2. If nested inside else part
3. If nested inside both if part and else part

The syntax of the nested if:

If nested inside if part

```
if (expression-1)
{
    if (expression-2)
    {
     True_Part_Statements;
    }
    else
    {
     False_Part_Statements;
    }
}
else
     body of else part;
```

If nested inside else part

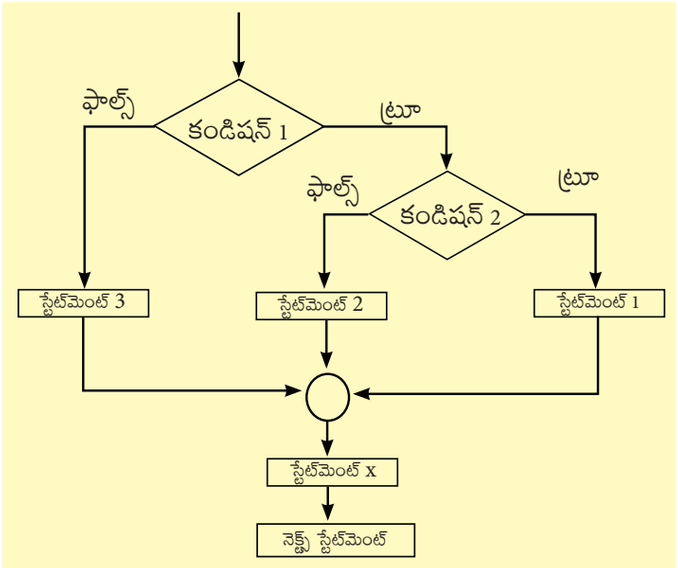
```
if (expression-1)
{
     body of true part;
}
else
{
     if (expression-2)
     {
      True_Part_Statements;
     }
     else
     {
      False_Part_Statements;
     }
}
```

If nested inside both if part and else part

```
if (expression)
{
     if (expression)
     {
      True_Part_Statements;
     }
     else
     {
      False_Part_Statements;
     }
}
else
{
     if (expression)
     {
      True_Part_Statements;
     }
     else
     {
      False_Part_Statements;
     }
}
```

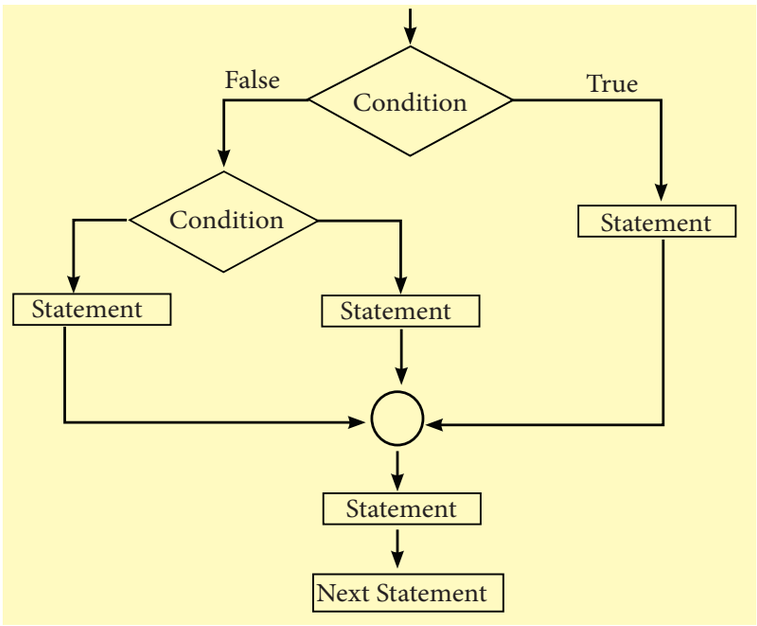
పైన పేర్కొనిన 1వ సమాసంలో nested if యొక్క మొదటి syntax మూల్యాంకనం చేయబడినపుడు సమాస ఫలితము తప్పుగా ఉండినచో. కంట్రోల్ తర్వాత స్టేట్‌మెంట్ -m కు పాస్ అగును. ఒకవేళ 2వ సమాసం మూల్యాంకనం చేయబడి, కండిషన్ "true" గా ఉన్నట్లయితే Nested-True-block తొలగింపబడును. తర్వాతి స్టేట్‌మెంట్ -n కూడా తొలగింపబడును. లేకుంటే Nested-False-Block స్టేట్‌మెంట్ -n మరియు స్టేట్‌మెంట్ -m తొలగింపబడును.

పైన పేర్కొనిన if..else structures యొక్క ప్రక్రియ విధానము క్రింది ఫ్లో చార్టులో ఇవ్వబడింది.

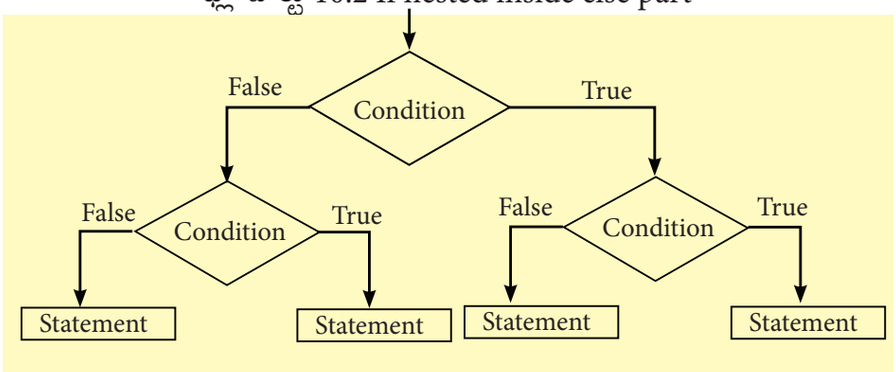


ఫ్లో చార్టు 10.1 if nested inside if Part

11th Computer Science_Chapter-10 Telugu.indd 162 30-03-2023 18:47:07



ఫ్లో చార్టు 10.2 If nested inside else part



ఫ్లో చార్టు 10.3 If part మరియు else part రెండింటిలోను if nested

ఉదాహరణ 10.3 Nested if Statement ను ఉపయోగించి గ్రేడ్ ఆధారిత కమిషన్ లెక్కించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main()
{
        int sales, commission;
        char grade;
        cout << "\n Enter Sales amount: ";
        cin >> sales;
        cout << "\n Enter Grade: ";
        cin >> grade;
        if (sales > 5000)
        {
                commission = sales * 0.10;
                cout << "\n Commission: " << commission;
        }
```


11th Computer Science_Chapter-10 Telugu.indd 163 30-03-2023 18:47:07

```
        else
        {
                commission = sales * 0.05;
                cout << "\n Commission: " << commission;
        }
        cout << "\n Good Job ..... ";
        return 0;
}
```

Output:

```
Enter Sales amount: 6000
Enter Grade: A
Commission: 600
Good Job .....
```

10.4.4 if -else-if ladder

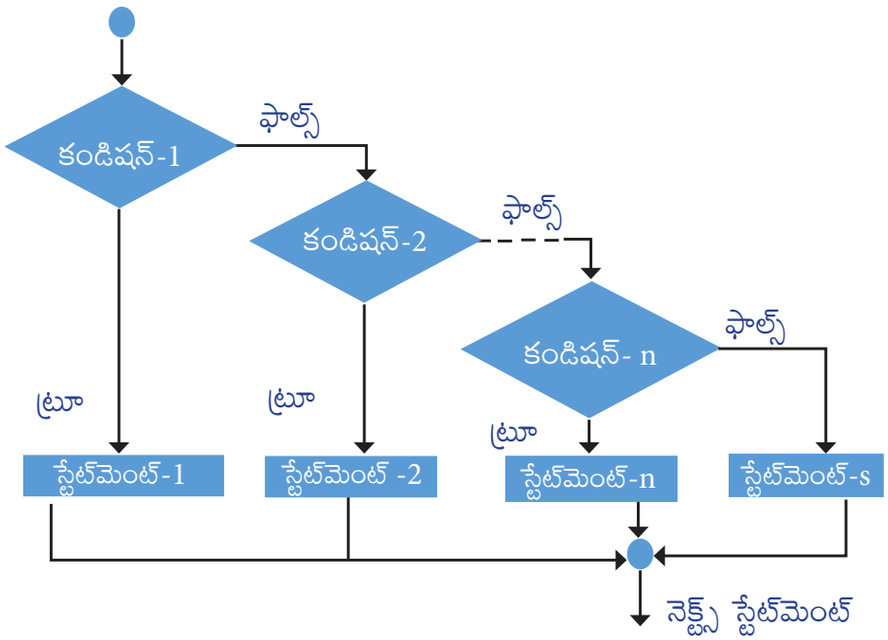
if-else ladder అనేది multi-path నిర్ణయం తీసుకొనే statement. ఈ రకమైన statement లో **if** statements అయితే ఒకటి లేదా అంతకంటే ఎక్కువ అనుసరించబడి చివరికి వేరే statements తో ముగుస్తుంది.

The syntax of if-else ladder:

```
if (expression 1)
{
  Statement-1
}
else
        if( expression 2)
        {
                Statement-2
        }
        else
                if ( expression 3)
                {
                        Statement-3
                }
                else
                {
                        Statement-4
                }
```

సంబంధిత expression నిజం అయితే block తో అనుబంధించిన statement అమలు చేయబడుతుంది. మరియు మిగిలిన ladder bypassed చేయబడుతుంది. Condition ఏది నిజం కాకపోతే చివరి statement అమలు చేయబడుతుంది.

11th Computer Science_Chapter-10 Telugu.indd 164 30-03-2023 18:47:07



ఫ్లో చార్టు 10.4 if-else ladder ఫ్లో చార్టు

ఉదాహరణ 10.4 If-else ladder ను ఉపయోగించి మీ గ్రేడ్‌ను కనుగొనెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main ()
{
int marks;
cout<<" Enter the Marks  :";
cin>>marks;
if( marks >= 60 )
        cout<< "Your grade is 1st class !!" <<endl;
                else if( marks >= 50 && marks < 60)
                        cout<< "your grade is 2nd class !!" <<endl;
                                else if( marks >= 40 && marks < 50)
                                        cout<< "your grade is 3rd class !!" <<endl;
else
        cout<< "You are fail !!" <<endl;
return 0;
}
```

Output

```
Enter the Marks  :60
Your grade is 1st class !!
```

60 కంటే ఎక్కువ లేదా సమానంగా ఉన్నప్పుడు **"Your grade is 1st class !!"** అని ప్రదర్శించబడుతుంది. మరియు మిగిలిన ladder bypassed చేయబడుతుంది. marks 50 మరియు 59 ల మధ్య ఉన్నప్పుడు **"Your grade is 2nd class !!"** అని ప్రదర్శించబడుతుంది. మరియు ఇతర ladder bypass చేయబడుతుంది. 40 నుండి 49 mark మధ్యలో ఉన్నప్పుడు **"Your grade is 3rd class !!"** ప్రదర్శించబడుతుంది. లేకపోతే "You are fail !!" అని ప్రదర్శించబడుతుంది.

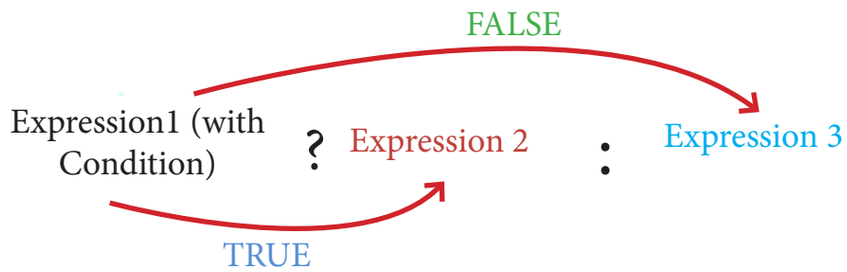
11th Computer Science_Chapter-10 Telugu.indd 165 30-03-2023 18:47:07

10.4.5 The ?: Alternative to if- else

conditional operator (or Ternary operator) అనేది if else statement కు ప్రత్యామ్నాయం. రెండు చిహ్నాలను (?:) కలిగి ఉంటే conditional operator కి మూడు arguments అవసరం. conditional operator యొక్క నియంత్రణ ప్రవాహం క్రింద ఇవ్వబడినది.

conditional operator యొక్క syntax:

```
expression 1? expression 2 : expression 3
```



పై syntax లో expression 1 అనునది ఒక condition ఇది మూల్యాంకనం చేయబడుతుంది. ఈ condition నిజం అయితే తరువాత control expression - 2 కు బదిలీ అవుతుంది. లేకపోతే control expression - 3 కి వెళుతుంది.

ఉదాహరణ 10.5 – కండిషనల్‌ను ఉపయోగించి రెండు సంఖ్యలో గరిష్ట సంఖ్యను గుర్తించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main()
{
        int a, b, largest;
        cout << "\n Enter any two numbers: ";
        cin >> a >> b;
         largest = (a>b)? a : b;
        cout << "\n Largest number : " << largest;
        return 0;
}
```

Output:

```
Enter any two numbers: 12 98
Largest number : 98
```

10.4.6 Switch statement

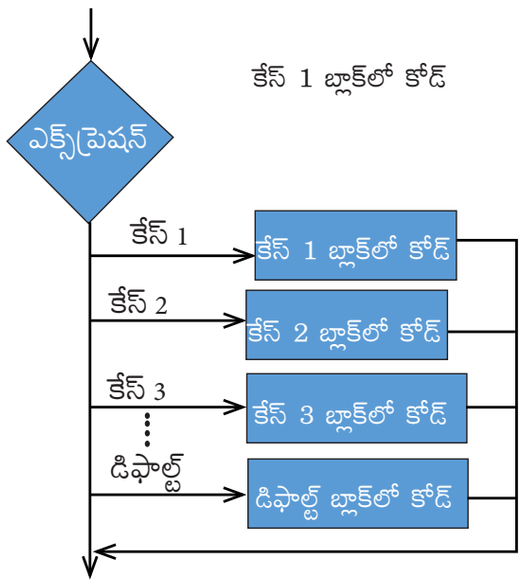
switch statement అనునది multi-way branch statement. execution విలువ ఆధారంగా code యొక్క వివిధ భాగాలకు execution పంపడానికి ఇది సులభమైన మార్గాన్ని అందిస్తుంది. switch statement క్రమాన్ని భర్తీ చేస్తుంది.

11th Computer Science_Chapter-10 Telugu.indd 166 30-03-2023 18:47:07

switch statement యొక్క syntax;

```
switch(expression)
{
        case constant 1:
                statement(s);
                break;
        case constant 2:
                statement(s);
                break;
        .
        .
        .
        .
        default:
                statement(s);
}
```

పై syntaxలో expression మూల్యాంకనం చేయబడుతుంది. మరియు దాని విలువ case statementsలలో ఒకదానిలో పేర్కొన్న constant value తో సరిపోతే సంబంధిత statementsల set అమలు చేయబడుతుంది. లేకపోతే default ఎంపిక క్రింద ఉన్న statements లు అమలు చేయబడుతాయి. switch statement మరియు flow chart యొక్క workflow క్రింద చూపబడినది.



ఫ్లో చార్టు10.5: స్విచ్ అండ్ ఫ్లో చార్టు యొక్క ప్రక్రియ

నియమాలు:

1. switch లో అందించబడిన expression స్థిరమైన విలువను కలిగి ఉండాలి. లేకుంటే అది చెల్లదు.
2. Duplicate విలువలు అనుమతించబడవు.
3. default ఐచ్ఛికం
4. statement sequence లు ముగించడానికి switch లోపల break statement ఉపయోగించబడుతుంది. break statement చేరుకొన్నప్పుడు switch ముగుస్తుంది మరియు switch statement ను అనుసరించి తరువాత పంక్తికి మారును.

11th Computer Science_Chapter-10 Telugu.indd 167 30-03-2023 18:47:07

5. break statement ఐచ్ఛికం ఇది విస్మరించబడితే తదుపరి సందర్భాలలో execution కొనసాగుతుంది. విరామం వచ్చే వరకు control will తదుపరి case ల ద్వారా వస్తుంది.
6. switch statements కూడా అనుమతించబడుతుంది.

ఉదాహరణ 10.6 – స్విచ్ స్టేట్‌మెంట్‌ను రూపొందించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main()
{
        int num;
        cout << "\n Enter week day number: ";
        cin >> num;
        switch (num)
        {
                case 1 : cout << "\n Sunday"; break;
                case 2 : cout << "\n Monday"; break;
                case 3 : cout << "\n Tuesday"; break;
                case 4 : cout << "\n Wednessday"; break;
                case 5 : cout << "\n Thursday"; break;
                case 6 : cout << "\n Friday"; break;
                case 7 : cout << "\n Saturday"; break;
                default: cout << "\n Wrong input....";
        }
}
```

Output:

```
Enter week day number: 6
Friday
```

10.4.7 Switch vs if-else

"if-else" మరియు "switch" అను రెండు selection statements లు. ఈ selection statements లు condition నిజం లేదా తప్పు అనే దాని ఆధారంగా నిర్దిష్ట statements block particular యొక్క How ను బదిలీ చేస్తుంది. అయితే, వారి కార్యకలాపాలలో కొన్ని తేడాలు ఉన్నాయి. ఇవి క్రింద ఇవ్వబడ్డాయి.

Key Differences మరియు switch ల మధ్య తేడాలు ఏవి?

వరుసల సంఖ్య	if-else	Switch
1	if block ను అమలు చేయాలి లేదా else block క్రింద అమలు చేయాలా వద్దా అని if statement లోని Expression నిర్ణయిస్తుంది.	switch statement లోప expression ఏ case ని execute చేయాలో నిర్ణయించుకుంటుంది.
2	if-else statement multiple choices కోసం multiple statements ని ఉపయోగిస్తుంది.	switch statement multiple choices కోసం single expression ను ఉపయోగిస్తుంది.
3	If-else statement equality కోసం అలాగే logical expression కోసం check చేస్తుంది.	equality కోసం మాత్రమే checks చేస్తుంది
4	**if** statement integer character pointer లేదా floating-point type లేదా Boolean type ను అంచనా వేస్తుంది.	switch statement character లేదా integer data type ను మాత్రమే అంచనా వేస్తుంది.
5	condition తప్పు అయితే else block statement అమలు చేయబడతాయి.	condition తప్పు అయితే default statements అమలు చేయబడతాయి.

11th Computer Science_Chapter-10 Telugu.indd 168 30-03-2023 18:47:07

switch statement ల కంటే if statement మరింత అనువైనది

10.5 Iteration statements

Iteration అనేది condition సంతృప్తి చెందే వరకు పదేపదే (లేదా) అంతకంటే ఎక్కువ అమలు చేయబడు statements క్రమం. ఈ statements లను control flow statements అని కూడా అంటారు. ఇది code పొడవును తగ్గించడానికి, సమయాన్ని తగ్గించడానికి, program execute చేయడానికి మరియు తక్కువ memory space ను తీసుకోవడానికి ఉపయోగించబడుతుంది. C++ మూడు రకాల iteration statements లకు మద్దతు ఇస్తుంది.

- for statement
- while statement
- do-while statement

specified condition నిజమైనంత వరకు అన్ని looping statements లు set statements లను పునరావృతం చేస్తాయి. specified condition ని loop నియంత్రణగా సూచిస్తాయి. మూడు loop statements కోసం నిజమైన condition ఏదైనా non-zero విలువ మరియు సున్న విలువ false చూపుతుంది.

10.5.1 లూప్ యొక్క భాగాలు

ప్రతి లూప్‌లో వివిధ ప్రయోజనాల కోసం ఉపయోగించే 4 అంశాలు ఉంటాయి. అవి.

- ప్రారంభ వ్యక్తీకరణ (Initialization expression)
- పరీక్ష వ్యక్తీకరణ (Test expression)
- నవీకరణ వ్యక్తీకరణ (Update expression)
- లూప్ యొక్క బాడీ (The body of the loop)

ప్రారంభ వ్యక్తీకరణ: control loop లోకి ప్రవేశించే ముందు control variable తప్పనిసరిగా ప్రారంభించబడాలి. control variable యొక్క initialization ప్రారంభ వ్యక్తీకరణ క్రింద జరుగుతుంది. ప్రారంభ వ్యక్తీకరణ loop ప్రారంభంలో ఒకసారి మాత్రమే అమలు చేయబడుతుంది.

పరీక్ష వ్యక్తీకరణ (Test Expression): Test expression అనేది condition లేదా దీనివలన loop-body ని అమలు చేయాలా వద్దా అని నిర్ణయిస్తుంది. expression నిజమని మూల్యాంకనం చేస్తే loop యొక్క body అమలు చేయబడుతుంది. లేకుంటే loop నిలిపివేయబడుతుంది. entry-controlled loop లో loop లోకి ప్రవేశించే ముందు పరీక్ష వ్యక్తీకరణ మూల్యాంకనం చేయబడుతుంది. అయితే exit-controlled loop లో loop నుండి నిష్క్రమించే ముందు పరీక్ష వ్యక్తీకరణ మూల్యాంకనం చేయబడుతుంది.

నవీకరణ – వ్యక్తీకరణ: ఇది loop variable విలువను మార్చడానికి ఉపయోగించబడుతుంది. loop యొక్క body executive చేయబడిన తరువాత loop చివరిలో ఈ statements executive చేయబడుతాయి.

లూప్ యొక్క బాడీ: statement లేదా statement ల సమితి పునరావృతంగా అమలు చేయబడి loop యొక్క body ని ఏర్పరుస్తుంది. entry-controlled loop లో ముందుగా test-expression మూల్యాంకనం చేయబడుతుంది మరియు non-zero అయితే loop యొక్క body executive చేయబడుతుంది. లేకపోతే loop నిలిపివేయబడుతుంది. exit-controlled loop లో body యొక్క మొదట అమలు చేయబడుతుంది. ఆపైన test-expression మూల్యాంకనం చేయబడుతుంది. test-expression నిజమైతే loop యొక్క body పునరావృతం అవుతుంది. లేకపోతే loop నిలిపివేయబడుతుంది.

10.5.2 for loop

for loop అనేది entry- controlled loop మరియు ఇది code ను పదే పదే అమలు చేయడానికి అనుమతించే looping statement. ఇది semicolons తో వేరుచేయబడిన మూడు వేరు వేరు statement ను (ప్రారంభం condition లేదా పరీక్ష వ్యక్తీకరణ మరియు నవీకరణ వ్యక్తీకరణలు) కలిగి ఉంది.

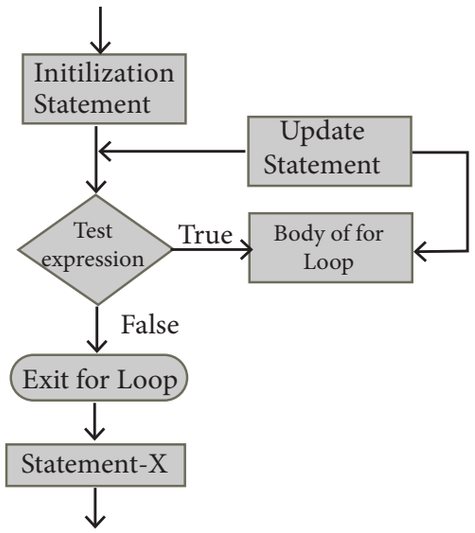
సాధారణ syntax

```
for (initialization(s); test-expression; update expression(s))
{
        Statement 1;
        Statement 2;
        ..............
}
Statement-x;
```


11th Computer Science_Chapter-10 Telugu.indd 169 30-03-2023 18:47:08

ఇది సాధారణ syntax:

Initialization part variables ని ప్రారంభించేందుకు లేదా ఒకసారి మాత్రమే executive చేయబడిన variable ను ప్రకటించడానికి ఉపయోగించబడుతుంది. ఆపైన test-expression కు వెళుతుంది. Test-expression యొక్క మూల్యాంకనం తర్వాత ఫలితం తప్పు అయితే control statement కి బదిలీ చేయబడుతుంది. ఫలితం నిజమైతే for loop యొక్క body అమలు చేయబడుతుంది. control expression ను update చేయడానికి బదిలీ చేయబడుతుంది. update expression భాగాన్ని మూల్యాంకనం చేసిన తరువాత control Test-expression భాగానికి బదిలీ చేయబడుతుంది. తరువాత మూడు నుండి ఐదు దశలు పునరావృతం అవుతాయి. workflow యొక్క for loop మరియు flow chart క్రింద ఇవ్వబడినది.



ఫ్లోచార్టు 10.6: ఫర్ లూప్ మరియు ఫ్లోచార్టు యొక్క క్రియా విధానం

ఉదాహరణ 10.7 for loop ను ఉపయోగించి 0 నుండి 9 వరకు సంఖ్యలను ప్రదర్శించెడి C++ ప్రోగ్రాం

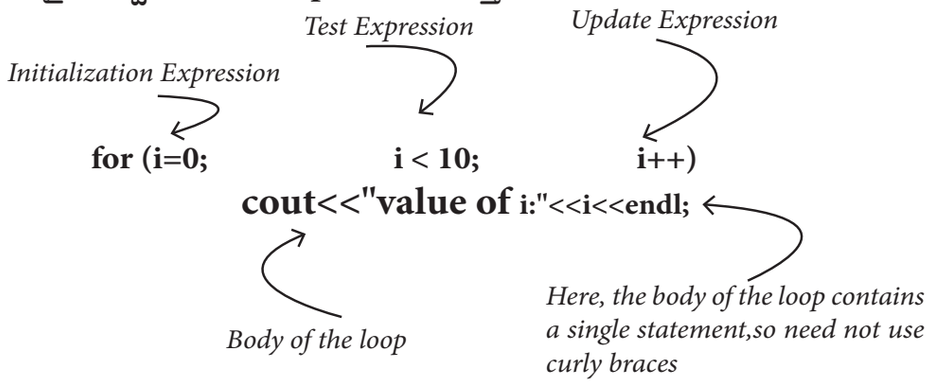
```
#include <iostream>
using namespace std;
int main ()
{
int i;
for(i = 0; i< 10; i ++ )
        cout<< "value of i : " <<i<<endl;
return 0;
}
```

Output

```
value of i : 0
value of i : 1
value of i : 2
value of i : 3
value of i : 4
value of i : 5
value of i : 6
value of i : 7
value of i : 8
value of i : 9
```


11th Computer Science_Chapter-10 Telugu.indd 170 30-03-2023 18:47:08

క్రింది వాక్యాలు పైన ఇవ్వబడిన **for loop** పనిని వివరిస్తుంది:



పై ప్రోగ్రామ్‌లో మొదటి variable ప్రారంభించబడుతుంది. తరువాత i ను 10 తో పోల్చబడుతుంది. i అనునది 10 కన్న తక్కువ ఉంటే i విలువ పెరుగుతుంది. ఈ విధంగా 0 నుండి 9 సంఖ్యలు ప్రదర్శించబడతాయి. i అనునది 10 అయిన తరువాత ఇకపై < 10 కాదు కాబట్టి నియంత్రణ for loop నుండి బయటికి వస్తుంది.

ఉదాహరణ 10.8 for loop ను ఉపయోగించి 1 నుండి 10 వరకు గల సంఖ్యల మొత్తాన్ని లెక్కించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main ()
{
int i,sum=0;
for(i=1; i<=10;i++)
 {
        sum=sum+i;
 }
cout<<"The sum of 1 to 10 is "<<sum;
return 0;
}
```

Output

```
The sum of 1 to 10 is 55
```

for loop యొక్క వైవిధ్యాలు

C++ లో అత్యంత ముఖ్యమైన looping statement లలో **for** loop ఒకటి. ఎందుకంటే ఇది అనేక వైవిధ్యాలను అనుమతిస్తుంది. ఈ వైవిధ్యాలు **for** loop యొక్క సౌలభ్యాన్ని మరియు అనువర్తనాన్ని పెంచుతాయి. ఈ వైవిధ్యాలు క్రింద చర్చించబడుతాయి.

బహుళ ప్రారంభీకరణ మరియు బహుళ నవీకరణ వ్యక్తీకరణలు

for loop యొక్క ప్రారంభం మరియు update expressions లో multiple statements లు ఉపయోగించవచ్చు. ఈ multiple initialization మరియు ఈ బహుళ నవీకరణ వ్యక్తీకరణ multiple update expressions, commas తో వేరుచేయబడుతాయి. ఉదాహరణకు

```
#include<iostream>
using namespace std;
int main()
{
        int i, j;
                for(i=0, j=10 ;  i<j  ;  i++,j--)
                {
                cout<<"\nThe value of i is"<<i<<" The value of j is "<<j;
```

Multiple initialization expressions (separated by commas)

Multiple update expressions (separated by commas)

11th Computer Science_Chapter-10 Telugu.indd 171 30-03-2023 18:47:08

```
}
return 0;
}
```

Output

```
The value of i is 0  The value of j is 10
The value of i is 1  The value of j is 9
The value of i is 2  The value of j is 8
The value of i is 3  The value of j is 7
The value of i is 4  The value of j is 6
```

పై ఉదాహరణలో initialization part i మరియు j అను రెండు variables ను కలిగి ఉంటుంది. update expression i++, j++ ఈ రెండు variables ',' లతో వేరుచేయబడుతుంది. ఇవి sequential order లో అమలుచేయబడుతాయి. అంటే ప్రారంభ సమయంలో మొదట i=0 తరువాత j=10 అదే విధంగా expression మొదట i++ మూల్యాంకనం చేయబడుతుంది మరియు j++ మూల్యాంకనం చేయబడుతుంది.

postfix కంటే prefix ఆపరేటర్ కి ప్రాధాన్యత ఇవ్వండి

సాధారణంగా update expression ఇంక్రిమెంట్/డిక్రిమెంట్ ఆపరేటర్‌ను కలిగి ఉంటుంది (++ or --). ఈ భాగంలో ఒంటరిగా ఉపయోగించాల్సినపుడు postfix కంటే prefix ని ఎల్లప్పుడు ఇష్టపడుతారు. దీని వెనుక కారణం ఎమిటి అంటే ఒంటరిగా వేగంగా executed చేయబడుతాయి.

ఐచ్ఛిక వ్యక్తీకరణలు (Optional expressions)

సాధారణంగా for loop లు మూడు భాగాలు కలిగి ఉంటుంది. అనగా initialization expressions కోసం ఈ మూడు test expressions లు update expressions మాత్రమే.

ఉదాహరణ 10.9: 1 నుండి 'n' వరకు గల సంఖ్యల మొత్తాన్ని లెక్కించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main ()
{
int i, sum=0, n;
cout<<"\n Enter The value of n";
cin>>n;
i =1;
for (  ; i<=n;  )
 {
        sum += i;
        ++i;
 }
cout<<"\n The sum of 1 to " <<n<<"is "<<sum;
return 0;
}
```

Output

```
Enter the value of n 5
The sum of 1 to 5 is 15
```


11th Computer Science_Chapter-10 Telugu.indd 172 30-03-2023 18:47:08

పై code లో update expression ఇవ్వబడలేదు. కానీ update expression ముందు semicolon అవసరం.

```
for (    ;        i<=n;   )
```

Initialization expression and update expressions are skipped

పై code లో loop లో initialization లేదా update expression ఇవ్వబడలేదు. రెండు లేదా ఏవైన expressions లేకుంటే control అనునది conditional తో కూడిన భాగానికి బదిలీ చేయబడుతుంది.

infinite loop

For loop లో test-expression లేనట్లు అయితే infinite loop ఏర్పడుతుంది.

test - expression is skipped

```
for( i=0 ;   ;    ++i)
    cout<<"\n Welcome";
```

This statement is displayed infinitely

అదే విధంగా క్రింది for loop కూడా infinite loop ని ఏర్పరుస్తుంది.

All three expressions are skipped

```
for(     ;        ;        )
    cout<<"\n Welcome";
```

This statement is displayed infinitely

Empty loop

Empty loop అంటే దాని body లో ఎటువంటి statement లేని loop ని empty అంటారు. క్రింది for loop అనునది empty loop:

```
for( i=0        ;        i<=5;   +=i) ;
```

The body of for loop contains a null statement

పై కోడ్‌లో for loop empty statement ను కలిగి ఉంది. ఇది ఒక empty loop అదే విధంగా క్రింది for loop కూడా empty loop ను ఏర్పరుస్తుంది.

```
int i;
for( i=0        ;        i<=5;   ++i) ;
{
        cout<<"\nWe are Indians";
}
```

The body of for loop contains a null statement

The body of for loop is not executed because semicolon(;) is given at the end of for loop.

11th Computer Science_Chapter-10 Telugu.indd 173 30-03-2023 18:47:08

పై code లో for loop తరువాత semicolon ఇవ్వబడినందున braces తో జతచేయబడిన for loop యొక్క body executive చేయబడదు.

for loop కోసం Declaration of variable

C++ లో variables ని for loop లో కూడా declared చేయవచ్చు. ఉదాహరణకు

```
int main ()
{
   int sum = 0;
   for(int i=0; i<=5; ++i )
   {
        sum = sum + i;
   }
   cout<<"\nThe variable i cannot be accessed here";
   cout<<"\n The variable sum can be accessed here";
}
```

Variable (i)is declared within the for loop.

The variable i can be accessed only within the body of loop.

main() బ్లాక్ లోపల declared చేయబడిన variable in main() లోపల ఎక్కడైన accessed చేయవచ్చును. అనగా main() లోని variable పరిధి.

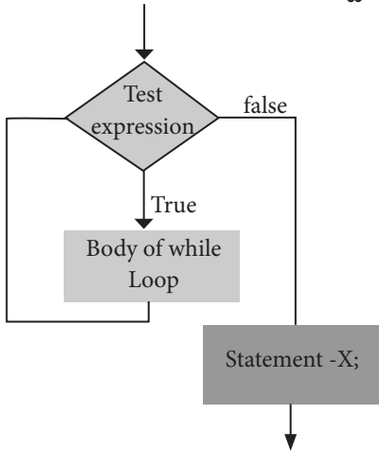
10.5.3 While loop

while loop అనేది control flow statement. ఇది condition నిజమైనంత వరకు loop statements లను execute చేయడానికి అనుమతిస్తుంది. అయితే loop అనేది entry ఎందుకంటే loop లోపలికి ప్రవేశించే ముందు test మూల్యాంకనం చేయబడుతుంది.

while loop యొక్క syntax :

```
while ( Test expression )
{
        Body of the loop;
}
Statement-x;
```

while loop యొక్క control flow మరియు flow chart క్రింద ఇవ్వబడినది



ఫ్లోచార్టు 10.7: while loop control flow and ఫ్లో చార్టు

11th Computer Science_Chapter-10 Telugu.indd 174 30-03-2023 18:47:08

while loop లో test expression మూల్యాంకనం చేయబడుతుంది మరియు test expression ఫలితం నిజమైతే loop యొక్క body executed చేయబడుతుంది. మరియు మళ్ళీ control loop కి బదిలీ చేయబడుతుంది. test expression ఫలితం తప్పు అయినప్పుడు -x కు బదిలీ చేయబడుతుంది.

ఉదాహరణ 10.10: while loop ను ఉపయోగించి 1 నుండి 10 వరకు గల సంఖ్యల మొత్తాన్ని గణించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main ()
{
int i=1,sum=0;
while(i<=10)
{
        sum=sum+i;
        i++;
}
cout<<"The sum of 1 to 10 is "<<sum;
return 0;
}
```

Output

```
The sum of 1 to 10 is 55
```

పై program, integer variable i కి 1ని మరియు variable మొత్తం కి0 తో ప్రారంభించబడుతాయి. while loop condition ను check చేస్తుంది. i < 10 condition నిజం అయితే i యొక్క విలువ ఇది మొత్తం. మరియు i కి జోడించబడుతుంది మరియు 1 పెంచబడుతుంది. తరువాత i < 10 చేయబడుతుంది. check కాబట్టి మొత్తానికి మునుపటి విలువకు 2 కూడబడుతుంది. ఇది i అనునది 11 అయ్యేంత వరకు కొనసాగుతుంది. 11 < 10 తప్పుగా అంచనా వేస్తుంది. మరియు loop ముగుస్తుంది. while loop ముగింపు తరువాత మొత్తం విలువ ప్రదర్శించబడుతుంది.

ఉదాహరణ 10.11: while loop ద్వారా 5 సంఖ్యల మొత్తము మరియు సరాసరిని లెక్కించు C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main ()
{
int i=1,num,avg,sum=0;
while(i<=5)
{
        cout<<"Enter the number : ";
        cin>>num;
        sum=sum+num;
        i++;
}
avg=sum/5;
cout<<"The sum is "<<sum<<endl;
cout<<"The average is "<<avg;
return 0;
}
```

Output

```
Enter the number : 1
Enter the number : 2
Enter the number : 3
Enter the number : 4
Enter the number : 5
The sum is 15
The average is 3
```


11th Computer Science_Chapter-10 Telugu.indd 175 30-03-2023 18:47:08

పై program, integer variables **num** మరియు **avg** ప్రకటించబడ్డాయి మరియు variable i కి ఒకటిని మరియు మొత్తంకి 0 ను ప్రారంభించబడుతుంది. while loop condition ను తనిఖి చేస్తుంది. i <= 5 condition నిజం అయినందున user నుండి ఒక సంఖ్య చదువబడుతుంది. మరియు ఇది మొత్తానికి జోడించబడుతుంది మరియు i ను ఒకటిగా పెంచబడింది. ఇప్పుడు condition i <= 5 మళ్ళీ తనిఖి చేయబడుతుంది. 2 <=5 నుండి రెండవ సంఖ్య పొందబడింది మరియు అది మొత్తానికి జోడించబడుతుంది. ఇది i అనునది 6 అయ్యే వరకు కొనసాగుతుంది. సమయంలో while loop ముగుస్తుంది. loop ముగింపు తరువాత avg గణించబడుతుంది మరియు sum మరియు avg రెండు ప్రదర్శించబడతాయి.

While loop వైవిధ్యాలు

while loop అనేక వైవిధ్యాలను కలిగి ఉండవచ్చు. ఇది empty loop లేదా infinite loop కావచ్చు. empty while loop null statement లను తప్ప body ఎటువంటి statement ను కలిగి ఉండదు. అంటే statement తప్ప. ఉదాహరణకి

```
int main()
{

    int i=0;
    while(++i < 10000 )
                          ;
    return 0;

}
```

This is an empty loop because the while loop does not contain any statement

పై code లో loop అనేది time delay loop. program కొంత సమయం పాటు pausing చేయడానికి time delay loop ఉపయోగపడుతుంది. loop delay లోపల update statement ఇవ్వనప్పుడు while loop అనేది infinite కావచ్చు. ఉదాహరణకి

```
int main()
{

    int i = 0;
    while(i < =10)
      cout <<"The value of i is "<<i;
      i++;

    return 0;

}
```

This statement will be displayed infinitely because no update statement inside the body of the loop

This is not a part of the while loop statement because of missing curly braces

11th Computer Science_Chapter-10 Telugu.indd 176 30-03-2023 18:47:08

10.5.4 do-while loop

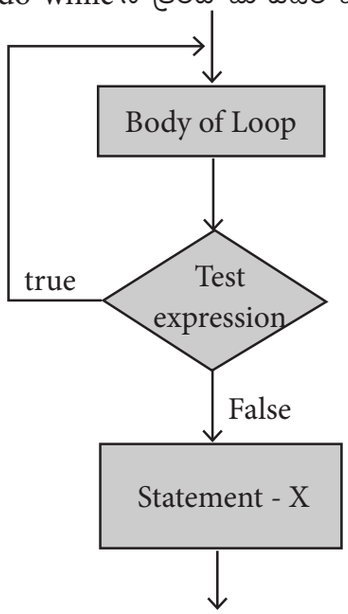
do-while loop అనేది exit-controlled loop do-while loop యొక్క body ని executing చేసిన తరువాత condition loop దిగువన మూల్యాంకనం చేయబడుతుంది. దీని అర్థం పునరావృత్త సమయంలో condition తప్పుగా అంచనా వేసినప్పటికి loop యొక్క body కనీసం ఒక్కసారి అయిన execute చేయబడుతుంది.

The do-while loop syntax is:

```
do
{
        Body of the loop;

} while(condition);
```

Flow control మరియు ఫ్లో చార్టు for do-while ని క్రింద చూపడం జరిగినది



ఫ్లోచార్టు 10.8 : do-while loop control flow and ఫ్లో చార్టు

ఉదాహరణ 10.12: do-while loop ను ఉపయోగించి 10 నుండి 1 వరకు సంఖ్యలను ప్రదర్శించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main ()
{
int n = 10;
do
{
        cout<<n<<", ";
        n--;
}while (n>0) ;
}
```

Output

10, 9, 8, 7, 6, 5, 4, 3, 2, 1

11th Computer Science_Chapter-10 Telugu.indd 177 30-03-2023 18:47:08

పై program లో integer variable **n** 10 కి ప్రారంభించబడుతుంది. తరువాత **n** విలువ **10** గా ప్రదర్శించబడుతుంది మరియు **n** కు **B** 1 తగ్గించబడుతుంది. ఇప్పుడు **9 > 0** మూల్యాంకనం చేయబడుతుంది ఎందుకంటే **n** మళ్ళీ ప్రదర్శించబడుతుంది మరియు **n** అనునది **8** కి తగ్గుతుంది. **n** అనునది **0** అయ్యేంత వరకు కొనసాగుతుంది. ఆ సమయంలో condition **n > 0** తప్పుగా అంచనా వేయబడుతుంది మరియు do-while loop ముగుస్తుంది.

10.5.5 Nesting of loops

మరొక loop ను కలిగి ఉన్న loop ను nested loop అంటారు. దీని syntax క్రింద ఇవ్వబడినది.

```
for (initialization(s); test-expression; update expression(s))
{
        for (initialization(s); test-expression; update expression(s)
        {
        statement(s);
         }
statement(s);
}
```

```
while(condition)
{
        while(condition)
        {
        statement(s);
        }
statement(s);
}
```

```
do
{
statement(s);
        do
        {
        statement(s);
        }while(condition);
} while( condition );
```

ఉదాహరణ 10.13: nested for loop ను ఉపయోగించి మాత్రికల గుణకారపట్టికను ప్రదర్శించెడి C++ ప్రోగ్రాం

```
#include<iostream>
using namespace std;
int main(void)
{
    cout<< "A multiplication table:" <<endl <<"  1\t2\t3\t4\t5\t6\t7\t8\t9" <<endl<< "" <<endl;
    for(int c = 1; c < 10; c++)
    {
    cout<< c << "| ";
    for(int i = 1; i< 10; i++)
    {
    cout<<i * c << '\t';
    }
    cout<<endl;
    }
return 0;
}
```


11th Computer Science_Chapter-10 Telugu.indd 178 30-03-2023 18:47:08

Output

A multiplication table:

1	2	3	4	5	6	7	8	9
1| 1	2	3	4	5	6	7	8	9
2| 2	4	6	8	10	12	14	16	18
3| 3	6	9	12	15	18	21	24	27
4| 4	8	12	16	20	24	28	32	36
5| 5	10	15	20	25	30	35	40	45
6| 6	12	18	24	30	36	42	48	54
7| 7	14	21	28	35	42	49	56	63
8| 8	16	24	32	40	48	56	64	72
9| 9	18	27	36	45	54	63	72	81

10.6 Jump statements

program యొక్క సాధారణ flow కి అంతరాయం కలిగించడానికి Jump statements లు ఉపయోగపడుతాయి. Jump statements యొక్క రకాలు

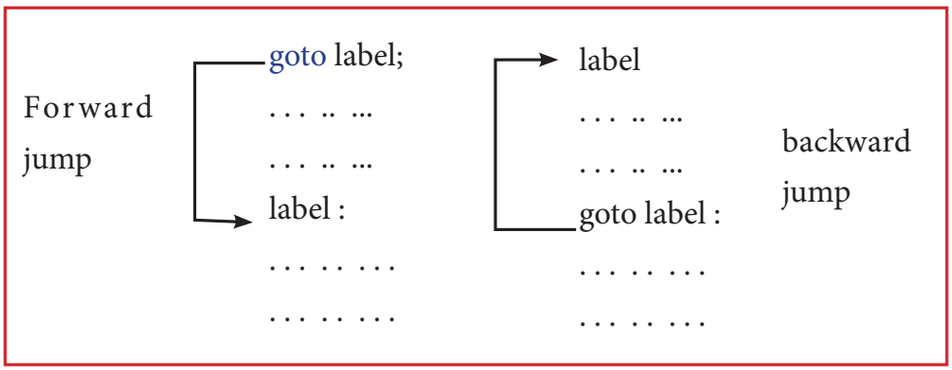
- goto statement
- break statement
- continue statement

10.6.1 goto statement

goto statement అనేది control statement ఇది program లో ఎటువంటి condition లేకుండా control ను ఒక place నుండి మరొక place కు బదిలీ చేయబడుతుంది.

Goto statement యొక్క syntax;

Syntax1	**Syntax2**
goto label; ---------- ---------- ---------- **label:**	**label:** ---------- ---------- ---------- **goto label;**



11th Computer Science_Chapter-10 Telugu.indd 179 30-03-2023 18:47:08

పై program లో 5 ఒక identifier ఇప్పుడు goto label; అనునది program jumps to label: కు దూకుతుంది మరియు క్రింద ఉన్న code ను అమలు చేస్తుంది.

ఉదాహరణ 10.14: goto statement ద్వారా మొదటి ఐదు బేసి సంఖ్యలను ప్రదర్శించెడి C++ ప్రోగ్రాం

```
# include <iostream>
using namespace std;
int main()
{
int n=1;
jump:
{
 if(n<10)
 { // Control of the program move to jump:
        cout<<n<<'\t';
        n+=2;
        goto jump;
 }
 else
        return 0;
} }
```

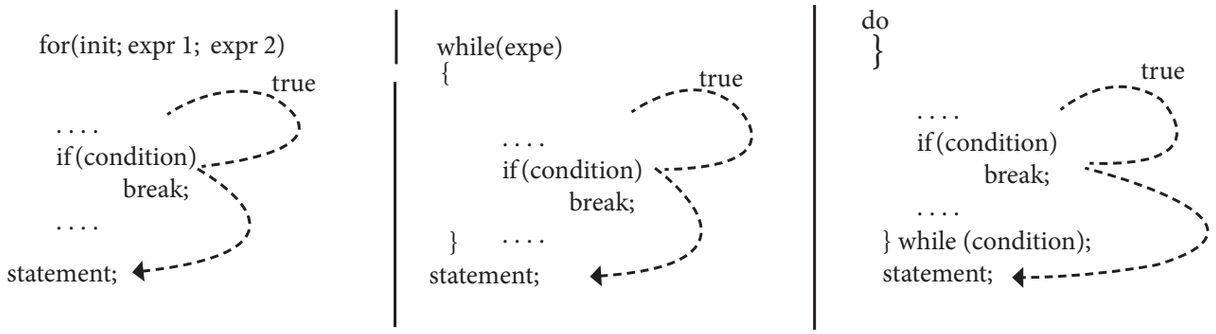
Output

```
1       3       5       7       9
```

పై program లో మొదటి 5 బేసి సంఖ్యలు ప్రదర్శించబడతాయి. n అనునది 10 కన్నా తక్కువ అయితే goto control ని jump statement ని బదిలీ చేస్తుంది. n అనునది 10 కన్నా ఎక్కువ అయితే control loop నుంచి బయట వస్తుంది.

10.6.2 break statement

break statement అనునది ఒక జంప్ స్టేట్‌మెంట్. ఇది లూప్ యొక్క execution ను terminate చేసి కంట్రోల్‌ను రెస్యూం నార్మల్ ఎక్సిక్యూషన్‌కు మార్పిడి చేయును. లూపింగ్ స్టేట్‌మెంట్‌లతో పనిచేసెడి బ్రేక్ స్టేట్‌మెంట్‌ల చిత్రణము క్రింద ఇవ్వబడినది.



break statement in for, while and do-while loop

11th Computer Science_Chapter-10 Telugu.indd 180 30-03-2023 18:47:08

ఉదాహరణ 10.15 break statement ను ఉపయోగించి N సంఖ్యలను లెక్కించు C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
int main ()
{
int count = 1;
do
{
cout<< "Count : " << count <<endl;
if( count > 3)
 {
        break;
 }
count ++;
}while( count < 20 );
return 0;
}
```

Output

```
Count : 1
Count : 2
Count : 3
```

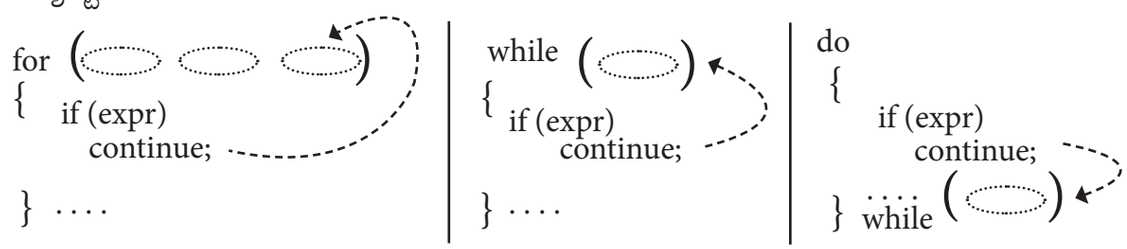
పై ఉదాహరణలో while loop 20 సార్లు పునరావృతం అవుతుంది. అయితే count 3 కు చేరుకొన్న వెంటనే break statement కారణంగా loop నిలిపి వేయబడుతుంది.

10.6.3 continue statement

continue statement break statement తో సమానంగా పనిచేస్తుంది. loop ను ముగించే బదులు break statement, continue statement loop ను కొనసాగించడానికి లేదా తదుపరి పునరావృత్తాన్ని అమలు చేయడానికి forces loop చేస్తుంది.

బ్రేక్‌స్టేట్‌మెంట్‌లో continue statement executive చేయబడినప్పుడు continue statement ను అనుసరించి loop లోని code చేయబడుతుంది మరియు loop యొక్క తదుపరి పునరావృతం ప్రారంభమవుతుంది.

కంటిన్యూ స్టేట్‌మెంట్ పనితీరును వివరించెడి పటము



పై ఉదాహరణలో loop 10 సార్లు పునరావృత్తం అవుతుంది. అయితే I విలువ 6 కు చేరుకొనట్లు అయితే continue statement కారణంగా control loop కోసం బదిలీ చేయబడుతుంది.

11th Computer Science_Chapter-10 Telugu.indd 181 30-03-2023 18:47:08

ఉదాహరణ 10.16 కంటిన్యూ స్టేట్‌మెంట్‌ను ఉపయోగించి **6** తప్ప తక్కిన **1** నుండి **10** వరకు సంఖ్యలను ప్రదర్శించెడి C++ ప్రోగ్రాం

```
#include <iostream>
using namespace std;
 int main()
{
for (int i = 1; i<= 10; i++) {
if (i == 6)
continue;
else
cout<<i<< " ";
}
return 0;
}
```

Output

1 2 3 4 5 7 8 9 10

Break మరియు Continue మధ్య గల వ్యత్యాసం

Break	Continue
loop యొక్క execution ను **ముగించడానికి Break** ఉపయోగపడుతుంది.	loop యొక్క execution ను ముగించడానికి Continue ఉపయోగించబడదు.
ఇది పునరావృత్తాన్ని **break** చేస్తుంది.	ఇది పునరావృతాన్ని **skip** చేస్తుంది.
ఈ statement లు execute చేయబడినపుడు control loop నుండి బయటకు వస్తుంది మరియు loop తరువాత వెంటనే statement ను execute చేస్తుంది.	ఈ statement ను execute చేసినపుడు loop నుండి బయటకు రాదు కాని loop యొక్క తదుపరి పునరావృతానికి కదలుతుంది / jump అవుతుంది.
Break ను loop లతో పాటు switch case లతో ఉపయోగించబడుతుంది.	Continue ను loop లలో మాత్రమే ఉపయోగించబడుతుంది. ఇది switch case లో ఉపయోగించబడదు.



ప్రయోగ అభ్యాసాలు:

క్రింది సమస్యలను పరిష్కరించడానికి C++ ప్రోగ్రామ్‌ను వ్రాయండి

1. ఒక character ని input చేయడానికి మరియు ఇచ్చిన character ఒకే alphabet, digit or ఏదైన ఒక character అని print చేయడానికి Program.
2. ఇచ్చిన character ఒక uppercase లేదా lowercase character లేదా ఒక digit లేదా వేరొక character అని ప్రింట్ చేయడానికి Program. దాని కోసం ASCII కోడ్‌లను ఉపయోగించండి. ASCII కోడ్‌లు క్రింద ఇవ్వబడ్డాయి.

Characters	ASCII Range
'0' - '9'	48 - 57
'A' - 'Z'	65 - 90
'a' - 'z'	97 - 122
other characters	0- 255 మినహా పైన పేర్కొన్న కోడ్‌లు.

11th Computer Science_Chapter-10 Telugu.indd 182 30-03-2023 18:47:08

3. పూర్ణాంకం యొక్క కారకాన్ని లెక్కించడానికి ప్రోగ్రామ్

4. fibonacci series i.e., 0 1 1 2 3 5 8 లను ప్రింట్ చేయడానికి ప్రోగ్రామ్

5. nested loop ను ఉపయోగించి క్రింద డిజైన్‌ను రూపొందించడానికి Programs

(a)	(b)
A A B A B C A B C D A B C D E A B C D E F	5 4 3 2 1 5 4 3 2 5 4 3 5 4 5

గుర్తుంచుకోవలసిన అంశాలు:

- computer program అనేది ఒక నిర్ధిష్ట పనిని నిర్వహించడానికి statements లేదా instructions.
- C++, నందు Null మరియు Compound Statement అనే రెండు రకాలు statements ఉపయోగించబడుతాయి.
- Control Statement అనునది flow of instaructions యొక్క క్రమాన్ని మార్చే Statements.
- C++ లో మూడు రకాల control statement ఉపయోగించబడుతాయి. (1) Sequence Statement (2) Selection Statement (3) Iteration Statement.
- *If* మరియు *Switch* అనేవి Selection Statements.
- Conditional Operator 'if else Statement' కు ప్రత్యామ్నాయం
- Switch Statment అనేది బహుళ మార్గం శాఖల statement.
- condition సంతృప్తి చెందేవరకు పదే పదే statement ల set ని అమలు చేయడానికి Iteration Statement (looping) ఉపయోగించబడుతుంది.
- మూడు రకాల Iteration Statements మద్ధతు కొరకు (1) *for* (2) *While* (3) *do-While*.
- C++లో మూడు Jump Statment ఉపయోగించి (1) *goto* (2) *break* (3) *continue*.

మూల్యాంకనం

84AUJ

విభాగం – అ

సరైన సమాధానం ఎంచుకోండి

1. null statement యొక్క ప్రత్యామ్నాయం పేరు

   (అ) No statement
   (ఆ) Empty statement
   (ఇ) Void statement
   (ఈ) Zero statement

2. C++, లో statement ల సమూహాన్ని ఇందులో చేర్చాలి:

   (అ) { } (ఆ) [ ]
   (ఇ) ( ) (ఈ) < >

3. iteration లో మళ్ళీ మళ్ళీ అమలు చేయడానికి statement ల సమితిని ఇలా అంటారు.

   (అ) condition (ఆ) loop
   (ఇ) statement (ఈ) body of loop

11th Computer Science_Chapter-10 Telugu.indd 183 30-03-2023 18:47:09

4. బహుళ మార్గాల శాఖ statement:
(అ) if (ఆ) if … else
(ఇ) switch (ఈ) for

5. iteration statement అనేవి ఎన్ని రకాలు?
(అ) 2 (ఆ) 3
(ఇ) 4 (ఈ) 5

6. (int i=0; i<10; i++) కోసం క్రింది loop లను ఎన్ని సార్లు అమలు చేస్తాయి?
(అ) 0 (ఆ) 10
(ఇ) 9 (ఈ) 11

7. క్రింది వాటిలో exit control loop ఏది?
(అ) for (ఆ) while
(ఇ) do…while (ఈ) if…else

8. jump statement నందు కీలక పదాల నుండి odd one ను గుర్తించండి.
(అ) break (ఆ) switch
(ఇ) goto (ఈ) continue

9. క్రింది వానిలో దేనిని entry control loop అంటారు?
(అ) do-while (ఆ) switch
(ఇ) while (ఈ) if-else

10. body లో మరొక loop ను కలిగి ఉన్న loop
(అ) Nested loop (ఆ) Inner loop
(ఇ) Inline loop (ఈ) Nesting of loop

విభాగం-ఆ

లఘు సమాధాన ప్రశ్నలు

1. null statement మరియు compound statement అంటే ఏమిటి?
2. selection statement అనగానేమి? అందులోని రకాలను తెలుపుము.
3. క్రింది code sigment ను సరిచేయండి.

```
if (x=1)
  p= 100;
else
  p = 10;
```

4. క్రింది code యొక్క output ను తెలపండి.

```
int year;
cin >> year;
  if (year % 100 == 0)
        if ( year % 400 == 0)
              cout << "Leap";
  else
        cout << "Not Leap year";
```

ఇవ్వబడిన input (i) 2000 (ii) 2003 (iii) 2010?

5. క్రింది code యొక్క output ను తెలుపుము.

```
for (int i=2; i<=10 ; i+=2)
  cout << i;
```

6. 21 నుండి 30 వరకు ఉన్న సంఖ్యలను ప్రదర్శించే while loop కోసం వ్రాయండి.
7. 2, 4, 6, 8.....20 ను సంఖ్యలను ప్రదర్శించే while loop ను వ్రాయండి.
8. if else మరియు 2: (fernary) ఆపరేటర్‌ని సరిపోల్చండి.

విభాగం-ఇ

సంక్షిప్త సమాధాన ప్రశ్నలు

1. క్రింది if-else ని ఏక conditional statement గా మార్చండి.

```
if (x >= 10)
        a = m + 5;
else
        a = m;
```

2. క్రింది కోడ్‌ని మళ్ళీ వ్రాయండి తద్వారా ఇది క్రియాత్మకంగా ఉంటుంది.

```
v = 5;
do;
{
  total += v;
  cout << total;
while v <= 10
```


11th Computer Science_Chapter-10 Telugu.indd 184 30-03-2023 18:47:09

3. ఇచ్చిన సంఖ్య యొక్క గుణకార పట్టికను ముద్రించడానికి C++ ప్రోగ్రామ్‌ను వ్రాయండి.

4. switch statement యొక్క syntax మరియు ప్రయోజనాలను వ్రాయండి.

5. క్రింది శ్రే ణులను ముదించుటకు ఒక చిన్న program ను వ్రాయండి.

   (a) 1 4 7 10..... 40

విభాగం - ఈ

విపుల సమాధాన ప్రశ్నలు

1. తగిన ఉదాహరణలో control statement ను వివరించండి.

2. entry control loop అంటే ఏమిటి? తగిన ఉదాహరణలలో ఏదైనా ఒక entry controlled loop ను దానిని వివరించండి.

3. రెండు సంఖ్యల LCM మరియు GCD లను కనుగొనడానికి program ను వ్రాయండి.

4. క్రింది series ల మొత్తాన్ని కనుగొనడానికి program ను వ్రాయండి.

   (అ) $x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \frac{x^5}{5!} - \frac{x^6}{6!}$

   (ఆ) $x + \frac{x^2}{2} + \frac{x^3}{3} + ... + \frac{x^n}{n}$

5. క్రింది series ల మొత్తాన్ని కనుగొనడానికి program ను వ్రాయండి.

   $S = 1 + x + x^2 + ... + x^n$

సూచన పుస్తకాలు

(1) Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.

(2) The Complete Reference C++ (Forth Edition), Herbert Schildt. Mc.Graw Hills.

(3) Computer Science with C++ (A text book of CBSE XI and XII), Sumita Arora, Dhanpat Rai & Co.

11th Computer Science_Chapter-10 Telugu.indd 185 30-03-2023 18:47:09

అధ్యాయము 11

ఫంక్షన్స్

అభ్యసన లక్ష్యాలు



ఈ పాఠము చదివిన తరువాత విద్యార్థుల తెలుసుకోవలసినవి.

- ఫంక్షన్లు నిర్వచనాలు మరియు ఫంక్షన్ల ఉపయోగాలను అవగాహన చేసుకోవడం.
- ఫంక్షన్ల రకాలు అర్థం చేసుకోవడం ముందే నిర్వచించబడిన మరియు వినియోగదారు నిర్వచించిన విధులను అర్థం చేసుకోవడం.
- సమస్యలను పరిష్కరించడానికి గణిత ఫంక్షనలను వర్తింప చేయడం.
- స్ట్రింగ్ మరియు క్యారెక్టర్ డేటా మానిప్యులేషన్ కోసం స్ట్రింగ్ మరియు క్యారెక్టర్ ఫంక్షన్లను ఉపయోగించడం.
- ఫంక్షన్లను సృష్టించడం ద్వారా programming ను అమలు పరచడం
- Role of argument లను అర్థం చేసుకోవడం మరియు argument యొక్క వివిన్న పద్ధతులను సరిపోల్చడం.
- ప్రోగ్రామ్‌లోని వేరియబుల్స్ మరియు ఫంక్షన్లు పరిధిని గుర్తించడం.

11.1 పరిచయం (INTRODUCTION)

ఒక పెద్ద ప్రోగ్రామ్‌ను చిన్న ఉప ప్రోగ్రామ్ (Blocks) గా విభజించవచ్చును. వీటిని ఫంక్షన్‌లు పిలుస్తారు. ఇక్కడ ప్రతి (Sub-program) కొన్ని నిర్ధిష్ట కార్యాచరణలను చేయగలదు. ఇవి Program యొక్క పరిమాణం మరియు సంక్లిష్టతను తగ్గిస్తాయి. అర్థం చేసుకోవడం పరీక్షించడం మరియు లోపాల కోసం తనిఖీ చేయడం మొదలైవి సులభతరం చేస్తాయి. డిఫాల్ట్‌గా అందుబాటులో ఉండే ఫంక్షన్‌లను **"Built-in"** ఫంక్షన్‌లుగా పిలుస్తారు మరియు వినియోగదారుల వారి సొంత ఫంక్షన్‌లను **"User-defined"** ఫంక్షన్‌లు అని పిలుస్తారు.

- Built-in functions – C++ బాషా ప్రమాణంలో అందుబాటులో ఉండే standard library.
- User-defined functions – వినియోగదారులు సృష్టించిన విధులు.

11.2 ఫంక్షన్ల అవసరాలు (Need for Functions)

Program యొక్క పరిమాణం మరియు సంక్లిష్టతను తగ్గించడానికి మనము ఫంక్షన్లను ఉపయోగిస్తాము. Programmers తమ స్వంత ఫంక్షన్‌లను వ్రాయడం లేదా ప్రామాణిక లైబ్రరీ నుండి కాల్ చేయడం ద్వారా sub programs లను ఉపయోగించుకోవచ్చు.

1. విభజించుట మరియు చేర్చుట (Divide and Conquer)

- సంక్లిష్టమైన ప్రోగ్రామ్‌లను ఫంక్షన్లు అని పిలువబడే నిర్వహించదగిన ఉప ప్రోగ్రామ్‌లుగా విభజించవచ్చు.
- ఒక ప్రోగ్రామర్ డెవలపింగ్, డీబగ్గింగ్ మరియు ప్రత్యేక ఫంక్షన్లపై దృష్టిని పరీక్షించడం చేయవచ్చు.
- చాలా మంది ప్రోగ్రామర్లు ఏకకాలంలో వివిధ ఫంక్షన్లలో పనిచేయవచ్చు.

2. పునర్వినియోగం (Reusability)

- కోడ్ యొక్క కొన్ని పంక్తులు పదేపదే వివిధ సందర్భాలలో ఉపయోగించబడవచ్చు. నిర్వహణను మెరుగుపరిచే మరియు ప్రోగ్రామ్ పరిమాణాన్ని తగ్గించే ఫంక్షన్‌లను ఉపయోగించడం ద్వారా ఒకే code యొక్క నకిలీని తొలగించవచ్చు.
- కొన్ని ఫంక్షన్‌లను వేర్వేరు Input లలో అనేక సార్లు పిలవవచ్చు.

11th Computer Science_Chapter-11 Telugu.indd 186 30-03-2023 18:48:43

11.3 ఫంక్షన్ల యొక్క రకాలు (Types of Functions)

ఫంక్షన్లను రెండు రకాలుగా వర్గీకరించబడతాయి.

1. Pre-defined or Built-in or Library Functions
2. User-defined Function.

C++ వివిధ పనుల కోసం ఉపయోగించడానికి సిద్ధంగా ఉన్న ఫంక్షన్ల యొక్క గొప్ప సేకరణను అందిస్తాయి. వీటిలో ప్రతి ఒక్కటి నిర్వర్తించాల్సిన పనులు ఇప్పటికే వ్రాయబడ్డాయి. debugg చేయబడ్డాయి మరియు సంకలనం చేయబడ్డాయి. వాటి నిర్వచనాలను మాత్రమే సమూహపరచి **"Header file"** అని పిలువబడే ఫైల్‌లలో నిల్వ చేయబడతాయి. ఇటువంటి వాటిని ఉపయోగించడానికి సిద్ధంగా ఉన్న సబ్ ప్రోగ్రామ్‌లను **pre-defined functions** లేదా **built-in functions** అంటారు.

C++ వినియోగదారు అవసరాలకు అనుగుణంగా నిర్దిష్ట పని కోసం కొత్త ఫంక్షన్‌లను సృష్టించే సదుపాయాన్ని కూడా అందిస్తుంది. పని పేరు మరియు అవసరమైన డేటా (Arguments) వినియోగదారునిచే నిర్ణయించబడతాయి మరియు అందువల్ల వాటిని **User-defined functions** అంటారు.

11.4 C++ Header Files మరియు Built-in ఫంక్షన్లు

లైబ్రరీ ఫంక్షన్ల కోసం నమూనా (prototype) మరియు నిర్వచనాలను ఫంక్షన్ల header file అందిస్తాయి. లైబ్రరీ ఫంక్షన్లులలో ఉపయోగించే డేటా రకాలు మరియు స్థిరాంకాలు కూడా వాటిలో నిర్వచించబడ్డాయి. Header files ని వాటి file extension **.h.** ద్వారా గుర్తించవచ్చు. ఏక హెడర్ ఫైల్ multiple built-in ఫంక్షన్‌లను కలిగి ఉండవచ్చు.

ఉదాహరణకు **stdio.h** ముందుగా నిర్వచించబడిన **"standard input/output"** ఫంక్షన్‌లను కలిగి ఉన్న header file.

11.4.1 ప్రామాణిక ఇన్‌పుట్ / అవుట్‌పుట్ (stdio.h)

ఈ హెడర్ ఫైల్ ప్రామాణిక I/O ముందే నిర్వచించిన **getchar(), putchar(), gets(), puts()** మొదలైన ఫంక్షన్లలను నిర్వచిస్తుంది.

11.4.1.1 getchar() మరియు putchar() ఫంక్షన్‌లు

కీబోర్డ్ నుండి ఒకే అక్షరాన్ని పొందడానికి ముందే నిర్వచించిన ఫంక్షన్ అయిన **getchar()** ఉపయోగపడుతుంది. మరియు దానిని ప్రదర్శించడానికి **putchar()** ఫంక్షన్‌ను ఉపయోగించబడుతుంది.

ప్రోగ్రామ్ 11.1 C++ code to accept a character and display it

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main()
{
        cout<<"\n Type a Character : ";
        char ch = getchar();
        cout << "\n The entered Character is: ";
        putchar(ch);
        return 0;
}
```

Output:

```
Type a Character : T
The entered Character is: T
```


11th Computer Science_Chapter-11 Telugu.indd 187 30-03-2023 18:48:43

11.4.1.2. gets() మరియు puts() functions

ఫంక్షన్ gets() ప్రామాణిక ఇన్‌పుట్ నుండి string ను రీడ్ చేస్తుంది మరియు దానిని వేరియబుల్ ద్వారా సూచించబడిన string లో నిల్వచేస్తుంది. ఫంక్షన్ **puts()** కొత్తలైన్‌లో **gets()** ఫంక్షన్ ద్వారా చదవబడిన string ను ప్రింట్ చేస్తుంది.

ప్రోగ్రామ్ 11.2 C++ code to accept and display a string

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main()
{
        char str[50];
        cout<<"Enter a string : ";
        gets(str);
        cout<<"You entered: "
        puts(str);
        return(0);
}
```

Output :

```
Enter a string : Computer Science
You entered: Computer Science
```

11.4.2 Character functions (ctype.h)

ఈ Header file అక్షరాలపై వివిధ కార్యకలాపాలను (operations) ను నిర్వచిస్తుంది. C++ అందుబాటులో ఉన్న వివిధ character functions క్రింద ఇవ్వబడినవి. హెడర్ ఫైల్ అయిన ctype.h లో ఉండవలెను.

11.4.2.1.isalnum()

Character **Alphanumeric (or) not** అని తనిఖీ చేయడానికి ఈ ఫంక్షన్ ఉపయోగించబడుతుంది. ఈ c. అనే ఒక అంకె (లేదా) క్షరం అయితే ఫంక్షన్ సున్నా కాని విలువను అందిస్తుంది. లేకుంటే '0' ను అందిస్తుంది.

సాధారణ రూపం:

int isalnum (char c)

ఉదాహరణ:

int r = isalnum('5');

cout << isalnum('A') <<'\t'<<r;

కానీ క్రింద ఇవ్వబడిన statement n వేరియబుల్స్‌కు 0 ని కేటాయించాయి, ఎందుకంటే ఇచ్చిన character ఒక అంకె లేదా వర్ణమాల కాదు

char c = '$';

int n = isalnum(c);

cout<<c;

Output:

0

11th Computer Science_Chapter-11 Telugu.indd 188 30-03-2023 18:48:43

ప్రోగ్రామ్ 11.3

```
#include<iostream>
#include<stdio.h>
#include<ctype.h>
using namespace std;
int main()
{
        char ch;
        int r;
        cout<<"\n Type a Character :";
        ch = getchar();
        r = isalnum(ch);
        cout<<"\nThe Return Value of isalnum(ch) is :"<<r;
}
```

Output-1:

```
Type a Character :A
The Return Value of isalnum(ch) is :1
```

Output-2:

```
Type a Character :?
The Return Value of isalnum(ch) is :0
```

11.4.2.2 isalpha()

ఇవ్వబడిన character ఒక అక్షరమా కాదా అని తనిఖీ చేయడానికి isalpha() ఫంక్షన్‌ను ఉపయోగించబడుతుంది.

సాధారణ రూపం:

isalpha(char c)

ఇవ్వబడిన character ఒక alphabet అయితే ఈ ఫంక్షన్ 1 ని అందిస్తుంది మరియు 0 లేకపోతే '0' గా సూచిస్తుంది. క్రింది statement 0 ని వేరియబుల్ 'n' కి కేటాయిస్తుంది. ఎందుకంటే ఇచ్చిన character ఒకే alphabet.

int n = isalpha('3');

కానీ క్రింద ఇవ్వబడిన statement 1 ని ప్రదర్శిస్తుంది. ఎందుకంటే ఇచ్చిన character ఒక alphabet

cout << isalpha('a');

11th Computer Science_Chapter-11 Telugu.indd 189 30-03-2023 18:48:43

ప్రోగ్రామ్ 11.4

```
#include<iostream>
#include<stdio.h>
#include<ctype.h>
using namespace std;
int main()
{
        char ch;
        cout << "\n Enter a charater: ";
        ch = getchar();
        cout<<"\n The Return Value of isalpha(ch) is :" << isalpha(ch) ;
}
```

Output-1:

```
        Enter a charater: A
        The Return Value of isalpha(ch) is :1
```

Output-2:

```
        Enter a charater: 7
        The Return Value of isalpha(ch) is :0
```

11.4.2.3 isdigit()

ఇచ్చిన characterఒక digit అవునా? కాదా? అని తనిఖీ చేయడానికి ఈ ఫంక్షన్ ఉపయోగపడుతుంది. ఈ ఫంక్షన్ ఇచ్చిన character digit అయితే 1 ని మరియు digit కాకపోతే 0 ని అందిస్తుంది.

సాధారణ రూపం:

isdigit(char c)

ప్రోగ్రామ్ 11.5

```
using namespace std;
#include<iostream>
#include<ctype.h>
int main()
{
        char ch;
        cout << "\n Enter a Character: ";
        cin >> ch;
        cout<<"\n The Return Value of isdigit(ch) is :" << isdigit(ch) ;
}
```


11th Computer Science_Chapter-11 Telugu.indd 190 30-03-2023 18:48:43

Output-1

Enter a Character: 3
The Return Value of isdigit(ch) is :1

Output-2

Enter a Character: A
The Return Value of isdigit(ch) is :0

***Return 0; (Not Compulsory in latest compilers)**

11.4.2.4. islower()

Character, lower case (చిన్న అక్షరంలో) ఉందో లేదో తనిఖీ చేయడానికి ఈ ఫంక్షన్ ఉపయోగపడుతుంది. ఇచ్చిన character lower case alphabet అయితే ఈ ఫంక్షన్‌లు సున్నా కాని విలువలను అందిస్తాయి మరియు లేకపోతే 0ను అందించును.

సాధారణ రూపం:

islower(char c)

క్రింది statements ను అమలు చేసిన తరువాత ఇచ్చిన character lower case ఉన్నందున వేరియబుల్ **n** విలువ 1 అవుతుంది.

```
char ch = 'n';
int n = islower(ch);
```

క్రింద ఇవ్వబడిన స్టేట్‌మెంట్ విలువ n చలరాశికి '0' గా ఉండును. ఇవ్వబడిన క్యారెక్టర్ అప్పర్‌కేస్ అల్ఫాబెట్‌గా (పెద్దక్షరాలు) ఉన్నంతవరకు మాత్రమే.

```
int n = islower('P');
```

11.4.2.5. isupper()

ఇచ్చిన character uppercase అని తనిఖీ చేయడానికి ఈ ఫంక్షన్ ఉపయోగపడుతుంది. ఈ ఫంక్షన్ ఒప్పు అయితే 1 ని అందిస్తుంది. లేకపోతే '0' ను అందిస్తుంది.

సాధారణ రూపం:

isupper(char c)

క్రింది ఉదాహరణలకు కోసం **n** కి విలువ **1** ని మరియు **m** కి విలువ **0** ను కేటాయించబడును.

```
int n=isupper('A');
int m=isupper('a');
```

11.4.2.6. toupper()

ఇచ్చిన character ని దాని uppercase లోకి మార్చడానికి ఈ ఫంక్షన్ ఉపయోగించబడుతుంది. ఈ ఫంక్షన్ ఇవ్వబడిన character కు సమానమైన upper case ఉ అందిస్తుంది. ఇవబడిన character లో upper case లో ఉంటే Output కూడా upper case లో ఉండును.

సాధారణ రూపం:

char toupper(char c);

ఇచ్చిన statement variable **'K'** character స్థిరాంకాన్ని కేటాయిస్తోంది.

char c = toupper('k');

కాని క్రింద ఇవ్వబడిన statement యొక్క output **'B'** అవుతుంది.

cout <<toupper('B');

11.4.2.7. tolower()

ఇచ్చిన character ని దాని lowercase లోకి మార్చడానికి ఈ ఫంక్షన్ ఉపయోగించబడుతుంది. ఈ ఫంక్షన్ ఇచ్చిన character కి సమానమైన lower case ని అందిస్తుంది. ఇచ్చిన character lower case లో ఉంటే అప్పుడు output ఒకే విధంగా ఉంటుంది.

సాధారణ రూపం:

char tolower(char c)

ఈ క్రింది Statement variable కి స్థిరమైన 'c' కు k ను character ను కేటాయించబడుతుంది.

char c = tolower('K');

cout << c ;

11.4.3 String manipulation (string.h)

లైబ్రరీ **string.h** (cstring అని కూడా పిలుస్తారు). Array of characters లో నిల్వ చేయబడిన స్ట్రింగ్‌లతో వ్యవహరించడానికి అనేక సాధారణ విధులను కలిగి ఉంటుంది. ఏదైన string function ని ఉపయోగించే ముందు string.h అనే header file ని చేర్చాలి.

11th Computer Science_Chapter-11 Telugu.indd 191 30-03-2023 18:48:43

11.4.3.1 strcpy()

సాధారణ రూపం:

strcpy(Target String, Source String)

strcpy() ఫంక్షన్ రెండు arguments తీసుకుంటుంది. అవి Target మరియు source ఇది source ద్వారా సూచించబడిన character string ను Target సూచించిన మెమరీ స్థానానికి కాపీ చేస్తుంది. string ను జోడించబడిన null terminating character **(\0)** కూడా కాపీ చేయబడింది.

ప్రోగ్రామ్ 11.6

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
        char source[] = "Computer Science";
        char target[20]="target";
        cout<<"\n String in Source Before Copied :"<<source;
        cout<<"\n String in Target Before Copied :"<<target;
        strcpy(target,source);
        cout<<"\n String in Target After strcpy function Executed :"<<target;
        return 0;
}
```

Output:

```
String in Source Before Copied :Computer Science
String in Target Before Copied :target
String in Target After strcpy function Executed :Computer Science
```

11.4.3.2 strlen()

strlen() ఒక null terminated string ను దాని Argument గా తీసుకొంటుంది మరియు దాని పొడవును అందిస్తుంది. పొడవు null(\0)లో ఉండదు.

సాధారణ రూపం:

strlen(string)

ప్రోగ్రామ్ 11.7

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
        char source[ ] = "Computer Science";
        cout<<"\n Given String is "<<source<<" its Length is "<<strlen(source);
        return 0;
}
```

Output:

```
Given String is Computer Science its Length is 16
```


11th Computer Science_Chapter-11 Telugu.indd 192 30-03-2023 18:48:43

11.4.3.3 strcmp()

strcmp() ఫంక్షన్ రెండు arguments లను తీసుకొంటుంది. string1 మరియు string2 ఇది string1 మరియు string2 లోని విషయాలను నిఘంటుపరంగా ఇది పోల్చుతుంది.

సాధారణ రూపం:

strcpy(String1, String2)

The strcmp() function returns a:

- String 1 లోని మొదటి విభిన్నమైన character string 2 లోని సంబంధిత character కంటే ఎక్కువ ఉంటే అది positive value (ASCII విలువలు పోల్చబడ్డాయి)
- String 1 లోని మొదటి భిన్నమైన character string 2 లోని సంబంధిత character string 2 లోని సంబంధిత character కంటే తక్కువగా ఉంటే అది negative value
- String1 మరియు string 2 లు సమానంగా వుంటే అప్పుడు '0'

ప్రోగ్రామ్ 11.8

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
        char string1[] = "Computer";
        char string2[] = "Science";
        int result;
        result = strcmp(string1,string2);
        if(result==0)
        {
        cout<<"String1 : "<<string1<<" and String2 : "<<string2 <<"Are Equal";
        }
        if (result<0)
        {
        cout<<"String1 :"<<string1<<" and String2 : "<<string2 <<" Are Not Equal";
        }
}
```

Output

String1 : Computer and String2 : Science Are Not Equal

11.4.3.4 strcat()

strcat() ఫంక్షన్ రెండు ఆర్గ్యుమెంట్‌లను తీసుకొంటుంది. అవి Target మరియు Source ఈ ఫంక్షన్ Source ద్వారా సూచించబడిన character string యొక్క కాపీని Target ద్వారా సూచించబడిన string ముగింపునకు జోడిస్తుంది.

11th Computer Science_Chapter-11 Telugu.indd 193 30-03-2023 18:48:43

సాధారణ రూపం:

strcat(Target, source)

ప్రోగ్రామ్ 11.9

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
        char target[50] = "Learning C++ is fun";
        char source[50] = " , easy and Very useful";
        strcat(target, source);
        cout << target ;
        return 0;
}
```

Output

```
        Learning C++ is fun , easy and Very useful
```

11.4.3.5 strupr()

ఇచ్చిన string ను uppercase letters గా మార్చడానికి **strupr()** ఫంక్షన్ ఉపయోగించబడుతుంది.

సాధారణ రూపం:

strcat(string)

ప్రోగ్రామ్ 11.10

```
using namespace std;
#include<iostream>
#include<ctype.h>
#include<string.h>
int main()
{
        char str1[50];
        cout<<"\nType any string in Lower case :";
        gets(str1);
        cout<<"\n Converted the Source string “<<str1<<into Upper Case is "<<strupr(str1);
        return 0;
}
```

Output:

```
        Type any string in Lower case : computer science
Converted the Source string computer science into Upper Case is COMPUTER SCIENCE
```

11.4.3.6 strlwr()

ఇచ్చిన string ను lowercase letters గా మార్చడానికి **strlwr()** ఫంక్షన్ ఉపయోగించబడుతుంది.

సాధారణ రూపం:

strlwr(string)

11th Computer Science_Chapter-11 Telugu.indd 194 30-03-2023 18:48:43

ప్రోగ్రామ్ 11.11

```
using namespace std;
#include<iostream>
#include<ctype.h>
#include<string.h>
int main()
{
        char str1[50];
        cout<<"\nType any string in Upper case :";
        gets(str1);
        cout<<"\n Converted the Source string “<<str1<<into Lower Case is "<<strlwr(str1);
}
```

Output:

```
        Type any string in Upper case : COMPUTER SCIENCE
Converted the Source string COMPUTER SCIENCE into lower Case is computer science
```

11.4.4 మేథమేటికల్ ఫంక్షన్స్ (Mathematical functions - math.h)

ప్రధానంగా అన్ని రకాల మేథమేటికల్ ఫంక్షన్‌లన్నియూ ప్రాథమిక గణిత ప్రక్రియలు కల్గిన math.h హెడర్‌ఫైల్‌నందు నిర్వచించబడను.

11.4.4.1 cos() ఫంక్షన్

cos() ఫంక్షన్ అనునది అర్ధకోణాలలో (రేడియన్స్) సింగిల్ ఆర్గుమెంట్‌ను నిర్వహించును. దీని రిటర్న్ విలువ [-1, 1] పరిధిలో ఉండును. రిటర్న్ విలువ అనునది డబుల్, ఫ్లోట్ లేదా లాంగ్ డబుల్‌గా ఉండును.

ప్రోగ్రామ్ 11.12

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
        double x = 0.5, result;
        result = cos(x);
        cout << "COS("<<x<<")= "<<result;
}
```

Output:

```
        COS(0.5)= 0.877583
```

11.4.4.2 sqrt() ఫంక్షన్

ఇవ్వబడిన విలువకు వర్గమూలమును **sqrt()** ఫంక్షన్ అందించును. ఇది ఒక సింగల్ నాన్ నెగటివ్ ఆర్గ్యుమెంట్‌ను తీసుకొనును. **sqrt()** ఫంక్షన్ నందు ఒక **negative value** ఉంచినచో, **domain error లభించును**.

11th Computer Science_Chapter-11 Telugu.indd 195 30-03-2023 18:48:44

ప్రోగ్రామ్ 11.13

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
        double x = 625, result;
        result = sqrt(x);
        cout << "sqrt("<<x<<") = "<<result;
        return 0;
}
```

Output:

```
        sqrt(625) = 25
```

11.4.4.3 sin() ఫంక్షన్

sin() ఫంక్షన్ రేడియన్లలో ఒకే Argument తీసుకుంటుంది. **sin()** ఫంక్షన్ range [-1, 1] లో విలువలను అందిస్తుంది. తిరిగి వచ్చిన విలువ double, float లేదా long double లో ఉంటుంది.

11.4.4.4 pow() ఫంక్షన్

pow() ఫంక్షన్ Power of an exponent కి పెరిగిన Base ని అందిస్తుంది. ఏదైన Argument **pow()** కి long double అయితే Return type long double కి ప్రమోట్ చేయబడుతుంది. కాకపోతే, తిరిగి వచ్చే రెండింతలు అగును. **pow()** ఫంక్షన్ రెండు argument లను తీసుకొంటుంది.

- **Base** - ఆధార విలువ
- **exponent** - ఆధారం యొక్క ఘాతాంకం

ప్రోగ్రామ్ 11.14

```
#include <iostream>
#include <math.h>
using namespace std;
int main ()
{
        double base, exponent, result;
        base = 5;
        exponent = 4;
        result = pow(base, exponent);
        cout << "pow("<<base << "^" << exponent << ") = " << result;
        double x = 25;;
        result = sin(x);
        cout << "\nsin("<<x<<")= "<<result;
        return 0;
}
```

Output:

```
        pow(5^4) = 625
        sin(25)= -0.132352
```


11th Computer Science_Chapter-11 Telugu.indd 196 30-03-2023 18:48:44

11.5 వినియోగదారుని నిర్వచించిన విధులు (User-defined Functions)

11.5.1 పరిచయం (Introduction)

ఒక నిర్దిష్ట పనిని నిర్వహించడానికి కొన్ని క్రొత్త ఫంక్షన్లను నిర్వచించుటనే వినియోగదారుని నిర్వచించిన ఫంక్షన్లు అంటారు. **వినియోగదారు నిర్వచన ఫంక్షన్లు** వినియోగదారునిచే సృష్టించబడతాయి. ఒక ఫంక్షన్ ఐచ్ఛికంగా ఇన్‌పుట్ పారామితులను నిర్వహించగలదు. ఇది కాల్స్ ఫంక్షన్‌లోనికి argument పంపుటకు వీలు కల్పిస్తుంది. ఒక ఫంక్షన్ ఐచ్ఛికంగా విలువలను తిరిగి ఇవ్వగలదు. **single reusable block** లోని సాధారణ కార్యక్రమాలను జతచేయడానికి ఫంక్షను ఉపయోగపడుతాయి. ఫంక్షన్ ఏమి చేస్తున్నదని ఇది స్పష్టంగా వివరిస్తుంది.

11.5.2 ఫంక్షన్ నిర్వచనం (Function Defi nition)

C++ ప్రోగ్రామ్‌లో ఎక్కడైనా ఉపయోగించబడటానికి ముందు ఒక ఫంక్షన్‌ను నిర్వహించాలి. ఫంక్షన్ నిర్వచనం యొక్క

Return_Data_Type Function_name(parameter list)

{

Body of the function

}

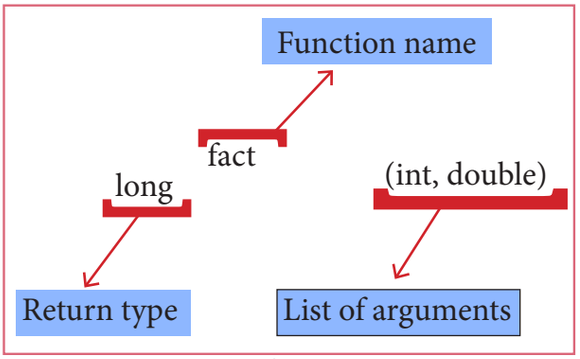
గమనిక:

1. Return_Data_Type అనునది C++ యొక్క ఏదైన చెల్లుబాటు అయ్యే డేటా రకం.
2. Function_name నిర్వచించిన వినియోగదారుని identifier.
3. పారామితి జాబితా ఇది ఐచ్ఛికం ఇది పారామితుల జాబితా అనగా డేటా రకాల ముందు మరియు కామాలతో వేరుచేయబడిన వేరియబుల్స్ జాబితా.
4. Body of the function ఉద్దేశించిన పనిని నిర్వహించడానికి అవసరమైన C++ స్టేట్మెంట్‌లను కలిగి ఉంటుంది.

11.5.3 ఫంక్షన్ Prototype

C++ ప్రోగ్రాం ఏదైనా ఫంక్షన్‌ను కలిగి ఉంటుంది. అయితే ఇది ఎల్లప్పుడూ **only one main() function** ప్రోగ్రామ్‌ను అమలు చేయడం ప్రారంభిస్తుంది. ఫంక్షన్ యొక్క నిర్వచనాలను మనం కోరుకున్న విధంగా ఏ క్రమంలోనైనా వ్రాయవచ్చు. main() ఫంక్షన్‌ను ముందుగా నిర్వచించవచ్చు మరియు దాని తరువాత అన్ని ఇతర ఫంక్షన్లలను నిర్వచించవచ్చు. ఫంక్షన్‌కి ముందు అవసరమైన అన్నీ ఫంక్షన్లను నిర్వహించగలము. వేరియబుల్ డిక్లరేషన్ వలే ప్రోగ్రాంలో ఉపయోగించబడే ముందు ఒక ఫంక్షన్ డిక్లేర్ చేయాలి. main() ఫంక్షన్ వెలుపల డిక్లరేషన్ స్టేట్‌మెంట్ ఇవ్వబడవచ్చు.

long fact (int, double)



పటము 11.1

పై పటములో ఉన్న prototype కంపైలర్ ఈ క్రింది సమాచారాన్ని అందిస్తుంది.

- ఫంక్షన్ యొక్క return value చాలా పొడవుగా ఉంటుంది.
- **Fact** అనేది ఫంక్షన్ పేరు
- ఫంక్షన్ రెండు argument తో పిలువబడుతుంది.

మొదటి argument **int డేటా** రకానికి చెందినది.

రెండవది **double** డేటా రకానికి చెందినది.

int display(int, int) // function prototype//

పై ఫంక్షన్ ప్రోటో టైపు రిటర్న్ డేటా రకం ఫంక్షన్ పేరు మరియు list of formal parameters లేదా arguments జాబితాల గురించి వివరాలను అందిస్తుంది.

11.5.4 Void command యొక్క ఉపయోగం (Use of void command)

void type రెండు ముఖ్యమైన ప్రయోజనాలను కలిగి ఉంటుంది.

- విలువను చూపించకుండా ఉండుటకు ఫంక్షన్‌ను సూచిస్తాయి.
- సాధారణ పాయింటర్‌ను ప్రకటించడానికి

11th Computer Science_Chapter-11 Telugu.indd 197 30-03-2023 18:48:44

గమనిక

void data కంపైలర్ ఫంక్షన్ విలువను ఇవ్వదని సూచిస్తుంది, లేక ఎక్కువ సందర్భాలలో శూన్యత ఏ విలువలను అంగీకరించదు.

ఉదాహరణకి:

void fun(void)

ఫంక్షన్ **fun**() కాలింగ్ ప్రోగ్రామ్ నుండి పొందదు లేదా కాలిగ్ ప్రోగ్రామ్‌కు విలువను తిరిగి ఇవ్వదని పై కనబరచిన ఫంక్షన్ ప్రోటో టైప్ కంపైలర్‌కు చెబుతుంది

11.5.5 ఫంక్షన్ యాక్సెస్ (Accessing a function)

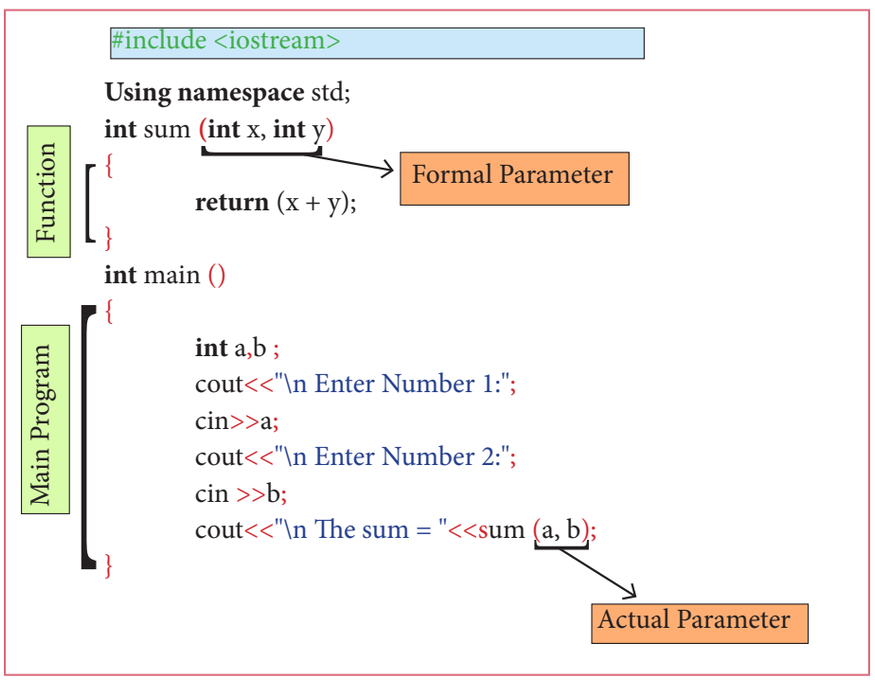
వినియోగదారు నిర్వచించిన ఫంక్షన్‌లు దాని పేరును మరియు arguments చేయడానికి అవసరమైన ఉపయోగించి స్పష్టంగా పిలవాలి. ఫంక్షన్ సరిగ్గా కాల్ చేయబడిందో లేదో అని తనిఖీ చేయడానికి కంపైలర్ ఫంక్షన్ ప్రోటో టైప్‌ను సూచిస్తుంది. ప్రోటో టైపులో నిర్వహించిన డేటా రకంతో argument రకం సరిగ్గా సరిపోలికపోతే కంపైలర్ మార్చిడి చేస్తుంది. type conversion అసాధ్యం అయితే కంపైలర్ దోష సందేశాన్ని రూపొందిస్తుంది.

ఉదాహరణ.

1	display()	calling the function without a return value and without any argument
2	display (x, y)	calling the function without a return value and with arguments
3	x = display()	calling the function with a return value and without any argument
4	x = display (x, y)	calling the function with a return value and with arguments

11.5.5.1 అధికారిక పారామితులు మరియు వాస్తవ పారామితులు లేదా వాదనలు (Formal Parameters and Actual Parameters or Arguments)

కాలింగ్ ఫంక్షన్ నుండి పిలువబడే ఫంక్షన్‌కు విలువలను పంపే మార్గాలు ఆర్గ్యుమెంట్ లేదా పారామితులు పారామితులుగా ఫంక్షన్ నిర్వచనములో ఉపయోగించే స్థిరాంకాలు. వేరియబుల్స్ లేదా వ్యక్తీకరణలను వాస్తవ పారామితులు అంటారు.



పటము 11.2 సాధారణ మరియు వాస్తవ పారామితులు

11th Computer Science_Chapter-11 Telugu.indd 198 30-03-2023 18:48:44

11.5.5.2 డిఫాల్ట్ ఆర్గ్యుమెంట్ (Default arguments)

C++, భాషలోని ఫంక్షన్ ప్రోటో టైపు యొక్క అధికారిక పారామితులకు డిఫాల్ట్ విలువలను కేటాయించవచ్చు. డిఫాల్ట్ ఆర్గ్యుమెంటు ఫంక్షను పిలిచేటప్పుడు కొన్ని విలువలను వదిలివేయడానికి అనుమతిస్తుంది. ఫంక్షను పిలిచినపుడు ఏదైనా కార్యచరణలకు విలువలను ఇవ్వకపోతే ప్రోగ్రామ్ డిఫాల్ట్ అని పిలువబడే ఫంక్షన్ కొరకు ఫంక్షన్ విలువలను అంగీకరిస్తుంది.

ఫంక్షను పిలుచునపుడు,

- తప్పిపోయిన ఏదైన ఆర్గ్యుమెంట్స్‌ల కొరకు కంప్లైయర్ అని పిలువబడే ఫంక్షన్ కోసం డిఫాల్ట్ ఆర్గ్యుమెంట్స్‌లలో విలువలను ఉపయోగిస్తుంది.
- డిఫాల్ట్ విలువ వేరియబుల్ ప్రారంభ రూపంలో ఇవ్వబడుతుంది.

ఉదాహరణ: void defaultvalue(int n1=10, n2=100);

- డిఫాల్ట్ ఆర్గ్యుమెంట్స్ పాక్షిక లేదా Argument లేకుండా ఫంక్షన్ call statements సులభతరం చేస్తాయి.

ఉదాహరణ: **defaultvalue(x,y);**
defaultvalue(200,150);
defaultvalue(150);
defaultvalue(x,150);

- డిఫాల్ట్ విలువల ఫంక్షను ప్రోటో టైపులో కుడి నుండి ఎడమకు చేర్చవచ్చు. అనగా ఆర్గ్యుమెంటు జాబితాల మధ్య ఆర్గ్యుమెంటు డిఫాల్ట్ విలువలను కలిగి ఉండకూడదు.

ఉదాహరణ: **void defaultvalue(int n1=10, n2);**//invalid prototype

void defaultvalue(int n1, n2 = 10);//valid prototype

11.5.5.3 స్థిరమైన ఆర్గ్యుమెంట్స్ (Constant Arguments)

Const keyword ఉపయోగించి constant వేరియబుల్ ప్రకటించవచ్చు. **const** keyword వేరియబుల్ విలువను స్థిరంగా చేస్తుంది. ప్రకటన చేసేటప్పుడు స్థిరమైన వేరియబుల్ ప్రారంభించబడాలి. ఫంక్షన్ **const** modifier ప్రారంభ విలువను కేటాయించడానికి అనుమతిస్తుంది. అది ఫంక్షన్ యొక్క అంతర్ భాగాన్ని మార్చబడదు.

Syntax :

<returntype><functionname> (const <datatype variable=value>)

ఉదాహరణ:

- int minimum(const int a=10);
- float area(const float pi=3.14, int r=5);

ప్రోగ్రామ్ 11.16

```
#include <iostream>
using namespace std;
double area(const double r,const double pi=3.14)
{
        return(pi*r*r);
}
int main ()
{
        double rad,res;
        cout<<"\nEnter Radius :";
        cin>>rad;
        res=area(rad);
        cout << "\nThe Area of Circle ="<<res;
        return 0;
}
```

Output:

```
Enter Radius :5
The Area of Circle =78.5
```


11th Computer Science_Chapter-11 Telugu.indd 199 30-03-2023 18:48:44

వేరియబుల్ విలువ "r" ను **r = 25** గా మార్చినట్లయితే ఫంక్షన్ **"area"** యొక్క అంతర్గతంలో కంపైలర్ ఒక దోషాన్ని **"assignment of read-only parameter 'r'"** వివరిస్తుంది.

```
double area(const double r,const double pi=3.14)
{
        r=25;
        return(pi*r*r);
}
```

11.6 కాల్ ఫంక్షన్ పద్ధతులు (Methods of calling functions)

C++ లో ఆర్గ్యుమెంట్సు రెండు విధాలుగా ఒక ఫంక్షనుకు పంపబడుతాయి. ఆర్గ్యుమెంటు దాటిని పద్ధతి ఆధారంగా ఫంక్షన్ కాలింగ్ పద్ధతులను **Call by Value method** మరియు **Call by Reference** లేదా **Address method** వర్గీకరించవచ్చు.

11.6.1 కాల్ బై వాల్యూ పద్ధతి (Call by value Method)

ఈ పద్ధతి Actual parameter యొక్క విలువను ఫంక్షన్ యొక్క formal parameter కాపీ చేస్తుంది. ఈ సందర్భంలో ఫంక్షన్‌లోని అధికారిక పారామితిలో చేసిన మార్పులు వాస్తవ పారామితిపై ప్రభావం చూపుతాయి.

ప్రోగ్రామ్ 11.17

```
#include<iostream>
using namespace std;
void display(int x)
{
        int a=x*x;
        cout<<"\n\nThe Value inside display function (a * a):"<<a;
}
int main()
{
         int a;
        cout<<"\nExample : Function call by value:";
         cout<<"\n\nEnter the Value for A :";
         cin>>a;
        display(a);
        cout<<"\n\nThe Value inside main function "<<a;
        return(0);
}
```

Output :

```
Example : Function call by value
Enter the Value for A : 5
The Value inside display function (a * a) : 25
The Value inside main function 5
```


11th Computer Science_Chapter-11 Telugu.indd 200 30-03-2023 18:48:44

11.6.2 సూచన లేదా చిరునామా ద్వారా కాల్ చేయు పద్ధతి (Call by reference or address Method)

ఈ పద్ధతి వాస్తవ ఆర్గ్యుమెంట్స్ యొక్క చిరునామాను అధికారిక పారామితిగా కాపీ చేస్తుంది. ఆర్గ్యుమెంట్ యొక్క చిరునామా ఆమోదించబడినందున అధికారిక పారామితిలో చేసిన మార్పు వాస్తవ పారామితిలోకి తిరిగి ప్రతిబింబిస్తుంది.

ప్రోగ్రామ్ 11.18

```
#include<iostream>
using namespace std;
void display(int &x) //passing address of a//
{
        x=x*x;
        cout<<"\n\nThe Value inside display function (n1 x n1) :"<<x ;
 }
int main()
{
int n1;
cout<<"\nEnter the Value for N1 :";
cin>>n1;
cout<<"\nThe Value of N1 is inside main function Before passing : "<< n1;
display(n1);
cout<<"\nThe Value of N1 is inside main function After passing (n1 x n1)  : "<< n1; return(0);
}
```

Output :

```
Enter the Value for N1 :45
The Value of N1 is inside main function Before passing : 45
The Value inside display function (n1 x n1) :2025
The Value of N1 is inside main function After passing (n1 x n1)  : 2025
```

display() ఫంక్షన్‌లో గల ఫంక్షన్ హెడర్‌లో మార్పు మాత్రమే ఉందని గమనించండి. **x** పారామితి యొక్క ప్రకటనలోని **&** గుర్తులు అంటే ఆర్గ్యుమెంట్ సూచన వేరియబుల్ మరియు ఫంక్షను సూచనలు ద్వారా పిలుస్తారు. అందువల్ల **n1** అనే ఆర్గుమెంట్స్ **display()** ఫంక్షన్‌కు పంపినప్పుడు వేరియబుల్ **xn1** యొక్క చిరునామాను పొందుతుంది, తద్వారా ఉనికిని భాగస్వామ్యం చేయబడుతుంది. మరొక మాటలో చెప్పాలంటే **x** మరియు **n1** వేరియబుల్స్ ఒకే మెమోరి స్థానాన్ని సూచిస్తుంది. మనం main() ఫంక్షన్‌లో **n1** పేరును అదే నిల్వ స్థానాన్ని సూచించడానికి **display()** ఫంక్షన్‌లో **x** పేరును ఉపయోగిస్తాము. కాబట్టి మనం **x** విలువను మార్చినప్పుడు వాస్తవానికి మనం **n1** విలువను మారుస్తున్నాము.

11.6.3 ఇన్‌లైన్ ఫంక్షన్ (Inline function)

సాధారణంగా ఒక ఫంక్షను యొక్క కాల్ స్టేట్‌మెంట్ కంపైలర్‌ను ఫంక్షన్లకు వెళ్ళునట్లు చేస్తుంది. (ఫంక్షన్ యొక్క నిర్వచనాన్ని STACKS లో నిల్వచేయబడుతుంది) మరియు కాల్స్ statement ను అనుసరించి సూచనలకు తిరిగి వెళ్ళండి. ఇది ప్రోగ్రాం వేగాన్ని తగ్గిస్తుంది. చిన్న ఫంక్షన్ నిర్వచనం కొరకు STACKS వంటి ఓవర్ హెడ్లను తగ్గించడానికి ఇన్‌లైన్ ఫంక్షను (Inline function) ని ఉపయోగించవచ్చు. ఇన్‌లైన్ ఫంక్షన్ సోర్స్‌ఫైల్‌లో సాధారణ ఫంక్షన్ లాగా కనిపిస్తుంది. కానీ ఫంక్షన్ కోడును నేరుగా కాలింగ్ ప్రోగ్రామ్‌లోకి ఇన్సర్ట్ చేస్తుంది. ఫంక్షన్ ఇన్‌లైన్ చేయడానికి ఫంక్షన్ హెడర్లో ఇన్లైను కీ వర్డ్‌ను ఇన్సర్ట్ చేయాలి.

11th Computer Science_Chapter-11 Telugu.indd 201 30-03-2023 18:48:44

ఒక **inline** ఫంక్షన్ అనునది సోర్స్‌ఫైలునందు నార్మల్ ఫంక్షన్ వలె ఉండిననూ, కాలింగ్ ప్రోగ్రాంలోకి ప్రత్యక్షంగా ఫంక్షన్ కోడ్‌లను చొప్పించును. ఒక వ్యక్తి ఇన్‌లైన్ ఫంక్షన్‌ను నిర్వహించుటకు **inline** అను కీవర్‌ను చొప్పించవలెను.

Syntax :

inline returntype functionname(datatype parameter 1, … datatype parameter n)

ఇన్‌లైన్ ఫంక్షన్ యొక్క ప్రయోజనాలు:

- ఇన్‌లైన్ ఫంక్షన్లు వేగంగా పనిచేస్తాయి. కాని ఎక్కువ మెమోరీ స్థలం అవసరం.
- STACKS ఉపయోగించడం యొక్క సంక్లిష్టతను తగ్గించండి.

ప్రోగ్రామ్ 11.19

```
#include <iostream>
using namespace std;
inline int add (int a , int b)
{
        int c=a+b;
        return(c);
}
int main ()
{
        int x,y,z;
        cout<<"\nEnter the First Number :";
        cin>>x;
        cout<<"\nEnter the second Number     :";
        cin>>y;
        z=add(x,y);
        cout << "\n sum of "<<x<<"+"<<y<<"="<<z;
        return 0;
}
```

Output:

```
Enter the First Number :10
Enter the second Number     :20
sum of 10+20=30
```

పై ప్రోగ్రాం సంకలనం సమయంలో సాధారణ ఫంక్షన్ డెఫినేషన్ ఫార్మాట్‌లో వ్రాయబడినప్పటికి ఫంక్షన్ కోడ్ statement నేరుగా కాలింగ్ **a+b** చేర్చబడుతుంది. **z=add(x,y);** ఇది కాలింగ్ **z = a+b** గా మార్చడానికి చేస్తుంది.

11.7 యూసర్ నిర్వచన ఫంక్షన్ ప్రకటన యొక్క వివిధ రూపాలు (Different forms of User-defined Function declarations)

11.7.1 రిటర్న్ విలువ లేకుండా మరియు పారామితి లేకుండా ఒక ఫంక్షన్ (A Function without return value and without parameter)

రిటర్న్ మరియు ఆర్గ్యుమెంట్స్ లేని ఫంక్షన్‌కు క్రింది ప్రోగ్రామ్ ఒక ఉదాహరణ.

11th Computer Science_Chapter-11 Telugu.indd 202 30-03-2023 18:48:44

ఫంక్షన్ పేరు **display(),** దిని రిటర్న్ డేటా రకం శూన్యం మరియు దానికి ఏ విధమైన ఆర్గ్యుమెంటు లేదు.

ప్రోగ్రామ్ 11.20

```
#include<iostream>
using namespace std;
void display()
{       cout<<"First C++ Program with Function"; }
int main()
{       display(); // Function calling statement//
        return(0);
}
```

Output :

```
        First C++ Program with Function
```

11.7.2 రిటర్న్ విలువకలిగిన మరియు పారామితి లేని ఒక ఫంక్షన్ (A Function with return value and without parameter)

ఫంక్షన్ పేరు **display()** దాని రిటర్న్ రకం int మరియు దీనికి ఎటువంటి ఆర్గ్యుమెంటు లేదు. return స్టేట్మెంట్ కాలింగ్ ఫంక్షన్‌కు విలువను తిరిగి ఇస్తుంది మరియు ప్రోగ్రాం నియంత్రణను తిరిగి కాలింగ్ స్టేట్మెంట్ బదిలీ చేస్తుంది.

ప్రోగ్రామ్ 11.21

```
#include<iostream>
using namespace std;
int display()
{
        int a=10, b=5, s;
        s=a+b;
        return s;
}
int main()
{       int m=display();
        cout<<"\nThe Sum="<<m;
        return(0);
}
```

Output :

```
        The Sum=15
```


11th Computer Science_Chapter-11 Telugu.indd 203 30-03-2023 18:48:44

11.7.3 రిటర్న్ విలువ లేని మరియు పారామితితో కూడిన ఫంక్షన్లు (A Function without return value and with parameter)

ఫంక్షన్ యొక్క పేరు **display()** దాని return రకం శూన్యం మరియు ఇది రెండు విలువలను స్వీకరించడానికి రెండు పారామితులు లేదా ఆర్గ్యుమెంట్‌లు **x** మరియు **y** కలిగి ఉండును. రిటర్న్ statement నియంత్రణను కాలింగ్ స్టేట్ మెంట్ తిరిగి ఇస్తుంది.

ప్రోగ్రామ్ 11 .22

```
#include<iostream>
using namespace std;
void display(int x, int y)
{
        int s=x+y;
        cout<<"The Sum of Passed Values: "<<s;
}
int main()
{
        int a=50,b=45;
          display(a,b);
        return(0);
}
```

Output :

The Sum of Passed Values: 95

11.7.4 రిటర్న్ విలువల మరియు పారామితితో కూడిన ఫంక్షన్ (A Function with return value and with parameter)

ఫంక్షన్ యొక్క పేరు Display() దాని రిటర్న్ టైపు int ఇది రెండు విలువలను స్వీకరించడానికి రెండు పారామితులు లేదా arguments **x** మరియు **y** కలిగి ఉండును. రిటర్న్ statement, నియంత్రణను తిరిగి దాని కాలింగ్ statement కు తిరిగి ఇస్తుంది.

ప్రోగ్రామ్ 11.23

```
#include<iostream>
using namespace std;
int display(int x, int y)
{
        int s=x+y;
        return s;
}
int main()
{
        int a=45,b=20;
        int s=display(a,b);
        cout<<"\nExample:Function with Return Value and  with Arguments";
        cout<<"\nThe Sum of Passed Values: "<<s;
        return(0);
}
```


11th Computer Science_Chapter-11 Telugu.indd 204 30-03-2023 18:48:44

Output :

Example: Function with Return Value and with Arguments

The Sum of Passed Values: 65

11.8 ఫంక్షన్ నుండి తిరిగి వచ్చుట (Returning from function)

ఫంక్షన్ నుండి తిరిగి రావడం **Return** statement ఉపయోగించి జరుగుతుంది.

Return statement అమలును ఆపివేస్తుంది మరియు కాలింగ్ ఫంక్షన్‌కు తిరిగి వస్తుంది. **Return** statement ఎగ్జిక్యూట్ అయినప్పుడు ఆ సమయంలో ఫంక్షన్ వెంటనే ముగుస్తుంది.

11.8.1 రిటర్న్ నివేదిక (The return statement)

Return statement ఒక ఫంక్షన్ నుండి తిరిగి రావడానికి ఉపయోగించబడుతుంది. ఇది jump statement గా వర్గీకరించబడింది ఎందుకంటే ఇది ఫంక్షన్ యొక్క అమలును ముగించి నియంత్రణను called statementకు బదిలీ చేస్తుంది. **Return** దానితో సంబంధం ఉన్న విలువ కలిగి ఉండవచ్చు లేదా కలిగి ఉండకపోవచ్చు. రిటర్న్‌తో అనుబంధించబడిన విలువ ఉంటే, ఆ విలువ calling statement return value అవుతుంది. పారామితి లేకుండా శూన్యమైన ఫంక్షన్‌ను return statement కొరకు ఫంక్షను ముగించడానికి ఉపయోగించవచ్చు.

Syntax:

return expression/variable;

ఉదాహరణ: return(a+b); return(a);

return; // to terminate the function

11.8.2 రిటర్న్‌ంగ్ విలువలు (Returning values):

విలువలు ఇవ్వని ఫంక్షన్లు శూన్యమైనవిగా ప్రకటించబడతాయి. డేటా రకం స్పష్టంగా పేర్కొనక పోయినట్లైతే ఫంక్షన్ యొక్క డేటా రకాన్ని **int** గా పరిగణిస్తారు.

ఉదాహరణ:

int add (int, int);

add (int, int);

రెండు ప్రోటో టైపులలోని రిటర్న్ విలువ int, ఎందుకంటే డిఫాల్ట్‌గా C++ లోని ఫంక్షన్ యొక్క రిటర్న్ విలువ స్పష్టంగా return value ఇవ్వనప్పుడు **int** రకంగా ఉంటుంది. క్రింది ఉదాహరణలు చూడండి.

Sl.No	Function Prototype	Return type
1	int sum(int, float)	int
2	float area(float, float)	float
3	char result()	char
4	double fact(int n)	double

11th Computer Science_Chapter-11 Telugu.indd 205 30-03-2023 18:48:44

పూర్ణాంకేతర విలువలను తిరిగి ఇవ్వడం

స్ట్రింగ్‌ను కాలింగ్ స్టేట్‌మెంట్‌కు కూడా తిరిగి ఇవ్వవచ్చు.

ప్రోగ్రామ్ 11.24

```
#include<iostream>
#include<string.h>
using namespace std;
char *display()
{       return (“chennai”);  }
int main()
{
        char s[50];
        strcpy(s,display());
        cout<<”\nExample:Function with Non Integer Return”<<s;
        return(0);}
```

Output :

```
Example: Function with Non Integer Return Chennai
```

11.9 పునరావృత ఫంక్షన్ (Recursive Function)

తనను తాను పిలిచే ఫంక్షను పునరావృత ఫంక్షన్ అంటారు. ఈ సాంకేతికను పునరావృత ఫంక్షన్‌ను పిలువబడును.

ఉదాహరణ 1: పునరావృత ఉపయోగించి సంఖ్య యొక్క కారకం

ప్రోగ్రామ్ 11.25

```
#include <iostream>
using namespace std;
int factorial(int); // Function prototype //
int main()
{
        int no;
        cout<<"\nEnter a number to find its factorial: ";
        cin >> no;
        cout << "\nFactorial of Number " << no <<" = " << factorial(no);
        return 0;
}
int factorial(int m)
{
        if (m > 1)
        {
        return m*factorial(m-1);
        }
        else
        {
        return 1;
}
}
```

Output :

```
Enter a number to find its factorial: 5
Factorial of Number 5 = 120
```


11th Computer Science_Chapter-11 Telugu.indd 206 30-03-2023 18:48:44

గమనిక: ఫంక్షన్ main() తర్వాత factorial() ఇవ్వబడినందున ఫంక్షన్ ప్రోటో టైపు తప్పనిసరి ఫంక్షన్

11.10 వేరియబుల్ యొక్క నియమాల పరిధి (Scope Rules of Variables)

పరిధి వేరియబుల్ యొక్క సౌలభ్యతను సూచిస్తుంది. C++ నందు నాలుగు రకాలు పరిధులు కలవు. అవి స్థానిక పరిధి, ఫంక్షన్ పరిధి, ఫైల్ పరిధి మరియు క్లాసు పరిధి.

11.10.1 పరిచయం (Introduction)

పరిధి అనునది వేరియబుల్ యొక్క ప్రాంతం లేదా జీవితం మరియు విస్తృతంగా టెప్పాలంటే మూడు స్థలాలలో వేరియబుల్స్ ప్రకటించబడతాయి.

- స్థానిక వేరియబుల్ అని పిలువబడే బ్లాక్ లోపల.
- ఫంక్షన్‌లలో ఉన్న వాటిని Function variables అంటారు.
- ఫంక్షన్లు వెలుపల ఉన్న వాటిని గ్లోబుల్ వేరియబుల్స్ అని పిలుస్తారు.
- క్లాసు లోపల క్లాస్ వేరియబుల్ లేదా డేటా సభ్యులు అంటారు.

11.10.2 స్థానిక పరిధి (Local Scope):

- స్థానిక వేరియబుల్ ఒక బ్లాక్‌లో నిర్వచించబడింది. కోడ్ యొక్క బ్లాక్ curly braces {} తో ప్రారంభమవుతుంది మరియు ముగిస్తుంది.
- స్థానిక వేరియబుల్ యొక్క పరిధి, నిర్వచించబడిన బ్లాక్‌తో సంబంధించియుండును
- స్థానిక వేరియబుల్, దాని డిక్లరేషన్ యొక్క బ్లాక్ వెలుపల నుండి యాక్సెస్ చేయబడదు.
- బ్లాక్‌లోకి ప్రవేశించిన తరువాత స్థానిక వేరియబుల్ సృష్టించబడుతుంది మరియు నిష్క్రమించిన తరువాత నాశనం చేయబడుతుంది.

11.10.3 ఫంక్షన్ పరిధి (Function Scope):

- ఒక ఫంక్షనులో ప్రకటించబడిన వేరియబుల్స్ యొక్క పరిధి ఫంక్షన్ బ్లాక్‌కు విస్తరించబడుతుంది మరియు అందులో అన్నీ ఉప బ్లాక్స్‌లు కలిగి ఉంటుంది.
- ఫంక్షన్ పరిధి వేరియబుల్ యొక్క జీవిత సమయం మరియు ఫంక్షన్ బ్లాక్ యొక్క జీవిత సమయం. పరిధి అధికారిక పారామితి యొక్క పరిధి ఫంక్షన్ స్కోపు.

11.10.4 ఫైల్ పరిధి (File Scope):

- అన్నీ బ్లాక్స్ మరియు ఫంక్షన్లు పైన ప్రకటించిన వేరియబుల్ (main ( ) తో సహా) ఫైల్ యొక్క పరిధిని కలిగి ఉంటుంది. ఫైల్ పరిధి వేరియబుల్ యొక్క జీవిత సమయం దాని ప్రోగ్రాం యొక్క జీవిత సమయం.
- ఫైల్ పరిధి వేరియబుల్‌ను **global variable** అని కూడా అంటారు.

ప్రోగ్రామ్ 11.26

```
//Demo to test all Scopes//
#include<iostream>
using namespace std;
int file_var=20;          //Declared within File - file scope variable
void add(int x)
{
        int m;            //Declaration of  variable m in add () - Function scope variable
        m=x+30+file_var;
        cout<<"\n The Sum = "<<m;
}
```


11th Computer Science_Chapter-11 Telugu.indd 207 30-03-2023 18:48:44

```
int main ( )
{
int a ;
a = 10;
if(a>b)
{
        int t;              // local to this if block - Local variable
        t=a+20;
 }
cout<<t;
add(a);
cout<<m;
cout<<"\nThe File Variable = "<<file_var;
return(0);
}
```

Error
In function 'int main()':
[Error] 't' was not declared in main()

On compilation the Program 11.28, the compiler prompts an error message: The variable **t** is not accessible. Because the life time of a local variable is the life time of a block in its state of execution.

[Error] 'm' was not declared in this scope

The variable **m i**s not accessible. Because the life time of the function scope variable is the life time of a block in its state of execution.

11.10.5 క్లాసు పరిధి (Class Scope):

- క్లాసు అనునది యూసర్ నిర్వచించిన డేటా రకమును సృష్టించడానికి మరియు అమలు చేయడానికి కొత్త మార్గం. క్లాసులు వివిధ రకాల డేటాను కలిసి కూర్చుటకు ఒక పద్ధతిని అందిస్తాయి.
- డేటా సభ్యులు ఒక క్లాసు యొక్క లక్షణాలను సూచించే డేటా వేరియబుల్స్.

class student
{
private :
int mark1, mark2, total;
};
	తరగతి విద్యార్థి కలిగిన మార్క్ 1 మార్క్ 2 మరియు మొత్తం డేటా వేరియబుల్స్. దీని పరిధి తరగతి విద్యార్థులో మాత్రమే ఉంటుంది.

గమనిక: క్లాసు పరిధి తదుపరి క్లాసులు మరియు ఆబ్జెక్టు అధ్యాయంలో చర్చించబడుతుంది.

11th Computer Science_Chapter-11 Telugu.indd 208 30-03-2023 18:48:44

11.10.6 పరిధి స్పష్టత ఆపరేటరు (Scope resolution operator)

పరిధి ఆపరేటరు వేరియబుల్ యొక్క దాగిన పరిధిని వెల్లడిస్తుంది. పరిధి స్పష్టత ఆపరేటర్ (::) క్రింది ప్రయోజనాల కోసం ఉపయోగించబడుతుంది.

- అదే పేరుతో స్థానిక వేరియబుల్ ఉన్నప్పుడు గ్లోబల్ వేరియబుల్ యాక్సెస్ చేయడానికి పరిధి స్పష్టత ఆపరేటరుకు ఉదాహరణ.

ప్రోగ్రామ్ 11.27

```
// Program to show that we can access a global variable
// using scope resolution operator :: when there is a local
// variable with same name //
#include<iostream>
using namespace std;
int x=45;  // Global Variable x
int main()
{
 int x = 10; // Local Variable x
 cout << "\nValue of global x is " << ::x;
 cout << "\nValue of local x is " << x;
 return 0;
}
```

Output:

```
Value of global x is 45
Value of local x is 10
```

గుర్తుంచుకొనవలసినది

- ఒక పెద్ద ప్రోగ్రామ్‌ను సాధారణంగా చిన్న బ్లాక్‌లుగా విభజించవచ్చు..
- ఫంక్షన్లు ముందుగా నిర్వచించిన లేదా అంతర నిర్మిత లేదా గ్రంథాలయ విధులు మరియు యూసర్ నిర్వచించిన ఫంక్షన్లుగా వర్గీకరించవచ్చు.
- యూసర్ నిర్వచించిన ఫంక్షన్లు వినియోగదారుల ద్వారా సృష్టించబడతాయి.
- శూ్య ఫంక్షన్ కంపైలర్‌కు ఫంక్షన్ ఏమి తిరిగి ఇవ్వదని చెబుతుంది.
- రిటర్న్ స్టేట్ మెంట్ కాలింగ్ ఫంక్షన్‌కు విలువను తిరిగి ఇస్తుంది మరియు ప్రోగ్రాం నియంత్రణను తిరిగి కాలింగ్‌కు బదిలీ చేస్తుంది.
- C++ లోని ఫంక్షన్ యొక్క డిఫాల్ట్ రిటర్న్ int రకం
- తనను తాను పిలిచే ఫంక్షన్లు పునరావృత ఫంక్షన్ అంటారు.
- పరిధి వేరియబుల్ యొక్క ప్రాముఖ్యతను సూచిస్తుంది.
- పరిధిలో నాలుగు రకాలున్నాయి. అవి: స్థానిక పరిధి, ఫంక్షన్ పరిధి ఫైల్ పరిధి మరియు క్లాసు పరిధి
- పరిధి ఆపరేటరు (::) వేరియబుల్ యొక్క దాగిన పరిధిని వెల్లడిస్తుంది.

11th Computer Science_Chapter-11 Telugu.indd 209 30-03-2023 18:48:44



చేసిచూడుము

క్రింది సమస్యలను పరిష్కరించడానికి C++ ప్రోగ్రామ్‌ను వ్రాయండి :

1. రెండు స్ట్రింగులను చదివే ప్రోగ్రామ్‌ను మరియు మొదటి స్ట్రింగ్‌ను రెండవ దానికి కలిపే ప్రోగ్రాం. ఉదాహరణు మొదటి స్ట్రింగ్ తమిళ్ మరియు రెండవ స్ట్రింగ్ నాడుగా నమోదు చేయబడితే ప్రోగ్రాం తమిళనాడును ముద్రించాలి. స్ట్రింగ్ లైబ్రరీ హెడర్‌ను ఉపయోగించాలి.
2. స్ట్రింగ్ చదివి పెద్ద అక్షరానికి మార్చే ప్రోగ్రామ్ అవసరమైన హెడర్ ఫైళ్ళను చేర్చండి
3. ఇచ్చిన అక్షరం వర్ణమాల కాదా అని తనిఖీ చేసే ప్రోగ్రాం ఇది వర్ణమాల అయితే అది చిన్న అక్షరం లేదా పెద్ద అక్షరరా అవసరమైన హెడర్ ఫైళ్ళను చేర్చండి
4. c++ లో ఫంక్షన్ సమ్మరీస్ ( ) కోసం రెండు ఆర్గుమెంట్స్ / పారామితులతో double x మరియు int n తో నిర్వచనం రాయండి ఫంక్షన్ రకం రెట్టింపు విలువను తిరిగి ఇవ్వాలి మరియు ఇది క్రింది శ్రేణి మొత్తాన్ని ప్రదర్శించాలి.

   x-x2 /3! + x3 / 5! - x4 / 7! + x5 / 9! -... నిబంధనల వరకు.

5. ఫంక్షన్ ( ) ను ప్రారంభించే ప్రోగ్రామ్ ఇది రెండు పూర్ణాంకాలు మరియు అంకగణిత ఆపరేటర్లను తీసుకొంటుంది మరియు సంబంధిత ఫలితాన్ని ముద్రిస్తుంది.



మూల్యాంకనము



విభాగం – అ

సరైన జవాబును ఎన్నుకొనుము

1. క్రింది వాటిలో ఏది హెడర్ ఫైల్ ప్రామాణిక I/O ముందే నిర్వచించిన విధులను నిర్వచిస్తుంది:
   అ) stdio.h ఆ) math.h ఇ)) string.h ఈ) ctype.h
2. అక్షరాలని వర్ణమాలా కాదా అని తనిఖీ చేయడానికి ఏ ఫంక్షన్ ఉపయోగించబడుతుంది?
   అ) isalpha() ఆ) isdigit() ఇ) isalnum() ఈ) islower()
3. ప్రోగ్రామ్ అమలును ఏ ఫంక్షను ప్రారంభిస్తుంది?
   A) isalpha() ఆ) isdigit() ఇ) main() ఈ) islower()
4. క్రింది వాటిలో ఏది రిటర్న్ విలువతో మరియు ఎటువంటి ఆర్గ్యుమెంటు లేకుండా ఉంటుంది?
   అ) x=display(int, int) ఆ) x=display() ఇ) y=display(float) ఈ) display(int)
5. ఇది (int, int); ఫంక్షన్ ప్రోటో టైపు యొక్క రిటర్న్ డేటా రకం?
   అ) int ఆ) float ఇ) char ఈ) double
6. కింది వాటికి పరిధి ఆపరేటర్ ఎవరు?
   అ) > ఆ) & ఇ) % ఈ) ::

11th Computer Science_Chapter-11 Telugu.indd 210 30-03-2023 18:48:44

విభాగం-ఆ

లఘు సమాధాన ప్రశ్నలు

1. "ఫంక్షన్" నిర్వచనం తెల్పుము.
2. strlen() ఫంక్షన్ గురించి వ్రాయుము.
3. void డేటా టైప్ యొక్క ముఖ్యత్వం ఏమిటి?
4. పారామితి (Parameter) అనగానేమి? అందులోని రకాలేవి?
5. 'లోకల్ స్కోప్' గురించి ఒక జ్ఞాపిక వ్రాయుము.

విభాగం-ఇ

సంక్షిప్త సమాధాన ప్రశ్నలు

1. బిల్ట్–ఇన్ ఫంక్షన్స్ అంటే ఏమిటి?
2. isupper() మరియు toupper() ఫంక్షన్స్ మధ్యగల వ్యత్యాసాలేవి?
3. strcmp() ఫంక్షన్ గురించి వ్రాయుము.
4. C++ యందలి pow() ఫంక్షన్ గురించి వ్రాయుము.
5. కంపైలర్‌కు ప్రోటోటైప్ అందించెడి సమాచారాలేవి?
6. డిఫాల్ట్ ఆర్గ్యుమెంట్స్ అంటే ఏమిటి? ఉదాహరణలిమ్ము.

విభాగం - ఈ

విపుల సమాధాన ప్రశ్నలు

1. తగిన ఉదాహరణలతో Call by value పద్ధతిని వివరించుము.
2. రికర్షన్ (Recursion) అనగానేమి? రికర్షన్‌ను ఉపయోగించి ఇవ్వబడిన సంఖ్య యొక్క factorial ను కనుగొనెడి ప్రొగ్రాంను రచించుము.
3. ఫంక్షన్ రిటర్న్ యొక్క వివిధ రూపాలను ఉదాహరణ సహితంగా వివరింపుము.
4. ఉదాహరణతో 'స్కోప్ ఆఫ్ వేరియబుల్' ను వివరింపుము.
5. ఏదేని పూర్ణ సంఖ్యను ఆమోదించెడి మరియు దానిని మరల్చెడి ప్రొగ్రాంను వ్రాయుము.

11th Computer Science_Chapter-11 Telugu.indd 211 30-03-2023 18:48:44

అధ్యాయము 12

అరేస్ మరియు నిర్మాణాలు (Arrays and Structures)

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును అభ్యసించిన తరువాత విద్యార్థులు నేర్చుకొను అంశాలు

- Arrays ఉపయోగించి నిర్మాణాత్మక డేటా రకాన్ని తెలుసుకోవడం
- Arrays రకాలను తెలుసుకోవడం
- వివిధ రకాల Arrays మార్చటానికి ప్రోగ్రామ్‌లను వ్రాయడం

12.1 పరిచయం (Introduction)

డేటాను నిల్వచేయడానికి వేరియబుల్స్ ఉపయోగించబడతాయి. ఈ వేరియబుల్స్ C++ లోని ప్రాథమిక బిల్డింగ్ బ్లాకులలో ఒకటి. మెమరీలో ఎక్కడైనా ఉపయోగించగల ఒకే విలువను నిల్వ చేయడానికి ఒకే వేరియబుల్ ఉపయోగించబడుతుంది. కొన్ని సందర్భాల్లో మనము ఒకే రకమైన బహుళ విలువలను నిల్వ చేయాలి. అలాంటప్పుడు దీనికి ఒకే డేటా రకం యొక్క బహుళ వేరియబుల్స్ అవసరం. అన్ని విలువలు మెమరీలో ఎక్కడైనా యాదృచ్ఛికంగా నిల్వ చేయబడతాయి.

ఉదాహరణకు 100 మంది విద్యార్థుల వరుస సంఖ్యలను నిల్వచేయడానికి దీనికి roll1, roll2, roll3,…….roll100 అని పేరు పెట్టబడిన 100 వేరియబుల్స్ అవసరం. 100 వేరియబుల్స్ తెలియజేయడం మరియు అన్ని రోల్ సంఖ్యలను నిల్వచేయడం చాలా కష్టం అవుతుంది. C++ లో బహుళ విలువలను వేరియబుల్‌లో నిల్వచేయడానికి ఈ Array భావన ఉపయోగపడుతుంది. అక్షరాలా Array అనగా “ఒకటి కంటే ఎక్కువ”. మరో మాటలో Array అనేది ఒక సాధారణ పేరు ద్వారా సూచించబడిన బహుళ విలువలను నిల్వచేయడానికి ఉపయోగించే సులభమైన పేరు. Array అనేది C++ లో వర్గీకరించబడిన డేటా రకం.

Array అనేది ఒక సాధారణ పేరు ద్వారా సూచించబడే ఒకే రకమైన వేరియబుల్స్ యొక్క సమాహారం. Array విలువలు ఒకే రకమైన మూలకాల యొక్క స్థిర సంఖ్యలో వరుసగా మెమరీలో నిల్వ చేయబడతాయి. అందువల్ల, పూర్ణాంక Array పూర్ణాంకాల క్రమాన్ని కలిగి ఉంటుంది. అక్షర శ్రేణి అక్షరాల క్రమాన్ని కలిగి ఉంటుంది. Array యొక్క పరిమాణాన్ని దాని కొలతలుగా సూచిస్తారు.

12.2 శ్రేణి రకాలు (Types of Arrays)

C++ లో వివిధ రకాల Arrays ఉపయోగించబడతాయి.

- ఏక – పరిమాణ Array
- ద్వి – పరిమాణ Array
- బహుళ – పరిమాణ Array

12.2.1 ఏక – పరిమాణ array (One-dimensional array)

ఇది Array యొక్క సరళమైన రూపం. One dimensional array అనునది విలువలను ఒకే అడ్డు వరుసలో లేదా ఒకే నిలువ వరుసలో నిల్వ చేయబడుతుంది.

Declaration

Syntax:

<data type><array_name> [<array_size>];

data_type అనునది array యొక్క ప్రాథమిక రకాన్ని ప్రకటిస్తుంది. ఇది array లోని ప్రతి మూలకం యొక్క రకం

Array_name ప్రస్తావించబడే పేరును నిర్దేశిస్తుంది.

Array_size పరిమాణం ఎన్ని అంశాలను కలిగి ఉందో నిర్వచిస్తుంది. పరిమాణాన్ని చదరపు బ్రాకెట్ల [ ] తో పేర్కొనాలి.

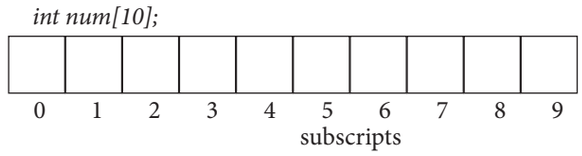
ఉదాహరణ:

```
int num[10];
```


11th Computer Science_Chapter-12 Telugu.indd 212 23-03-2023 15:38:16

పై ప్రకటనలో "num" అనే శ్రేణి 10 అంశాలతో పూర్ణాంక రకంగా ప్రకటించబడుతుంది. (10 వేర్వేరు విలువలను నిల్వచేయడానికి మెమరీ స్థలం)

పై ప్రకటనలో ఈ కంపైలర్ ఒక common name మెమరీ locations ని కేటాయిస్తుంది.



ప్రతి మూలకం (మెమరీ బాక్స్) 0 నుండి ప్రారంభమయ్యే ప్రత్యేక సూచిక సంఖ్యను "సబ్ స్క్రిప్ట్" అంటారు. సబ్‌స్క్రిప్ట్ ఎల్లప్పుడూ 0 తో మొదలవుతుంది మరియు ఇది జతచేయని పూర్ణాంక విలువగా ఉండాలి. శ్రేణి యొక్క ప్రతి మూలకం చదరపు బ్రాకెట్‌లోని సబ్‌స్క్రిప్ట్ సూచికతో దాని పేరుతో సూచించబడుతుంది. ఉదాహరణకు num[3] నాల్గవ మూలకాన్ని సూచిస్తుంది.

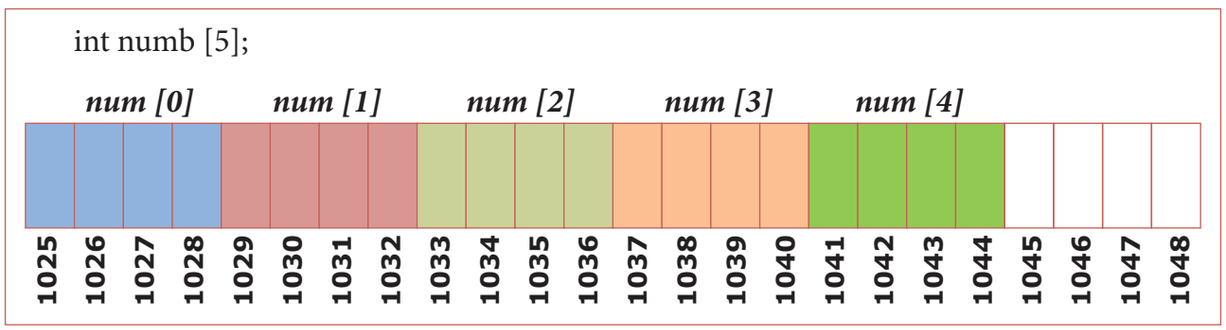
వివిధ డేటా రకాలతో మరికొన్ని array ప్రకటన:
char emp_name[25]; //; emp_name అనునది పరిమాణం 25 తో కూడిన అక్షర శ్రేణి.

float salary[20]; //; salary అనునది పరిమాణం 20 తో వున్న ఫ్లోటింగ్ పాయింట్ Array

int a[5], b[10], c[15]; // int రకం అనునది బహుళ Array గా ప్రకటించబడుతుంది.

One dimensional array యొక్క మెమరీ ప్రాతినిధ్యం

Array ఉంచడానికి అవసరమైన నిల్వ మొత్తం నేరుగా రకం మరియు పరిమాణంతో సంబంధం కలిగి వుంటుంది. ఈ క్రింది పటము ఐదు అంశాలతో కూడిన Array యొక్క మెమరీ కేటాయింపును చూపుతుంది.



పైన చూపిన పటములో num అనే Array 5 మూలకాలతో కూడిన పూర్ణాంక శ్రేణిగా చూపిస్తుంది. Dev-C++ కంపైలర్ ప్రకారం ప్రతి int రకం వేరియబుల్‌కు 4 బైట్లు కేటాయించబడతాయి. ఇక్కడ array or 5 మూలకాలు ఉన్నాయి. ఇక్కడ ప్రతి మూలకానికి 4 బైట్లు కేటాయించబడతాయి. పూర్తిగా ఈ శ్రేణికి 20 బైట్లు కేటాయించబడతాయి.

Datatype	Turbo C++	Dev C++
char	1	1
int	2	4
float	4	4
long	4	4
double	8	8
long double	10	10

శ్రేణి కోసం కేటాయించిన మెమరీ స్థలాన్ని ఈ క్రింది సూత్రాన్ని ఉపయోగించి లెక్కించవచ్చు. శ్రేణి రకం కోసం కేటాయించిన బైట్ల సంఖ్య × మూలకాల సంఖ్య.

11th Computer Science_Chapter-12 Telugu.indd 213 23-03-2023 15:38:16

ఇన్సిలైజేషన్

Array ని ప్రకటన చేసే సమయంలో ఇన్సిలైజేషన్ చేయవలెను. అనవసరంగా array ని ఇన్సిలైజ్డ్ చేయడం వలన అన్ని array మూలకాలు array విలువలను కలిగి ఉంటుంది.

ఉదాహరణ:

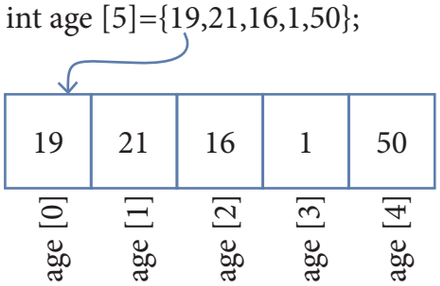
<datatype> <array_name> [size] = {value-1,value-2,…… ,value-n};

ఉదాహరణ:

int age[5]={19,21,16,1,50};

పై ఉదాహరణలో array పేరు 'age'. దీని పరిమాణం 5. ఈ సందర్భంలో మొదట మూలకం age [0] లో మూలకం 19 అనే సంఖ్య నిల్వచేయపడుతుంది. రెండవ మూలకం age [1] లో నిల్వచేయబడుతుంది. ఇదే విధంగా పటం 12.1 లో చూపించబడుతుంది.

int age [5]={19,21,16,1,50};



పటము 12.1

arrayవిలువలనుప్రకటించేటప్పుడుమరియు ప్రారంభించేటప్పుడు విలువలు curly braces లో { ….. } ఇస్తారు.

Array ప్రకటించే సమయంలో array size అనునది ఐచ్చికం మాత్రమే.

ఉదాహరణ:

int age[]={ 19,21,16,1,50};

పై ఇన్సిలైజేషన్‌లో ఇన్సిలైజేషన్‌తో ప్రకటించబడిన array యొక్క పరిమాణము నేరుగా వర్గీకరించలేదు. కాబట్టి కంపైలర్ దీని మొత్తం విలువలపై ఆధారపడి పరిమాణమును నిర్ధారిస్తుంది. ఈ సందర్భంలో array యొక్క పరిమాణం '5'.

array ప్రారంభానికి మరిన్ని ఉదాహరణలు

float x[5] = {5.6, 5.7, 5.8, 5.9, 6.1};

char vowel[6] = {'a', 'e', 'i', 'o', 'u', '\0'};

రన్‌టైమ్‌లో విలువలను array అంగీకరించడం :

ప్రోగ్రామ్‌ను అమలు చేసేటప్పుడు array ఒక విలువను మాత్రమే ఇన్‌పుట్ చేయగలదు. మీరు ఒకటి కంటే ఎక్కువ విలువలను ఇన్పుట్ చేయాలనుకుంటే మీరు ఏదైనా ఒక లూపును ఉపయోగించవచ్చు. శ్రేణి మూలకాల ద్వారా మళ్ళించడానికి array లూప్ ఆదర్శంగా సరిపోతుంది.

```
// ప్రోగ్రామ్ రన్‌టైమ్‌లో విలువలను ఇన్‌పుట్‌గా పొందడం
#include <iostream>
using namespace std;
int main()
{
        int num[5];
        for(int i=0; i<5; i++)
        {
                cout<< "\n Enter value " << i+1 << "= ";
                cin>>num[i];
        }
}
```

పై ప్రోగ్రామ్‌లో లూప్‌లోని స్టేట్‌మెంట్‌లను 5 సార్లు అమలు చేయడానికి ఒక for లూప్ నిర్మించబడింది. లూప్ రన్ అయిన ప్రతిసారీ *cout* స్టేట్మెంట్ "Enter value ……" స్టేట్మెంట్‌ను తిరిగి ఇస్తుంది మరియు ఇన్పుట్ విలువ *cin* ద్వారా సంగ్రహించి num[i] లో నిల్వచేయబడుతుంది.

11th Computer Science_Chapter-12 Telugu.indd 214 23-03-2023 15:38:16

ఈ క్రింది పట్టిక పై కోడ్ బ్లాక్ యొక్క అమలును చూపుతుంది.

Iteration	i <5	cout << "\n Enter value " << i+1 << "= ";	cin>>num [i];	Received value stored in memory		i++ (i=i+1)
1	5 > 0 (T)	Enter value 1 =	num[0] = 5	num[0]	5	1
2	5 > 1 (T)	Enter value 2 =	num[1] = 10	num[1]	10	2
3	5 > 2 (T)	Enter value 3 =	num[2] = 15	num[2]	15	3
4	5 > 3 (T)	Enter value 4 =	num[3] = 20	num[3]	20	4
5	5 > 4 (T)	Enter value 4 =	num[25 = [4	num[4]	25	5
6	5 > 5 (F)	Exit from Loop				

గమనిక:

లూప్ కోసం సూచిక ***i*** ప్రారంభ విలువ 0 (సున్నా) తో ప్రకటించబడుతుంది. చాలా సందర్భాలలో లూప్ సూచిక యొక్క ప్రారంభ విలువ array సబ్‌స్క్రిప్ట్ ప్రాతినిధ్యంగా ఉపయోగించబడుతుంది.

array మూలకాలను యాక్సెస్ చేయడం:

మనము సాధారణ వేరియబుల్స్ మాదిరిగానే ప్రోగ్రామ్‌లో ఎక్కడైనా array అంశాలను ఉపయోగించవచ్చు. array యొక్క అంశాలను స్వీకరించడానికి మీరు array పేరుతో చదరపు బ్రాకెట్లలో సబ్‌స్క్రిప్ట్ ఇండెక్స్ విలువను ఇవ్వాలి.

ఉదాహరణ:

cout<<num[3];

పై వివరణలో సంఖ్య [3] శ్రేణి యొక్క 4 వ మూలకాన్ని సూచిస్తుంది మరియు *cout* స్టేట్‌మెంట్ num[3] విలువను ప్రదర్శిస్తుంది.

గమనిక:

బ్రాకెట్‌లో ఉన్న సబ్‌స్క్రిప్ట్ అనునది ఒక వేరియబుల్ ఒక స్థిరాంకం లేదా పూర్ణాంకం యొక్క expression ని మూల్యాంకనం చేయును.

// శ్రేణి మూలకాలను యాక్సెస్ చేయడం

```
#include <iostream>
using namespace std;
int main()
{
        int num[5] = {10, 20, 30, 40, 50};
        int t=2;
        cout<<num[2] <<endl; // S1
        cout<<num[3+1] <<endl; // S2
        cout<<num[t=t+1]; // S3
}
```

output:

```
30
50
40
```


11th Computer Science_Chapter-12 Telugu.indd 215 23-03-2023 15:38:16

పై ప్రోగ్రామ్‌లో స్టేట్‌మెంట్ **S1** అనేది 3 వ మూలకం యొక్క విలువను ప్రదర్శిస్తుంది. (సబ్‌స్క్రిప్ట్ ఇన్‌డెక్స్ 2) **S2** అనునది 5 వ మూలకం యొక్క విలువను ప్రదర్శిస్తుంది. (సబ్‌స్క్రిప్ట్ ఇండెక్స్ 3+1 = 4) అదే విధంగా స్టేట్‌మెంట్ **S3** నాల్గవ మూలకం యొక్క విలువను ప్రదర్శిస్తుంది. (C++ భాష అనేది చదరపు బ్రాకెట్లలో ఇచ్చిన వ్యక్తీకరణను t=2+1 = 3 గా లెక్కిస్తుంది)

C++ ప్రోగ్రామ్‌లో 10 విలువలను ఇన్పుట్ చేయడానికి మరియు బేసి, సరి సంఖ్యలను లెక్కించడం.

```
#include <iostream>
using namespace std;
int main()
{
        int num[10], even=0, odd=0;
        for (int i=0; i<10; i++)
        {
                cout<< "\n Enter Number " << i+1 <<"= ";
                cin>>num[i];
                if (num[i] % 2 == 0)
                 ++even;
                else
                 ++odd;
        }
        cout << "\n There are "<< even <<" Even Numbers";
        cout << "\n There are "<< odd <<" Odd Numbers";
}
```

Output:

```
Enter Number 1= 78
Enter Number 2= 51
Enter Number 3= 32
Enter Number 4= 66
Enter Number 5= 41
Enter Number 6= 68
Enter Number 7= 27
Enter Number 8= 65
Enter Number 9= 28
Enter Number 10= 94
There are 6 Even Numbers
There are 4 Odd Numbers
```

(HOTS : పైన పేర్కొన్న ప్రోగ్రామ్‌ను if స్టేట్‌మెంట్‌కు బదులుగా కండిషనల్ ఆపరేటర్ ఉపయోగించి వ్రాయండి)

one dimensional array ను శోధించడం:

ఇచ్చిన సంఖ్యల సమితి నుండి నిర్ధిష్ట విలువ కనుగొనే ప్రక్రియను శోధన అంటారు. ఒక సరళ శోధన లేదా సీక్వెన్షియల్ సెర్చ్ శోధించిన విలువను జాబితాలోని మూలకాల విలువతో పోల్చి ఆ విలువ కనుగొనబడే వరకు జట్టులోని అన్ని మూలకాలను శోధిస్తుంది.

11th Computer Science_Chapter-12 Telugu.indd 216 23-03-2023 15:38:16

లీనియర్ సెర్చ్ కోసం ప్రోగ్రామ్

```
#include <iostream>
using namespace std;
int main()
{
        int num[10], val, id=-1;
        for (int i=0; i<10; i++)
        {
        cout<< "\n Enter value " << i+1 <<"= ";
        cin>>num[i];
        }
        cout<< "\n Enter a value to be searched: ";
        cin>>val;
    for (int i=0; i<size; i++)
    {
          if (arr[i] == value)
          {    id= i;
               break;
          }
    }
        if(id==-1)
        cout<< "\n Given value is not found in the array..";
        else
        cout<< "\n The value is found at the position" << id+1;
        return 0;
}
```

పై ప్రోగ్రామ్, శోధించిన విలువను జాబితాలోని మూలకాల విలువతో పోల్చి ఆ విలువను కనుగొనే వరకు జట్టులోని అన్ని అంశాలను దాటుతూ ప్రయాణిస్తుంది.

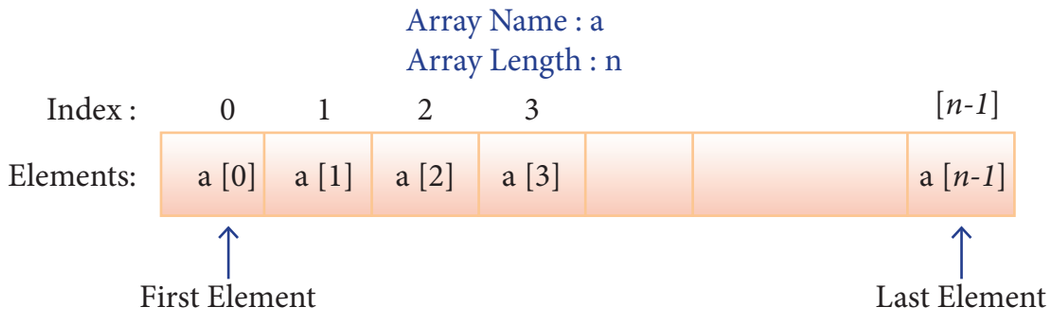
స్ట్రింగ్స్

చిహ్నాల క్రమాన్ని స్ట్రింగ్ అని పిలుస్తారు. ఇక్కడ చిహ్నం అనేది అక్షరం సంఖ్య లేదా చిహ్నంగా ఉండవచ్చు ప్రతి మూలకం ఒక బైట్ మెమరీని ఆక్రమిస్తుంది. ప్రతి స్ట్రింగ్ శూన్య ('\0', ASCII కోడ్ 0) అక్షరంతో ముగించబడుతుంది. ఇది స్ట్రింగ్ చివరిలో జతచేయబడాలి. C++ లో స్ట్రింగ్‌ను సూచించడానికి ప్రత్యేక డేటా రకం లేదు. బదులుగా స్ట్రింగ్‌నుone-dimensional character array గా అమలుచేస్తుంది. character array ని ప్రకటించేటప్పుడు ఇది null character ను చివరలో నిలిపి ఉంచుతుంది. మరియు character array యొక్క పరిమాణం స్ట్రింగ్ యొక్క పొడవు కన్నా ఒక character పొడవుగా ఉంటుంది.

Character Array (స్ట్రింగ్) సృష్టి:

Array, సృష్టించే ముందు దాని పరిమాణం (పొడవు) తెలుసుకోవాలి. తద్వారా శ్రేణి యొక్క పరిమాణానికి అనుగుణంగా మెమరీ స్థానాలు కేటాయించబడుతుంది. Array సృష్టించబడిన తర్వాత దాని పొడవు స్థిరంగా ఉండును. ప్రోగ్రామ్ రన్ సమయంలో దీనిని మార్చుటకు వీలుకాదు. (ఇది పటము 12.2 లో చూపబడింది)

11th Computer Science_Chapter-12 Telugu.indd 217 23-03-2023 15:38:16



పటము 12.2

సింటాక్స్

Array ప్రకటన:

char array_name[size];

పైన ఇచ్చిన ప్రకటనలో జట్టు పరిమాణం ఎల్లప్పుడూ గుర్తించబడని పూర్ణాంక విలువగా ఉండాలి.

ఉదాహరణ,

char country[6];

ఇక్కడ character array నిల్వ చేయడానికి, array 6 బైట్ల మెమరీని కలిగి వుంటుంది. స్ట్రింగ్ యొక్క పొడవు 5 అక్షరాల కంటే ఎక్కువ ఉండకూడదు మరియు చివరిలో శూన్య అక్షరం కోసం ఒక స్థానం కేటాయించబడినది.

// అక్షర శ్రేణిని ప్రదర్శించే ప్రోగ్రామ్:

```
#include <iostream>
using namespace std;
int main()
 {
        char country[6];
        cout<< "Enter the name of the country: ";
        cin>>country;
        cout<<" The name of the country is "<<country;
}
```

OUTPUT

```
Enter country the name: INDIA
The country name is INDIA
```

ప్రారంభించడం

character arrayను డిక్లరేషన్ సమయంలో ప్రారంభించవచ్చు. దాని వాక్య నిర్మాణం క్రింద చూపబడింది:

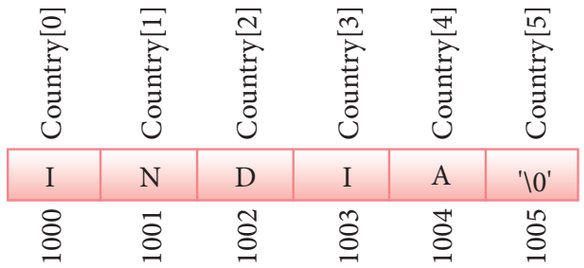
char array_name[size]={ list of characters separated by comma or a string } ;

ఉదాహరణకు,

char country[6]="INDIA";

పై ఉదాహరణలో "INDIA" అను పదంలో 5 అక్షరాలు వున్నాయి. ఇవి country అనెడి శ్రేణి కి ప్రారంభ విలువగా కేటాయించబడతాయి. టెక్స్ట్ డబుల్ కోట్స్‌లో జతచేయబడింది. మెమరీ ప్రాతినిధ్యం పటము 13.3 లో చూపబడింది.

11th Computer Science_Chapter-12 Telugu.indd 218 23-03-2023 15:38:16



పటము 12.3

పై మెమరీ ప్రాతినిధ్యంలో ప్రతి అక్షరం ఒక బైట్ మెమరీని ఆక్రమిస్తుంది. స్ట్రింగ్ చివరిలో కంపైలర్ చేత null character స్వయంచాలకంగా జోడించబడుతుంది.

C++ character array ప్రారంభించడానికి ఇతర మార్గాలను కూడా అందిస్తుంది.:

char country[6]={‘I’, ‘N’, ‘D’, ‘I’, ‘A’, ‘\0’};

char country[]=“INDIA”;

char country[]={‘I’, ‘N’, ‘D’, ‘I’, ‘A’, ‘\0’};

శ్రేణి యొక్క పరిమాణం స్పష్టంగా పేర్కొనకపోతే, కంపైలర్ స్వయంచాలకంగా జాబితాలోని మూలకాల సంఖ్య ఆధారంగా శ్రేణి యొక్క పరిమాణాన్ని లెక్కిస్తుంది మరియు తదనుగుణంగా స్థలాన్ని కేటాయిస్తుంది.

స్ట్రింగ్ యొక్క ప్రారంభంలో అన్ని అక్షరాలు ప్రారంభించబడకపోతే మిగిలిన అక్షరాలు శూన్యంతో (null) నిండి వుంటాయి.

ఉదాహరణ:

char str[5]={'5','+','A'};

str[0]; ---> 5

str[1]; ---> +

str[2]; ---> A

str[3]; ---> NULL

str[4]; ---> NULL

గమనిక:

ప్రారంభించేటప్పుడు మూలకాల శ్రేణి దాని పరిమాణం కంటే ఎక్కువ ప్రారంభించబడదు.

ఉదాహరణకు

char str[2]={'5','+','A','B'}; // Invalid

పై ఉదాహరణలో కంపైలర్ “initialize-string for array of chars is too long” దోష సందేశాన్ని ప్రదర్శిస్తుంది.

పాలిండ్రోమ్‌ను తనిఖీ చేయడానికి ప్రోగ్రామ్ వ్రాయండి

```
#include<iostream>
using namespace std;
int main( )
{
        int i, j, len, flag =1;
        char a [20];
        cout<<"Enter a string:";
        cin>>a;
        for(len=0;a[len]!='\0';++len)
        for(!=0,j=len-1;i<len/2;++i,--j)
        {
        if(a[j]!=a[i])
                        flag=0;
        }
```


11th Computer Science_Chapter-12 Telugu.indd 219 23-03-2023 15:38:16

```
        if(flag==1)
                cout<<"\n The String is palindrome";
        else
                cout<<"\n The String is not palindrome";
        return 0;
}
```

Output:

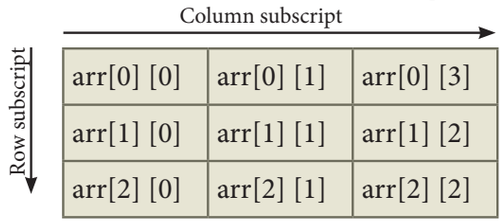
```
        Enter a string : madam
        The String is palindrome
```

12.3 ద్వి - పరిమాణ శ్రేణి Two-dimensional array

Two-dimensional (2D) arrays ఒకే రకమైన మూలకాల సేకరణ, ఇక్కడ మూలకాలు నిర్ధిష్టమై అడ్డు వరుసలు మరియు నిలువు వరుసలలో నిల్వ చేయబడతాయి. ఉదాహరణ m × n మాత్రిక ఇక్కడ m అనునది అడ్డు వరుసల సంఖ్యను సూచిస్తుంది. మరియు n అనునది నిలువ సంఖ్య పటము 12.4 లో చూపబడుతుంది.

int arr[3][3];

2D array సంభావిత మెమరీ ప్రాతినిధ్యం



array అను శ్రేణి మూడు అడ్డు వరుసలు మరియు మూడు నిలువ వరుసలతో మాత్రిక రూపంలో చూడవచ్చు. ఇక్కడ గమనించవలసిన విషయం ఏమిటంటే సబ్ స్క్రిప్ట్ 0 శ్రేణితో మొదలవుతుంది. [0][0] మొదటి మూలకాన్ని సూచిస్తుంది.

పటము 12.4

12.3.1 2-D Array యొక్క ప్రకటన

2-D శ్రేణి యొక్క ప్రకటన అనేది

data-type array_name[row-size][col-size];

పై డిక్లరేషన్‌లో data-type అనునది దానికి తగిన C++ డేటా రకాన్ని సూచిస్తుంది. array name అనునది 2-D array, యొక్క పేరును సూచిస్తుంది. rows-size అనునది అడ్డు వరుసల సంఖ్యను సూచిస్తుంది మరియు col-size అనునది నిలువ వరుసల సంఖ్యను సూచిస్తుంది.

ఉదాహరణకి

```
int A[3][4];
```

పై ఉదాహరణలో A అనునది 2-D array, 3 అనునది వరుసలు సంఖ్య మరియు 4 అనునది నిలువ వరుసలను సూచిస్తుంది. ఈ array గరిష్టంగా 12 మూలకాలను నిలువ వుంచును.

గమనిక:

Array పరిమాణం తప్పనిసరిగా పూర్ణాంక విలువ 0 కంటే ఎక్కువగా ఉండాలి. Array నిలువ వరుసల పరిమాణం తప్పనిసరి కాని అడ్డు వరుస పరిమాణం ఐచ్ఛికం మాత్రమే

11th Computer Science_Chapter-12 Telugu.indd 220 23-03-2023 15:38:16

2-D శ్రేణి యొక్క ఇతర ఉదాహరణలు:

int A[3][3];

float x[2][3];

char name[5][20];

12.3.2 Two-Dimensional array ను Initialization చేయును

2-D శ్రేణి ప్రకటన సమయంలో శ్రేణిని ఒకటి కంటే ఎక్కువ మార్గాల్లో ప్రారంభించవచ్చు.

ఉదాహరణకు

```
int matrix[4][3]={
{10,20,30},// Initializes row 0
{40,50,60},// Initializes row 1
{70,80,90},// Initializes row 2
{100,110,120}// Initializes row 3
};
int matrix[4][3]={10,20,30,40,50,60,70,80,90,100,110,120};
```

Array's లో అడ్డు వరుస పరిమాణం ఐచ్ఛికం అయితే నిలువ వరుస పరిమాణం తప్పనిసరి.

ఉదాహరణకు

```
int matrix[][3]={
{10,20,30},// row 0
{40,50,60},// row 1
{70,80,90},// row 2
{100,110,120}// row 3
};
```

12.3.3 Accessing the Two-dimensional array

Two-dimensional array ఒక నిర్ధిష్టమైన మూలకాన్ని యాక్సెస్ చేయడానికి రెండు ఇండెక్స్ విలువలను ఉపయోగిస్తుంది. ఇక్కడ మొదటి ఇండెక్స్ అడ్డు వరుస విలువను మరియు రెండవ ఇండెక్స్ నిలువ వరుస విలువను నిర్దేశిస్తుంది.

matrix[0][0]=10;// మొదటి అడ్డు వరుస యొక్క మొదటి మూలకానికి 10 ని కేటాయించండి

matrix[0][1]=20;// మొదటి అడ్డు వరుస యొక్క రెండవ మూలకానికి 20 ని కేటాయించండి.

matrix[1][2]=60;// రెండవ వరుస యొక్క మూడవ మూలకానికి 60 ని కేటాయించండి

matrix[3][0]=100;// నాల్గవ వరుస యొక్క మొదటి మూలకానికి 100 ని కేటాయించండి.

11th Computer Science_Chapter-12 Telugu.indd 221 23-03-2023 15:38:16

రెండు మాత్రికలను కలపడానికి C++ ప్రోగ్రామ్ వ్రాయండి

```
#include<iostream>
#include<conio>
using namespace std;
int main()
{
        int row, col, m1[10][10], m2[10][10], sum[10][10];
        cout<<"Enter the number of rows : ";
        cin>>row;
        cout<<"Enter the number of columns : ";
        cin>>col;
        cout<< "Enter the elements of first matrix: "<<endl;
        for (int i = 0;i<row;i++ )
        for (int j = 0;j <col;j++ )
        cin>>m1[i][j];
        cout<< "Enter the elements of second matrix: "<<endl;
        for (int i = 0;i<row;i++ )
        for (int j = 0;j<col;j++ )
        cin>>m2[i][j];
        cout<<"Output: "<<endl;
        for (int i = 0;i<row;i++ )
        for (int j = 0;j<col;j++ )
        {
        sum[i][j]=m1[i][j]+m2[i][j];
        cout<<sum[i][j]<<" ";
        }
        cout<<endl<<endl;
        }
getch();
return 0;
}
```

```
Enter the number of rows :  2
Enter the number of column : 2
Enter the elements of first matrix:
1
1
1
1
Enter the elements of second matrix:
1
1
1
1
Output:
2  2
2  2
```


11th Computer Science_Chapter-12 Telugu.indd 222 23-03-2023 15:38:17

12.3.4 2-D array యొక్క మెమరీ ప్రాతినిధ్యం:

సాధారణంగా two-dimensional array మాత్రిక రూపంలో చూడవచ్చు. 2-D శ్రేణి యొక్క conceptual view దృశ్యం క్రింది విధంగా ఉండును:

int A[4][3];

A[0][0]	A[0][1]	A[0][2]
A[1][0]	A[1][1]	A[1][2]
A[2][0]	A[2][1]	A[2][2]
A[3][0]	A[3][1]	A[3][2]

పై ఉదాహరణలో 2-D array యొక్క పేరు A లో 4 అడ్డు వరుసలు మరియు 3 నిలువ వరుసలు ఉన్నాయి. one-dimensional array లోని 2-D array కూడా మూలకాలను నిరంతర మెమరీలో నిల్వచేయబడతాయి. 2-D array మెమరీ ప్రాతినిధ్యాలు రెండు రకాలు వున్నాయి. అవి:

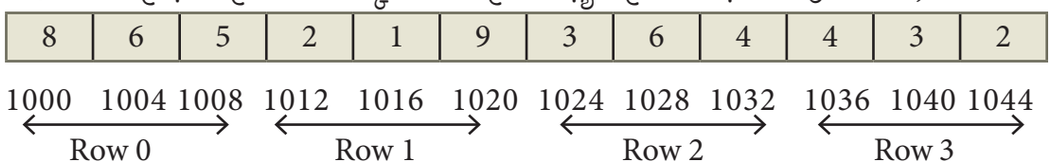
- Row-Major order (అడ్డు వరుస ప్రధాన క్రమం)
- Column-Major order (నిలువ వరుస ప్రధాన క్రమం)

ఉదాహరణ

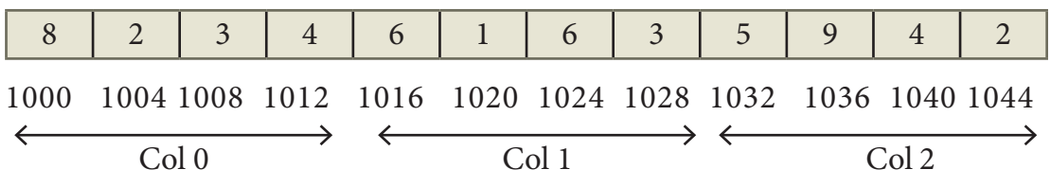
int A[4][3]={ { 8,6,5}, { 2,1,9}, {3,6,4}, {4,3,2} }

అడ్డు వరుస ప్రధాన క్రమం:

అడ్డు వరుస ప్రధాన క్రమంలో అన్ని మూలకాలు వరుస వరుసలుగా నిరంతర మెమరీ స్థానాల్లో నిల్వ చేయబడతాయి. అనగా మొదటి అడ్డు వరుసలోని అన్ని అంశాలు తరువాత రెండవ అడ్డు వరుస మరియు మొదలైనవి. వరుస ప్రధాన క్రమం యొక్క మెమరీ ప్రాతినిధ్యం క్రింది విధంగా వుంటుంది;



నిలువ వరుస ప్రధాన క్రమం:



12.4 స్ట్రింగ్స్ Array

స్ట్రింగ్ array అనునది two-dimensional character array. మొదటి సూచిక (అడ్డు వరుసలు) యొక్క పరిమాణం స్ట్రింగ్ సంఖ్యను సూచిస్తుంది మరియు రెండవ సూచిక (నిలువ వరుసలు) పరిమాణం స్ట్రింగ్ యొక్క గరిష్ట పొడవును సూచిస్తుంది. సాధారణంగా స్ట్రింగ్ చివరిలో శూన్య అక్షరానికి అనుగుణంగా స్ట్రింగ్ 2-D array ను ప్రకటిస్తారు. ఉదాహరణకు 2-D array డిక్లరేషన్.

char Name[6][10];

పై వివరణలో 2-D array రెండు సూచికలు వున్నాయి, ఇవి అడ్డు వరుస పరిమాణం మరియు నిలువ వరుస పరిమాణాన్ని సూచిస్తాయి. అంటే 6 అడ్డు వరుసల సంఖ్యను మరియు 10 నిలువ వరుసల సంఖ్యను సూచిస్తుంది.

11th Computer Science_Chapter-12 Telugu.indd 223 23-03-2023 15:38:17

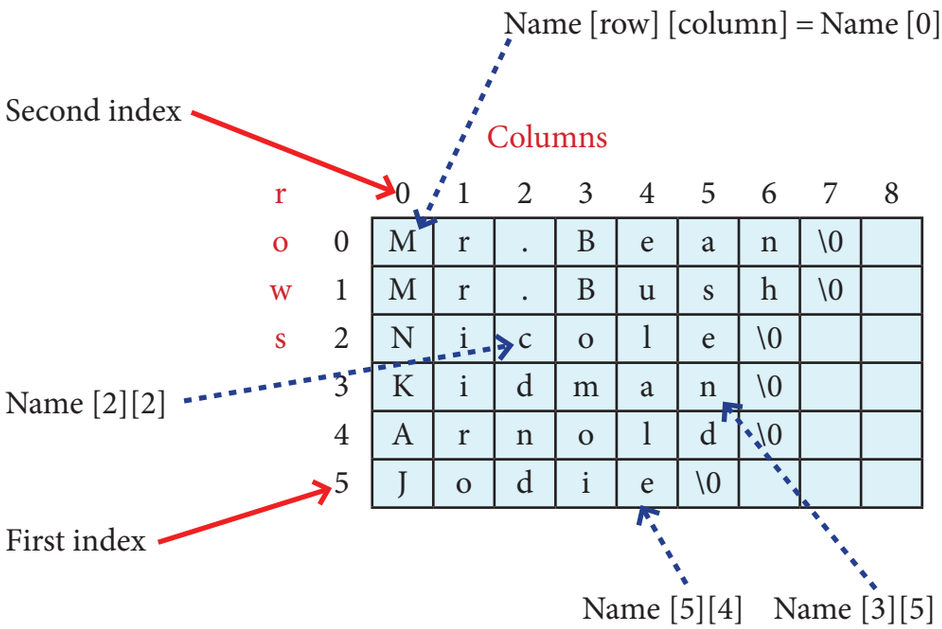
12.4.1 Initialization

ఉదాహరణకు

char Name[6][10] = {"Mr. Bean", "Mr.Bush", "Nicole", "Kidman", "Arnold", "Jodie"};

పై ఉదాహరణలో 2-D array 6 స్ట్రింగ్స్‌తో ప్రారంభించబడుతుంది, ఇక్కడ ప్రతి స్ట్రింగ్ గరిష్టంగా 9 అక్షరాల పొడవు వుంటుంది. ఎందుకంటే చివరి అక్షరం శూన్యం.

2-D arrayల. యొక్క మెమరీ అమరిక క్రింద చూపబడింది. మరియు అన్ని స్ట్రింగ్స్ నిరంతర ప్రదేశాలలో నిల్వ చేస్తారు.



row	0	1	2	3	4	5	6	7	8
0	M	r	.	B	e	a	n	\0	
1	M	r	.	B	u	s	h	\0	
2	N	i	c	o	l	e	\0		
3	K	i	d	m	a	n	\0		
4	A	r	n	o	l	d	\0		
5	J	o	d	i	e	\0			

2-D క్యారెక్టర్ అర్రేను ఉపయోగించి స్ట్రింగ్స్ అర్రేను ప్రదర్శించెడి C++ ప్రోగ్రాం.

```
#include<iostream>
using namespace std;
int main( )
{
        // initialize 2d array
        char colour [4][10]={"Blue","Red","Orange",
                        "yellow"};

        // printing strings stored in 2d array
        for (int i=0;  i <4; i++)
        cout  <<  colour [i] << "\n";
}
```

Output:

```
Blue
Red
Orange
Yellow
```


11th Computer Science_Chapter-12 Telugu.indd 224 23-03-2023 15:38:17

కేస్‌స్టడీ (Case Study):

(1) 10 మంది విద్యార్థుల మార్కులను అంగీకరించడానికి మరియు సగటు గరిష్ట మరియు కనిష్ట మార్కులను కనుగొనడానికి ప్రోగ్రామ్‌ను వ్రాయండి.

నిర్మాణాలు (Structures)

12.5 నిర్మాణాల పరిచయం

నిర్మాణం అనేది వినియోగదారుడు చేత నిర్వచించబడనది, ఇది వివిధ డేటా రకాల అంశాల కలయికను కలిగి వుంటుంది. మిశ్రమ డేటా రకాల వేరియబుల్స్‌ను ఒకే యూనిట్‌గా సమూహపరచడానికి ఇది అనుమతిస్తుంది.

12.5.1 నిర్మాణాల ఆవశ్యకత (Purpose of Structures)

ఏ పరిస్థితిలోనైన ఏకరీతి డేటా టైప్‌ల వస్తువులను సూచించడానికి ఒకటి కంటే ఎక్కువ వేరియబుల్ అవసరం అయినప్పుడు శ్రేణిని ఉపయోగించవచ్చు. మూలకాలు వేర్వేరు డేటా రకాలు అయితే శ్రేణి మద్ధతు ఇవ్వదు. ఒకటి కంటే ఎక్కువ వేరియబుల్స్ ఉపయోగించినట్లయితే, వాటిని మెమరీలో నిల్వచేయవచ్చు కాని ప్రక్కనే వున్న ప్రదేశాలలో కాదు. ఇది శోధిస్తున్నప్పుడు సమయాన్ని పెంచుతుంది. నిర్మాణం అనునది ఒకదానికొకటి ప్రక్కనే వున్న ఒక మెమరీ భాగంలో ఒకే తార్కిక మూలకంలో భాగంగా వేర్వేరు డేటా రకాలను నిల్వ చేయడానికి ఒక సదుపాయాన్ని అందిస్తుంది.

12.5.2 నిర్మాణాలను ప్రకటించడం మరియు నిర్వచించడం (Declaring and defining structures)

'struct' అనే కీవర్డ్ ఉపయోగించి structure ప్రకటించబడింది. నిర్మాణమును సృష్టించే వాక్య నిర్మాణం క్రింద ఇవ్వబడింది.

```
struct structure_name {
        type member_name1;
        type member_name2;
    } reference_name;
```

> structure నిర్వచనంతో పాటు ప్రకటించిన వస్తువును గ్లోబల్ ఆబ్జక్ట్స్ అంటారు.

నిర్మాణ రకం యొక్క వస్తువులను నేరుగా ప్రకటించడానికి ఐచ్ఛిక ఫీల్డ్ రిఫరెన్స్ పేరును ఉపయోగించవచ్చు.

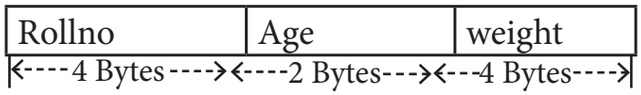
ఉదాహరణ:

```
struct Student
{
    long rollno;
    int age;
    float weight;
} ;
```

struct యొక్క పై ప్రకటనలో మూడు వేరియబుల్స్ రోల్ సంఖ్య, వయస్సు మరియు బరువు ఉపయోగించబడతాయి. నిర్మాణంలోని ఈ వేరియబుల్స్ (డేటా ఎలిమెంట్) ను సభ్యులు (లేదా ఫీల్డ్స్) అంటారు.

11th Computer Science_Chapter-12 Telugu.indd 225 23-03-2023 15:38:17

విద్యార్థి నిర్మాణాన్ని ఉపయోగించడానికి స్టూడెంట్ యొక్క వేరియబుల్ డిక్లేర్ చేయబడింది మరియు మెమరీ కేటాయింపు పటము 12.5 లో చూపబడింది.



పటము 12.5 మెమరీ కేటాయింపు

struct Student balu; // బాలు కోసం విద్యార్థి నిర్మాణాన్ని సృష్టించండి.

ఇది బాలు అని పిలువబడే టైప్ స్టూడెంట్ యొక్క వేరియబుల్‌ను నిర్వచిస్తుంది. ఒకే నిర్మాణ రకం యొక్క బహుళ వేరియబుల్స్ నిర్వచించడం సాధ్యపడుతుంది.

struct Student frank; // విద్యార్థి Frank కోసం ఒక నిర్మాణాన్ని సృష్టించండి.

ఉదాహరణకు, నిర్మాణ వస్తువులు balu మరియు frank లను కూడా నిర్మాణ డేటా రకంగా ప్రకటించవచ్చు:

```
struct Student
{
longrollno;
int age;
float weight;
} balu, frank;
```

12.5.3 Structure అంశాలను సూచిస్తుంది

విద్యార్థి structure రకానికి చెందిన రెండు వస్తువులను ప్రకటించిన తర్వాత (బాలు మరియు ఫ్రాంక్) వారి సభ్యులు నేరుగా యాక్సెస్ చేయవచ్చు. దాని కోసం వాక్య నిర్మాణం ఆబ్జెక్ట్ పేరు మరియు సభ్యుల పేరు మధ్య dot (.) ను ఉపయోగిస్తుంది. ఉదాహరణకు Student యొక్క నిర్మాణ అంశాలను ఈ క్రింది విధంగా యాక్సెస్ చేయవచ్చు:

```
balu.rollno
balu.age
balu.weight
frank.rollno
frank.age
frank.weight
```


11th Computer Science_Chapter-12 Telugu.indd 226 23-03-2023 15:38:17

Anonymus Structure Vs Named Structure

ఎలాంటి name/tag లేని స్ట్రక్చర్‌ను 'Anonymus' స్ట్రక్చర్ అంటారు.

```
struct
{
long rollno;
        int age;
        float weight;
} student;
```

ఇక్కడ విద్యార్థి (స్టూడెంట్) అనునది పైన పేర్కొనిన నిర్మాణానికి సూచనగా వాడబడిన పదం. ఇందులోని మూలాంశాలు (ఎలిమెంట్స్) student, rollno, student age and student weight మొదలగు విధంగా ప్రాప్యతను పొందును.

12.5.4 నిర్మాణ అంశాలను ప్రారంభించడం

వేరియబుల్స్‌కు విలువలను కేటాయించడం మాదిరిగానే నిర్మాణ మూలకాలకు విలువలను కేటాయించవచ్చు.

ఉదాహరణ

```
balu.rollno= "702016";
balu.age= 18;
balu.weight= 48.5;
```

Arrays విలువలను కేటాయించినట్లే విలువలను నేరుగా కేటాయించవచ్చు.

```
balu={702016, 18, 48.5};
```

12.5.5 స్ట్రక్చర్ అసైన్‌మెంట్

మూలకాల విలువలను కేటాయించడానికి బదులుగా నిర్మాణాలను నేరుగా కేటాయించవచ్చు.

ఉదాహరణ

మహేష్ మరియు ప్రవీణ్ ఒకే వయస్సు మరియు ఒకే ఎత్తు మరియు బరువు వుంటే మహేష్ విలువలను ప్రవీణ్‌కు కాపీ చేయవచ్చు.

```
struct Student
{
        int  age;
        float height, weight;
}mahesh;
```

స్ట్రక్చర్ వేరియబుల్స్ / ఆబెక్ట్ రెండూ ఒకే రకంగా వుంటేనే స్ట్రక్చర్ అసైన్‌మెంట్ సాధ్యమవుతుంది.

మహేష్ వయస్సు 17 మరియు ఎత్తు మరియు బరువులు వరుసగా 164.7 మరియు 52.5. ఈ క్రింది స్టేట్‌మెంట్ విషయములను తెలియచేస్తుంది.

```
mahesh = {17, 164.5, 52.5};
praveen =mahesh;
```

అదే వయస్సు ఎత్తు మరియు బరువును ప్రవీణ్‌కు ఆపాదించడం జరుగును.

11th Computer Science_Chapter-12 Telugu.indd 227 23-03-2023 15:38:17

ఉదాహరణ:

క్రింది C++ ప్రోగ్రామ్ విద్యార్థుల సమాచారాన్ని కీబోర్డ్ ద్వారా చదువుతుంది మరియు దానినే ప్రదర్శిస్తుంది.

```
#include <iostream>
using namespace std;
struct Student
{
        int  age;
        float height, weight;
} mahesh;
void main( )
{
        cout<< " Enter the age:"<<endl;
        cin>>mahesh.age;
        cout<< "Enter the height:"<<endl;
        cin>>mahesh.height;
        cout<< "Enter the weight:"<<endl;
        cin>>mahesh.weight;
        cout<< "The values entered  for Age, height and weight are"<<endl;
        cout<<mahesh.age<<     "\t"<<mahesh.height<<     "\t"<<Mahesh.
weight;
}
```

Output:

```
Enter the age:
18
Enter the height:
160.5
Enter the weight:
46.5
The values entered  for Age, height and weight are
18      160.5           46.5
```

గుర్తుంచుకోవలసిన విషయములు

- Structure అనునది వినియోగదారు నిర్వచించినది. ఇది వివిధ డేటా రకాల అంశాల కలయికను కలిగి వుంటుంది.
- Structure అనే పద Structure కీ పదాన్ని వుపయోగించి నిర్మాణం ప్రకటించబడింది.
- దీని అంశాలు దాని ఆబ్జెక్ట్ పేరును వుపయోగించి డాట్ (.) ను సూచిస్తాయి. ఆపరేటర్ మరియు సంఖ్య పేరు
- పేరు / ట్యాగ్ లేని నిర్మాణాన్ని అనామక నిర్మాణం అంటారు.
- ప్రత్యేకమైన అసైన్‌మెంట్ స్టేట్‌మెంట్‌లను వుపయోగించడం ద్వారా లేదా డిక్లరేషన్ సమయంలో దాని విలువలను బ్రేసస్ ద్వారా నిర్మాణ అంశాలను ప్రారంభించవచ్చు.
- రెండు వస్తువులు ఒకే నిర్మాణ రకానికి చెందినవి అయితే మాత్రమే నిర్మాణ వస్తువును మరొక నిర్మాణ వస్తువుకు కేటాయించవచ్చు.
- మరొక నిర్మాణంలో ప్రకటించిన నిర్మాణాన్ని సమూహ నిర్మాణం అంటారు.
- ఒక నిర్మాణం దాని సభ్యుల మూలకం వలె శ్రేణిని కలిగి వుంటుంది.
- స్ట్రక్చర్ వేరియబుల్ యొక్క శ్రేణిని కూడా సృష్టించవచ్చు.

11th Computer Science_Chapter-12 Telugu.indd 228 23-03-2023 15:38:17

మూల్యాంకనం





భాగము – అ

సరైన జవాబును ఎంచుకోండి

1. క్రింది వాటిలో సాధారణ పేరుతో సూచించబడిన ఒకే రకమైన వేరియబుల్స్ సేకరణ ఏది?

   (అ) int (ఆ) float (ఇ) Array (ఈ) class

2. int age[]={6,90,20,18,2}; లో ఎన్ని మూలకాలు వున్నాయి?

   (అ) 2 (ఆ) 5 (ఇ) 6 (ఈ) 4

3. cin>>n[3]; ఈ ప్రకటన విలువలను ఏ మూలకానికి అంగీకరిస్తుంది?

   (అ) 2 (ఆ) 3 (ఇ) 4 (ఈ) 5

4. అప్రమేయంగా స్ట్రింగ్ ఏ అక్షరంతో ముగుస్తుంది?

   (అ) \o (ఆ) \t (ఇ) \n (ఈ) \b

5. Structure నిర్వచనం దేని ద్వారా ముగించబడుతుంది?

   (అ) : (ఆ) } (ఇ) ; (ఈ) ::

6. Structure ప్రకటించబడినప్పుడు ఏమి జరుగుతుంది?

   అ) ఇది ఏ మెమరీని కేటాయించదు

   (ఆ) ఇది మెమరీని కేటాయిస్తుంది

   ఇ) ఇది ప్రకటించబడుతుంది మరియు ప్రారంభించబడుతుంది

   ఈ) ఇది మాత్రమే ప్రకటించబడుతుంది

7. నిర్మాణ ప్రకటన క్రింద ఇవ్వబడింది.

```
struct Time
{
int hours;
int minutes;
int seconds;
}t;
```

   పై డిక్లరేషన్ వుపయోగించి క్రింది వాటిలో ఏది సెకన్లను సూచించుచున్నది.

   (అ) సమయం, సెకన్లు (ఆ) సమయం::సెకన్లు (ఇ) సెకన్లు (ఈ) t. సెకన్లు

8. ఈ క్రింది వాటిలో ఏది సరైన నిర్మాణం?

   (అ) struct {int num;} (ఆ) struct sum {int num;}

   (ఇ) struct sum int sum; (ఈ) struct sum {int num;};

11th Computer Science_Chapter-12 Telugu.indd 229 23-03-2023 15:38:17

9. నిర్మాణ ప్రకటన క్రింద ఇవ్వబడినది

```
struct employee
{
int empno;
char ename[10];
}e[5];
```

పై ప్రకటన వుపయోగించి క్రింది వాటిలో ఏది సరైనది?

(అ) cout<<e[0].empno<<e[0].ename; (ఆ) cout<<e[0].empno<<ename;

(ఇ)cout<<e[0]->empno<<e[0]->ename; (ఈ) cout<<e.empno<<e.ename;

10. నిర్మాణ సభ్యుడిని యాక్సెస్ చేసేటప్పుడు డాట్ ఆపరేటర్ యొక్క ఎడమ వైపున ఉన్న ఐడెంటిఫైయర్ పేరు

(అ) స్ట్రక్చర్ వేరియబుల్ (ఆ) నిర్మాణం బ్యాగ్

(ఇ) నిర్మాణ సభ్యుడు (ఈ) నిర్మాణం ఫంక్షన్

విభాగం – ఆ

క్రింది ప్రశ్నలకు క్లుప్తంగా జవాబులు వ్రాయండి

1. Array లో ట్రావెర్సల్ అంటే ఏమిట?
2. స్ట్రింగ్స్ అంటే ఏమిటి?
3. two – dimensional array ను నిర్మించడానికి వాక్య నిర్మాణం (syntax) ను వ్రాయుము.
4. Structure నిర్వచనం మరియు వీటి ఉపయోగాలు తెలుపుము?
5. క్రింది structure నిర్వచనంలో దోషం ఏమిటి?

```
struct   employee{ inteno;charename[20];char dept;}
Employee e1,e2;
```

విభాగం – ఇ

క్రింది ప్రశ్నలకు సంక్షిప్తంగా జవాబులు వ్రాయండి

1. Array ను నిర్వచించండి. వాటి రకాలు ఏమిటి?
2. Array యొక్క స్ట్రింగ్‌ను గురించి వ్రాయండి
3. క్రింది కోడ్ 'S' తో ప్రారంభమయ్యే మొత్తం విద్యార్థుల పేరును సంక్షిప్తీకరిస్తుంది మరియు దానిని ప్రదర్శిస్తుంది. అవసరమైన స్టేట్‌మెంట్లలో ఖాళీలను పూరించండి.

```
struct student {int exam no,lang,eng,phy,che,mat,csc,total;char name[15];};
int main()
{
student s[20];
for(int i=0;i<20;i++)
{        …….. //accept student details                }
for(int i=0;i<20;i++)
{
…….. //check for name starts with letter "S"
```


11th Computer Science_Chapter-12 Telugu.indd 230 23-03-2023 15:38:17

```
…….  // display the detail of the checked name
}
return 0;
}
```

4. Structure లోని members ఎలా యాక్సెస్ చేస్తారు? ఉదాహరణ ఇవ్వండి.

5. Anonymous structure అంటే ఏమిటి? ఉదాహరణ ఇవ్వండి.

విభాగం - ఈ

క్రింది ప్రశ్నలను వివరించండి

1. రెండు మాత్రికల మధ్య వ్యత్యాసాన్ని కనుగొనడానికి C++ ప్రోగ్రామ్‌ను వ్రాయండి.

2. క్రింది నిర్మాణ నిర్వచనాన్ని ఉపయోగించి రెండు దూరాలను జోడించడానికి C++ ప్రోగ్రామ్‌ను వ్రాయండి.

```
struct Distance{
int feet;
float inch;
}d1 , d2, sum;
```

3. క్రింది c++ ప్రోగ్రామ్ యొక్క అవుట్‌పుట్‌ను వ్రాయండి

```
#include<iostream>
#include<stdio>
#include <string>
#include<conio>
using namespace std;
struct books {
char name[20], author[20];
} a[2];
int main()
{  cout<< "Details of Book No " << 1 << "\n";
cout<< "----------------------\n";
cout<< "Book Name :"<<strcpy(a[0].name,"Programming  ")<<endl;
cout<< "Book Author :"<<strcpy(a[0].author,"Dromy")<<endl;
cout<< "\nDetails of Book No " << 2 << "\n";
cout<< "----------------------\n";
cout<< "Book Name :"<<strcpy(a[1].name,"C++programming" )<<endl;
cout<< "Book Author :"<<strcpy(a[1].author,"BjarneStroustrup ")<<endl;
cout<<"\n\n";
cout<< "================================================\n";
cout<< " S.No\t| Book Name\t|author\n";
cout<< "================================================";
for (int i = 0; i < 2; i++) {
```


11th Computer Science_Chapter-12 Telugu.indd 231 23-03-2023 15:38:17

```
cout<< "\n  " << i + 1 << "\t|" << a[i].name << "\t| " << a[i].author;
}
cout<< "\n================================================";
return 0;
}
```

4. క్రింది c++ ప్రోగ్రామ్ యొక్క అవుట్పుట్ వ్రాయండి

```
#include <iostream>
#include <string>
using namespace std;
struct student
{
        introll_no;
        char name[10];
        long phone_number;
};
int main(){
student p1 = {1,"Brown",123443},p2;
p2.roll_no = 2;
strcpy(p2.name ,"Sam");
p2.phone_number = 1234567822;
cout<< "First Student" <<endl;
cout<< "roll no : " << p1.roll_no <<endl<< "name : " << p1.name <<endl;
cout<< "phone no : " << p1.phone_number <<endl;
cout<< "Second Student" <<endl;
cout<< "roll no : " << p2.roll_no <<endl<< "name : " << p2.name <<endl;
cout<< "phone no : " << p2.phone_number <<endl;
return 0;
}
```

5. క్రింది ప్రోగ్రామ్‌లో లోపాన్ని డీబగ్ చేయండి.

```
#include <istream.h>
structPersonRec
{
```


11th Computer Science_Chapter-12 Telugu.indd 232 23-03-2023 15:38:17

```
        charlastName[10];
        chaefirstName[10];
    int age;
    }
    PersonRecPeopleArrayType[10];
    void main()
    {
        PersonRecord people;
    for (i = 0; i < 10; i++)
    {
        cout<<people.firstName<< ' ' <<people.lastName  <<people.age;
    }
    for (int i = 0; i < 10; i++)
    {
        cout<< "Enter first name: ";        cin<<peop[i].firstName;
        cout<< "Enter last name: ";         cin>>peop[i].lastName;
        cout<< "Enter age: ";          cin>> people[i].age;}
}
```

సూచన పుస్తకాలు:

1. Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.
2. The Complete Reference C++ (Forth Edition), Herbert Schildt.Mc.Graw Hills.
3. Computer Science with C++ (A text book of CBSE XI and XII), SumitaArora, DhanpatRai& Co.
4. The C++ Programming Language, Bjarne Stroustrup
5. https://www.tutorialspoint.com
6. http://www.cs.princeton.edu
7. https://www.programiz.com

11th Computer Science_Chapter-12 Telugu.indd 233 23-03-2023 15:38:17

ఆబ్జెక్టు ఓరియంటడ్ ప్రోగ్రామింగ్ సాంకేతిక పద్ధతులు పరిచయం
(Introduction to Object Oriented Programming Techniques)

అభ్యసన లక్ష్యాలు

ఈ పాఠమును చదివిన తరువాత ఈ క్రింది విషయాలను నేర్చుకుంటారు.

- **OOPS** భావనను అర్థంచేసుకొనుట
- ప్రొసీజిరల్ మాడులర్ ఆబ్జెక్టు ఓరియంటడ్ ప్రోగ్రామింగ్ (*Procedural, Modular and object oriented programming*) మరియు **OOPS** కి మధ్య గల బేధాలను గూర్చి తెలుసుకొనుట.
- OOP యొక్క ప్రయోజనాలు మరియు ప్రతికూలతలు గూర్చి అర్థం చేసుకొనుట.

13.1 పరిచయం (Introduction)

(OOP - Object-Oriented Programming) అనునవి వస్తువులు మరియు తరగతులు వారిగా ప్రోగ్రాం విధానాన్ని వివరించడానికి ఉపయోగించే పదం. డేటా మరియు ప్రవర్తన ఈ రెండింటిని కలిగియున్న దానిని సమహారంగా సాఫ్ట్‌వేర్‌గా నిర్వచించడానికి అబ్జెక్టు ఓరియంటడ్ రూపావళి అనుమతిస్తుంది. ఇది సాంప్రదాయక ఫంక్షన్ ప్రోగ్రామింగ్ అభ్యాసానికి విరుద్ధంగా ఉంది. డేటా మరియు ప్రవర్తనం 1989 ల నుండి ఆబ్జెక్టు (Object) అనే పదం ప్రోగ్రాం భాషలకు సంబంధించి కనిపించింది. ఇది 1990 నుండి అభివృద్ధి చేయబడిన దాదాపు అన్నీ భాషలలో **ఆబ్జెక్టు** ఓరియంటడ్ లక్షణాలను కలిగి ఉంది. ఈ అధ్యాయం సాధారణమైన OOP భావనలను పరిచయం చేస్తుంది.

13.2 ప్రోగ్రామింగ్ భావజాలం (Programming Paradigms)

భావజాలం (Paradigm) అనగా ప్రోగ్రాం యొక్క సూత్రాన్ని నిర్వహించడం ఇది ప్రోగ్రామింగ్ ఒక విధానం అగును. సంగణక యంత్రాన్ని (computer) ఉపయోగించి సమస్యలను పరిష్కరించడానికి అనేక రకములైన మార్గములు కలవు. అవి: విధానపరమైన (Procedural programming) మాడులర్ ప్రోగ్రామింగ్ మరియు ఆబ్జెక్టు ఓరియంటడ్ ప్రోగ్రామింగ్ (Object Oriented Programming)

13.2.1 విధానపరమైన ప్రోగ్రామింగ్ (Procedural programming)

విధానపరమైన ప్రోగ్రామింగ్ అనగా ఏదైనా చేయమని సంగణక యంత్రానికి (computer) సూచనల పట్టికలను ఇచ్చుట. ఇది (Procedural programming) విధానాలను లక్ష్యంగా పెట్టుకుంది. మరియు పనిచేయడానికి ఒత్తిడి ఇచ్చును.

విధాన పరమైన కార్యక్రమం యొక్క ముఖ్య లక్షణాలు (Important features of procedural programming)

- కార్యక్రమాలు subroutines లేదా subprogrammes రూపంతో నిర్వహించబడుతాయి.
- అన్నీ డేటా అంశాలు ప్రపంచవ్యాప్తంగా ఉన్నాయి.

11th Computer Science_Chapter-13 Telugu.indd 234 23-03-2023 16:44:47

- కనిష్ట పరిమాణ సాఫ్ట్‌వేర్ (software) అనువర్తనాలకి అనుకూలమైనది.
- డేటా రకంలో ఏదైనా మార్పు ఒకే డేటా రకాన్ని ఉపయోగించే అన్ని (subroutines) ప్రచారం చేయాల్సిన అవసరం ఉన్నందున ప్రోగ్రామ్ కోడ్‌ను నిర్వహించడం మరియు మెరుగుపరచడం కష్టం. ఇది అధిక సమయాన్ని తీసుకొంటుంది.
- ఉదాహరణ: **FORTRAN** మరియు **COBOL**

13.2.2 మాడ్యులర్ ప్రోగ్రామ్ (Modular programming)

మాడ్యులర్ ప్రోగ్రామింగ్ కంప్యూటర్‌ను ఏదైన చేయమని సూచించే సూచన పట్టికను కలిగి ఉంటుంది. **అయితే ఈ భావజాలంలో (ఫంక్షన్‌ఆధారిత) బహుళ మాడ్యుల్‌ని కలిగి ఉంటుంది. ప్రతి ఒక్క మాడ్యుల్స్ సంబంధిత రకాల సమితిని కలిగి మాడ్యూల్స్ క్రింద దాచబడుతుంది.** మాడ్యూల్స్‌ను సవరించడం ద్వారా మాత్రమే డేటా అమరికను మార్చవచ్చు.

మాడ్యూలర్ కార్యక్రమం యొక్క ముఖ్య లక్షణాలు (Important features of Modular programming)

- డేటా కంటే అల్గోరిథం (algorithm) కు ఎక్కువ ప్రాధాన్యతనిచ్చును.
- కార్యక్రమాలను (Programs) వ్యక్తిగత గుణాంకములుగా Modulas విభజించబడ్డాయి.
- ప్రతీ నమూనాలు (Modulas) ఒకదానికి ఒక్కటి స్వతంత్రముగా ఉంటాయి మరియు అవి వాటి యొక్క సొంత (local) డేటాను కలిగి ఉంటాయి.
- మాడ్యూల్స్ వాటి యొక్క సొంత డేటాతో పాటు దానికి పంపిన డేటాతో పనిచేస్తుంది.
- ఉదాహరణ: **పాస్కేల్ (Pascal) మరియు C**

13.2.3 ఆబ్జెక్టు ఆధారిత ప్రోగ్రామ్ (Object Oriented Programming)

Object Oriented Programming నమూనా అల్గోరిథమ్ కంటే డేటాపై ప్రాధాన్యతనిస్తుంది. ఇది **classes** మరియు **objects** ను ఉపయోగించి programs ని అమలు చేస్తుంది.

తరగతి: క్లాసు అనునది C++ లో ఒక నిర్మాణం ఇది డేటాను మరియు దాని అనుబంధ ఫంక్షన్‌ను ఎన్‌కాప్సులేషన్ (Class) భావనను ఉపయోగించి ఒకే యూనిట్‌గా బంధించడానికి ఉపయోగపడుతుంది. క్లాసు అనేది వినియోగదారు నిర్వచించిన డేటా రకం. క్లాసు ఒకే విధమైన objects సమూహాన్ని సూచిస్తుంది.

సంబంధాన్ని పంచుకునే వస్తువుల సాధారణ లక్షణాలు మరియు సమూహాన్ని సూచించే టెంప్లేట్ లేదా బ్లూప్రింట్‌గా కూడా దీనిని నిర్వచించవచ్చు.

ఆబ్జెక్టు : డేటా మరియు దాని అనుబంధ ఫంక్షన్‌ను (Objects) ఒకే యూనిట్‌గా సూచిస్తుంది. ఆబ్జెక్టులు OOP యొక్క ప్రాథమిక యూనిట్. అవి class variables అని కూడా పిలువబడే తరగతికి సంబంధించిన ఉదాహరణలుగా పిలువబడుతుంది.

కొన్ని లక్షణాలు మరియు ప్రవర్తనల తత్వములు ఆబ్జెక్టు అని పిలువబడుతుంది.

OOP యొక్క ముఖ్య లక్షణాలు

- అల్గోరిథం కన్నా డేటాపై ఎక్కువ దృష్టి పెడుతుంది. డేటాకు ముఖ్యత్వం ఇస్తుంది.
- విధాన సంగ్రహణకు అదనంగా సంగ్రహణను ప్రవేశపెట్టారు.
- డేటాని మరియు దాని అనుబంధ కార్యకలాపాలు ఒకే యూనిట్‌లో సమూహంగా చేయబడ్డాయి.
- నిర్వహించబడుతున్న డేటాతో ప్రోగ్రాంలు రూపొందించబడ్డాయి.
- సారూప్యమైన మరియు విభిన్నమైన డేటా రకాల మధ్య సంబంధాలు సృష్టించబడును.
- ఉదాహరణ: **C++, Java, VB.Net, Python** మొదలగునవి.

13.3 OOP యొక్క ప్రాథమిక భావనలు (Basic Concepts of OOP)

విధానపరమైన (procedural) మరియు మాడ్యూలర్ ప్రోగ్రామింగ్ యొక్క లోపాలను

11th Computer Science_Chapter-13 Telugu.indd 235 23-03-2023 16:44:47

అధిగమించడానికి OOP ని అభివృద్ధి చేయబడింది. (software) రూపొందించడానికి OOP ముఖ్యమైన మరియు శక్తివంతమైన మార్గం అని విస్తృతంగా ఆమోదించబడింది.

OOP ప్రధానంగా క్రింది వాటిని ప్రోత్సహిస్తుంది. అంశాలు :

- **మాడ్యూలరైసేషన్ :** ప్రోగ్రాములను **మాడ్యూల్స్‌గా** విభజిస్తుంది.
- **సాఫ్ట్‌వేర్ పునఃప్రయోజనం :** ముందుగానే అమరియున్న లేక కొత్త (Software re-use) మాడ్యూల్స్ నుండి ప్రోగ్రామును కంపోస్ చేయవచ్చు.

OOP యొక్క ప్రధాన లక్షణాలు:

- డేటా సంగ్రహణ
- ఎన్‌కాప్సులేషన్
- మాడ్యులారిటీ
- వారసత్వం
- బహురూపవాదం

13.3.1 ఎన్‌కాప్సులేషన్ (Encapsulation)

డేటా మరియు ఫంక్షనులను ఒకే యూనిట్‌గా బంధించే యంత్రాగాన్ని **ఎన్‌కాప్సులేషన్** అంటారు. ఇది సంగ్రహణను అమలు చేస్తుంది.

ఎన్‌కాప్సులేషన్ అనేది డేటా వేరియబుల్స్ మరియు ఫంక్షన్లును క్లాసులో బంధించడాన్ని **డేటా బైండింగ్** అని కూడా పిలుస్తారు.

ఎన్‌కాప్సులేషన్ అనునది class యొక్క అత్యంత అద్భుతమైన లక్షణం. డేటా బయట ప్రపంచానికి అందుబాటులో ఉండదు. class లో ఉన్నటువంటి ఫంక్షన్లును మాత్రమే access చేయగలవు. ఈ ఫంక్షన్లు ఆబ్జెక్టు మరియు ప్రోగ్రామ్ మధ్య interface సమన్వయములను అందిస్తాయి. **ఈ ప్రోగ్రామ్ ద్వారా ప్రత్యక్ష ఫఎఇవం నుండి డేటా యొక్క ఈ ఎన్‌కాప్సులేషన్ డేటాను హైడింగ్ లేదా సమాచార హైడింగ్ అని అంటారు.**

13.3.2 డేటా సంగ్రహణ (Data Abstraction)

సమాచార సంగ్రహణ అనేది నేపథ్య (background) వివరాలను బహిర్గతం చేయకుండా అవసరమైన లక్షణాలను మాత్రమే చూపించడాన్ని సూచిస్తుంది. ఈ లక్షణాలపై పనిచేసే సారుప్య గుణాలు మరియు ఫంక్షన్ల పట్టికను నిర్వహించడానికి సమాచార సంగ్రహణ భావనను (data abstraction) ను ఉపయోగిస్తారు. ఇది సృష్టించవలసిన ఆబ్జెక్టు యొక్క ప్రధానమైన లక్షణాలను కలుపుతాయి. ఈ లక్షణాలను **డేటా సభ్యులు** అని పిలుస్తారు. ఎందుకంటే వారు సమాచారాన్ని కలిగి ఉంటారు. ఈ డేటాపై పనిచేసే విధులను **పద్ధతి లేదా సభ్యుల ఫంక్షన్** (member function) అంటారు.

13.3.3 మాడ్యులారిటీ (Modularity)

మాడ్యులారిటీ అంటే ఒక ఫంక్షన్ యూనిట్ల సమితి (named modules)గా విభజించబడిన ఒక సిస్టమ్‌ని design చేయడం. ఇది ఒక పెద్ద అప్లికేషన్ (application) గా కూర్చబడుతుంది.

13.3.4 వారసత్వం (Inheritance)

వారసత్వం అనేది **ప్రస్తుత తరగతి** నుండి కొత్త తరగతులను నిర్మించే సాంకేతికత. దీని యొక్క అతి ముఖ్యమైన ప్రయోజనం **కోడ్ పునర్వినియోగం**.

13.3.5 బహురూపం (Polymorphism)

బహురూపం అనునది ఒక సందేశం యొక్క పనితీరు ఒకటి కంటే ఎక్కువ రూపాల్లో ప్రదర్శించబడే సామర్థ్యం.

13.4 OOP యొక్క ప్రయోజనాలు (Advantages of OOP)

పునర్వినియోగం:

ఒక్కసారి వ్రాసి అనేక సార్లు క్లాసును ఉపయోగించడం ద్వారా మీరు దీనిని సాధించవచ్చు.

11th Computer Science_Chapter-13 Telugu.indd 236 23-03-2023 16:44:47

పునరావృతం:

డేటా పునరావృతానికి వారసత్వం ఒక మంచి (redundancy) లక్షణం మీకు బహుళ తరగతిలో ఒకే కార్యచరణ అవసరమైతే అదే ఒకే కార్యచరణ కోసం ఒక సాధారణ క్లాసుని వ్రాయవచ్చు మరియు ఆ క్లాసుని ఉపవర్గానికి వారసత్వంగా పొందవచ్చు.

సులభముగా నిర్వహించుట:

మునుపే ఉన్న వాటికి చిన్న తేడాలతో కొత్త వస్తువులను సృష్టించవచ్చు కాబట్టి మునుపే ఉన్న కోడును నిర్వహించడం మరియు సవరించడం సులభం.

భద్రత:

డేటా దాచడం మరియు సంగ్రహణను ఉపయోగించి డేటా యొక్క భద్రతను నిర్వహించి అవసరమైన డేటాను మాత్రమే అందించబడుతుంది.

13.5 OOP యొక్క నష్టాలు (Disadvantages of OOP)

పరిమాణం:

ఆబ్జెక్టు ఓరియంటడ్ ప్రోగ్రాం ఇతర ప్రోగ్రామ్‌ల కంటే చాలా పెద్దది.

ప్రయత్నం:

ఆబ్జెక్టు ఓరియంటడ్‌ను సృష్టించడానికి ప్రయత్నం చాలా అవసరం.

వేగం:

ఆబ్జెక్టు ఓరియంటడ్ ప్రోగ్రాములు వాటి పరిమాణం కారణంగా ఇతర ప్రోగ్రాంల కంటే నెమ్మదిగా ఉంటాయి.

గుర్తుంచుకోవాల్సిన అంశాలు

- పారా డిగ్మ్ అగా ప్రోగ్రాం యొక్క సూత్రాన్ని నిర్వహించడం. ఇది ఒక ప్రోగ్రామింగ్ విధానం.
- ప్రొసీజురల్ లేదా మాడ్యూలర్ ప్రోగ్రామింగ్ అంటే సూచనల జాబితాను ఇవ్వడం మరియు కంప్యూటరును ఏదైనా చేయమని సూచనలను ఇవ్వడం.
- ప్రోసీజురల్ ప్రోగ్రామింగ్ విధులు మీద ఎక్కువ అవగాహనను ఉంచుతుంది. ఇది ప్రోగ్రామ్‌ల సబ్‌రోటీన్స్ లేదా సబ్ ప్రోగ్రామ్ రూపంలో నిర్వహించబడతాయి.
- మాడ్యూలర్ ప్రోగ్రామింగ్ మాడ్యూల్‌లో సంబంధిత విధానాలను మిళితం చేస్తుంది మరియు మాడ్యూల్స్ క్రింద డేటాను దాచిపెడుతుంది.
- OOP పారా డిగ్మ్ అల్గోరిథం కంటే డేటాపై నొక్కి చెబుతుంది. ఇది క్లాసులను మరియు ఆబ్జెక్టులను ఉపయోగించి ప్రోగ్రామ్‌లను అమలు చేస్తుంది.
- క్లాసు అనునది యూసర్ (user) నిర్వహించిన డేటా రకం. ఇది సారూప్య వస్తువుల సమూహాన్ని సూచిస్తుంది.
- ఆబ్జెక్టు, OOP యొక్క ప్రాథమిక యూనిట్ ఇది డేటా మరియు ఫంక్షన్‌ను ఒకే యూనిట్‌తో కలిసి సూచిస్తుంది.
- డేటా మరియు ఫంక్షన్‌ను ఒకే యూనిట్‌లో బంధించేయంత్రాంగాన్నిENCAPSULATION అంటారు. ఇది సంగ్రహణను అమలు చేస్తుంది.
- సంగ్రహణ అనునది నేపథ్య వివరాలను వెల్లడించకుండా అవసరమైన లక్షణాలను మాత్రమే చూపించడాన్ని వెల్లడిస్తుంది.
- మాడ్యులారిటీ అనేది ఒక పెద్ద అప్లికేషన్‌లో కంపోజ్ చేయగల సెటప్ ఫంక్షన్‌లో యూనిట్‌లుగా విభజించబడిన సిస్టమ్‌ను డిజైన్ చేయడం.
- పాలిమార్ఫిజం అంటే ఒకటి లేదా ఒకటి కంటే ఎక్కువ రూపాల్లో ప్రదర్శించబడే సందేశం లేదా ఫంక్షన్ యొక్క సామర్థ్యం.
- ఇప్పటి వరకే ఉన్న క్లాసుల నుండి క్రొత్త & క్లాసులను (ఉత్పన్నమైన క్లాసు) నిర్మించే సాంకేతికత వారసత్వం. వారసత్వం యొక్క అతి ముఖ్యమైన ప్రయోజనం కోడ్ పునర్వినియోగం, ఇది ప్రకృతిలో పరివర్తన చెందును.

11th Computer Science_Chapter-13 Telugu.indd 237 23-03-2023 16:44:47

మూల్యాంకనము



విభాగము - అ

సరైన జవాబులను ఎన్నుకొనుము

1. క్లాసులు మరియు వస్తువుల ఆధారంగా ప్రోగ్రామింగ్ విధానాన్ని వివరించడానికి ఏ పదాన్ని ఉపయోగిస్తారు.

   (అ) OOP (ఆ) POP (ఇ) ADT (ఈ) SOP

2. ప్రోసీజురస్‌లను మరింత లక్ష్యంగా చేసుకొనే పారాడైమ్

   (అ) ఆబ్జెక్టు ఓరియంటడ్ ప్రోగ్రాం (ఆ) ప్రోసీజురల్ ప్రోగ్రామింగ్

   (ఇ) మాడ్యూలర్ ప్రోగ్రాం (ఈ) స్ట్రక్చరల్ ప్రోగ్రాం

3. క్రింది వాటిలో యూసర్ డిఫైన్డ్ రకం ఏది?

   (అ) class (ఆ) float (ఇ) int (ఈ) object

4. కొన్ని లక్షణాలు మరియు ప్రవర్తనతో గుర్తించదగిన అంశం

   (అ) క్లాసు (ఆ) ఆబ్జెక్టు (ఇ) నిర్మాణం (ఈ) సభ్యుడు (member)

5. డేటాను మరియు ఫంక్షను ఒకే యూనిట్‌గా బంధించే విధానం

   (అ) వారసత్వం (ఆ) ఎన్‌కాప్సులేషన్

   (ఇ) పాలిమార్ఫిజం (ఈ) సంగ్రహణ

6. ప్రోగ్రామ్ ద్వారా direct access నుండి డేటా యొక్క ఇన్సులేషన్‌ను ఏమి అంటారు.

   (అ) డేటా హైడింగ్ (ఆ) ఎన్‌కాప్సులేషన్

   (ఇ) పాలిమార్ఫిజం (ఈ) సంగ్రహణ

7. క్రింది వాటిలో ఏ భావన సృష్టించబడవలసిన ఆబ్జెక్టు యొక్క అన్ని ముఖ్యమైన లక్షణాలను దాచియుండును

   (అ) క్లాసు (ఆ) ఎన్‌కాప్సులేషన్

   (ఇ) పాలిమార్ఫిజం (ఈ) సంగ్రహణ

8. క్రింది వాటిలో inheritance యొక్క ముఖ్య ప్రయోజనం ఏది?

   (అ) డేటా హైడింగ్ (ఆ) కోడ్ పునర్వినియోగం

   (ఇ) కోడ్ సవరణ (ఈ) ప్రాప్యత

9. "ఒక్కసారి వ్రాసి అనేక సార్లు ఉపయోగించు" అనుటను దేని ద్వారా సాధించవచ్చు?

   (అ) రిడండెన్సీ (ఆ) పునర్వినియోగం

   (ఇ) మార్పు (ఈ) కూర్పు

10. క్రింది వాటిలో ఏది డేటా యొక్క పరివర్తన స్వభావానికి మద్ధతు ఇస్తుంది?

    (అ) వారసత్వం (ఆ) ఎన్‌కాప్సులేషన్ (ఇ) పాలిమార్ఫిజం (ఈ) సంగ్రహణ

11th Computer Science_Chapter-13 Telugu.indd 238 23-03-2023 16:44:47

విభాగం - ఆ

సంక్షిప్తముగా జవాబులిమ్ము

1. విధానపరమైన ప్రోగ్రామింగ్ నమూనా మాడ్యూలర్ ప్రోగ్రామింగ్ ఏ విధంగా భిన్నంగా ఉంటుంది?
2. క్లాసులను మరియు ఆబ్జెక్టులను వేరు చేయండి.
3. పాలిమార్ఫిజం అంటే ఏమిటి?
4. ఎన్కాప్స్యులేషన్ మరియు సంగ్రహణ ఎలా పరస్పర సంబంధం కలిగి ఉన్నాయి?
5. OOP యొక్క నష్టాలను వ్రాయండి.

విభాగం - ఇ

క్లుప్తముగా జవాబులిమ్ము

1. పారాడైమ్ (paradigm) అనగానేమి? పారాడైమ్‌లో రకములను గుర్తించుము.
2. విధానపరమైన ప్రోగ్రామ్ యొక్క లక్షణాలను వ్రాయుము.
3. మాడ్యూలర్ ప్రోగ్రామింగ్ యొక్క కొన్ని లక్షణాలను పట్టికపరచుము.
4. మాడ్యూలైజేషన్ మరియు సాఫ్ట్‌వేర్ పునర్వినియోగం అంటే ఏమిటి?
5. సమాచారం దాచడం (information hiding) నిర్వచించండి.

విభాగం - ఈ

వివరంగా వ్రాయండి

1. ఆబ్జెక్టు ఓరియంటడ్‌కు మరియు ప్రోసీజురుల్ ప్రోగ్రామ్‌కు మధ్య గల తేడాలను వివరించండి.
2. OOPs యొక్క ఫలితాలను వివరింపుము.
3. OOPs కి మద్ధతు ఇచ్చే ప్రాథమిక భావనలపై ముఖ్య విషయములు రాయండి.

సూచన పుస్తకాలు (Reference):

(1) Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.

(2) The Complete Reference C++ (Forth Edition), Herbert Schildt.Mc.Graw Hills.

(3) Computer Science with C++ (A text book of CBSE XI and XII), SumitaArora, DhanpatRai& Co.

(4) A text book of CBSE XI and XII computer science by PreetiArora and Pinky Gupta.

(5) Computer Science with C++ Reeta shoo and Gagansahoo

(6) The C++ Programming Language,BjarneStroustrup

(7) C++ Primer (5th Edition) by S. B. Lippman, J. Lajoie

11th Computer Science_Chapter-13 Telugu.indd 239 23-03-2023 16:44:47

అధ్యాయము 14

క్లాస్ మరియు లక్ష్యాలు



అభ్యసనా లక్ష్యాలు

ఈ పాఠ్యాంశమును అభ్యసించిన విద్యార్థులు కింది విషయముపై అవగాహన పొందెదరు

- క్లాసస్ యొక్క ప్రయోజనాలను, కన్స్ట్రక్టర్లు మరియు డిస్ట్రక్టర్లు యొక్క లక్ష్యాలను అవగాహన పరుచుట.
- కన్స్ట్రక్టర్ మరియు డిస్ట్రక్టర్లును క్లాసస్‌ను ఉపయోగించి C++ ప్రోగ్రామ్‌ను నిర్మించుట.
- కన్స్ట్రక్టర్లు మరియు డిస్ట్రక్టర్లు ద్వారా క్లాస్ ప్రోగ్రామ్‌ను అమలు పరచుట మరియు డీబగ్ చేయుట.

14.1 క్లాసెస్ యొక్క పరిచయం (Introduction to Classes)

C++ యొక్క అతి ముఖ్యమైన లక్షము "క్లాస్". దీనిని Bjarne Stroustrup అనునతడు ప్రాముఖ్యత వాస్తవం ద్వారా 'C with classes'. ను మొదట ఇవ్వబడినది. C++ ద్వారా నాలుగు లక్షణాలు సాధారణంగా OOP నందు **సంగ్రహణ ఎన్కాప్స్యులేషన్, వారసత్వము మరియు inheritance పాలిమార్ఫిజం**.

14.1.1 క్లాస్ యొక్క అవసరాలు (Need for Class)

క్లాస్ అనేది డేటా మరియు దాని అనుబంధ విధులను ఒకదానితో ఒకటి ముడిపడి ఉంటుంది. డేటా రకం లక్షణాలను మాత్రమే కాకుండా అనుబంధ కార్యకలాపాలను కలిగి ఉన్న వాస్తవ ప్రపంచ సంస్థలను సూచించడానికి క్లాసెస్ అవసరం. ఇది వినియోగదారు నిర్వచించిన డేటా రకాన్నీ సృష్టించడానికి ఉపయోగపడుతుంది.

14.1.2 క్లాస్ ప్రకటన (Declaration of a class)

C++ నందు keyword **class** ను ఉపయోగించి క్లాస్ పేరు మరియు క్లాస్ నిర్వచించబడుతుంది. క్లాస్ యొక్క మొత్తం భాగమును curly brackets తో నిర్వచించబడుతుంది మరియు సెమికోలన్ చేత చివరి ప్రకటనల జాబితాను ముగించుదురు.

గమనిక

నిర్మాణము మరియు క్లాస్‌ల మధ్య ఉన్న ఏకైక వ్యత్యాసం ఏమిటంటే నిర్మాణ సభ్యులు అప్రమేయంగా **పబ్లిక్**. ఇక్కడ క్లాస్‌లో **ప్రైవేట్‌గా** ఉంటుంది.

సాధారణ రూపంలో క్లాస్ నిర్వచనము

```
class class-name
{
private:
        variable declaration;
        function declaration;
protected:
        variable declaration;
        function declaration;
public:
        variable declaration;
        function declaration;
};
```

- క్లాస్ అనునది సభ్యుల ప్రకటనను కలిగి ఉంటుంది. (డేటా సభ్యులు మరియు ప్రమేయ సభ్యులు)

11th Computer Science_Chapter-14 Telugu.indd 240 24-03-2023 10:35:48

- క్లాస్ నిర్మాణములో మూడు ప్రత్యేక భాగాలు కలవు. (దృష్టి ద్వారా ప్రత్యేక విభాగాలు) private , public మరియు protected

14.1.3 క్లాస్ ప్రాప్యత నిర్దేశకులు (Class Access Specifiers)

ఆబ్జెట్ ఓరియంటెడ్ ప్రోగ్రామింగ్ యొక్క ముఖ్యమైన లక్షణాలలో డేటా దాచడం **Data hiding** ఒకటి. ఇది క్లాస్ రకం యొక్క అంతర్గత ప్రాతినిధ్యాన్ని నేరుగా తీసుకోవడానికి ఆ ప్రోగ్రామ్ యొక్క ఫంక్షన్లను నిరోధించడానికి అనుమతినిస్తుంది. క్లాస్ సభ్యులకు ప్రాప్యత పరిమితి క్లాస్ యొక్క పబ్లిక్ ప్రైవేట్ మరియు ప్రొటెక్టడ్ విభాగాలచే పేర్కొనబడింది. ప్రభుత్వ, ప్రైవేట్ మరియు ప్రొటెక్టడ్ విభాగాలను ప్రాప్యత పరిమితి విభాగాలు అంటారు.

పబ్లిక్ సభ్యులు (The Public Members)

పబ్లిక్ సభ్యులను క్లాస్ వెలుపల ఎక్కిడి నుండైన యాక్సెస్ చేయవచ్చు, కానీ అది ప్రోగ్రామ్ లోపలనే కలిగి ఉండవలయును. మీరు ఏ సభ్యుల ఫంక్షన్‌ను ఉపయోగించకుండా కూడా పబ్లిక్ డేటా సభ్యుల విలువలను పొందవచ్చును.

ప్రైవేట్ సభ్యులు (The Private Members)

ప్రైవేట్ సభ్యులను క్లాస్ వెలుపల నుండి తీసుకోలేరు. క్లాస్ సభ్యుల విధులు మాత్రమే ప్రైవేట్ సభ్యులను తీసుకోవడం చేయగలవు. అప్రమేయంగా క్లాస్‌లోని సభ్యులందరూ ప్రైవేట్‌గా ఉంటారు.

ప్రొటెక్టడ్ సభ్యులు (The Protected Members)

ఒక protected సభ్యుడు ఒక private సభ్యుడితో సమానంగా ఉంటుంది. కానీ ఇది చిన్న క్లాస్‌లలో తీసుకొని చేయగల ఒక అదనపు ప్రయోజనాన్ని అందిస్తుంది. వీటిని ఉత్పన్న తరగతులు అని పిలుస్తారు (వారసత్వ క్లాసులు)

ఉదాహరణ

class అనే *Keyword,* క్లాస్ నిర్వచనాన్ని *ompiler* ద్వారా చూపును.
క్లాసు పేరు *tag name* వినియోగదారు నిర్వచించిన డేటా రకంగా పనిచేస్తుంది.
దీన్ని ఉపయోగించి ఒక క్లాస్ రకం సృష్టించబడుతుంది.

```
class student
{
private:
        char name [10];
        int rollno, mark1, mark2, total;
protected:
        void accept();
        void compute();
public:
        void display();
};
```

ఇది *private* సభ్యులతో కూడిన ప్రత్యేక సభ్యులు అదే విధంగా ఈ సభ్యులను క్లాస్ నందు బయట నుండి పొందలేము.

ఇందు *protected* సభ్యులతో కూడిన ప్రత్యేక సభ్యులు అదే విధంగా ఈ సభ్యులను క్లాస్ నందు బయటి నుండి పొందలేము.

ఇందులోని సభ్యులను వెలుపలి నుండి కూడా యాక్సెస్ చేయవచ్చు.

11th Computer Science_Chapter-14 Telugu.indd 241 24-03-2023 10:35:48

గమనిక

ఒక్కవేళ అన్ని క్లాస్ సభ్యులను నుండి private నిర్వచించినట్లు అయితే, క్లాస్ యొక్క object క్లాస్ నుండి దేనిని యాక్సెస్ చేయదు.

కార్యాచరణ I

క్రింది కోడ్ ఫ్రాగ్మెంట్ చెల్లుబాటుకానందుకు, కారణము తెలుపును.

(i)	(ii)

class count
{
int first;
int second;
public:
int first;
};
	class item
	{
	int prd;
	};
	item int prdno;

14.1.4 క్లాస్ సభ్యుల నిర్వచనము (Definition of class members)

క్లాస్ అనునది సభ్యులలో కూడినది. వీరు సభ్యులు డేటా సభ్యులు మరియు ఫంక్షన్‌గా విభజించబడుదురు. డేటా సభ్యులు అనునది డేటా చరరాశులు అగును. ఇవి క్లాస్ యొక్క లక్షణాలను సూచిస్తుంది. సభ్యుల ప్రక్రియ (methods) అనునది ఒక క్లాస్ యొక్క కొన్ని నిర్దిష్టమైన ప్రక్రియలు చేయడానికి సహాయపడే ఒక నిర్వచనము (attributes). ఫంక్షన్ సభ్యులను పద్ధతులు అని మరియు డేటా సభ్యులను attributes అని పిలిచెదరు.

ఉదాహరణ

```
Class result
{
Private;
        char name [10];
        int rollno,mark1, mark2, total;

Public:
        void accept();
        void display();
};
```

Data members

Member functions

గమనిక

క్లాసెస్ కొన్ని ప్రత్యేక ఫంక్షన్లను కలిగి ఉండును. అవి కన్స్ట్రక్టర్లు మరియు డిస్ట్రక్టర్లు.

14.1.5 క్లాస్‌లోని పద్ధతులను నిర్వచించుట (Defining methods of a class)

క్లాస్ యొక్క పద్ధతులు ఫంక్షన్లను నిర్వచించకుండా క్లాస్ యొక్క నిర్వచనం అసంపూర్తిగా ఉండును. సభ్యుల ఫంక్షన్ రెండు విధాలుగా నిర్వచించబడును.

(1) క్లాస్ లోపల నిర్వచించుట

(2) క్లాస్ బయట నిర్వచించుట

(1) క్లాస్ లోపల నిర్వచించుట

సభ్యుల ఫంక్షన్ క్లాస్ లోపల నిర్వచించునపుడు ఇది inline ఫంక్షన్‌లాగా పనిచేయును. అవి inline సభ్యుల ఫంక్షన్ అని పిలువబడును.

గమనిక

ఫంక్షన్‌ను inline ఫంక్షన్‌గా నిర్వచించునపుడు అవి ప్రోగ్రామ్‌ను compile చేయునపుడు ఆ నిర్వచనము పిలువబడు అన్ని చోట్ల దాని యొక్క copyని చేర్చును.

(2) క్లాస్ వెలుపల నిర్వచనం

సభ్యుల ఫంక్షన్లను క్లాస్ వెలుపల సాధారణ ఫంక్షన్ నిర్వచనం వలే నిర్వచించినపుడు (మీకు తెలసిన ఫంక్షన్ నిర్వచనాలు) దీనిని **non-inline** సభ్యుల ఫంక్షన్ లేదా అవుట్‌లైన్ సభ్యుల ఫంక్షన్ కానీ సభ్యుల ఫంక్షన్ అంటారు. ఈ ప్రయోజనం కోసం **(Scope resolution operator) [::]** ను ఉపయోగిస్తారు. Out line సభ్యుల ఫంక్షన్‌కు నిర్మించినచో క్రింది వాక్యనిర్మాణం,

11th Computer Science_Chapter-14 Telugu.indd 242 24-03-2023 10:35:48

Syntax

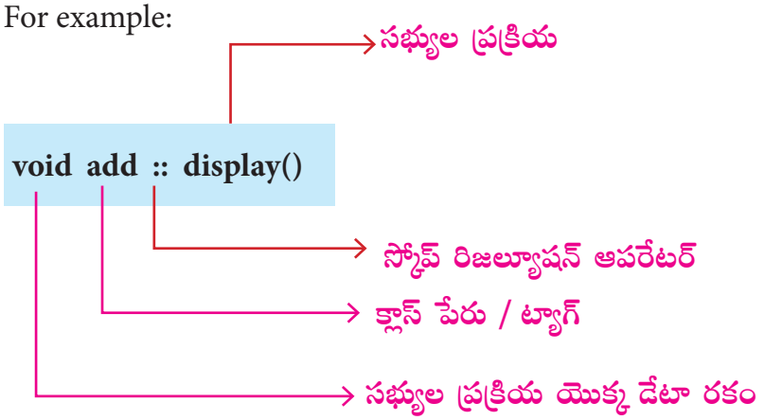
return_type class_name :: function_name (parameter list)

{

function definition

}

For example:



ఉదాహరణ 14.1 Inline and Outline సభ్యుల ప్రక్రియ

యాక్సెస్ స్పెసిఫైయర్ లేకపోతే ఆ సభ్యులను private గా సూచించబడును.

```
# include <iostream>
using namespace std;
class Box
{
        double width;          // no access specifier mentioned
public:
        double length;
        void printWidth( )              //inline member function definition
        {
                cout<<"\n The width of the box is..."<<width;
        }
        void  setWidth( double w );          //prototype of the function
};
void Box :: setWidth(double w)      // outline member function definition
{
        width=w;
}
int main( )
{
Box b;          // object for class Box
b.setWidth(10.0);      // Use member function to set the width.
b.printWidth( );                //Use member function to print the width.
return 0;
}
```

Output:

The width of the box is... 10

11th Computer Science_Chapter-14 Telugu.indd 243 24-03-2023 10:35:48

గమనిక

సభ్యుల ఫంక్షన్‌ను ప్రకటించునపుడు loop, switch లేదా goto స్టేట్‌మెంట్‌లను inline ఫంక్షన్‌గా తీసుకోవడం సరి అయినది కాదు

14.2 ఆబ్జెక్ట్‌లను సృష్టించడం (Creating Objects)

స్పెసిఫికేషన్ క్లాస్ యొక్క లక్షణాలను నిర్వచిస్తుంది. క్లాస్‌ను ఉపయోగించడానికి ఆ క్లాస్ యొక్క చరరాశులను ప్రకటించాలి. క్లాస్ లోపల ప్రకటించబడు చరరాశులను Objects అంటారు. Objects ను క్లాస్ యొక్క *instance* గా పిలుస్తారు.

ఉదాహరణ

student s;

పై స్టేట్‌మెంట్‌లో **'s'** అనునది **student**. క్లాస్ యొక్క instance అగును.

క్లాస్ యొక్క ఆబ్జెక్టులను రెండు విధాలుగా సృష్టించవచ్చు.

(1) గ్లోబల్ ఆబ్జెక్టు

(2) లోకల్ ఆబ్జెక్టు

(1) గ్లోబల్ ఆబ్జెక్టు

ఒక ఆబ్జెక్టు అన్ని ఫంక్షన్‌ను వెలుపల ప్రకటించబడితే లేదా క్లాస్ నిర్వచనం యొక్క ముగింపును సూచించే Flower bracket తరువాత వెంటనే వారి పేర్లను ఉంచితే దానిని గ్లోబుల్ ఆబ్జెక్టు అంటారు.

ఈ ఆబ్జెక్ట్‌లను క్లాస్‌లోని అన్ని ఫంక్షన్‌లు ద్వారా ఉపయోగించవచ్చు.

(2) లోకల్ ఆబ్జెక్టు

ఒక ప్రక్రియలో ఒక ఆబ్జెక్టును ప్రకటిస్తే దానిని లోకల్ ఆబ్జెక్టు అంటారు. దీని ఫంక్షన్‌ను బయట నుండి యాక్సెస్ చేయలేరు.

14.2 లోకల్ ఆబ్జెక్టు మరియు గ్లోబల్ ఆబ్జెక్టును నిర్వచించే ప్రోగ్రామ్

```
# include <iostream>
# include <conio>
using namespace std
class add
{
  int a,b;
  public:
   int sum;
  void getdata()
  {
   a=5;
   b=10;
   sum = a+b;
  }
} a1;                    //global object
add a2;                  //global object
int main()
{
add a3;          // Local object
a1.getdata();
a2.getdata();
a3.getdata();
cout<<a1.sum;
cout<<a2.sum;
cout<<a3.sum;
return 0;
}
```

Output:

151515

కార్యాచరణ 2

క్రింది ప్రోగ్రామ్‌లోని లోపాలను గుర్తించండి error

```
class A
{
        float x;
        void init()
        {
        A a1;
        X1.5=1;
        }};
void main()
{    A1.init(); }
```


11th Computer Science_Chapter-14 Telugu.indd 244 24-03-2023 10:35:48

14.3 ఆబ్జెక్టులకు మెమరీ స్థలం కేటాయించుట (Memory allocation of objects)

సభ్యుల ఫంక్షన్‌లో క్లాస్ నిర్వచనంలో భాగంగా నిర్వచించబడినపుడే దానిని మెమరీ స్థలంలో సృష్టించబడును. మరియు నిల్వచేయబడుతాయి. ఎందుకంటే ఆ క్లాసుకు చెందిన అన్ని ఆబ్జెక్టులు ఒకే సభ్యుల ఫంక్షన్‌ను ఉపయోగిస్తారు. కాబట్టి ప్రతి ఆబ్జెక్టును సృష్టించినపుడు సభ్యుల ఫంక్షన్‌కు ప్రత్యేక మెమరీ స్థలం కేటాయించబడుతుంది. ఎందువలన అనగా డేటా చరరాశి వేరువేరు ఆబ్జెక్టులకు వేర్వేరు డేటా విలువలను కలిగి ఉంటాయి.

14.3 ఆబ్జెక్టులకు మెమరీ కేటాయింపును వివరించే ప్రోగ్రామ్

```
# include <iostream>
using namespace std;

class product
{
        int code, quantity;
        float price;
        public:
        void assignData();
        void Print();
};
int main()
{
        product p1, p2;
        cout<<"\n Memory allocation for object p1 " <<sizeof(p1);
        cout<<"\n Memory allocation for object p2 " <<sizeof(p2);
        return 0;
}
```

Output:

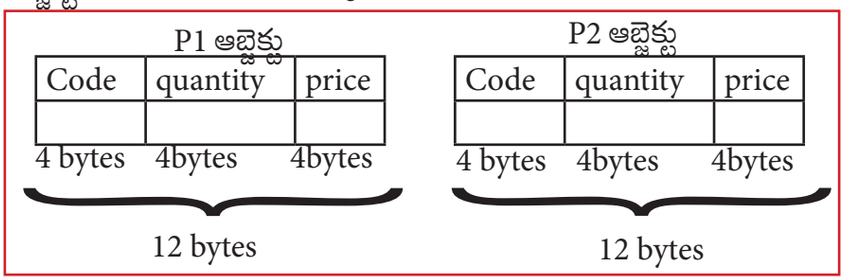
```
Memory allocation for object p1  12
Memory allocation for object p2  12
```

సభ్యుల ఫంక్షన్‌లో assignData( ) మరియు Print( ) సాధారణ మెమరీ స్థలాన్ని కేటాయించబడుతుంది. p1 మరియు p2 రెండు ఆబ్జెక్టులకు సంబంధించిన డేటా విలువలు అన్ని సాధారణ ప్రాంతంలో కోడ్ ప్రాంతానికి ప్రాప్యతను కలిగి ఉంటాయి.

గమనిక

సభ్యులకు మెమరీ స్థలం కేటాయించడానికి క్లాస్ యొక్క ఆబ్జెక్టును సృష్టించాలి.

p1 మరియు p2 ఆబ్జెక్టులకు మెమరీ కేటాయింపు వివరణ



11th Computer Science_Chapter-14 Telugu.indd 245 24-03-2023 10:35:48

కార్యాచరణ 3

క్రింది ప్రోగ్రామ్‌లో s1 మరియు s2 ఆబ్జెక్టు యొక్క మెమరీ పరిమాణం ఎంత?

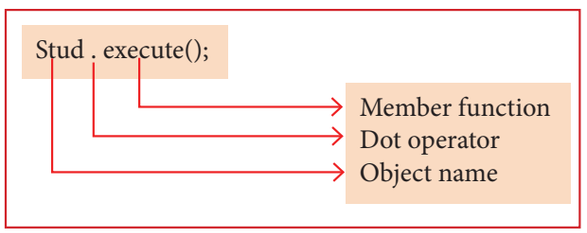
```
class sum
{
        int n1,n2;
        public:
        void add(){int n3=10;n1=n2=10;}
} s1,s2;
```

14.4 క్లాస్ సభ్యులను ప్రస్తావించడం (Referencing class members)

ఒక క్లాస్ యొక్క సభ్యులను వినియోగించడానికి క్లాస్ ఆబ్జెక్ట్ ను ఉపయోగించిన పిదప సభ్యుల ఫంక్షన్‌ను పిలవడానికి సాధారణ వాక్య నిర్మాణం

Object_name . function_name(actual parameter);

ఈ క్రింది ఉదాహరణ తీసుకొనుము.



14.4 ఒక ఆబ్జెక్టు యొక్క పరస్పర చర్యను వివరించే **C++** ప్రోగ్రామ్

```
#include<iostream>
using namespace std;
class compute
{
 int n1,n2;      //private by default
 public :
 int n;
 int add (int a, int b)   //inline member  function
 {
 int c=a+b;      //int c ; local variable for this function
 return c;
 }

}c1,c2;
int main()
{
 c1.n =c1.add(12,15); //member function is called
 c2.n =c2.add(8,4);
 cout<<"\n Sum of object-1  "<<c1.n;
 cout<<"\n Sum of object-2  "<<c2.n;
 cout<<"\n Sum of the two objects are  "<<c1.n+c2.n;
          return 0;
}
```

Output:

```
Sum of object-1  27
Sum of object-2  12
Sum of the two objects are  39
```

గమనిక

ఒక క్లాస్ కోసం ఆబ్జెక్టుల శ్రేణిని కూడా సృష్టించవచ్చును. ఇది రకాల శ్రేణి మాదిరిగానే ప్రకటించబడుతుంది మరియు నిర్వచించబడుతుంది.

ఉదాహరణ:

student s[10];

ఇక్కడ student క్లాస్ పేరు మరియు s[10] అనేది student క్లాస్ కోసం 10 ఆబ్జెక్టులను సృష్టిస్తుంది.

14.5 కన్స్‌ట్రక్టర్లు - పరిచయం (Introduction to Constructors)

క్లాస్ యొక్క నిర్వచనం ఒక కొత్త వినియోగదారు నిర్వచించిన డేటా రకాన్నీ మాత్రమే సృష్టిస్తుంది. క్లాస్ యొక్క దృవీకరణను దృవీకరించడం (ఉత్పత్తి మరియు ప్రారంభించడం) అవసరం. కన్స్‌ట్రక్టర్లను ఉపయోగించి ఆబ్జెక్టులను దృవీకరించవచ్చును.

11th Computer Science_Chapter-14 Telugu.indd 246 24-03-2023 10:35:48

14.5.1 కన్స్ట్రక్టర్ల యొక్క అవసరాలు (Need for Constructors)

c++ లోని శ్రేణి లేదా నిర్మాణాన్ని డిక్లరేషన్ సమయంలో ప్రారంభించవచ్చు.

ఉదాహరణకు

```
struct sum
  {
        int n1,n2;
  };
class add
  {
        int num1,num2;
  };
int  main()
  {
        int arr[]={1,2,3};        //declaration and initialization of array
        sum s1={1,1};            //declaration and initialization of structure object
        add  a1={0,0};          // class object declaration and initialization throws
                                                              compilation error
   }
```

క్లాస్ యొక్క సభ్యుల ప్రక్రియ ఎటువంటి సభ్యుల సంబంధం లేకుండా ఏదైననూ యాక్సెస్ చేయవచ్చును.

నిర్మాణం లేదా శ్రేణికి సమానమైన class type object ను డిక్లరేషన్ సమయంలో ప్రారంభించడం సాధ్యం కాదు. ఎందువలన అనగా క్లాస్ ఆబ్జెక్టులు వారి అనుబంధ ప్రాప్యత నిర్దేశకులు (private or protected or public) ఉన్నాయి. అందువల్ల క్లాసెస్‌లో ప్రత్యేక సభ్యుల ఫంక్షన్ కలిగియుండును. కావున దానిని కన్స్ట్రక్టరు అందురు. కన్స్ట్రక్టరు ఫంక్షన్ క్లాసు ఆబ్జెక్టును ప్రారంభిస్తుంది.

14.6 ప్రకటన మరియు నిర్వచనం (Declaration and Definition)

క్లాస్ యొక్క పరిధిలోకి వచ్చినపుడు కన్స్ట్రక్టర్ అను పిలువబడు ఒక ప్రత్యేక ఫంక్షన్ అమలవుతుంది. కన్స్ట్రక్టర్ ఫంక్షన్ పేరు క్లాసు పేరుకు సమానమైన పేరును కలిగి ఉంటుంది. కన్స్ట్రక్టర్ ఏమి తిరిగి ఇవ్వదు. అవి ఏ డేటా రకంతోను సంబంధం కలిగి ఉండదు. ఇది క్లాస్ నిర్వచనం లోపల లేదా వెలుపల నిర్మించిచబడవచ్చు.

ఉదాహరణ 1:

14.5 క్లాస్ స్పెసిఫికేషన్ లోపల నిర్వచించిన కన్స్ట్రక్టర్

```
#include<iostream>
using namespace std;
class Sample
{
        int i,j;
        public :
          int k;
          Sample()
          {
             i=j=k=0;   //constructor defined inside the class
          }
};
```


11th Computer Science_Chapter-14 Telugu.indd 247 24-03-2023 10:35:48

14.6.1 కన్స్‌ట్రక్టర్ యొక్క విధులు (Functions of constructor)

కన్స్‌ట్రక్టర్ అనేది క్లాస్ యొక్క ప్రత్యేక ప్రారంభ సభ్యుల ప్రక్రియ అని మనకు ఇప్పుడు తెలుసు. కానీ, క్లాస్ యొక్క ఉదాహరణ ప్రకటించునపుడు లేదా సృష్టించినపుడల్లా స్వయంచాలకంగా పిలుస్తారు. కన్స్‌ట్రక్టర్ యొక్క ప్రధాన విధులు

1) ఆబ్జెక్టుకు మెమరీ స్థలాన్ని కేటాయించడం

2) క్లాస్ ఆబ్జెక్టు యొక్క డేటా సభ్యుని ప్రారంభించడం

తరగతి ఆబ్జెక్టును ప్రారంభించడానికి మరొక మార్గం కలదు. కానీ ఆ సందర్భంలో మనం సభ్యుల ఫంక్షన్‌ను స్పష్టంగా పిలువగలము.

14.7 కన్స్ట్రక్టర్ల రకాలు (Types of constructors)

కన్స్ట్రక్టర్ల నందు చాలా రకాలు కలవు.

- **డిఫాల్ట్ కన్స్ట్రక్టర్లు**

పారామీటరును అంగీకరించని కన్స్ట్రక్టర్లను డిఫాల్ట్ కన్స్ట్రక్టర్లు అంటారు. ఉదాహరణకు క్లాస్ డేటాలో ఉన్న డేటా :: డేటా() Data ::Data() డీఫాల్ట్ కన్స్ట్రక్టర్లు. ఈ కన్స్ట్రక్టర్లను ఉపయోగించి ఇతర డేటా రకాల చలరాశులు సృష్టించబడిన విధంగానే ఆబ్జెక్టులు సృష్టించబడతాయి. ఒక క్లాస్‌లో స్పష్టమైన కన్స్‌ట్రక్టర్ (యూజర్ డిఫైన్స్ కన్స్‌ట్రక్టర్) లేకపోతే కంపైలర్ దానంతట అవే డిఫాల్ట్ కన్స్‌ట్రక్టర్‌ను ఉత్పత్తి చేస్తుంది.

- **పారామీటర్ చేయబడిన కన్స్‌ట్రక్టర్**

వాదనలు తీసుకోగలిగిన కన్స్‌ట్రక్టర్‌లను పారామీటర్ కన్స్‌ట్రక్టర్ అని అంటారు. ఈ రకమైన కన్స్‌ట్రక్టర్‌లు విభిన్న ప్రారంభ విలువలతో ఆబ్జెక్టును సృష్టించడానికి సహాయపడుతుంది. ఫంక్షన్‌కు పారామీటర్‌ను పంపడం ద్వారా ఇది సాధించబడుతుంది.

ఉదాహరణ:

```
Data :: Data(int,int);
```

- **కాపీ కన్స్ట్రక్టర్లు (Copy Constructors)**

స్వంత క్లాస్‌లో ఉన్న ఆబ్జెక్టును ఇప్పటికే ఉన్న క్లాసును సూచించివున్న కన్స్‌ట్రక్టర్‌ను కాపీ కన్స్‌ట్రక్టర్ అని అంటారు. సాధారణంగా డేటా (డేటా &) రూపంలో ఉంటుంది.

ఇక్కడ డేటా క్లాస్ యొక్క పేరు

1) ఒక ఆబ్జెక్టు ఏదైన సభ్యుని ఫంక్షన్‌కు పారామీటర్‌గా పంపబడినపుడు

ఉదాహరణకు void Data::putdata(Data x);

2) సభ్యుల ఫంక్షన్‌కు ఒక ఆబ్జెక్టును తిరిగి ఇచ్చినప్పుడు

ఉదాహరణ Data getdata() { }

3) ఒక ఆబ్జెక్టు దాని own class యొక్క ఉదాహరణను సూచించినపుడు, పంపబడునపుడు
ఉదాహరణ Data d1, d2 (d1); // d2(d1) ను కాపీ కన్స్‌ట్రక్టర్ అని పిలుస్తారు.

11th Computer Science_Chapter-14 Telugu.indd 248 24-03-2023 10:35:48

ఉదాహరణ 14.6 కన్స్ట్రక్టర్ రకాలు

```
#include<iostream>
using namespace std;
class Data
 {
       int i, j;
     public:
       int k;
        Data()
        {
             cout<<"Non Parametrerized constructor";
              i=0;
               j=0'
         }
         Data(int a,int b)
         {
             cout<<"Parametrerized constructor";
              i=a;
               j=b'
         }
        Data(Data &a)
         {
             cout<<"Copy constructor";
              i=a.i;
               j=b.j'
         }

          void display()            //member function
           {
              cout<< i<<j;
            }
       };
int main()
{
          Data  d1,d2(10,20),d3(d2);
          d1.display();
          d2.display();
          d3.display();
           return 0;

}
```

14.8 కన్స్ట్రక్టర్ల ఆహ్వానం (Invocation of constructors)

పారామీటర్ చేయబడిన కన్స్ట్రక్టర్‌ను ఉపయోగించి ఆబ్జెక్టులను సృష్టించడానికి రెండు మార్గాలు ఉన్నాయి.

- Implicit కాల్
- Explicit కాల్

11th Computer Science_Chapter-14 Telugu.indd 249 24-03-2023 10:35:48

14.8.1 Implicit కాల్ (Implicit call)

ఈ పద్ధతిలో ఒక ఆబ్జెక్టును సృష్టించినప్పుడల్లా పారామీటర్ కన్స్ట్రక్టర్ చేయబడిన కన్స్ట్రక్టర్ దానంతట అదే ఆబ్జెక్టును సృష్టించును. ఉదాహరణకు simple s1(10,20) ఆబ్జెక్టు s1 పారామీటర్ చేయబడిన కన్స్ట్రక్టర్‌ను సృష్టించడానికి దానంతట అదే ప్రారంభించబడుతుంది.

14.8.2 Explicit కాల్ (Explicit call)

స్పష్టమైన కాల్ పద్ధతిలో ఒక తాత్కాలిక ఆబ్జెక్టును సృష్టించుటకు ఇది చాలా సరి అయిన పద్ధతి. డేటాను కోల్పోయే అవకాశం లేదు. ఒక తాత్కాలిక ఆబ్జెక్టు దీనిలో ఒక expression ను ఉపయోగించబడుతున్నంత కాలం ఇది మెమరీలో ఉంటుంది. ఆ తరువాత ఇది నాశనం అవుతుంది.

14.9 ఆబ్జెక్టులను క్రియాశీలకంగా ప్రారంభించడం (Dynamic initialization of Objects)

రన్‌టైమ్‌లో ప్రాథమిక విలువలను ఇచ్చినపుడు దానిని క్రియాశీలక ప్రారంభం అంటారు.

క్రియాశీలక ప్రారంభాన్ని వివరించడానికి ఉదాహరణ 14.7

```
#include<iostream>
using namespace std;
class X
{   int n;
    float avg;
    public:
    X(int p,float q)
    {  n=p;
       avg=q;  }
    void disp()
    {    cout<<"\n Roll numbe:- " <<n;
         cout<<"\nAverage :- "<<avg;        } };
int main()
{
int a ; float b;
         cout<<"\nEnter the Roll Number";
         cin>>a;
         cout<<"\nEnter the Average";
         cin>>b;
         X x(a,b);       // dynamic initialization
         x.disp();
         return 0;
}
```

Output:

```
Enter the Roll Number 1201
Enter the Average 98.6
 Roll numbe:- 1201
Average :- 98.6
```


11th Computer Science_Chapter-14 Telugu.indd 250 24-03-2023 10:35:48

14.10 కన్స్ట్రక్టర్ల లక్షణాలు (Characteristics of Constructors)

- కన్స్ట్రక్టర్ పేరు తరగతికి సమానంగా ఉండాలి.
- కన్స్ట్రక్టర్‌ను తిరిగి పొందే రకం పేర్కొనబడదు.
- ఒక కన్స్ట్రక్టర్ పారామీటర్ జాబితాను కలిగి ఉంటుంది.
- కన్స్ట్రక్టర్ ఫంక్షన్‌కు అధిక శ్రమ కలిగించవచ్చును.
- దానిని స్వతహాగా పొందలేము. కానీ దాని నుండి ఉత్పన్నమైన క్లాస్‌ను బేస్ క్లాస్ కన్స్ట్రక్టర్ అని పిలుస్తాము.
- వినియోగదారు నిర్వచించిన కన్స్ట్రక్టర్ లేకపోవడం వలన కంపైలర్ ఒక కన్స్ట్రక్టర్‌ను ఉత్పత్తి చేస్తుంది.
- కంపైలర్ సృష్టించిన కన్స్ట్రక్టర్ ఒక పబ్లిక్ సభ్యుల ఫంక్షన్.
- ఆబ్జెక్టులు సృష్టించినపుడు కన్స్ట్రక్టరు దానంతట అదే అమలు అవుతుంది.
- ఒక కన్స్ట్రక్టర్ దాని క్లాస్ టైప్ (class type) యొక్క ఆబ్జెక్టును సృష్టించడానికి స్పష్టంగా ఉపయోగించవచ్చును.

14.11 డిస్ట్రక్టర్స్ (Destructors)

క్లాస్ ఆబ్జెక్టు అనునది దాని పరిధి నుండి బయటకు వెళ్ళినపుడు డిస్ట్రక్టర్ అని పిలువబడే ఒక ప్రత్యేక ఫంక్షన్ అమలు అవుతుంది. డిస్ట్రక్టర్ క్లాస్ ట్యాగ్ వలె అదే పేరును కలిగి ఉంది. కానీ, '~' **(tilde)** తో prefix చేయబడి ఉన్నది. డిస్ట్రక్టర్ ఫంక్షన్ కూడా ఏమీ ఇవ్వదు మరియు ఇది ఏ డేటా రకంతో సంబంధం కలిగి ఉండదు.

14.11.1 డిస్ట్రక్టర్ల అవసరం (Need of Destructors)

క్లాస్ ఆబ్జెక్టు దాని పరిధిని దాటినచో ఆ ప్రత్యేక ప్రక్రియను డిస్ట్రక్టర్ అంటారు. డిస్ట్రక్టర్ ప్రక్రియ ఒక ఆబ్జెక్టు యొక్క మెమోరిని తొలగిస్తుందో ఆ ఆబ్జెక్టును సృష్టించే సమయంలో కన్స్ట్రక్టర్ కేటాయించింది.

14.11.2 డిక్లరేషన్ మరియు నిర్వచనము (Declaration and Definition)

డిస్ట్రక్టర్ అనేది ఒక ప్రత్యేక సభ్యుల ప్రక్రియ. ఇది ఒక ఆబ్జెక్టు యొక్క జీవితకాలం ముగిసినపుడు మరియు కన్స్ట్రక్టర్ నిర్మించిన ఆబ్జెక్టును నాశనం చేసినపుడు పిలుస్తారు. సాధారణంగా public క్రింద ప్రకటిస్తారు.

ఉదాహరణ 14.8 ఒక క్లాస్‌లో డిస్ట్రక్టర్ ఫంక్షన్‌ను వివరించడం

```
#include<iostream>
using namespace std;
class simple
{
private:
int a, b;
public:
simple()
{
a= 0 ;
b= 0;
cout<< "\n Constructor of class-simple ";
}
void getdata()
{
cout<<"\n Enter values for a and b ";
cin>>a>>b;
}
void putdata()
{
cout<<"\nThe two integers are .. ";
cout<<<<a<<'\t'<< b<<endl;
cout<<"\n The sum = "<<a+b;
}
~simple()
{   cout<<"\n Destructor is executed ";}
};
int main()
{
simple s;
s.getdata();
s.putdata();
return 0;
}
```

Output:

```
Constructor of class-simple
Enter values for a and b  6 7
The two integers are .. 6      7
The sum = 13
Destructor is executed
```

14.12 డిస్ట్రక్టర్ల లక్షణాలు (Characteristics of Destructors)

- tilde క్యారక్టర్ '~' ద్వారా ప్రిఫిక్స్ చేసిన క్లాస్ మాదిరిగానే డిస్ట్రక్టర్‌కు అదే పేరు కలదు.
- డిస్ట్రక్టర్లకు వాదనలు ఉండదు
- దీనికి రిటర్న్ టైప్ లేదు.
- డిస్ట్రక్టర్లను ఓవర్‌లోడ్ చేయలేము.
- యూజర్ నిర్వహించిన డిస్ట్రక్టర్ లేనప్పుడు ఇది కంపైలర్ ద్వారా ఉత్పత్తి అవుతుంది.
- డిస్ట్రక్టర్ స్వయంచాలకంగా అమలు చేయబడుతుంది. నియంత్రణ ఆబ్జెక్టు పరిధిని నాశనం చేయడానికి తరగతి పరిధికి చేరుతుంది.
- దీనిని వారసత్వం inheritance పొందలేము.

11th Computer Science_Chapter-14 Telugu.indd 251 24-03-2023 10:35:48

గుర్తుంచుకోవలసిన విషయాలు

- ఒక క్లాస్ డేటా మరియు అనుబంధ ఫంక్షన్లను కలుపుతుంది.
- C++ లోని ఒక క్లాస్ ఈ రకమైన ఏ ఆబ్జెక్టును సృష్టించవచ్చో ఉపయోగించి వినియోగదారు నిర్వహించిన డేటా రకాన్నీ చేస్తుంది.
- క్లాస్ డేటా సభ్యులు మరియు సభ్యుల ఫంక్షన్ యాక్సెస్ స్పెసిఫైయర్లు మరియు క్లాస్ టాగ్ పేరు ఇవ్వబడుతుంది.
- క్లాస్ యొక్క సభ్యుల ఫంక్షన్ క్లాస్ (ఇన్లైన్) నిర్వచనంలో లేదా క్లాస్ నిర్వచనం వెలుపల నిర్వచించవచ్చు.
- ఈ క్లాస్ రకం యొక్క ఆబ్జెక్టును ఉపయోగించడం ద్వారా క్లాస్ యొక్క public సభ్యులను క్లాస్ వెలుపల నుండి నేరుగా పొందవచ్చును.
- ఒక క్లాస్ డేటా మరియు ఫంక్షన్‌ను ఒక దానిఇతో ఒకటి సంబంధం ఉండడం ద్వారా OOP లక్షణాలను ENCAPSULATION కి మద్ధత్తు ఇస్తుంది.
- క్లాస్ బయట గల సమాచారం ప్రపంచం నుండి private మరియు protected సభ్యుల ద్వారా దాచే డేటాకు మద్ధత్తు ఇస్తుంది.
- సభ్యుల ఫంక్షన్‌ను అదే క్లాసులోని మరొక సభ్యుల ఫంక్షన్ ద్వారా పిలిచినప్పుడు దీనిని సభ్యుల ఫంక్షన్ల గూడు (nesting of member function) అని పిలుస్తారు.
- అదే క్లాసుకు చెందిన మెంబర్లు కల్గిన క్లాసుపేరుతోనున్న స్కోప్ రిజల్యూషన్ ఆపరేటర్ (::) ఉపయోగించినపుడు క్లాస్-పేరు :: ఫంక్షన్ -పేరు వచ్చును. :: వేరియబుల్ - నేమ్‌నందలి వేరియబుల్‌నేమ్ గ్లోబల్ వేరియబుల్ ను అందించును, (ఫైల్ స్కోప్‌తో ఉండునది).
- క్లాస్ ఆబ్జెక్టు పరిధి నుండి బయటకు వెళ్ళినపుడు డిస్ట్రక్టర్ అని పిలువబడే ఒక ప్రత్యేక ఫంక్షన్ అమలు అవుతుంది.
- కన్స్‌ట్రక్టర్ ఫంక్షన్ మెమరీను కలిగియుండి క్లాస్ ఆబ్జెక్టును అనుమతించును.
- క్లాస్ ఇంటెన్స్ స్కోప్‌లోకి వచ్చినప్పుడు, కన్స్‌ట్రక్టర్‌ను నిలిపియంచెడి స్పెషల్ పుంక్షన్ రూపొందును.
- క్లాస్ ఆబ్జెక్టు స్కోప్‌గు దాటి వెళ్ళినపుడు, డిస్‌ట్రక్టర్‌ను నిలిపియుంచెడి స్పెషల్ ఫంక్షన్ రూపొందును.
- కన్స్‌ట్రక్టర్ ఫంక్షన్ పేరు మరియు డిస్ట్రక్టర్ క్లాస్ ట్యాగ్‌కు సమానమైన పేరును కలిగి ఉన్నాయి.
- పారామీటర్లు లేని కన్స్‌ట్రక్టరును డిఫాల్ట్ కన్స్‌ట్రక్టర్ అని అంటారు.
- డిఫాల్ట్ ఆర్గ్యుమెంట్ ఉన్న కన్స్‌ట్రక్టర్ డిఫాల్ట్ కన్స్‌ట్రక్టర్‌కు సమానం
- కన్స్‌ట్రక్టర్‌లు మరియు డిస్ట్రక్టర్‌లు రెండునూ ఏమీ పొందలేవు. అవి ఏ డేటా రకంతోను సంబంధం కలిగి ఉండదు.
- ఆబ్జెక్టును క్రియాశీలకంగా ప్రారంభించవచ్చును.

11th Computer Science_Chapter-14 Telugu.indd 252 24-03-2023 10:35:48



అభ్యాసము:

1 క్లాస్ ఎంప్లాయీని ఈ క్రింది స్పెసిఫికేషన్‌తో నిర్వచించండి.

private members of class Employee
empno- integer
ename – 20 characters
basic – float
netpay, hra, da, - float
calculate () – A function to find the basic+hra+da with float return type
public member functions of class employee
havedata() – A function to accept values for empno, ename, basic, hra, da and
call calculate() to compute netpay
dispdata() – A function to display all the data members on the screen

మూల్యాంకనం

విభాగం – అ

సరైన జవాబులు ఎన్నుకోండి.

1. క్లాస్ లోపల ప్రకటించిన చరరాశులు

   (అ) డేటా (ఆ) ఇన్లైన్
   (ఇ) పద్ధతి (ఈ) గుణాలు

2. సభ్యుల ఫంక్షన్లు గురించి క్రింది వాటిలో ఏది నిజం (లేదా) తప్పు?

   i) డాట్ ఆపరేటర్‌ను ఉపయోగించి సభ్యుల ఫంక్షన్ మరొక ఫంక్షన్‌కు నేరుగా కాల్ చేయవచ్చు.

   ii) సభ్యుల ఫంక్షన్ క్లాస్ యొక్క ప్రైవేట్ డేటాను యాక్సెస్ చేయవచ్చు.

   (అ) i) నిజం, ii) నిజం (ఆ) i) తప్పు, ii) నిజం
   (ఇ) i) నిజం, ii) తప్పు (ఈ) i) తప్పు, ii) తప్పు

3. సభ్యుల ఫంక్షన్ అని పిలువబడే డాట్ ఆపరేటర్‌ను ఉపయోగించకుండా నేరుగా మరొక సభ్యుల ప్రక్రియకు కాల్ చేయవచ్చు

   (అ) ఉప ఫంక్షన్
   (ఆ) ఉప సభ్యుడు
   (ఇ) సభ్యుల ఫంక్షన్ యొక్క నెస్టింగ్ (nesting)
   (ఈ) సభ్యుల ఫంక్షన్ యొక్క sibling

4. క్లాస్ లోపల నిర్వచించిన సభ్యుల ఫంక్షన్ _____ వలె ప్రవర్తిస్తుంది.

   (అ) ఇన్లైన్ (ఆ) నాన్ ఇన్లైన్
   (ఇ) అవుట్‌లైన్ (ఈ) డేటా

5. ఈ క్రింది వాటిలో ఏది యాక్సెస్ స్పెసిఫైయర్ అనుకోకుండా మార్పు డేటాను రక్షిస్తుంది?

   (అ) ప్రైవేట్ (ఆ) ప్రొటెక్టెడ్
   (ఇ) పబ్లిక్ (ఈ) గ్లోబల్

6.
```
class x
 {
     int y;
     public:
     x(int z){y=z;}
     } x1[4];
int main()
{  x  x2(10);
  return 0;}
```

పై ప్రోగ్రాం కోసం ఎన్ని ఆబ్జెక్టులు సృష్టించబడతాయి?
(అ) 10 (ఆ) 14 (ఇ) 5 (ఈ) 2

7. కన్స్‌ట్రక్టర్ గురించి క్రింది స్టేట్‌మెంట్ నిజమా కాదా పేర్కొనండి.

   i) కన్స్‌ట్రక్టర్‌లును ప్రైవేట్ విభాగంలో ప్రకటించాలి.

   ii) ఆబ్జెక్టులు సృష్టించబడినప్పుడు కన్స్‌ట్రక్టర్‌లు స్వయంచాలకంగా ప్రారంభించబడుతాయి.

   (అ) నిజం, నిజం (ఆ) నిజం, తప్పు
   (ఇ) తప్పు, నిజం (ఈ) తప్పు, తప్పు

11th Computer Science_Chapter-14 Telugu.indd 253 24-03-2023 10:35:48

8. క్రింది ప్రోటోటైప్ కోసం క్రింది వాటిలో ఏ కన్స్ట్రక్టర్ అమలు చేయబడుతుంది?

add display(add &); // add is a class name

(అ) డిఫాల్ట్ కన్స్ట్రక్టర్

(ఆ) పారామీటర్ చేయబడిన కన్స్ట్రక్టర్

(ఇ) కాపీ కన్స్ట్రక్టర్

(ఈ) పారామీటర్ కానీ కన్స్ట్రక్టర్

విభాగం - ఆ

క్లుప్తముగా సమాధానము లిమ్ము.

1. సభ్యులు అనగా ఎవరు?
2. రెండు యూజర్ డిఫైన్డు డేటా రకం ద్వారా నిర్మాణం మరియు క్లాస్‌ను వేరు చేయండి.
3. OOP పరంగా క్లాస్ మరియు ఆబ్జెక్టుల మధ్య గల తేడాను తెలుపుము.
4. కంపైలర్ స్వయంచాలకంగా కన్స్ట్రక్టర్‌ను ఉత్పత్తి చేయగలిగినప్పటికే కన్స్ట్రక్టర్‌ను నిర్వచించడం మంచి అభ్యాసంగా ఎందుకు పరిగణించబడుతుంది?
5. డిస్ట్రక్టర్ యొక్క ప్రాముఖ్యతను వ్రాయండి.

విభాగం - ఇ

సంక్షిప్తంగా సమాధానము లిమ్ము.

1. వాక్య నిర్మాణ లోపాలు ఏవైన ఉంటే వాటిని తొలగించి లోపాలను underline చేయండి.

```
#include<iostream>
$include<stdio>
class mystud
{       int studid =1001;
        char name[20];
   public
        mystud( )     { }
        void register ( )
        {cin>>stdid; gets(name);   }
void display ( )
{cout<<studid<<": "<<name<<endl;}
}
int main( )
{ mystud MS;
  register.MS( );
  MS.display( );
}
```

2. ఆబ్జెక్టును క్రియాశీలకంగా ఎలా ప్రారంభిస్తారో ఉదాహరణతో రాయండి.
3. పబ్లిక్ యాక్సెస్బిలిటీ క్రింద కన్స్ట్రక్టర్లను మరియు డిస్ట్రక్టర్‌లను ప్రకటించడం వల్ల కలిగే ప్రయోజనాలు తెలుపుము.
4. C++ కోడ్ ఇచ్చినప్పుడు క్రింది ప్రశ్నలకు సమాధానములు ఇవ్వండి. (i) & (ii).ఆబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రామింగ్‌లో అంటే ఏమిటి మరియు ఎప్పుడు సహాయం కోరుతుంది / ఎప్పుడు పిలవబడుతుంది.

```
class TestMeOut
{
 public:
 ~TestMeOut()        //Function 1
  {cout<<"Leaving the exam hall"<<endl;}
  TestMeOut()        //Function 2
  {cout<<"Appearing for exam"<<endl;}
   void MyWork()   //Function 3
  {cout<<"Answering"<<endl;}    };
```

(i) In Object Oriented Programming, what is Function 1 referred as and when doesit get invoked / called ?

(ii) In Object Oriented Programming, what is Function 2 referred as and when doesit get invoked / called ?

విభాగం - డి

విపులముగా సమాధానము లిమ్ము.

1. కన్స్ట్రక్టర్ మరియు డిస్ట్రక్టర్‌ల మధ్య గల తేడాను పేర్కొనండి

11th Computer Science_Chapter-14 Telugu.indd 254 24-03-2023 10:35:48

2. క్రింది C++ వివరణతో క్లాస్ RESORT ను నిర్వచించుము.

Private members:

Rno // Data member to store room number

Name //Data member to store user name

Charges //Data member to store per day charge

Days //Data member to store the number of days

Compute( )/*A function to calculate total amount as Days * Charges and if the total amount exceeds 11000 then total amount is 1.02 * Days *Charges */

Public member:

GetInfo( ) /* Function to Read the information like name , room no, charges and days*/

DispInfo( )/* Function to display all entered details and total amount calculated using COMPUTE function*/

3. క్రింది వాటికి అవుట్‌పుట్‌ను వ్రాయండి.

```
#include<iostream>
using namespace std;
class student
{
        int rno, marks;
        public:
        student(int r,int m)
        { cout<<"Constructor "<<endl;
          rno=r;
          marks=m;
        }
        void printdet()
        {
          marks=marks+30;
          cout<<"Name: Bharathi"<<endl;
          cout<<"Roll no : "<<rno<<"\n";
          cout<<"Marks : "<<marks<<endl;
          }
};
int main()
{
        student s(14,70);
        s.printdet();
        cout<< "Back to Main";
        return 0;
}
```

సూచన పుస్తకాలు:

(1) Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.

(2) The Complete Reference C++ (Forth Edition), Herbert Schildt.Mc.Graw Hills.

11th Computer Science_Chapter-14 Telugu.indd 255 24-03-2023 10:35:49

అధ్యాయము 15

ఫాలిమార్ఫిజం (Polymorphism)



అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశాన్ని అభ్యసించిన తరువాత విద్యార్థులు నేర్చుకొను అంశాలు

- ఓవర్‌లోడింగ్ యొక్క ఆవశ్యకతను అర్థం చేసుకోగలరు
- ఫంక్షన్, కన్స్ట్రక్టర్ మరియు ఆపరేటర్ ఓవర్‌లోడింగ్‌ను ఉపయోగించి C++ ప్రోగ్రామ్‌లను రూపొందించగలరు.
- Polymorphism భావనను ఉపయోగించి ప్రోగ్రామ్‌లను అమలు మరియు డీబగ్ చేయగలరు.

15.1 పరిచయం

polymorphism అనే పదానికి అనేక రూపాలు అని అర్థం poly రూపాలు morph అంటే ఒకటి కంటే ఎక్కువ రూపాలలో ప్రదర్శించబడే ఒక సందేశం లేదా ఫంక్షన్ యొక్క సామర్థ్యం. C++ లో ఫంక్షన్ ఓవర్‌లోడింగ్ మరియు ఆపరేటర్ ఓవర్‌లోడింగ్ ద్వారా polymorphism సాధించబడుతుంది. ఓవర్‌లోడింగ్ అనే పదానికి రెండు లేదా అంతకంటే ఎక్కువ విభిన్న అర్థాలు గల పేరు అని అర్థం. అందువల్ల "**ఓవర్ లోడ్ ఫంక్షన్**" అంటే ఒకటి కంటే ఎక్కువ విభిన్న అర్థాలు గల ఫంక్షన్‌ను సూచిస్తుంది.

15.2 ఫంక్షన్ ఓవర్‌లోడింగ్

సందేశం లేదా డేటాను ఒకటి కంటే ఎక్కువ రూపాలలో ప్రాసెస్ చేసే ఫంక్షన్ సామర్థ్యాన్ని ఫంక్షన్ ఓవర్‌లోడింగ్ అంటారు. మరోమాటలో చెప్పాలంటే ఫంక్షన్ ఓవర్‌లోడింగ్ అంటే ఒకే పరిధిలోని రెండు లేదా అంతకంటే ఎక్కువ ఫంక్షన్‌లు ఒకే పేరును పంచుకుంటాయి కానీ వాటి పారామితులు (parameters) భిన్నంగా ఉంటాయి. ఈ పరిస్థితిలో ఒకే పేరును పంచుకొనే ఫంక్షన్‌లు ఓవర్‌లోడ్ చేయబడ్డాయి అని అంటారు మరియు ఈ ఫంక్షన్‌ను ఫంక్షన్ ఓవర్‌లోడింగ్ అంటారు. ఫంక్షన్ యొక్క పారామితుల సంఖ్య మరియు రకాలను **function's signature** అంటారు. ఓవర్‌లోడెడ్ ఫంక్షన్ ఉపయోగించినపుడు నిర్వచనాలలో పేర్కొన్న పారామితి రకాలతో ఫంక్షన్‌ను కాల్ చేయడానికి వాడిన argument రకాలను పోల్చడం ద్వారా కంపైలర్ (compiler) ను ఉపయోగించడానికి అత్యంత సరైన నిర్వచనాన్ని నిర్ణయిస్తుంది. అత్యంత సరైన ఓవర్‌లోడ్ ఫంక్షన్ లేదా ఆపరేటర్‌ను ఎంచుకొనే ఫంక్షన్‌ను ఓవర్‌లోడ్ రిసొల్యూషన్ అంటారు.

15.2.1 ఓవర్‌లోడింగ్ ఫంక్షన్ యొక్క ఆవశ్యకత

కొన్నిసార్లు ఒకే చర్యకోసం అనేక అర్థవంతమైన విభిన్న పేర్లను కనుగొనడం కష్టం. వృత్తం, త్రిభుజం మరియు దీర్ఘచతురస్రం యొక్క వైశాల్యాలను కనుగొనవలసిన పరిస్థితులలో ఈ క్రింది ఫంక్షన్ నమూనాను పరిగణించండి.

```
float area_circle(float radius)                    // to calculate the area of a circle
float area_triangle(float half,floatbase,float height)   // to calculate the area of a triangle
```


11th Computer Science_Chapter-15 Telugu.indd 256 24-03-2023 12:03:03

float area_rectangle(float length , float breadth) // to calculate the area of a rectangle

దీనిని ఒకే ఫంక్షన్ header నుపయోగించి ఈ క్రింది విధంగా కూడా వ్రాయవచ్చు.

float area (float radius);

float area (float half, float base, float height);

float area (float length , float breadth);

వివర ఉదాహరణ 15.1 ఓవర్‌లోడింగ్ ఫంక్షన్‌ను ప్రదర్శించడానికి C++ ప్రోగ్రామ్

```
#include <iostream>
using namespace std;
void print(int i)
 {cout<< " It is integer " << i <<endl;}
void print(double  f)
{  cout<< " It is float " << f <<endl;}
void print(string c)
{  cout<< " It is string " << c <<endl;}
int main() {
        print(10);
        print(10.10);
        print("Ten");
        return 0;
}
```

Output:

```
It is integer 10
 It is float 10.1
It is string Ten
```

గమనిక

ఫంక్షన్ ఓవర్‌లోడింగ్ అనేది polymorphism అమలు చేయడమే కాకుండా ప్రోగ్రాంలోని పోలికల సంఖ్యను తగ్గించి ప్రోగ్రాంను వేగంగా అమలుచేస్తుంది. ఇది గుర్తించుకోవలసిన ఫంక్షన్ పేర్ల సంఖ్యను తగ్గించడం ద్వారా ప్రోగ్రామర్‌కు కూడా సహాయపడుతుంది.

15.2.2 ఫంక్షన్ ఓవర్‌లోడింగ్ యొక్క నిబంధనలు

1. ఓవర్‌లోడ్ ఫంక్షన్ దాని యొక్క ఆర్గ్యుమెంట్ల సంఖ్యలోగాని, డేటా రకాలలోగాని విభిన్నంగా ఉండాలి.
2. ఓవర్‌లోడ్ చేయబడిన return type ఫంక్షను రకం ఒకే డేటా రకం ఓవర్‌లోడింగ్ కోసం పరిగణించబడదు.
3. ఓవర్‌లోడ్ డిఫాల్ట్ ఆర్గ్యుమెంట్లు ఫంక్షన్‌లు, ఓవర్‌లోడింగ్‌లో పారామితి (parameter) జాబితాలో భాగంగా పరిగణించబడవు.

11th Computer Science_Chapter-15 Telugu.indd 257 24-03-2023 12:03:03

వివర ఉదాహరణ 15.2 ఫంక్షన్ ఓవర్‌లోడింగ్‌ను ప్రదర్శించడానికి C++ ప్రోగ్రామ్

```
#include <iostream>
using namespace std;
long add(long, long);
long add(long,long,long);
float add(float, float);
int main()
{
 long a, b, c,d;
 float e, f, g;
 cout << "Enter three integers\n";
cin >> a >> b>>c;
d=add(a,b,c);          //number of arguments different  but  same data type
cout << "Sum of 3 integers: " << d << endl;
cout << "Enter two integers\n";
cin >> a >> b;
c = add(a, b);         //two arguments with same data type
cout << "Sum of 2 integers: " << c << endl;
cout << "Enter two floating point numbers\n";
 cin >> e >> f;
 g = add(e, f);        //two arguments with different data type
cout << "Sum of floats: " << g << endl;
}
long add(long c, long g)
{
 long sum;
 sum = c + g;
 return sum;
}
float add(float c, float g)
{
 float sum;
 sum = c + g;
 return sum;
}
long add(long c, long g,long h)
{
 long sum;
 sum = c + g+h;
 return sum;
}
```

Output

```
Enter three integers
3 4 5
Sum of 3 integers: 12
Enter two integers
4 6
Sum of 2 integers: 10
Enter two floating point numbers
2.1 3.1
Sum of floats: 5.2
```


11th Computer Science_Chapter-15 Telugu.indd 258 24-03-2023 12:03:03

15.3 కన్స్ట్రక్టర్ ఓవర్‌లోడింగ్

కన్స్ట్రక్టర్లు తరగతుల యొక్క ప్రత్యేక ఫంక్షన్లు కాబట్టి కన్స్ట్రక్టర్ల కోసం ఫంక్షన్ ఓవర్‌లోడింగ్‌ను వర్తింపజేయవచ్చును. ఒక తరగతి విభిన్న signature ఒకటి కంటే ఎక్కువ కన్స్ట్రక్టర్లను కలిగియుంటుంది. కన్స్ట్రక్టర్ ఓవర్‌లోడింగ్ ఒక తరగతి (class) కోసం బహుళ రకాల వస్తువులను సృష్టించే సౌలభ్యాన్ని అందిస్తుంది.

వివర ఉదాహరణ 15.3 కన్స్ట్రక్టర్ ఓవర్‌లోడింగ్

కంపైలర్ ఇచ్చిన నంబర్ ఫంక్షన్‌ని దానిపేరు మరియు రిటర్న్ రకం ద్వారా కన్స్ట్రక్టర్ గుర్తిస్తుంది.

```
#include<iostream>
using namespace std;
class add
{ int num1, num2, sum;
public:
add()
{       cout<<"\n Constructor without parameters.. ";
num1= 0;  num2= 0;  sum = 0; }
add ( int s1, int s2 )
{ cout<<"\n Parameterized constructor... ";
 num1= s1;  num2=s2; sum=0; }
add (add &a)
{ cout<<"\n Copy Constructor ... ";
 num1= a.num1;
 num2=a.num2;
 sum = 0; }
void getdata()
{ cout<<"\nEnter data ... "; cin>>num1>>num2; }
void addition()
{ sum=num1+num2; }
void putdata() {
cout<<"\n The numbers are..";
cout<<num1<<'\t'<<num2;
cout<<"\n The sum of the numbers are.. "<< sum; } };
int main() {
add a, b (10, 20) , c(b);
a.getdata();
a.addition();
b.addition();
c.addition();
cout<<"\n Object a : ";
a.putdata();
cout<<"\n Object b : ";
b.putdata();
cout<<"\n Object c.. ";
c.putdata();
return 0; }
```


11th Computer Science_Chapter-15 Telugu.indd 259 24-03-2023 12:03:04

Output

```
Constructor without parameters..
 Parameterized constructor...
 Copy Constructor ...
Enter data ...  20 30
 Object a :
 The numbers are..20    30
 The sum of the numbers are.. 50
 Object b :
 The numbers are..10    20
 The sum of the numbers are.. 30
 Object c..
 The numbers are..10    20
 The sum of the numbers are.. 30
```

గమనిక

బహుళ కన్స్ట్రక్టర్లు ఉన్నందున ఒక వస్తువును సృష్టించేటప్పుడు కన్స్ట్రక్టర్‌కు argument కూడా పంపాలి.

15.4 ఆపరేటర్ ఓవర్‌లోడింగ్

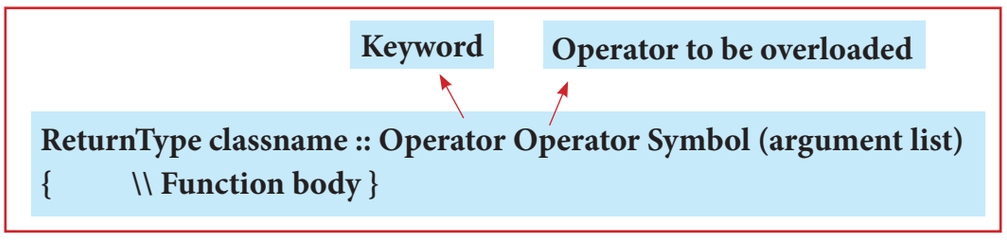
ఆపరేటర్ ఓవర్‌లోడింగ్ అనే పదం +,++,-,—,+=,-=,*.<,> వంటి సాధారణ C++ ఆపరేటర్‌లకు అదనపు కార్యాచరణను అందించడాన్ని సూచిస్తుంది. ఇది కూడా ఒక రకమైన polymorphism. దీనిలో వినియోగదారు నిర్వచించిన అర్థాన్ని అందించడానికి ఆపరేటర్ ఓవర్‌లోడ్ చేయబడతాడు.

ఉదాహరణకు Integer, String(concatenation) మొదలైన వివిధ రకాల డేటాపై అదనంగా చేయడానికి C++ ఆపరేటర్‌ని ఓవర్‌లోడ్ చేయవచ్చు.

దాదాపు అన్ని ఆపరేటర్లు C++ లో ఓవర్‌లోడ్ చేయబడతాయి. అయితే ఓవర్‌లోడ్ చేయలేని కొన్ని ఆపరేటర్లు ఉన్నాయి. అవి ఈ క్రింది విధంగా ఉన్నాయి.

- Scope operator (**::**)
- Sizeof
- Member selector (**.**)
- Member pointer selector (*)
- Ternary operator (**?:**)

ఆపరేటర్ ఓవర్‌లోడింగ్ నిర్మాణ నిబంధన



11th Computer Science_Chapter-15 Telugu.indd 260 24-03-2023 12:03:04

15.4.1 ఆపరేటర్ ఓవర్‌లోడింగ్ యొక్క పరిమితులు

ఆపరేటర్ ఓవర్‌లోడింగ్‌ను అమలుచేస్తున్నప్పుడు గుర్తుంచుకోవలసిన కొన్ని పరిమితులు క్రింది విధంగా ఉన్నాయి.

1. ఆపరేటర్ యొక్క ప్రాధాన్యత మరియు సాంగత్యమును మార్చలేము.
2. క్రొత్త ఆపరేటర్లు సృష్టించబడరు, ఇప్పటికే ఉన్న ఆపరేటర్లు మాత్రమే ఓవర్‌లోడ్ చేయబడతారు.
3. ఆపరేటర్ విధానం యొక్క అర్థాన్ని పునర్నిర్వచించలేము. పూర్ణాంకాలు జోడించబడుటను మార్చలేరు. ఒక ఆపరేటర్‌కు మాత్రమే అదనపు విధులు ఇవ్వబడతాయి.
4. ఓవర్‌లోడ్ చేయబడిన ఆపరేటర్లకు default ఆర్గ్యుమెంట్లు ఉండకూడదు
5. మైనరీ ఆపరేటర్లు ఓవర్‌లోడ్ అయినప్పుడు ఎడమ చేతి ఆబ్జెక్ట్ తప్పనిసరిగా సంబంధిత తరగతికి సంబంధించిన ఆబ్జెక్ట్‌గా ఉండాలి.

వివర ఉదాహరణ 15.5 సంక్లిష్ట సంఖ్యలో '+' ను ఉపయోగించి బైనరి ఆపరేటర్ ఓవర్‌లోడింగ్

```
#include<iostream>
using namespace std;
class complex
{  int real,img;
   public:
void read()
{
cout<<"\nEnter the REAL PART : ";
cin>>real;
cout<<"\nEnter the IMAGINARY PART : ";
cin>>img;
}
complex operator +(complex c2)
{
    complex c3;
    c3.real=real+c2.real;
    c3.img=img+c2.img;
    return c3;
}
    void display()
{  cout<<real<<"+"<<img<<"i";  }
};
int main()
{
    complex c1,c2,c3;
    int choice, cont;
    cout<<"\n\nEnter the First Complex Number";
    c1.read();
    cout<<"\n\nEnter the Second Complex Number";
    c2.read();
    c3=c1+c2;  // binary + overloaded
    cout<<"\n\nSUM = ";
    c3.display();
    return 0;
 }
```

Output

```
Enter the First Complex Number
Enter the REAL PART : 3
Enter the IMAGINARY PART : 4
Enter the Second Complex Number
Enter the REAL PART : 5
Enter the IMAGINARY PART : 8
SUM = 8+12i
```


11th Computer Science_Chapter-15 Telugu.indd 261 24-03-2023 12:03:04

వివర ఉదాహరణ 15.6 ఆపరేటర్ ఓవర్‌లోడింగ్‌ను ఉపయోగించి స్ట్రింగ్ యొక్క సంయోగం

```
#include<string.h>
#include<iostream>
using namespace std;
class strings
{
        public:
        char s[20];
        void getstring(char str[])
{
        strcpy(s,str);
  }
        void operator+(strings);
};
void strings::operator+(strings ob)
{
        strcat(s,ob.s);
        cout<<"\nConcatnated String is:"<<s;
}
int main()
{
        strings ob1, ob2;
        char string1[10], string2[10];
        cout<<"\nEnter First String:";
        cin>>string1;
        ob1.getstring(string1);
        cout<<"\nEnter Second String:";
        cin>>string2;
        ob2.getstring(string2);
        //Calling + operator to Join/Concatnate strings
        ob1+ob2;
        return 0;
}
```

Output

```
Enter First String:COMPUTER
Enter Second String:SCIENCE
Concatnated String is:COMPUTERSCIENCE
```


11th Computer Science_Chapter-15 Telugu.indd 262 24-03-2023 12:03:04

గుర్తుంచుకోవలసిన అంశాలు

- C++ ఫంక్షన్ ఓవర్‌లోడింగ్ మరియు ఆపరేటర్ ఓవర్‌లోడింగ్ ద్వారా పాలిమార్ఫిజమ్ సాధించబడుతుంది.
- ఓవర్‌లోడింగ్ అనే పదానికి రెండు లేదా అంతకంటే ఎక్కువ విభిన్న అర్థాలు కలిగిన పేరు అని అర్థం.
- ఓవర్‌లోడ్ ఫంక్షన్ అనేది ఒకటి కంటే ఎక్కువ విభిన్న అర్థాలను కలిగియుండే ఫంక్షన్‌ను సూచిస్తుంది.
- ఓవర్‌లోడ్ చేయబడిన ఫంక్షన్‌లు ఒకే పేరు మరియు వేర్వేరు సంతకాలు (ఆర్గ్యుమెంట్ సంఖ్య మరియు ఆర్గ్యుమెంట్ రకం) కలిగియుంటాయి.
- ఒక ఫంక్షన్ యొక్క ఆర్గ్యుమెంట్ జాబితాను function signature అంటారు.
- ఒక సూచిక పారామితిని (reference parameter) తీసుకొని మరొక సాధారణ కాల్ బై వాల్యూ (call-by-value) పారామితిని తీసుకొన్నప్పుడు రెండు ఫంక్షన్‌లు ఓవర్‌లోడ్ చేయబడవు.
- సాధారణ ఫంక్షన్‌లు అలాగే మెంబర్ ఫంక్షన్‌లు కూడా ఓవర్‌లోడ్ చేయబడతాయి.
- ఒక తరగతి ఓవర్‌లోడ్ చేయబడిన కన్స్ట్రక్టర్‌లను కలిగియుంటుంది. కానీ డిస్ట్రక్టర్ ఫంక్షన్ ఓవర్‌లోడ్ చేయబడదు.
- ఒక ఆపరేటర్‌కు ప్రత్యేక అర్థాన్ని ఇచ్చే విధానాన్ని ఆపరేటర్ ఓవర్‌లోడింగ్ అంటారు.
- ఆపరేటర్ ఓవర్‌లోడింగ్ అనేది చాలా వరకు C++ ఆపరేటర్‌లకు క్రొత్త నిర్వచనాలను అందిస్తుంది.
- యూసర్ డిఫైన్డ్ రకాలు (వస్తువులు) కూడా ఓవర్‌లోడ్ చేయబడవచ్చు.
- ఓవర్‌లోడ్ చేయబడిన ఆపరేటర్ యొక్క నిర్వచనం 'ఆపరేటర్' అనే మూలపదం keyword ను ఉపయోగించి అందించబడుతుంది, దాని తర్వాత ఒక ఆపరేటర్ గుర్తు ఉంటుంది.
- క్రింది వాటిని మినహాయించి అన్నీ C++ ఆపరేటర్లను ఓవర్‌లోడ్ చేయగలము.
- Scope resolution operator (::), sizeof (), Conditional operator (?:), Member selection(.) and Member pointer selector (*) operator

మూల్యాంకనము

3B0EM

భాగము – అ

సరైన సమాధానము ఎన్నుకొనుము

1. క్రింది వాటిలో ఒకటి కంటే ఎక్కువ విభిన్న అర్థాలను కలిగియుండే ఫంక్షన్‌ను సూచించేది ఏది?

   (అ) ఫంక్షన్ ఓవర్‌లోడింగ్ (ఆ) మెంబర్ ఓవర్‌లోడింగ్

   (ఇ) ఆపరేటర్ ఓవర్‌లోడింగ్ (ఈ) ఆపరేషన్స్ ఓవర్‌లోడింగ్

2. క్రింది వాటిలో ఏది ప్రోగ్రామ్‌లోని పోలికల సంఖ్యను తగ్గిస్తుంది

   (అ) ఆపరేటర్ ఓవర్‌లోడింగ్ (ఆ) ఆపరేషన్స్ ఓవర్‌లోడింగ్

   (ఇ) ఫంక్షన్ ఓవర్‌లోడింగ్ (ఈ) మెంబర్ ఓవర్‌లోడింగ్

3.
```
void dispchar(char ch='$',int size=10)
    {
      for(int i=1;i<=size;i++)
      cout<<ch;
    }
```


11th Computer Science_Chapter-15 Telugu.indd 263 24-03-2023 12:03:04

క్రింది ఇన్‌పుట్ కోసం మీరు $ ను 10 సార్లు ప్రింట్ చేయడానికి dispchar () ఫంక్షన్‌ను ఎలా అమలు చేస్తారు?

(అ) dispchar(); (ఆ) dispchar(ch,size);

(ఇ) dispchar($,10); (ఈ)dispchar('$',10 times);

4. ఫంక్షన్ ఓవర్‌లోడింగ్‌కు సంబంధించి క్రింది వాటిలో ఏది నిజం కాదు?

(అ) ఓవర్‌లోడ్ చేయబడిన ఫంక్షన్లు తప్పనిసరిగా వాటి signatureలో తేడా ఉండాలి.

(ఆ) ఫంక్షన్ ఓవర్‌లోడ్ చేయడం కోసం రిటన్ టైప్ కూడా పరిగణించబడుతుంది.

(ఇ) ఓవర్‌లోడడ్ ఫంక్షన్‌ల డిఫాల్ట్ ఆర్గ్యుమెంట్ ఓవర్‌లోడింగ్ కోసం పరిగణించబడవు.

(ఈ) డిస్ట్రక్టర్ ఫంక్షన్ ఓవర్‌లోడ్ చేయబడదు.

5. ఫంక్షన్ ఓవర్‌లోడింగ్ కోసం క్రింది వాటిలో invalid ప్రోటోటైప్ ఏది

(అ) void fun (intx);
void fun (char ch) ;

(ఆ) void fun (intx);
void fun (inty);

(ఇ) void fun (double d);
void fun (char ch);

(ఈ) void fun (double d);
void fun (inty);

భాగము - ఆ

లఘు సమాధానాలిమ్ము

1. ఫంక్షన్ ఓవర్‌లోడింగ్ అనగానేమి?
2. ఓవర్‌లోడడ్ కాని ఆపరేటర్లను జాబితా పరుచుము.
3. class add{int x; public: add(int)}; కన్స్ట్రక్టర్ కోసం అవుట్‌లైన నిర్వచనాన్ని వ్రాయండి.
4. ఫంక్షన్ యొక్క రిటన్ రకం ఫంక్షన్‌ను ఓవర్‌లోడ్ చేయడంలో సహాయపడుతుందా?
5. ఫంక్షన్‌ను ఓవర్‌లోడ్ చేయడం వల్ల ఉపయోగం ఏమిటి?

భాగము - ఇ

సంక్షిప్త సమాధానాలిమ్ము

1. ఫంక్షన్ ఓవర్‌లోడింగ్ యొక్క నియమాలు ఏవి?
2. అనేక ఫంక్షన్లు ఉన్నప్పుడు ఏ ఫంక్షన్‌ను ప్రారంభించాలని compiler ఎలా నిర్ణయిస్తుంది? ఒక ఉదాహరణను ఇవ్వండి.
3. ఆపరేటర్ ఓవర్‌లోడింగ్ అంటే ఏమిటి? ఓవర్‌లోడ్ చేయగల ఆపరేటర్లకు కొన్ని ఉదాహరణలు ఇవ్వండి.
4. కన్స్ట్రక్టర్ ఓవర్‌లోడింగ్ యొక్క ప్రయోజనాలను చర్చించండి.
5. class sale (int cost, discount ;public: sale(sale &) నిర్దేశించిన కన్స్ట్రక్టర్ కోసం non inline definition ను వ్రాయండి.

11th Computer Science_Chapter-15 Telugu.indd 264 24-03-2023 12:03:04

భాగము - ఈ

విపులంగా సమాధానాలివ్వుము

1. ఆపరేటర్ ఓవర్‌లోడింగ్ యొక్క నియమాలును పేర్కొనుము?

2. క్రింద ఇవ్వబడిన class ను గమనించి (i) నుండి (v) వరకు గల ప్రశ్నలకు సమాధానము ఇవ్వండి.

```
class Book {
int BookCode ; char Bookname[20];float fees;
public:
Book( )                         //Function 1
{ fees=1000;
  BookCode=1;
  strcpy(Bookname,"C++");  }
void display(float C)           //Function 2
{   cout<<BookCode<<":"<<Bookname<<":"<<fees<<endl;   }
~Book( )                        //Function 3
{  cout<<"End of Book Object"<<endl;   }
       Book (intSC,char S[ ],float F)  ;          //Function 4
};
```

(i) పై ప్రోగ్రామ్‌లో ఫంక్షన్ 1 మరియు ఫంక్షన్ 4 కలిపి దేనిని సూచిస్తాయి?

(ii) ఫంక్షన్ 3 ద్వారా ఏ భావన వివరించబడింది ? ఈ ఫంక్షన్‌ను ఎప్పుడు పిలుస్తారు / invoked చేస్తారు?

(iii) ఫంక్షన్ 3 యొక్క ఉపయోగమేమిటి?

(iv) ఫంక్షన్ 1 మరియు ఫంక్షన్ 2 లను అమలుచేయడానికి main లో statement లను రాయండి.

(v) ఫంక్షన్ 4 యొక్క నిర్వచనాన్ని రాయండి.

3. క్రింది ప్రోగ్రాంకు output ను రాయండి.

```
include<iostream>
using namespace std;
class Seminar
{ int Time;
public:
Seminar()
 {      Time=30;cout<<"Seminar starts now"<<endl;    }
void Lecture()
{ cout<<"Lectures in the seminar on"<<endl; }
Seminar(int Duration)
{   Time=Duration;cout<<"Welcome to Seminar "<<endl; }
Seminar(Seminar &D)
{   Time=D.Time;cout<<"Recap of Previous Seminar Content "<<endl;}
```


11th Computer Science_Chapter-15 Telugu.indd 265 24-03-2023 12:03:04

```
~Seminar()
{cout<<"Vote of thanks"<<endl; } };
int main()
{       Seminar s1,s2(2),s3(s2);
        s1.Lecture();
        return 0;
}
```

4. ఈ క్రింది ప్రోగ్రాం నుండి అడిగిన ప్రశ్నలకు సమాధానాలిమ్ము.

```
#include<iostream>
#include<string.h>
using namespace std;
class comp {
public:
char s[10];
void getstring(char str[10])
  { strcpy(s,str);    }
void operator==(comp);
};
void comp::operator==(comp ob)
{ if(strcmp(s,ob.s)==0)
 cout<<"\nStrings are Equal";
else
 cout<<"\nStrings are not Equal"; }
int main()
{ comp ob, ob1;
char string1[10], string2[10];
cout<<"Enter First String:";
cin>>string1;
ob.getstring(string1);
cout<<"\nEnter Second String:";
cin>>string2;
ob1.getstring(string2);
ob==ob1;
return 0; }
```

(i) ప్రోగ్రాం ముగిసే పరిధి scope వున్న ఆబ్జెక్ట్‌లను పేర్కొనండి.

(ii) ప్రోగ్రాం మధ్యలో వ్యర్థమయ్యే object పేరు తెల్పండి.

(iii) ఓవర్‌లోడ్ అయిన ఆపరేటర్ పేరు మరియు దానిని అమలు చేసే statement ను రాయండి.

(iv) ఓవర్‌లోడడ్ మెంబర్ ఫంక్షన్ యొక్క prototype ను రాయండి.

(v) ఓవర్‌లోడడ్ ఆపరేటర్ కోసం ఏ రకమైన ఆపరేండ్‌లు ఉపయోగించబడతాయి?

(vi) పై ప్రోగ్రాంలో ఏ కన్స్ట్రక్టర్ అమలుచేయబడింది? ప్రోగ్రాం యొక్క అవుట్‌పుట్‌ను రాయండి.

వివరణాత్మక అధ్యయనం (CASE STUDY)

500 రూపాయలు ప్రారంభ మొత్తంతో మీరు ఒక Kitty Bank ను కలిగయున్నారని అనుకొందాం మరియు మీరు దానికి మరికొంత మొత్తాన్ని జోడించాలి. 500 రూపాయలు ప్రారంభ విలువతో మొత్తం అనే డేటా మెంబర్‌తో 'Deposit' class ను సృష్టించండి. ఇప్పుడు ఈ తరగతికి చెందిన మూడు కన్స్ట్రక్టర్‌లను ఈ క్రింది విధంగా తయారు చేయండి.

1. ఎలాంటి పారామితి లేకుండా (without any parameter) ఎటువంటి మొత్తం జోడించబడదు.
2. పారామితితో కూడిన మొత్తం Kitty Bank కు జోడించబడుతుంది.
3. ఎప్పుడైతే మొత్తం జోడించబడిందో అదనంగా సమాన మొత్తం స్వయంచాలకంగా జమచేయబడుతుంది.

'Deposit' అనే ఆబ్జెక్ట్‌ను సృష్టించి చివరి మొత్తాన్ని Kitty Bank లో ప్రదర్శించండి.

పరిశీలనా గ్రంథాలు:

1. Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.
2. The Complete Reference C++ (Forth Edition), Herbert Schildt. Mc.Graw Hills.
3. The C++ Programming Language,BjarneStroustrup

11th Computer Science_Chapter-15 Telugu.indd 266 24-03-2023 12:03:04

అధ్యాయము 16

ఇన్‌హెరిటెన్స్ (Inheritance)

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశము పూర్తయిన తరువాత విద్యార్థులు నేర్చుకొను విషయాలు

- ఇన్‌హెరిటెన్స్ యొక్క ఆవశ్యకత గురించి అర్థం చేసుకొనుట.
- ఇన్‌హెరిటెన్స్‌లను ఉపయోగించి C++ ప్రోగ్రామ్‌ల నిర్మాణం
- ఇన్‌హెరిటెన్స్ భావనను కలిగి ఉన్న ప్రోగ్రామ్‌లను అమలు మరియు డీబగ్ చేయడం.

16.1 ఇన్‌హెరిటెన్స్ పరిచయం (Introduction to Inheritance)

ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామింగ్ యొక్క ముఖ్యమైన లక్షణాలలో ఇన్‌హెరిటెన్స్ ఒకటి. ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రాంలో ఇన్‌హెరిటెన్స్ కొత్త తరగతి (class) మరియు దాని ఆబ్జెక్ట్‌లను ఇప్పటికే ఉన్న తరగతుల లక్షణాల నుంచి స్వీకరిస్తుంది. కొత్త తరగతిని సృష్టించడానికి ప్రాతిపదికగా ఉపయోగించే తరగతిని సూపర్ క్లాస్ లేదా బేస్ క్లాస్ అంటారు. సూపర్ క్లాస్ నుండి వచ్చిన క్లాస్‌ను సబ్ క్లాస్ లేదా డిరైవుడ్ క్లాస్ అంటారు.

16.2 ఇన్‌హెరటెన్స్ అవసరం (Need for Inheritance)

ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామ్ యొక్క కోడ్ పునర్వినియోగం యొక్క ముఖ్యమైన లక్షణం ఇన్‌హెరిటెన్స్. ఇది ఇప్పటికే ఉన్న లేదా బేస్ తరగతుల నుండి derived క్లాస్ అని పిలువబడే కొత్త తరగతులను సృష్టించే ప్రక్రియ. ఇన్‌హెరిటెన్స్ ఒక తరగతి యొక్క అన్ని కోడ్‌లను ప్రైవేట్‌గా ప్రకటించినవి తప్ప మరొక క్లాస్‌కి ఇన్‌హెరిటెన్స్‌గా పొందటానికి అనుమతిస్తుంది. ఇన్‌హెరిటెన్స్ పొందవలసిన తరగతిని బేస్ క్లాస్ లేదా పేరెంట్ క్లాస్ అని పిలుస్తారు మరియు ఇతర క్లాస్‌ని ఇన్‌హెరిటెన్స్‌గా పొందిన క్లాస్‌ని derived క్లాస్ లేదా చైల్డ్ క్లాస్ అంటారు. derived క్లాస్ అనునది పవర్ ప్యాక్ (power packed) క్లాస్ ఎందుకంటే ఇది అదనపు లక్షణాలను మరియు పద్ధతులను జోడించగలదు మరియు దాని కార్యాచరణను మెరుగుపరుస్తుంది.

గమనిక

ఇన్‌హెరిటెన్స్ యొక్క ప్రధాన ప్రయోజనాలు:

- ఇది వాస్తవ ప్రపంచ సంబంధాలను బాగా సూచిస్తుంది
- ఇది కోడ్ యొక్క పునర్వినియోగాన్ని అందిస్తుంది.
- ఇది ట్రాన్సిటివిటికి మద్ధతిస్తుంది.

16.3 ఇన్‌హెరిటెన్స్ వివిధ రకాలు (Types of Inheritance)

కలవు. అవి ఒకే ఇన్‌హెరిటెన్స్, బహుళ ఇన్‌హెరిటెన్స్, బహుళస్థాయి ఇన్‌హెరిటెన్స్, హైబ్రిడ్ ఇన్‌హెడిటెన్స్ మరియు క్రమానుగత ఇన్‌హెరిటెన్స్.

1. ఏక ఇన్‌హెరిటెన్స్ (Single Inheritance)

బేస్ క్లాస్ నుండి సంక్రమించిన డిరైవుడ్ క్లాస్‌ను ఏక ఇన్‌హెరిటెన్స్ అంటారు.

2. మల్టీ ఇన్‌హెరిటెన్స్

మల్టిపుల్ బేస్ క్లాస్ నుండి సంక్రమించిన డిరైవుడ్ క్లాస్‌ను మల్టిపుల్ ఇన్‌హెరిటెన్స్ అంటారు.

11th Computer Science_Chapter-16 Telugu.indd 267 24-03-2023 13:58:39

3. హైరార్కికల్ ఇన్‌హెరిటెన్స్

ఒకే బేస్ క్లాస్ నుండి ఒకటి కంటే ఎక్కువ డిరైవుడ్ క్లాస్ సృష్టించినట్లు అయితే దీనిని హైరార్కికల్ ఇన్‌హెరిటెన్స్ అంటారు.

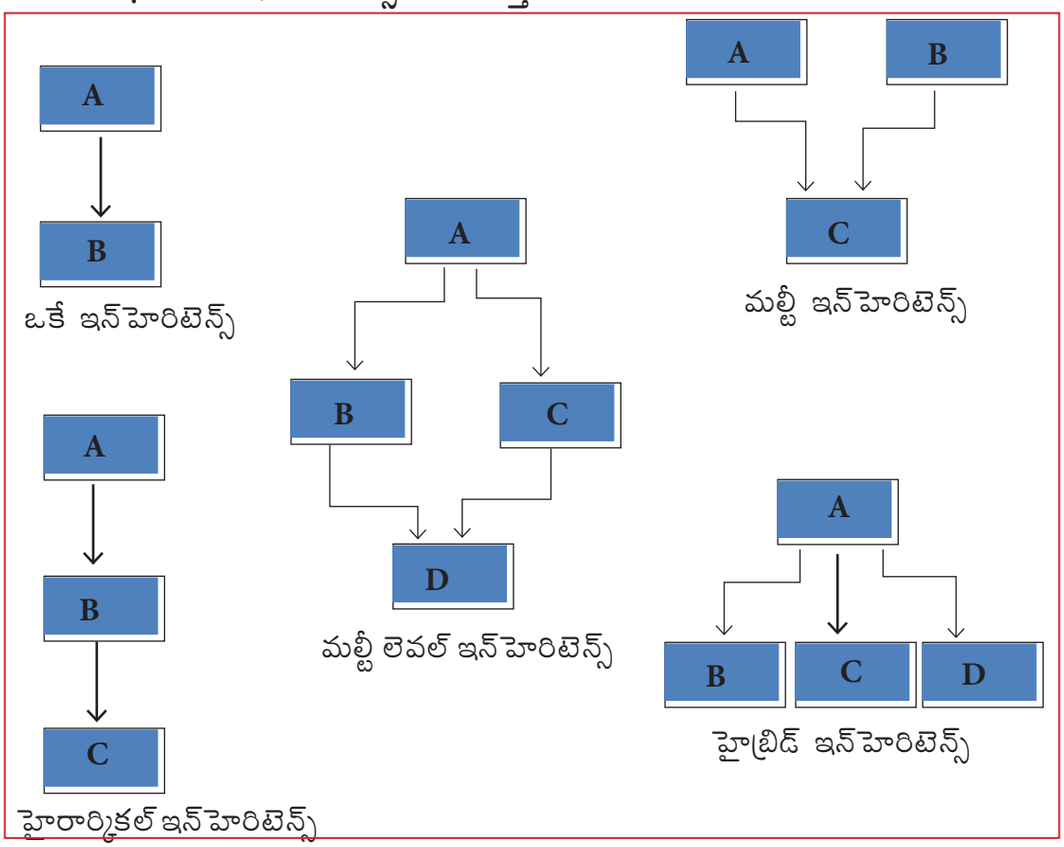
4. మల్టీ లెవల్ ఇన్‌హెరిటెన్స్

ఇన్‌హెరిటెన్స్ యొక్క పరివర్తన (ట్రాన్సిటివ్) స్వభావం ఈ ఇన్‌హెరిటెన్స్ రూపం ద్వారా ప్రతిబింబిస్తుంది. ఒక తరగతి నుండి మరొక తరగతిని సంగ్రహించుటను డిరైవుడ్ క్లాస్ అని, తరువాత దీనిని మల్టీ లెవల్ ఇన్‌హెరిటెన్స్ అని పరిగణిస్తారు.

5. హైబ్రిడ్ ఇన్‌హెరిటెన్స్

ఒకటి కంటే ఎక్కువ ఒకే రకమైన ఇన్‌హెరిటెన్స్‌ల కలయిక ఉన్నప్పుడు దానిని హైబ్రిడ్ ఇన్‌హెరిటెన్స్ అని అంటారు. ఇది ఒక వేల బహుళ స్థాయి మరియు బహుళార్ధక ఇన్‌హెరిటెన్స్ లేదా హైరార్కికల్ మరియు బహుళస్థాయి ఇన్‌హెరిటెన్స్ లేదా హైరార్కికల్ బహుళస్థాయి మరియు బహుళ ఇన్‌హెరిటెన్స్‌ల కలయిక అగును.

క్రింది పటం వివిధ రకాల ఇన్‌హెరిటెన్స్‌ని సూచిస్తుంది



16.4 డిరైవుడ్ క్లాస్ మరియు బేస్ క్లాస్

డిరైవుడ్ క్లాస్‌ని నిర్వచించేటప్పుడు అది derived తరగతి అని గుర్తించాలి. డిరైవుడ్ తరగతిని నిర్వచించడానికి ఈ క్రింది అంశాలను గమనించాలి.

i కీవర్డ్ క్లాస్‌ను ఉపయోగించాలి.

ii కీవర్డ్ క్లాస్ తర్వాత డిరైవుడ్ క్లాస్‌కి పేరు ఇవ్వవలెను.

iii ఇది సింగిల్ పంక్చుయేషన్ మార్క్ (:)

11th Computer Science_Chapter-16 Telugu.indd 268 24-03-2023 13:58:39

iv ప్రైవేట్ పబ్లిక్ లేదా ప్రొటెక్టెడ్ (protected) అనునవి డెరివేషన్ రకాలు (దృశ్యమానత మోడ్) విజిబిలిటి మోడ్ పేర్కొనబడకపోతే అప్రమేయంగా విజిబిలిటి మోడ్ ప్రైవేట్‌గా పరిగణించబడుతుంది.

v బేస్ క్లాస్ (పెరెంట్ క్లాస్) పేరు ఒక బేస్ క్లాస్ కంటే ఎక్కువ ఉన్నప్పుడు కామా ద్వారా వేరుచేయవచ్చును.

class derived_class_name :visibility_mode base_class_name

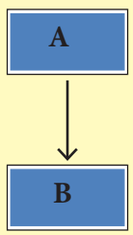
{ // members of derivedclass };

వివిధ రకాల ఇన్‌హెరిటెన్స్‌కి ఈ క్రింది కొన్ని ఉదాహరణలు

16.4.1 ఒకే ఇన్‌హెరిటెన్స్

ఉదాహరణ 16.1 ఏక ఇన్‌హెరిటెన్స్

```
# include <iostream>
using namespace std;
class student          //base class
{ private :
        char name[20];
        int rno;
   public:
        void  acceptname()
        { cout<<"\n Enter roll no and name .. ";    cin>>rno>>name;   }
        void  displayname()
        { cout<<"\n Roll no :-"<<rno;
          cout<<"\n Name :-"<<name<<endl;      } };
class exam : public student          //derived class with single base class
 {
     public:
       int  mark1,mark2,total;
       void acceptmark()
       {  cout<<"\n Enter mark1 and mark2.... ";    cin>>mark1>>mark2; }
void  displaymark()
{ cout<<"\n\t\t Marks Obtained ";
 cout<<"\n Subject1.. "<<mark1;
 cout<<"\n Subject2 .. "<<mark2;
 cout<<"\n Total .. "<<mark1+mark2;    } };
int main()
{  exam e1;
 e1.acceptname();     //calling base class function using derived class object
 e1.acceptmark();
 e1.displayname();              //calling base class function using derived class object
 e1.displaymark();
 return 0;
}
```



Single inheritance

11th Computer Science_Chapter-16 Telugu.indd 269 24-03-2023 13:58:39

Output

```
Enter roll no and name .. 1201 KANNAN
Enter lang,eng,phy,che,csc,mat marks.. 95 90 92 96 97 99
Roll no :-1201
Name :-KANNAN
        Marks Obtained
Language.. 95
English .. 90
Physics .. 92
Chemistry.. 96
Comp.sci.. 97
Maths ..  99
Total ..  569
```

పై ప్రోగ్రామ్‌లో derived క్లాస్ “exam” అనునది బేస్ క్లాస్ “student” లోని సభ్యులందరినీ ఇన్‌హెరిటెన్స్ పొందుతుంది. కానీ దీనియందు బేస్ క్లాస్ యొక్క ప్రైవేట్ కాని సభ్యులకు మాత్రమే యాక్సెస్ హక్కు ఉంది.

16.4.3 మల్టీ లెవల్ ఇన్‌హెరిటెన్స్

ఉదాహరణ 16.2 బహుళ స్థాయి ఇన్‌హెరిటెన్స్

```
# include <iostream>
using namespace std;
class student            //base class
{
private :
char name[20];
int rno;
public:
void acceptname()
{ cout<<"\n Enter roll no and name .. "; cin>>rno>>name;
}
void displayname()
{ cout<<"\n Roll no :-"<<rno;
cout<<"\n Name :-"<<name<<endl;  }};
 class exam : public student          //derived class with single base class
                {
                        total=mark1+mark2+mark3;
                        cout<<"\nTOTAL MARK SCORED   :  "<<total;
        }
 };
```


11th Computer Science_Chapter-16 Telugu.indd 270 24-03-2023 13:58:39

```
{
        public:
        int  mark1, mark2 ,mark3;
        void acceptmark()
        {       cout<<"\n Enter 3 subject marks.. ";
                cin>>mark1>>mark2>>mark3;     }
void displaymark(){
cout<<"\n\t\t Marks Obtained ";
cout<<"\n Subject1... "<<mark1;
cout<<"\n Subject2... "<<mark2;
cout<<"\n Subject3... "<<mark3;   }   };
class result : public exam
{
int total;
public:
void showresult()
{
total=mark1+mark2+mark3;
cout<<"\nTOTAL MARK SCORED : "<<total;
}
};
int main()
{
        result r1;
r1.acceptname();          //calling base class function using derived class object
r1.acceptmark();          //calling base class function which itself is a derived
                          // class  function using its derived class object
r1.displayname();         //calling base class function using derived class object
r1.displaymark();         /*calling base class function which itself is a derived
                          class  function using its derived class object*/
r1.showresult();          //calling the child class function
return 0;
}
```

Output:

```
Enter roll no and name .. 1201 Lohit Sarathi
 Enter 3 subject marks.. 98 100 100
 Roll no :-1201
 Name :-Lohith Sarathi
            Marks Obtained
Subject1 ...  98
Subject 2 ...  100
Subject 3... 100
TOTAL MARK SCORED    :  298
```


11th Computer Science_Chapter-16 Telugu.indd 271 24-03-2023 13:58:39

పై ప్రోగ్రామ్‌లో క్లాస్ "exam" నుండి క్లాస్ "result " ను తీసుకొనబడింది. ఇది స్వయంగా క్లాస్ student నుండి తీసుకొనబడింది.

గమనిక

మల్టీ లెవల్ ఇన్‌హెరిటెన్స్‌లో ఇన్‌హెరిటెన్స్ లెవల్‌ని ఏ స్థాయికైనా సంబంధాన్ని బట్టి విస్తరించవచ్చు. మల్టీ లెవల్ ఇన్‌హెరిటెన్స్ లేత తండ్రి మరియు పిల్లల మధ్య సంబంధాన్ని పోలి ఉండును.

ఏ డిక్లరేషన్ లేని తరగతికి బైట్ సైజు ఉంటుంది. class x{}; X occupies 1 బైట్‌ని ఆక్రమిస్తుంది.

పై ప్రోగ్రామ్ నందు derived క్లాస్ "result" క్లాస్ "detail" మరియు క్లాస్ "exam" యొక్క లక్షణాలను "student" నుండి తీసుకొనబడింది. కాబట్టి, ఈ ఇన్‌హెరిటెన్స్ మల్టీ లెవల్ మరియు మల్టీ ఇన్‌హెరిటెన్స్ అని పిలుస్తారు.

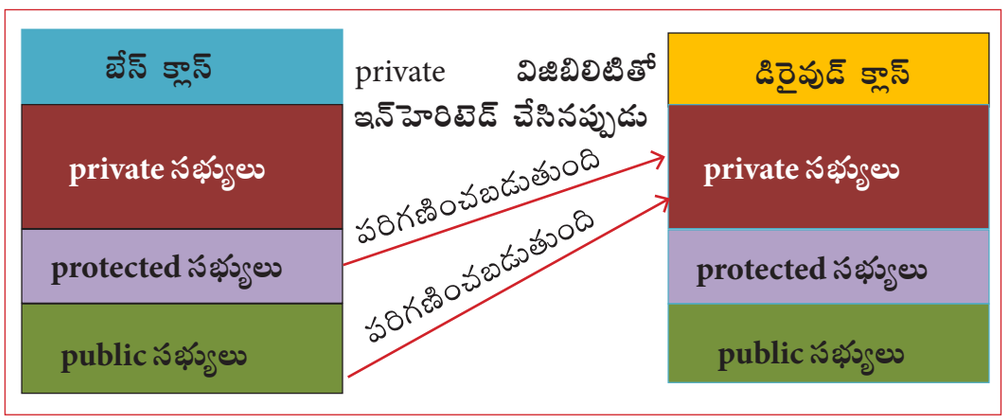
16.5 విజిబిలిటి మోడ్‌లు (VISIBILITY MODES)

ఇన్‌హెరిటెన్స్ యొక్క ముఖ్యమైన లక్షణం ఏమిటంటే బేస్ క్లాస్ అనునది సభ్యుడు. ఏ derived క్లాస్ నుండి సంగ్రహించబడినది అని విజిబిలిటి మోడ్ నుండి తెలుసుకొనుట.

డిరైవుడ్ క్లాస్ నుండి సౌలభ్యపరిచిన బేస్ క్లాస్‌ను విజిబిలిటి మోడ్ నియంత్రణ చేస్తుంది. private, protected మరియు public అనునవి మూడు విజిబిలిటి మోడ్‌లు private అనునది డిఫాల్ట్ విజిబిలిటి మోడ్ అగును. విజిబిలిటి మోడ్‌లు మరియు యాక్సెస్ స్పెసిఫైయర్‌లు సారూప్యంగా కనిపిస్తున్నప్పటికీ వాటి మధ్య ప్రధాన వ్యత్యాసం యాక్సెస్ ఫైయర్‌లు తరగతిలో ఉన్న సభ్యుల ప్రాప్యతను నియంత్రిస్తుంది. ఇక్కడ విజిబిలిటి మోడ్‌లు సభ్యుల ప్రాప్యతను నియంత్రిస్తాయి.

16.5.1 Private విజిబిలిటి మోడ్ (Private visibility mode)

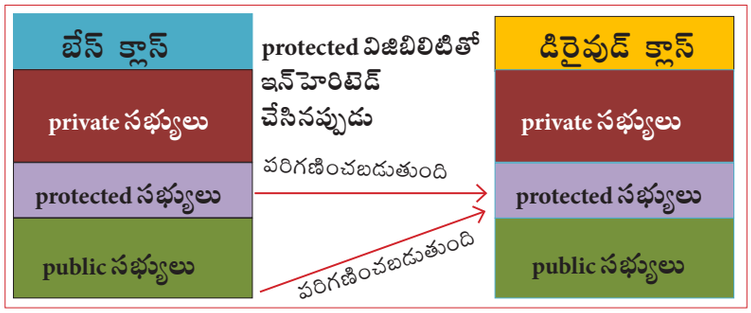
బేస్ క్లాస్ **private** విజిబిలిటి మోడ్‌తో ఇన్‌హెరిటెన్స్ పొందినప్పుడు బేస్ క్లాస్ యొక్క public మరియు protected సభ్యులు డిరైవుడ్ క్లాస్ యొక్క **private** సభ్యులవుతారు.



11th Computer Science_Chapter-16 Telugu.indd 272 24-03-2023 13:58:39

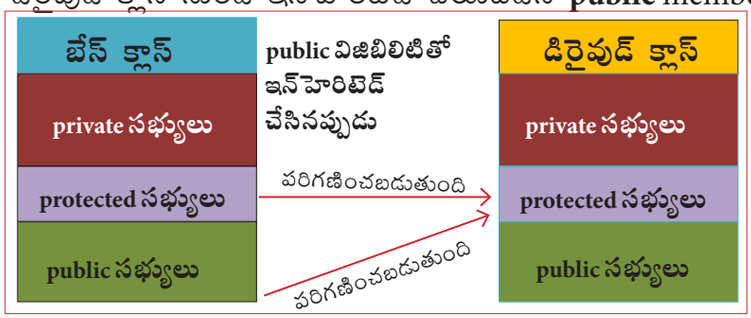
16.5.2 protected విజిబిలిటి మోడ్ (protected visibility mode)

protected విజిబిలిటి మోడ్‌తో బేస్ క్లాస్ ఇన్‌హెరిటెన్స్ పొందినపుడు బేస్ క్లాస్ యొక్క రక్షిత (protected) మరియు పబ్లిక్ (public) సభ్యులు డిరైవుడ్ క్లాస్ యొక్క protected సభ్యులవుతారు.



16.5.3 public విజిబిలిటి మోడ్ (public visibility mode)

public విజిబిలిటి మోడ్‌తో బేస్ క్లాస్‌ను ఇన్‌హెరిటెడ్ చేసినపుడు **protected** members యొక్క బేస్ క్లాస్‌లు అనునది డిరైవుడ్ క్లాస్ యొక్క **protected** members అగును మరియు **public** members యొక్క బేస్ క్లాస్‌లు డిరైవుడ్ క్లాస్ నుండి ఇన్‌హెరిటెడ్ చేయబడిన **public** members అగును.



గమనికలు

క్లాస్‌ని public, protected లేక private తో ఇన్‌హెరిటెన్స్ చేసినపుడు బేస్ క్లాస్ యొక్క సభ్యులు, private సభ్యులు యొక్క బేస్ క్లాసెస్ ఇన్‌హెరిటెన్స్ చేయబడవు అవి కనిపించును. అనగా డిరైవ్డ్ క్లాస్‌లో ఉండును మరియు యాక్సెస్ కావు.

ఉదాహరణ 16.3 విభిన్న విజిబిలిటి మోడ్‌ల యొక్క ప్రాముఖ్యతను వివరిస్తుంది.

```
//Implementation of Single Inheritance using public visibility mode
#include <iostream>
using namespace std;
class Shape
{   private:
        int count;
  protected:
        int width, height;
 public:
        void setWidth(int w)
        {       width = w;    }
void setHeight(int h)
{ height = h; } };
```


11th Computer Science_Chapter-16 Telugu.indd 273 24-03-2023 13:58:39

```
class Rectangle: public Shape
{
public:
int getArea()
{
return (width * height);
}
};
int main()
{
Rectangle Rect;
Rect.setWidth(5);
Rect.setHeight(7);
// Print the area of theobject.
cout<< "Total area: "<<Rect.getArea() <<endl;
return 0;
}
```

Output
Total area: 35

ఈ క్రింది పట్టిక ఇన్‌హెరిటెన్స్‌కు ముందు ప్రతి తరగతి లోపల నిర్వహించిన సభ్యులను కలిగి వుంటుంది.

MEMBERS of class	**విజిబిలిటి మోడ్**		
	Private	Protected	Public
Shape (బేస్ క్లాస్)	int count;	int width; int height;	void setWidth(int) void setHeight(int)
Rectangle (తరగతి నిర్దేశించిన members తో డిరైవుడ్ క్లాస్ మాత్రం)			int getArea();

క్రింది పట్టిక ఇన్‌హెరిటెన్స్ తరువాత నిర్దేశించిన సభ్యుల యొక్క వివరాలు.

MEMBERS of class	**విజిబిలిటి మోడ్ బేస్ క్లాస్ యొక్క లక్షణాలను పొందు public**		
	Private	Protected	Public
Shape (బేస్ క్లాస్)	int count;	int width; int height;	void setWidth(int) void setHeight(int)
Rectangle (Public విజిబిలిటితో బేస్ క్లాస్ యొక్క లక్షణాలను పొందిన base class)	బేస్ క్లాస్ యొక్క Private సభ్యులు derived class ద్వారా నేరుగా ప్రాప్యత చేయబడరు.	int width; int height;	int getArea(); void setWidth(int) void setHeight(int)

తరగతి **"rectangle" protected** విజిబిలిటితో డిరైవుడ్ అయినది అని అనుకుందాం. అప్పుడు క్లాస్ **"rectangle"** యొక్క లక్షణాలు క్రింది విధంగా మారుతాయి.

11th Computer Science_Chapter-16 Telugu.indd 274 24-03-2023 13:58:39

MEMBERS of class	విజిబిలిటి మోడ్ బేస్ క్లాస్ యొక్క లక్షణాలను పొందు –protected		
	Private	Protected	Public
Shape (బేస్ క్లాస్)	int count;	int width; int height;	void setWidth(int) void setHeight(int)
Rectangle (derived class protected విజిబిలిటితో బేస్‌క్లాస్ లక్షణాలను పొందుతుంది.)	బేస్ క్లాస్ యొక్క Private membersని derived class నేరుగా ప్రాప్యత చేయదు.	int width; int height; void setWidth(int) void setHeight(int)	int getArea();

ఒక వేళ తరగతి rectangle దాని బేస్ క్లాస్ shape నుండి private విజిబిలిటి మోడ్‌తో derived అయితే, తరగతి rectangle యొక్క లక్షణాలు క్రింది విధంగా మారుతుంది.

MEMBERS of class	విజిబిలిటి మోడ్ బేస్ క్లాస్ యొక్క లక్షణాలను పొందు –private		
	Private	Protected	Public
Shape (బేస్ క్లాస్)	int count;		void setWidth(int) void setHeight(int)
Rectangle (derived class బేస్ క్లాస్ లక్షణాలను పొందడ)	int width; int height; void setWidth(int) void setHeight(int)		int getArea();

మీరు ఇప్పటికే ఉన్న బేస్ క్లాస్ నుండి క్లాస్‌ను derived చేసినపుడు అది దాని విజిబిలిటి మోడ్ ఆధారంగా బేస్ క్లాస్ యొక్క లక్షణాలను ఇన్‌హెరిటెన్స్ పొందవచ్చు. కాబట్టి అవసరాన్ని బట్టి తగిన విజిబిలిటి మోడ్‌ను ఇవ్వాలి.

బేస్ క్లాస్ యొక్క లక్షణాలు derived class కి అందుబాటులో ఉండాలని మీరు కోరుకుంటున్నప్పుడు Private ఇన్‌హెరిటెన్స్‌ను ఉపయోగించాలి కాని derived క్లాస్ నుండి పొందిన తరగతికి కాదు. బేస్ క్లాస్ యొక్క లక్షణాలు కేవలం derived క్లాస్ సభ్యులకు మాత్రమే అందుబాటులో ఉండేలా బయటి ప్రపంచానికి అందుబాటులో లేనపుడు private ఇన్‌హెరిటెన్స్‌ని ఉపయోగించాలి.

బేస్ క్లాస్ యొక్క లక్షణాలు డిరైవుడ్ క్లాస్ సభ్యులకు మరియు బయటి ప్రపంచానికి అందుబాటులో ఉన్నప్పుడు Public ఇన్‌హెరిటెన్స్‌ను ఉపయోగించవచ్చు.

16.6 ఇన్‌హెరిటెన్స్ మరియు కన్స్‌ట్రక్టర్లు మరియు డిస్ట్రక్టర్లు (Inheritance and constructors and destructors)

Derived Class యొక్క Object సృష్టించబడినపుడు కంపైలర్ మొదటి బేస్ క్లాస్ కన్స్‌ట్రక్టర్‌ని పిలుస్తుంది మరియు ఆపైన Derived class యొక్క కన్స్‌ట్రక్టర్‌ను పిలుస్తుంది. ఎందుకనగా Derived Class బేస్ క్లాస్ సభ్యులపై నిర్మించబడింది. Derived Class యొక్క Object గడువు ముగిసినపుడు మొదట Derived Class డిస్ట్రక్టర్ తరువాత బేస్ క్లాస్ డిస్ట్రక్టర్ ప్రారంభించబడుతుంది.

11th Computer Science_Chapter-16 Telugu.indd 275 24-03-2023 13:58:39

ఉదాహరణ 16.4 కన్స్ట్రక్టర్లు మరియు డిస్ట్రక్లర్లు క్రమం

```
#include<iostream>
using namespace std;
class base
{
public:
base()
{ cout<<"\nConstructor of base class..."; }
~base()
{ cout<<"\nDestructor of base class.... "; }
};
class derived:public base
{
public :
derived()
{ cout << "\nConstructor of derived ..."; }
~derived()
{ cout << "\nDestructor of derived ..."; }};
class derived1 :public derived
{
public :
derived1()
{ cout << "\nConstructor of derived1 ...";}
~derived1()
{ cout << "\nDestructor of derived1 ...";}
};
int main()
{
derived1 x;
return 0;
}
```

Output:

```
Constructor of base class...
Constructor of derived ...
Constructor of derived1 ...
Destructor of derived1 ...
Destructor of derived ...
Destructor of base class....
```

గమనిక

- కన్స్ట్రక్టర్లు ఇన్‌హెరిటెన్స్‌గా వచ్చిన తరగతి క్రమంలో అమలు చేయబడతాయి. అనగా బేస్ కన్స్ట్రక్టర్లు నుండి డిరైవ్డ్ వరకు డిస్ట్రక్టర్లు రివర్స్ ఆర్డర్‌లో అమలు చేయబడుతుంది.
- Derived class object యొక్క పరిమాణము = అన్ని బేస్ క్లాస్‌ల డేటా members పరిమాణము + derived class లోని అన్ని డేటా members ల పరిమాణం.

16.7 Derived class లో బేస్ క్లాస్ ఫంక్షన్లను భర్తీ చేయడం మరియు Shadowing (Overriding / Shadowing Base class functions in derived class)

ఇన్‌హెరిటెన్స్ విషయంలో బేస్ క్లాస్ మరియు Derived class సభ్యుల ఫంక్షన్ ఒకే పేరును కలిగి ఉన్న సంధర్భాలు ఉన్నాయి. c ను ఆబ్జెక్ట్ ఓవర్‌లోడ్ మెంబర్ ఫంక్షన్ అని పిలిస్తే, కంపైలర్

11th Computer Science_Chapter-16 Telugu.indd 276 24-03-2023 13:58:39

గందరగోళానికి దారితీస్తుంది. Derived చేసిన మెంబర్ ఫంక్షన్ బేస్ క్లాస్ మెంబర్ ఫంక్షన్ కంటే ఎక్కువ ప్రాధాన్యతనిస్తుంది. బేస్ క్లాస్ సభ్యుల ఫంక్షన్ Derived class సభ్యుల ఫంక్షన్ వలె అదే పేరును కలిగి ఉండిన ఇది బేస్‌క్లాస్ యొక్క సభ్యుల ఫంక్షన్‌కు shadows ని ఇస్తుంది. స్కోప్ రిజల్యూషన్ (::) ఆపరేటర్ ఈ సమస్యను పరిష్కరిస్తుంది.

ఉదాహరణ 16.5 Derived class లో స్కోప్ రిజల్యూషన్ ఆపరేటర్ యొక్క ఉపయోగం

```
#include<iostream>
#include<string>
using namespace std;
class Employee
{
   private:
      char name[50];
      int code;
   public:
      void getdata();
      void display();
};
class staff: public Employee
{
   private:
      int ex;
   public:
         void getdata();
      void display();
};
void Employee::display()
{
         cout<<"Name:"<<name<<endl;
         cout<<"Code:"<<code<<endl;
}
void staff::display()
{        Employee :: display();//overriding
         cout<<"Experience:"<<ex<<" Years"<<endl;
}
int main()
{
         staff s;
         cout<<"Enter data"<<endl;
         s.getdata();
         cout<<endl<<endl<<"\tDisplay Data"<<endl;
         s.display();
         return 0;
}
```

Output

```
Enter data
Name: SUGANYA
Code: 1201
Experience: 30
      Display Data
Name: SUGANYA
Code:1201
Experience:30 Years
```


11th Computer Science_Chapter-16 Telugu.indd 277 24-03-2023 13:58:39

పై ప్రోగ్రామ్‌లో getdata() మరియు display() రెండునూ బేస్ మరియు derived class లో నిర్వచింపబడును. కాబట్టి derived class "staff" "Employee" తరగతి యొక్క లక్షణాలను వారసత్వంగా పొందినపుడు అది రెండు getdata() మరియు display() ని కలిగి వుంటుంది. derived getdata() మరియు display() ని వేరుచేయడానికి నిర్వచింపబడిన getdata() మరియు display() :: (స్కోప్ రిజల్యూషన్) ఆపరేటర్‌ను బేస్ క్లాస్ పేరుతో పాటు బేస్ క్లాస్ సభ్యులకు ఇవ్వబడుతుంది.

గమనిక

Derived class మెంబర్ ఫంక్షన్ దాని బేస్ క్లాస్ మెంబర్ ఫంక్షన్ యొక్క పేరును కలిగి ఉన్నప్పుడు Derived class మెంబర్ ఫంక్షన్ బేస్ క్లాస్ యొక్క ఇన్‌హెరిటెన్స్‌గా వచ్చిన ఫంక్షన్‌ను Shadow చేస్తుంది / దాచిపెడుతుంది. ఈ పరిస్థితిని ఫంక్షన్ ఓవరైడింగ్ అంటారు. దీనిని :: అనుసరించి బేస్ క్లాస్ పేరు మరియు సభ్యుల ఫంక్షన్ పేరు ఇవ్వడం ద్వారా పరిష్కరించవచ్చును.

గుర్తుంచుకోవలసిన అంశాలు:

- ఇప్పటికే ఉన్న తరగతి నుండి క్రొత్త తరగతిని పొందే విధానాన్ని ఇన్‌హెరిటెన్స్ అంటారు.
- ఇన్‌హెరిటెన్స్ యొక్క ప్రధాన ప్రయోజనం ఇది కోడ్ యొక్క పునర్వినియోగతకు మద్ధతు ఇస్తుంది.
- derived class బేస్ క్లాస్ యొక్క అన్ని లక్షణాలను ఇన్‌హెరిటెన్స్‌గా పొందుతుంది. ఇవి (పవర్ ప్యాక్డ్ క్లాస్) power packed class ఎందుకంటే ఇవి అదనపు లక్షణాలను మరియు పద్ధతులను జోడించగలరు మరియు దాని కార్యాచరణను మెరుగుపరుస్తుంది.
- సింగిల్ ఇన్‌హెరిటెన్స్ బహుళ ఇన్‌హెరిటెన్స్ బహుళ స్థాయి ఇన్‌హెరిటెన్స్ మొదలగునవి ఇన్‌హెరిటెన్స్ రకాలు.
- derived class ఒక బేస్ క్లాస్ నుండి మాత్రమే ఇన్‌హెరిటెన్స్‌గా పొందినపుడు దీనిని సింగిల్ ఇన్‌హెరిటెన్స్ అంటారు.
- derived class బహుళ బేస్ క్లాస్‌ల నుండి ఇన్‌హెరిటెన్స్‌గా పొందినపుడు దానిని బహుళ ఇన్‌హెరిటెన్స్ అంటారు.
- ఒక derived class నుండి ఒక తరగతి ఉద్భవించినపుడు దీనిని బహుళ స్థాయి ఇన్‌హెరిటెన్స్ అని పిలుస్తారు. ఇన్‌హెరిటెన్స్ యొక్క పరివర్తన స్వభావం ఈ ఇన్‌హెరిటెన్స్ రూపం ద్వారా ప్రతిబింబిస్తుంది.
- ఒకే బేస్ క్లాస్ నుండి ఒకటి కంటే ఎక్కువ derived class సృష్టించబడినపుడు దానిని క్రమానుగత ఇన్‌హెరిటెన్స్ అందురు.
- ఒకటి కంటే ఎక్కువ రకాల ఇన్‌హెరిటెన్స్ కలయిక ఉన్నప్పుడు దానిని హైబ్రిడ్ ఇన్‌హెరిటెన్స్ అంటారు.
- బహుళ ఇన్‌హెరిటెన్స్‌లో బేస్ క్లాస్‌లు అవి c ప్రకటనలో కనిపించే క్రమంలో నిర్మించబడతాయి.
- ఒక సబ్ క్లాస్ public, private లేక protected గా తనంతటతానుగా derived చేస్తారు.
- private సభ్యుని యొక్క తరగతిని ఇన్‌హెరిటెన్స్ చేయలేము.
- public గా derived class లో బేస్ క్లాస్‌లోని public సభ్యులు public గా ఉంటారు మరియు బేస్ క్లాస్ యొక్క protected సభ్యులు derived class లో protect చేయబడుతారు.
- private గా derived లో బేస్ క్లాస్ యొక్క public మరియు protected సభ్యులు derived class private గా మారుతారు.
- protected మోడ్‌లో తరగతి c అయినపుడు బేస్ క్లాస్ యొక్క public మరియు protected సభ్యులు derived class లోను protect చేయబడుతారు.
- బేస్ క్లాస్ యొక్క కన్స్‌ట్రక్టర్లు మరియు డిస్ట్రక్టర్లు ఇన్‌హెరిటెన్స్‌గా పొందబడవు, కాని derived class కోసం ఒక object ను సృష్టించేటపుడు బేస్ క్లాస్ యొక్క కన్స్‌ట్రక్టర్లు స్వయంచాలకంగా ప్రారంభించబడతాయి.
- డిస్ట్రక్టర్లును రివర్స్ ఆర్డర్‌లో ఇన్‌వోక్ చేయబడతాయి derived class లో మొదటగా derived class లు ఆ తరువాత బేస్ క్లాస్‌లు డిస్ట్రక్టర్లుగా ఇన్‌వోక్ చేయబడతాయి.
- derived class ఆబ్జెక్ట్ యొక్క పరిమాణం = మొత్తం బేస్ క్లాస్ డేటా సభ్యుల పరిమాణం + derived class మొత్తం డేటా సభ్యుల పరిమాణం
- స్కోప్ రిజల్యూషన్ ఆపరేటర్ (::) ను ఉపయోగించి సభ్యుల ఓవరైడింగ్ పరిష్కరించబడుతుంది
- ఈ పాయింటర్‌ను ఉపయోగించి ప్రస్తుత ఆబ్జెక్ట్ సభ్యులను రెఫర్ చేయవచ్చును.

11th Computer Science_Chapter-16 Telugu.indd 278 24-03-2023 13:58:40

మూల్యాంకనం

593PT

భాగము - అ

సరైన సమాధానాన్ని ఎన్నుకొనుము

1. క్రింది వాటిలో ఏది ఇప్పటికే ఉన్న తరగతి నుండి క్రొత్త తరగతిని సృష్టించే ప్రక్రియ?
   (అ) ఫాలిమార్ఫిజం (ఆ) ఇన్‌హెరిటెన్స్
   (ఇ) ఎన్కాప్యులేషన్ (ఈ) సూపర్ క్లాస్
2. క్రింది వాటిలో ఏది బేస్ క్లాస్ school నుండి a class student ను పొందును
   (అ) school: student
   (ఆ) class student : public school
   (ఇ) student : public school
   (ఈ) class school : public student
3. పరివర్తన స్వభావాన్ని ప్రతిబింబించే ఇన్‌హెరిటెన్స్ రకం
   (అ) ఒకే ఇన్‌హెరిటెన్స్
   (ఆ)బహుళ ఇన్‌హెరిటెన్స్
   (ఇ) ఒకే ఇన్‌హెరిటెన్స్
   (ఈ)హైబ్రిడ్ ఇన్‌హెరిటెన్స్
4. బేస్ క్లాస్ యొక్క లక్షణాలు derived class కి అందుబాటులో ఉండాలని మీరు కోరుకున్నపుడు కాని derived class నుండి derive చేయబడిన class కాకుండా ఏ విజిబిలిటి మోడన్ ఉపయోగించాలి?
   (అ) Private (ఆ) Public
   (ఇ) Protected (ఈ) All of these
5. ఇన్‌హెరిటెన్స్ దేనినుంచి క్రొత్త class ని సృష్టించే ప్రక్రియ?
   (అ) బేస్ క్లాస్ (ఆ) నైరూప్య
   (ఇ) derived class (ఈ) ఫంక్షన్
6. ఒక derived class నుండి class ఉద్భవించిన దానిని ఇలా సూచిస్తారు.
   (అ) బహుళ ఇన్‌హెరిటెన్స్
   (ఆ) బహుళస్థాయి ఇన్‌హెరిటెన్స్
   (ఇ) ఒకే ఇన్‌హెరిటెన్స్
   (ఈ) రెట్టింపు ఇన్‌హెరిటెన్స్
7. క్రింది వాటిలో ఏది ఇన్‌హెరిటెన్స్ క్రమంలో అమలు చేయబడుతుంది?
   (అ) డిస్ట్రక్టర్ (ఆ) మెంబర్ ఫంక్షన్
   (ఇ) కన్స్ట్రక్టర్ (ఈ) ఆబ్జెక్ట్
8. ఇన్‌హెరిటెన్స్‌కు సంబంధించి క్రింది వాటిలో ఏది నిజం?
   (అ) బేస్ క్లాస్ Private సభ్యులు derived class నందు private ఇన్‌హెరిటెన్స్‌ను పొందుతారు
   (ఆ) బేస్ క్లాస్ Private సభ్యులు derived class నందు private యాక్సెసిబిలిటితో ఇన్‌హెరిటెన్స్‌ను పొందుతారు
   (ఇ) బేస్ క్లాస్ Public సభ్యులు ఇన్‌హెరిటెన్స్‌ను పొందుతారు. కాని derived class కనిపించదు
   (ఈ) బేస్ క్లాస్ Protected సభ్యులు ఇన్‌హెరిటెన్స్‌ను పొందుతారు కాని బయటి class కనిపించరు
9. తరగతి డిక్లరేషన్ ఆధారంగా క్రింది ప్రశ్నలకు సమాధానములిమ్ము (9.1 నుండి 9.4 వరకు)

```
class vehicle
{ int wheels;
public:
void input_data(float,float);
void output_data();
protected:
int passenger;
 };
class heavy_vehicle : protected vehicle {
int diesel_petrol;
protected:
        int load;
public:
void read_data(float,float)
void write_data(); };
class bus: private heavy_vehicle {
char Ticket[20];
public:
void fetch_data(char);
void display_data(); };
```


11th Computer Science_Chapter-16 Telugu.indd 279 24-03-2023 13:58:40

9.1 heavy_vehicle తరగతి యొక్క బేస్ క్లాస్‌ను గుర్తించుము.

(అ) Bus (ఆ) heavy_vehicle

(ఇ) vehicle (ఈ) (అ) మరియు (ఇ) రెండునూ

9.2 displaydata() ఫంక్షన్ నుండి యాక్సెస్ చేయగల డేటా సభ్యులు

(అ) passenger (ఆ) load

(ఇ) Ticket (ఈ) ఇవన్నియూ

9.3 తరగతి bus ఆబ్జెక్టుల ద్వారా యాక్సెస్ చేయగల మెంబర్ ఫంక్షన్

(అ) input_data(), output_data()

(ఆ) read_data(), write_data()

(ఇ) fetch_data(), display_data()

(ఈ) ఇవన్నియూ

9.4 తరగతి Bus ద్వారా public ఇన్‌హెరిటెన్స్‌గా పొందిన మెంబర్ ఫంక్షన్

(అ) input_data(), output_data()

(ఆ) read_data(), write_data()

(ఇ) fetch_data(), display_data()

(ఈ) ఇవేవీ కావు

విభాగము - ఆ

క్లుప్తముగా సమాధానములను తెలుపుము

1. ఇన్‌హెరిటెన్స్ అనగానేమి?
2. బేస్ క్లాస్ అనగానేమి?
3. derived class ని ఎందువలన పవర్ ప్యాక్డ్ క్లాస్ అంటారు?
4. రెండింటిలో అనేక బేస్‌క్లాస్ ఉన్నప్పటికీ బహుళ స్థాయి మరియు బహుళ ఇన్‌హెరిటెన్స్ దేని పరంగా భిన్నంగా ఉంటాయి?
5. public మరియు private విజిబిలిటి మోడ్ మధ్య భేదాలు ఏమిటి?

విభాగము - ఇ

సంక్షిప్తంగా సమాధానములను తెలుపుము

1. క్రొత్త తరగతిని పొందేటప్పుడు గమనించవలసిన అంశాలు ఏమిటి?
2. private విజిబిలిటి మోడ్‌లో ఉన్న సభ్యులకు మరియు public విజిబిలిటి మోడ్‌లో ఉన్న సభ్యులకు మధ్య భేదములేవి?
3. కోడ్ యొక్క పునర్వినియోగం కోసం ఫాలిమార్ఫిజం మరియు ఇన్‌హెరిటెన్స్ మధ్య భేదములేమి?
4. ఓవర్ రైడింగ్ అంటే అర్థం ఏమిటి?
5. ఇన్‌హెరిటెన్స్‌లో కన్స్‌ట్రక్టర్లు మరియు డిస్ట్రక్టర్లు అమలు గురించి కొన్ని వాస్తవాలు వ్రాయండి

భాగము - ఈ

విపుల సమాధానాలు వ్రాయుము

1. వివిధ రకాల ఇన్‌హెరిటెన్స్ గురించి వివరించుము.
2. చిత్ర ప్రాతినిధ్యం ద్వారా విభిన్న విజిబిలిటి మోడ్‌లను వివరించండి.
3. క్రింది **c++** కోడ్‌ను పరిశీలించి ఇవ్వబడిన ప్రశ్నలకు సమాధానములను తెలపండి.

```
class Personal
{
int Class,Rno;
char Section;
protected:
char Name[20];
public:
personal();
void pentry();
void Pdisplay(); };
class Marks:private Personal
{ float M{5};
protected:
char Grade[5];
public:
Marks();
void Mentry();
void Mdisplay(); };
class Result:public Marks
{
float Total,Agg;
public:
char FinalGrade, Commence[20];
Result();
void Rcalculate();
void Rdisplay();
};
```


11th Computer Science_Chapter-16 Telugu.indd 280 24-03-2023 13:58:40

3.1 పైన ప్రోగ్రామ్‌లో ఏ రకమైన ఇన్‌హెరిటెన్స్ చూపబడినది?

3.2 బేస్ క్లాస్ యొక్క విజిబిలిటి మోడ్‌ను పేర్కొనండి.

3.3 తరగతి యొక్క ఆబ్జెక్టు Result సృష్టించబడినపుడు కన్స్ట్రక్టర్ డిస్ట్రక్టర్ ఇన్వోకేషన్ యొక్క క్రమాన్ని వ్రాయండి.

3.4 బేస్ క్లాస్‌లు మరియు derived class పేర్లును తెలపండి.

3.5 క్రింది తరగతి ఆబ్జెక్టులు ఆక్రమించిన బైట్ల సంఖ్యను తెలుపుము.

(అ) Personal (ఆ) Marks

(ఇ) Result

3.6 తరగతి ఆబ్జెక్ట్ Result నుండి యాక్సెస్ చేయగల సభ్యులందరి పేర్లను వ్రాయండి.

3.7 తరగతి ఆబ్జెక్ట్ Result నుండి యాక్సెస్ చేయగల మెంబర్ ఫంక్షన్ల పేర్లను వ్రాయండి.

3.8 తరగతి Result యొక్క సభ్యుల ఫంక్షన్ల నుండి యాక్సెస్ చేయగల అన్ని సభ్యుల పేర్లను వ్రాయండి.

4. క్రింది ప్రోగ్రామ్‌కు అవుట్‌పుట్‌ను తెలుపండి

```
#include<iostream>
using namespace std;
class A
{       protected:
      int x;
      public:
      void show()
      {cout<<"x = "<<x<<endl;}
      A()
      { cout<<endl<<" I am class A "<<endl;}
      ~A()
      { cout<<endl<<" Bye ";} };
class B : public A
{protected:
 int y;
 public:
 B(int x1, int y1)
 { x = x1;
   y = y1;       }
B()
{   cout<<endl<<" I am class B "<<endl; }
~B()
{   cout<<endl<<" Bye "; }
void show()
{ cout<<"x = "<<x<<endl;
  cout<<"y = "<<y<<endl;    } };
int main()
{A objA;
B objB(30, 20);
objB.show();
return 0;  }
```

5. క్రింది ప్రోగ్రామ్ డీబగ్ చేయండి

```
%include(iostream.h)
#include<conio.h>
class A()
{ public;
int a1,a2:a3;
void getdata[]
{       a1=15;  a2=13; a3=13; }         }
class B:: public A()
{       PUBLIC
        voidfunc()
        { int b1:b2:b3;
          A::getdata[];
          b1=a1;
          b2=a2;
          a3=a3;
          cout<<b1<<'\t'<<b2<<'t\'<<b3; }
void main()
{       B der;
        der1:func(); }
```

కేస్ స్టడీ

క్లాస్ స్టాక్ కోసం క్లాస్‌ను రాయండి. ప్రతి స్టాక్‌లో నికర ధరను కలిగియున్న డేటా మెంబర్ మరియు ఈ ధరను నిర్ణయించే కన్స్ట్రక్టర్ ఉంటుంది. ప్రతి స్టాక్‌లో get_Price() అనే పద్ధతి ఉంది ఇది స్థూల ధరను లెక్కించి తిరిగి ఇస్తుంది. (స్థూల ధరలో 21% వరకు వ్యాట్ కలదు)

సూచన పుస్తకాలు:

Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.

11th Computer Science_Chapter-16 Telugu.indd 281 24-03-2023 13:58:40

కంప్యూటర్ ఎథిక్స్ మరియు సైబర్ భద్రత (Computer Ethics And Cyber Security)

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును అభ్యసించిన తరువాత విద్యార్థులు నేర్చుకొను అంశాలు.

- సైబర్‌నేరాలు గురించి తెలుసుకోవడం.
- సైబర్ ప్రపంచంలోని నీతిశాస్త్రం యొక్క అవసరాలకు తగిన మార్గదర్శకాలను తెలుసుకోవడం.
- సైబర్ నేరాలకు సంబంధించిన సమస్యలను అర్థం చేసుకోవడం.
- ఫైర్ వాల్స్ మరియు ప్రాక్సీ సర్వర్లు యొక్క కార్యచరణ గురించి తెలుసుకోవడం.
- ఎన్‌క్రిప్షన్ మరియు డిక్రిప్షన్ గురించి నేర్చుకొనుట.
- ఐటి చట్టంపై పరిజ్ఞానం పొందడం.



17.1 పరిచయం (INTRODUCTION)

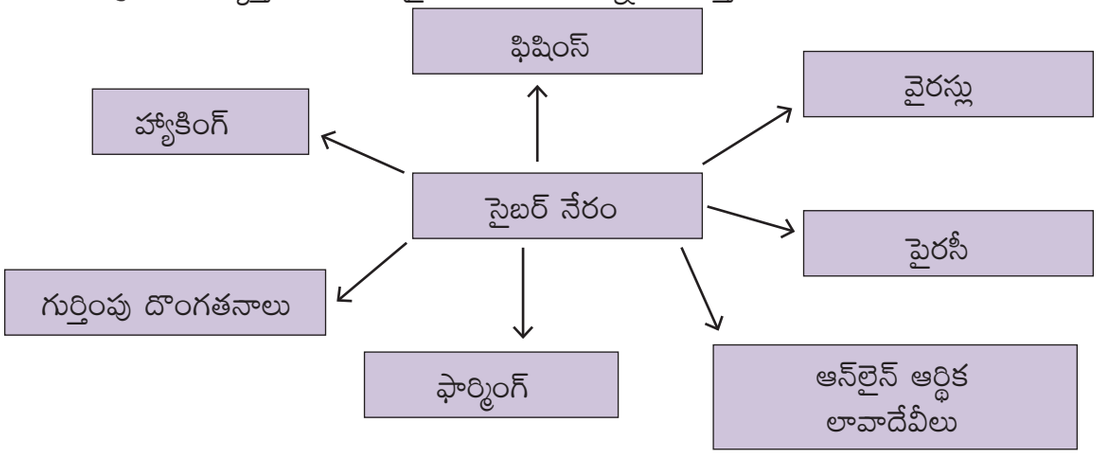
అంతర్జాలం (Internet) అనేది కమ్యూనికేషన్ మీడియా. ఇది అందరికీ సులభంగా అందుబాటులో ఉంటుంది సమాచార సాంకేతిక పరిజ్ఞానం కంప్యూటర్లు, చరవాణిలు, అంతర్జాలముల ద్వారా విస్తృతంగా వ్యాపించింది. సమాచార సాంకేతిక పరిజ్ఞానాన్ని దుర్వినియోగం చేయడానికి చాలా అవకాశాలు ఉన్నాయి.

కంప్యూటర్ వ్యవస్థలు సాధారణంగా కొంత హాని కలిగిస్తాయి. వ్యక్తులు, వ్యాపారాలలో రోజువారీ జీవితంలో అవి ముఖ్యమైన పాత్ర పోషిస్తాయి. విలువైన డేటా తప్పుడు చేతుల్లోకి రాకుండా ప్రత్యేక శ్రద్ధ తీసుకోవాలి. కాబట్టి సమాచారాన్ని రక్షించాల్సిన అవసరం ఉంది.

సైబర్‌నేరం అనేది కంప్యూటర్ మరియు నెట్‌వర్క్‌తో కూడిన నేరం. ఇది సమాజానికి పెరుగుతున్న ముప్పుగా మారుతోంది. ఇది మరియు నేరస్థులు లేదా అంతర్జాలం యొక్క విస్తృతమైన వినియోగాన్ని దోపిడీ చేస్తున్న వ్యక్తుల బాధ్యతా రహితచర్య వల్ల సంభవిస్తుంది. సమాచార సాంకేతిక పరిజ్ఞానం యొక్క నైతిక ఉపయోగానికి ఇది పెద్ద సవాళ్లను అందిస్తుంది. సైబర్ నేరం చాలా వ్యాపార వ్యవస్థల యొక్క సమగ్రత, భద్రత మరియు మనుగడకు కూడా ముప్పు కలిగిస్తుంది.

11th Computer Science_Chapter-17 Telugu.indd 282 24-03-2023 16:51:35

పటము 17.1 ప్రపంచ వ్యాప్తంగా జరిగే సైబర్ నేరాల రకాన్ని అందిస్తుంది.



పటము 17.1 సైబర్ నేరముల రకములు

ఎథిక్స్ (నీతి) (ETHICS)

ఎథిక్స్ అంటే "ఏది తప్పు మరియు ఏది సరైనది". ఇది కంప్యూటర్లను ఉపయోగించే వ్యక్తుల ప్రవర్తనను శాసించే నైతిక సూత్రాల సమితి. ఒక వ్యక్తి సరైన ప్రవర్తనను అనుసరించడానికి జ్ఞానాన్ని పొందుతాడు. నీతిని (Ethics) నైతికత అని కూడా పిలుస్తారు. సమాజంలో సరి మరియు తప్పు యొక్క సాధారణంగా అంగీకరించబడిన ప్రమాణాలను నైతికాలు సూచిస్తాయి. సైబర్ ప్రపంచంలో కొన్ని ప్రమాణాలు ఉన్నాయి.

- పైరేటెడ్ సాఫ్ట్‌వేర్‌ను ఉపయోగించవద్దు
- అనధికార వినియోగదారు ఖాతాలను ఉపయోగించవద్దు
- ఇతరుల పాస్‌వర్డ్‌లను దొంగిలించవద్దు
- హాక్ చేయవద్దు

కంప్యూటర్ ఎథిక్స్‌లోని ప్రధాన సమస్యలు గోప్యత, కాపీరైటెడ్ చేసిన కంటెంట్ ప్రచురణ, డిజిటల్ కంటెంట్ యొక్క అనధికార పంపిణీ మరియు వెబ్‌సైట్లు, సాఫ్ట్‌వేర్ మరియు సంబంధిత సేవలలో వినియోగదారుల ప్రవేశం వంటి అంతర్జాలం వాడకం వల్ల ఉత్పన్నమయ్యే దృశ్యాలపై ఆధారపడి ఉంటాయి.

కంప్యూటర్ ఎథిక్స్ (COMPUTER ETHICS)

ఇంటర్‌నెట్ సహాయంతో ప్రపంచం ఇప్పుడు గ్లోబల్ విలేజ్‌గా మారింది. వ్యక్తులతో పాటు వివిధ సంస్థలు మరియు వ్యాపారాలకు అంతర్జాలం ఒక వరం అని నిరూపించబడింది. వ్యాపారాలలో ఇ – వాణిజ్యం బాగా ప్రాచుర్యం పొందింది. ఎందుకంటే ఇది ఇతర మార్గాల కంటే విస్తృతమైన ఖాతాదార్లను చేరుకోవడానికి సహాయపడుతుంది.

కంప్యూటర్ ఎథిక్స్ ఏదైనా వ్యక్తి, సంస్థ లేదా సంస్థ యొక్క నైతిక విలువలు మరియు నమ్మకాలను దెబ్బదీయకుండా లేదా ఉల్లంఘించకుండా కంప్యూటర్ సాంకేతిక పరిజ్ఞానాన్ని మరియు దాని సంబంధిత విభాగాలను వినియోగించే, నియంత్రించే విధానాలు, విలువలు మరియు అభ్యాసాలతో వ్యవహరిస్తుంది.

ఎథిక్స్ మార్గదర్శకాలు (GUIDELINES OF ETHICS)

సాధారణంగా కంప్యూటర్ వినియోగదారులు ఈ క్రింది మార్గదర్శకాలను గమనించాలి.

1. **నిజాయితీ:** అంతర్జాలమును ఉపయోగిస్తున్నప్పుడు వినియోగదారులు నిజాయితీగా ఉండాలి.

2. **గోప్యత :** వినియోగదారులు అనధికార వ్యక్తులతో ఎటువంటి (Confidentiality) ముఖ్యమైన సమాచారాన్ని పంచుకోకూడదు.

3. **గౌరవం :** ప్రతి వినియోగదారు ఇతర వినియోగదారుల గోప్యతను గౌరవించాలి.

11th Computer Science_Chapter-17 Telugu.indd 283 24-03-2023 16:51:35

4. **వృత్తి నైపుణ్యం :** ప్రతి వినియోగదారు వృత్తిపరమైన ప్రవర్తనను కొనసాగించాలి.

5. **చట్టాన్ని అనుసరించుట :** కంప్యూటర్ వినియోగంలో వినియోగదారులు సైబర్ చట్టాన్ని ఖచ్చితంగా పాటింటాలి.

6. **బాధ్యత :** ప్రతి వినియోగదారు వారి చర్యలకు యాజమాన్యం మరియు బాధ్యత తీసుకోవాలి.



ఎథిక్స్ అనేది సమాజంలో ఒక వ్యక్తి యొక్క ప్రవర్తనను నియంత్రించే నైతిక సూత్రాల సమితి. కంప్యూటర్ ఎథిక్స్ అనేది వినియోగదారులచే కంప్యూటర్ల వాడకాన్ని నియంత్రించే నైతిక సూత్రాల సమితి.

17.2 నైతిక సమస్యలు (ETHICAL ISSUES)

నైతిక సమస్య అనేది ఒక సమస్య. ఇది ఒక వ్యక్తి లేదా సంస్థ ప్రత్యామ్నాయాల మధ్య ఎంచుకోవాలసిన అవసరం ఉంది, అది సరైన (నైతిక) లేదా తప్పు (అనైతిక) గా అంచనా వేయాలి. సమాజంలో సానుకూల ప్రభావం చూపడానికి ఈ సమస్యలను గుర్తించాలి మరియు పరిష్కరించాలి.

కొన్ని సాధారణ నైతిక సమస్యలు క్రింద ఇవ్వబడ్డాయి

- సైబర్ నేరం
- సాఫ్ట్‌వేర్ పైరసీ
- అనధికారిక ప్రవేశము
- హ్యాకింగ్
- మోసానికి కంప్యూటర్ల వాడకం
- వైరస్ల రూపంలో విధ్వంసం
- కంప్యూటర్లను ఉపయోగించి తప్పుడు వాదనలు చేయడం

సైబర్ నేరం (CYBER CRIME)

సైబర్ క్రైమ్ ఒక మేధో, వైట్ కాలర్ నేరం. ఇటువంటి నేరాలకు పాల్పడేవారు సాధారణంగా కంప్యూటర్ వ్యవస్థను తెలివిగా నిర్వహిస్తారు.

ఉదాహరణ – అంతర్జాలము (ఇంటర్‌నెట్) ద్వారా అక్రమ నగదు బదిలీ.

11th Computer Science_Chapter-17 Telugu.indd 284 24-03-2023 16:51:35

కొన్ని కంప్యూటర్ నేరాల ఉదాహరణలు మరియు వాటి విధులు క్రింది పట్టిక 17.1 లో ఇవ్వబడినది:

పట్టిక 17.1 కంప్యూటర్ నేరములు

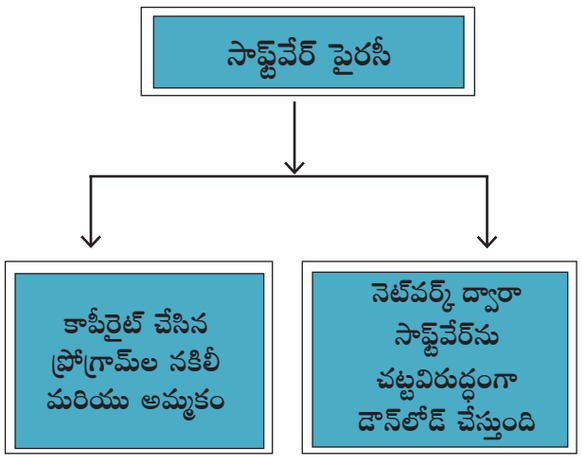
నేరం	పని విధానము
సైబర్ టెర్రిజం	వ్యాపారం లేదా వ్యక్తి పట్ల హ్యాకింగ్ బెదిరింపులు మరియు బ్లాక్ మెయిల్ చేయడం
సైబర్ స్టాకింగ్	ఆన్‌లైన్ ద్వారా వేదించడం.
మాల్‌వేర్	సున్నితమైన డేటాను దొంగిలించడం గుప్తీకరించడం లేదా తొలగించడం, కోర్ కంప్యూటింగ్ విధులను మార్చడం లేదా హైజాక్ చేయడం మరియు వారి అనుమతి లేకుండా వినియోగదారు కంప్యూటర్ కార్యాచరణను పర్యవేక్షించడం వంటి పలు విధులను నిర్వహించగల హానికరమైన ప్రోగ్రాములు.
సేవాదాడిని తిరస్కరించడం	నకిలీ అభ్యర్థనలతో సిస్టమ్‌ను ఓవర్‌లోడ్ చేస్తోంది. కాబట్టి ఇది సాధారణ చట్టబద్ధమైన అభ్యర్థనలను అందించదు.
మోసపూరితం	డేటాను మార్చడం, ఉదాహరణకు బ్యాంకింగ్ రికార్డులను అనధికార ఖాతాకు బదిలీ చేయడానికి మార్చడం
కోత (Harvesting)	ఇతరుల ఖాతా (ల) కు చట్టవిరుద్ధంగా ప్రాప్యత పొందడానికి ఒక వ్యక్తి లేదా ప్రోగ్రామ్ చట్టబద్ధమైన వినియోగదారు నుండి లాగిన్ మరియు పాస్‌వర్డ్ సమాచారాన్ని సేకరిస్తుంది.
గుర్తింపు దొంగతనం	నేరస్థులు ఆర్థిక లబ్ధి కొరకు ఒక వ్యక్తి పైన చేయు సాధారణ నేరం.
మేధో సంపత్తి దొంగతనం (Intellectual property theft)	మరొక వ్యక్తి లేదా సంస్థ అభివృద్ధి చేసిన ఆచరణాత్మక లేదా సంభావిత సమాచారాన్ని దొంగలించడం
సలామి ముక్కలు (Salami slicing)	ప్రతి లావాదేవీ నుండి చిన్న మొత్తంలో డబ్బును దొంగిలించడం.
స్కామ్ (Scam)	నిజం కానిదాన్ని నమ్మడానికి ప్రజలను మోసగించడం
స్పామ్	పెద్ద సంఖ్యలో ఇంటర్నెట్ వినియోగదారులకు అవాంఛిత ఇ-మెయిల్ పంపిణీ చేయడం
స్ఫూఫింగ్	ఇది ఒక హానికరమైన సాధన సమాచారము తెలియని వనరుల నుండి మారు వేషంలో తెలిసిన రిసీవర్‌కు పంపించడం

సాఫ్ట్‌వేర్ పైరసీ (SOFTWARE PIRACY)

సాఫ్ట్‌వేర్ పైరసీ అనేది ఒక వ్యక్తి లేదా సంస్థ సృష్టించిన సాఫ్ట్‌వేర్ యొక్క కాపీరైట్ ఉల్లంఘన. ఇది సంకేతాలు / ప్రోగ్రామ్‌లు మరియు ఇతర సమాచారాన్ని చట్టవిరుద్ధంగా దొంగలించడం మరియు అనధికార మార్గాల ద్వారా నకిలీ కాపీలను సృష్టించడం మరియు ఈ డేటాను ఒకరి స్వంత ప్రయోజనం కోసం లేదా వాణిజ్య లాభం కోసం ఉపయోగించడం.

సరళంగా చెప్పాలంటే సాఫ్ట్‌వేర్ పైరసీ, సాఫ్ట్‌వేర్ యొక్క అనధికార కాపీయింగ్. పటము 17.2 సాఫ్ట్‌వేర్ పైరసీ యొక్క రేఖాచిత్ర ప్రాతినిధ్యం చూపిస్తుంది.

11th Computer Science_Chapter-17 Telugu.indd 285 24-03-2023 16:51:35



పటము 7.2 సాఫ్ట్‌వేర్ పైరసీ యొక్క రేఖాచిత్ర ప్రాతినిధ్యం

సాఫ్ట్‌వేర్ పైరసీకి పూర్తిగా భిన్నమైన విధానాన్ని షేర్‌వేర్ అని పిలుస్తారు. సాఫ్ట్‌వేర్‌ను కాపీ చేయకుండా ప్రజలు ఆపడానికి ప్రయత్నించడం వ్యర్థమని ఇది అంగీకరిస్తుంది మరియు బదులుగా ప్రజల నిజాయితీపై ఆధారపడుతుంది. షేర్‌వేర్ ప్రచురణ కర్తలు ప్రోగ్రామ్‌ల కాపీలను స్నేహితులు మరియు సహోద్యోగులకు ఇవ్వమని వినియోగదారులను ప్రోత్సహిస్తారు, కానీ, ఆ ప్రోగ్రామ్‌ను క్రమం తప్పకుండా ఉపయోగించే ప్రతి ఒక్కరినీ నేరుగా ప్రోగ్రామ్ రచయితకు రిజిస్ట్రేషన్ ఫీజు చెల్లించమని అడుగుతారు. చట్టవిరుద్ధంగా ప్రజలకు అందుబాటులో ఉంచే వాణిజ్య కార్యక్రమాలను తరచూ (Warez) అని పిలుస్తారు.

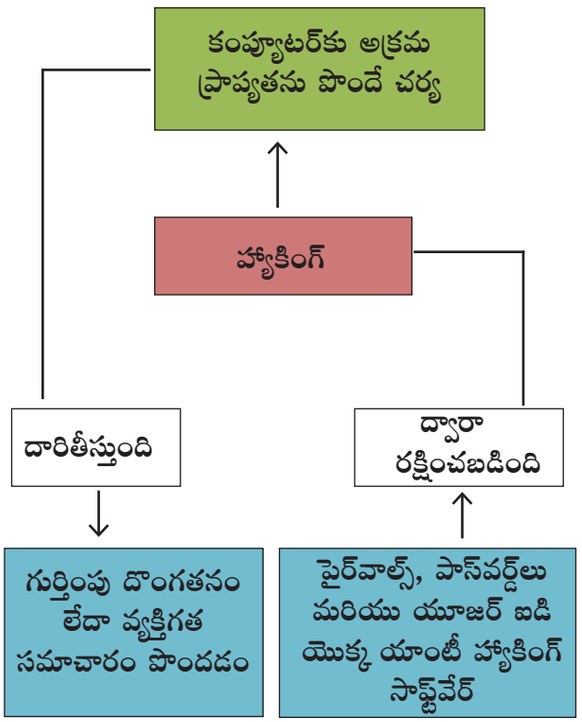
అనధికారిక ప్రవేశము (UNAUTHORIZED ACCESS)

ఎవరైనా చట్టబద్ధమైన వినియోగదారు ఖాతాలోకి ప్రవేశించడం ద్వారా వెబ్‌సైట్ ప్రోగ్రామ్ సర్వర్ సర్వీసు లేదా ఇతర సిస్టమ్‌కు ప్రాప్యత పొందడాన్ని అనధికార ప్రాప్యత అంటారు. ఉదాహరణకు ప్రాప్యత పొందే వరకు వారిది కాని ఖాతా కోసం ఎవరైన పాస్‌వర్డ్ లేదా యూజర్ నేమ్ (username) ఊహించడానికి ప్రయత్నించడం అనధికార ప్రాప్యతగా పరిగణించబడుతుంది.

అనధికార ప్రాప్యతను నివారించడానికి ఫైర్ వాల్స్, ఇంట్రూషన్ డిటెక్షన్ సిస్టమ్స్ (IDS) వైరస్ మరియు కంటెంట్ స్కానర్‌లు, పాచెస్ మరియు హాట్ ఫెక్స్‌సెస్ ఉపయోగించబడతాయి.

హ్యాకింగ్ (HACKING)

యజమానుల అనుమతి లేదా జ్ఞానం లేకుండా వ్యక్తిగత డేటాను దొంగలించడానికి కంప్యూటర్ సిస్టమ్‌లోకి చొరబడటం హ్యాకింగ్ (పాస్‌వర్డ్‌ను దొంగలించు విధం). ఇది కంప్యూటర్ సిస్టమ్‌కు అనధికార ప్రాప్యతను పొందుతోంది మరియు దాని విషయాలను మారుస్తుంది. ఇది నేరపూరిత కార్యకలాపాల కోసం చేయవచ్చు లేదా ఇది ఒక అభిరుచి కావచ్చు. వ్యవస్థల రక్షణను విచ్ఛిన్నం చేసే సవాలును మాత్రమే హ్యాకర్ అనుభవిస్తుంటే హ్యాకింగ్ ప్రమాదకరం కాదు. అయితే ఇటువంటి నైతిక హ్యాకింగ్‌ను నియంత్రిత ప్రయోగాలుగా మాత్రమే సాధన చేయాలి. పటము 17.3 హ్యాకింగ్ యొక్క రేఖాచిత్ర ప్రాతినిధ్యం చూపిస్తుంది.



పటము 17.3 హ్యాకింగ్ యొక్క రేఖాచిత్ర ప్రాతినిధ్యం

క్రాకింగ్ (CRACKING)

క్రాకింగ్ అంటే ఎవరైనా ప్రోగ్రామ్ మూలాన్ని సవరించి కోడ్ దోపిడీ చేయబడవచ్చు లేదా సవరించబడవచ్చు. క్రాకర్ (బ్లాక్ టోపీ లేదా డార్క్ సైడ్ హ్యాకర్ అని కూడా పిలుస్తారు)

11th Computer Science_Chapter-17 Telugu.indd 286 24-03-2023 16:51:35

హానికరమైన లేదా క్రిమినల్ హ్యాకర్ "క్రాకింగ్" అంటే డేటాను దొంగలించడానికి, అవినీతి చేయడానికి లేదా చట్టవిరుద్ధంగా వీక్షించడానికి కంప్యూటర్ సిస్టమ్స్‌లోకి ప్రవేశించడానికి ప్రయత్నించడం.

క్రాకర్ అంటే మరొకరి కంప్యూటర్ సిస్టమ్‌లోకి తరచుగా నెట్‌వర్క్‌లో కంప్యూటర్ ప్రోగ్రామ్‌లలో పాస్‌వర్డ్‌లు లేదా లైసెన్స్‌లను దాటవేసే వ్యక్తి.

వారు కొన్ని సున్నితమైన సమాచారాన్ని అభ్యర్థిస్తూ అధికారిక ఇ – మెయిల్ పంపవచ్చు. ఇది బ్యాంక్ లేదా ఇతర అధికారిక సంస్థ నుండి చట్టబద్ధమైన ఇ – మెయిల్‌లాగా ఉండవచ్చు.

17.3 సైబర్ భద్రత మరియు బెదిరింపులు (Cyber Security and Threats)

సైబర్ దాడులు ప్రధానంగా కంప్యూటర్ సిస్టమ్‌కు గణనీయమైన నష్టాన్ని కలిగించడం కోసం లేదా సంస్థ నుండి ముఖ్యమైన సమాచారాన్ని దొంగిలించడం కోసం ప్రారంభించబడతాయి. సైబర్ భద్రత అనేది సైబర్ దాడుల ప్రమాదాన్ని తగ్గించే సంస్థ మరియు వ్యక్తులను కంప్యూటర్ ఆధారిత బెదిరింపుల నుండి రక్షించే వివిధ సాంకేతికతలు, ప్రక్రియలు మరియు చర్యల సమాహారం.

సైబర్ అటాక్స్ రకాలు (TYPES OF CYBER ATTACKS)

మాల్వేర్ అనేది ఒక రకమైన సాఫ్ట్‌వేర్. దీని ద్వారా నేరస్థులు సాఫ్ట్‌వేర్‌కు అక్రమ ప్రాప్యత మరియు నష్టాన్ని పొందుతారు. వివిధ రకాల సైబర్ దాడులు మరియు వాటి విధులు పట్టిక 17.2 లో ఇవ్వబడ్డాయి.

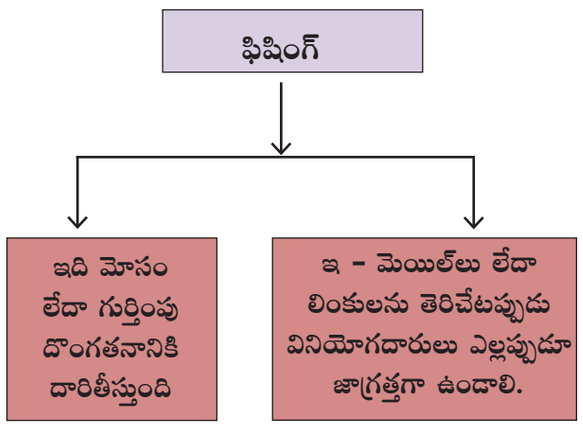
పట్టిక 17.2 సైబర్ దాడులు మరియు విధులు

సైబర్ భద్రతా బెదిరింపులు (Cyber Security Threats)

ఇటీవలి సంవత్సరాలలో భద్రతా వ్యవస్థలలో అంతర్లీనంగా ఉన్న బలహీనతలు మరియు రాజీ పడిన సంస్థాగత మౌలిక సదుపాయాల కారణంగా చాలా మంది వ్యక్తులు మరియు సంస్థలు సమస్యలను ఎదుర్కొంటున్నారు. వివిధ రకాల సైబర్ భద్రతా బెదిరింపులు ఈ క్రింది విధంగా వర్గీకరించబడ్డాయి.

ఫిషింగ్ (Phishing)

ఫిషింగ్ అనేది ఒక రకమైన కంప్యూటర్ నేరం. పాస్‌వర్డ్ మరియు క్రెడిట్ కార్డ్ నంబర్లతో సహా వినియోగదారు డేటాను దొంగిలించడానికి ఉపయోగిస్తారు. ఇ – మెయిల్ (లేదా) తక్షణ వచన సందేశపెట్టెను తెరిచినప్పుడు రాజీపడే హానికరమైన లింక్‌లు లేదా లింక్‌లను పంపిణీ చేయడానికి ఫిషింగ్ ఉపయోగించబడుతుంది. బాధితుల నుండి సున్నితమైన లాగిన్ ఆధారాలను సంగ్రహించడం సహా పలు విధులు నిర్వర్తించే హానికరమైన లింకులు లేదా జోడింపులను పంపిణీ చేయడానికి దాడి చేసేవాడు ఫిషింగ్‌ను ఉపయోగిస్తాడు

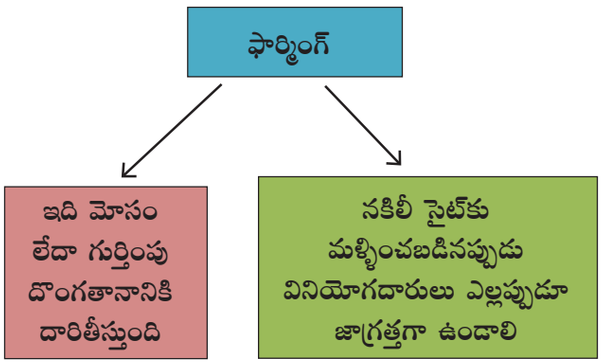


పటము 17.4 ఫిషింగ్ యొక్క రేఖాచిత్ర ప్రాతినిధ్యం

ఫార్మింగ్ (Pharming)

ఫార్మింగ్ అనేది స్కామింగ్ అభ్యాసం. దీనిలో హానికరమైన కోడ్ వ్యక్తిగత కంప్యూటర్ లేదా సర్వర్‌లో ఇన్‌స్టాల్ చేయబడి వినియోగదారులకు వారి జ్ఞానం లేదా అనుమతి లేకుండా మోసపూరిత వెబ్‌సైట్‌లకు తప్పుదారి పట్టిస్తుంది. ఫార్మింగ్‌ను **"ఫిషింగ్ విత్‌వుట్ ఎ"** ట్రాప్ అని పిలుస్తారు. ఇంటర్నెట్‌లో వినియోగదారులను మార్చడానికి హాకర్లు ప్రయత్నించే మరో మార్గం. ఇది వెబ్‌సైట్ యొక్క ట్రాఫిక్‌ను నకిలీ సైట్‌కు మళ్ళించడానికి ఉద్దేశించిన సైబర్ దాడి.

11th Computer Science_Chapter-17 Telugu.indd 287 24-03-2023 16:51:36

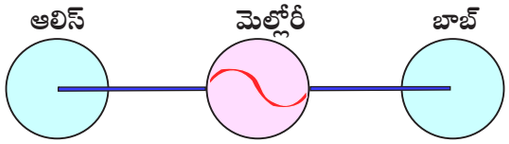


పటము 17.5 ఫార్మింగ్ యొక్క రేఖాచిత్ర ప్రాతినిధ్యం

మ్యాన్ - ఇన్ - ది - మిడిల్ (Man In The Middle (MITM))

మ్యాన్ - ఇన్ - ది - మిడిల్ అటాక్ (MITM అభిమానుల దాడి కూడా) ఈ దాడి రహస్యంగా ప్రసారం చేస్తుంది మరియు ఒకరితో ఒకరు నేరుగా కమ్యూనికేట్ చేస్తున్నారని నమ్మే రెండు పార్టీల మధ్య కమ్యూనికేషన్‌ను మారుస్తుంది.

ఉదాహరణ: ఆలిస్ బాబ్‌తో కమ్యూనికేట్ చేయాలనుకుంటున్నాడని అనుకుందాం. ఇంతలో, మల్లోరీ సంభాషణను వినడానికి మరియు ఐచ్ఛికంగా బాబ్‌కు తప్పుడు సందేశాన్ని అందించడానికి అడ్డగించాలని కోరుకుంటాడు.



పటము 17.6 మ్యాన్ - ఇన్ - ది - మిడిల్ యొక్క ఉదాహరణ

కుకీలు (Cookies)

కుకీ (HTTP కుకీ, వెబ్‌కుకీ, బ్రౌజర్ కుకీలు లేదా కుకీలు అని కూడా పిలుస్తారు) అనేది వెబ్‌సైట్ నుండి పంపబడిన ఒక చిన్న డేటా మరియు వినియోగదారు ఇంటర్నెట్ బ్రౌజ్ చేస్తున్నప్పుడు యూజర్ యొక్క వెబ్ బ్రౌజర్ ద్వారా వినియోగదారు కంప్యూటర్ మెమరీ (హార్డ్ డ్రైవ్) లో నిల్వ చేయబడుతుంది. వెబ్‌సైట్‌లకు స్థిరమైన సమాచారాన్ని (ఆన్‌లైన్ స్టోర్‌లోని షాపింగ్ కార్ట్‌లో జోడించిన అంశాలు వంటివి) గుర్తుంచుకోవడానికి లేదా యూజర్ యొక్క బ్రౌజింగ్ కార్యాచరణను రికార్డ్ చేయడానికి (నిర్దిష్ట బటన్లను క్లిక్ చేయడం, లాగిన్ చేయడం మొదలైనవి) కుకీలు నమ్మదగిన యంత్రాంగాన్ని రూపొందించారు. వినియోగదారుడు గతంలో పేర్లు, చిరునామాలు, పాస్‌వర్డ్‌లు మరియు క్రెడిట్ కార్డ్ నంబర్‌ల వంటి ఫారమ్ ఫీల్డ్‌లలోకి ప్రవేశించిన ఏక పక్ష సమాచార భాగాలను గుర్తుంచుకోవడానికి కూడా వీటిని ఉపయోగించవచ్చు. భద్రతా కోణం నుండి కుకీల డేటా గుప్తీకరించబడకపోతే ఏదైనా అనామక వినియోగదారు (హ్యాకర్ కుకీల సమాచారాన్ని యాక్సెస్ చేయవచ్చు మరియు దానిని దుర్వినియోగం చేయవచ్చు.

వెబ్‌సైట్ సాధారణంగా క్రింది కారణాల కోసం కుకీలను ఉపయోగిస్తాయి:

- వెబ్‌సైట్‌లు ఎవరు సందర్శించారనే వారి గురించి సమాచారాన్ని సేకరించడానికి.
- సందర్శకులు ఎంత తరచుగా సైట్‌కు వస్తారో మరియు వారు సైట్‌లో ఎంతకాలం ఉంటారో తెలుసుకోవడానికి సైట్‌లు తరచుగా ఈ సమాచారాన్ని ఉపయోగిస్తాయి.
- ఇది వెబ్‌సైట్‌లో వినియోగదారు అనుభవాన్ని వ్యక్తికరించడానికి సహాయపడుతుంది.
- వినియోగదారుల గురించి వ్యక్తిగత సమాచారాన్ని నిల్వచేయడానికి కుకీలు సహాయపడతాయి, తద్వారా వినియోగదారు తదనంతరం సైట్‌కు తిరిగి వచ్చినప్పుడు మరింత వ్యక్తికరించిన అనుభవం అందించబడుతుంది.

11th Computer Science_Chapter-17 Telugu.indd 288 24-03-2023 16:51:36

మీరు ఎప్పుడైనా ఒక సైట్‌కు తిరిగివచ్చి మీ పేరు రహస్యంగా తెరపై కనిపించడం చూస్తే మునుపటి సందర్శనలో మీరు మీ పేరును సైట్‌కు ఇచ్చారు మరియు అది కుకీలో నిల్వచేయబడింది. ఉదాహరణకు కొన్ని ఆన్‌లైన్ షాపింగ్ సైట్లు వారి మునుపటి కొనుగోళ్ల ఆధారంగా వినియోగదారులకు సిఫార్సులు చేసే విధానం. ఇది ప్రకటనలను పర్యవేక్షించడానికి సహాయపడుతుంది. కంప్యూటర్ సిస్టమ్‌లో కుకీలు హానికరంగా పనిచేయవు. అవి కేవలం ఎప్పుడైనా తొలగించగల టెక్స్ట్‌ఫైల్స్.

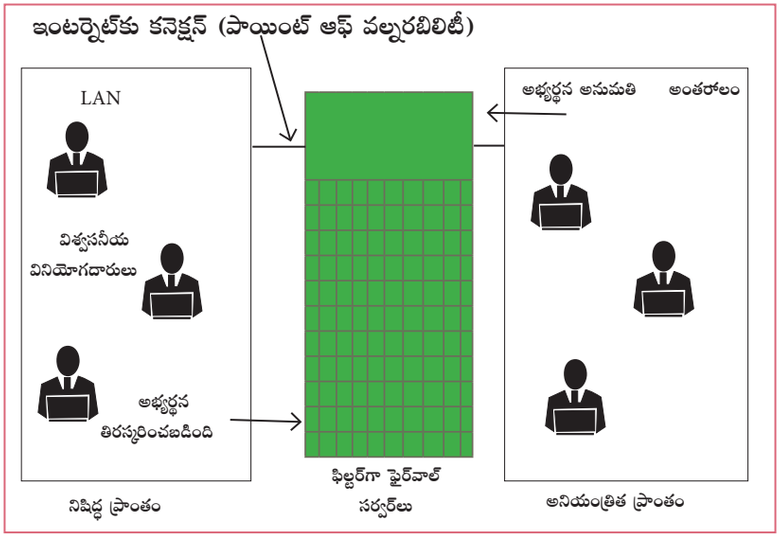
వైరస్లను వ్యాప్తి చేయడానికి కుకీలను ఉపయోగించలేరు మరియు అవి మీ హార్డ్ డ్రైవ్‌ను యాక్సెస్ చేయలేవు. అయినప్పటికీ, క్రెడిట్ కార్డ్ సమాచారంతో సహా మీరు వెబ్‌సైట్‌కు అందించే ఏదైన వ్యక్తిగత సమాచారం మీ బ్రౌజర్‌లో కుకీ ఫీచర్ స్పష్టంగా ఆపివేయబడకపోతే కుకీలో నిల్వ చేయబడుతుంది. గోప్యతను బెదిరించే మార్గంలో కుకీలు ఒకటి.

ఫైర్‌వాల్ మరియు ప్రాక్సీ సర్వర్‌లు (Firewall and Proxy Servers)

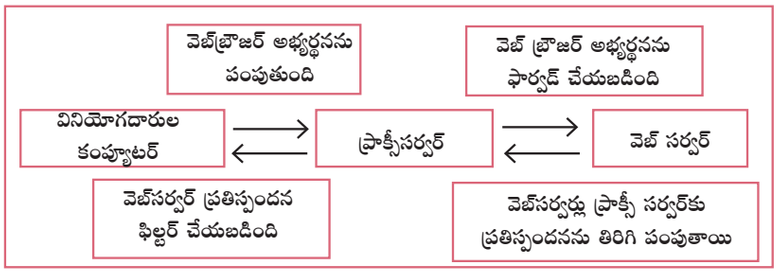
ఫైర్‌వాల్ అనేది కంప్యూటర్ నెట్‌వర్క్ భద్రతా ఆధారిత వ్యవస్థ. ఇది పర్యవేక్షిస్తుంది మరియు నిర్వహించిన భద్రతా నియమాల ఆధారంగా ఇన్‌కమింగ్ మరియు అవుట్‌గోయింగ్ నెట్‌వర్క్ ట్రాఫిక్‌ను నియంత్రిస్తుంది. ఫైర్‌వాల్ సాధారణంగా విశ్వసనీయ అంతర్గత కంప్యూటర్ నెట్‌వర్క్ మరియు నెట్‌వర్క్ అప్పగించిన కంప్యూటర్ మధ్య బ్లాక్‌ను ఏర్పాటు చేస్తుంది. పటము 17.7 ఫైర్‌వాల్ సర్వర్ యొక్క పనిని చూపిస్తుంది.

ప్రాక్సీ సర్వర్ తుది వినియోగదారులకు మరియు వెబ్ సర్వర్‌కు మధ్యవర్తిగా పనిచేస్తుంది. క్లయింట్ ప్రాక్సీ సర్వర్‌కు కనెక్ట్ అవుతుంది. ఫైల్ కనెక్షన్ వెబ్‌పేజీ లేదా వేరే సర్వర్ నుండి లభించే ఇతర వనరులు వంటి కొన్ని సేవలను అభ్యర్థిస్తుంది. ప్రాక్సీ సర్వర్ అభ్యర్థనను ప్రామాణీకతను తనిఖీ చేస్తుంది. మరియు దాని ఆధారంగా అభ్యర్థనను మంజూరు చేస్తుంది. ప్రాక్సీ సర్వర్‌లు సాధారణంగా తరచుగా సందర్శించే సైట్ చిరునామాలను దాని cacheలో ఉంచుతాయి. ఇది మెరుగైన ప్రతిస్పందన సమయానికి దారితీస్తుంది.

పటము 17.8 ప్రాక్సీ యొక్క పని తీరును చూపిస్తుంది



పటము 17.7 ఫైర్‌వాల్ సర్వర్



పటము 17.8 ప్రాక్సీ సర్వర్ యొక్క పని

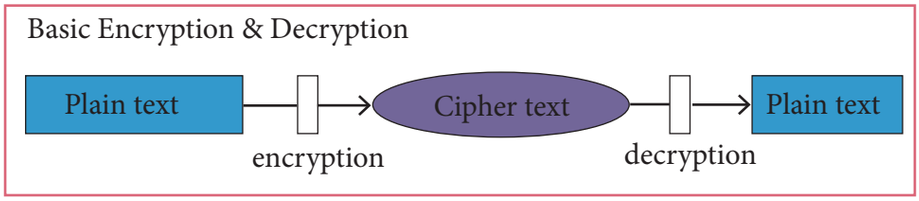
11th Computer Science_Chapter-17 Telugu.indd 289 24-03-2023 16:51:36

గుప్తీకరణ మరియు వ్యక్తీకరణ (Encryption and Decryption)

గుప్తీకరణ మరియు వ్యక్తీకరణ అనేది అధికారం ఉన్న వ్యక్తులకు మాత్రమే సమాచారాన్ని ప్రవేశసాధ్యం చేయగల గోప్యతను నిర్ధారించే ప్రక్రియలు.

గుప్తీకరణ అనేది సాధారణ వచన డేటాను (సాధారణ పాఠం) యాదృచ్ఛిక మరియు మాంగిల్డ్ డేటాగా (సైఫర్ టెక్స్ట్) అనువదించే ప్రక్రియ.

వ్యక్తీకరణ అనేది సంకేతలిపి వచనాన్ని తిరిగి సాధారణ వచనాలుగా మార్చే ప్రక్రియ. గుప్తీకరణ మరియు వ్యక్తీకరణలు క్రిప్టోగ్రఫీ (గూడాచారి లిపి) ద్వారా జరుగుతుంది. (cryptography) ఒక కీలక పదము అనేది క్రిప్టోగ్రాఫిక్ అల్గోరిథం యొక్క క్రియాత్మక ఉత్పత్తిని నిర్ణయించే సమాచార భాగం.



పటము 17.9 గుప్తీకరణ మరియు వ్యక్తీకరణ

రహస్య సంభాషణను సులభతరం చేయడానికి మిలిటరీలు మరియు ప్రభుత్వాలు ఎన్క్రిప్షన్ ఉపయోగిస్తాయి. ఇది ఇప్పుడు అనేక రకాల పౌర వ్యవస్థలలోని సమాచారాన్ని రక్షించడానికి సాధారణంగా ఉపయోగించబడుతుంది. కమ్యూనికేషన్ సిస్టమ్‌లో డేటాను రక్షించడానికి కూడా ఇది ఉపయోగపడుతుంది. ఉదాహరణకు డేటా నెట్‌వర్క్‌లు (ఉదా: ఇంటర్నెట్, ఇ కామర్స్) మొబైల్ టెలిఫోన్లు, వైర్‌లెస్ మైక్రోఫోన్లు, వైర్‌లెస్ ఇంటర్‌కామ్ సిస్టమ్స్, బ్లూటూత్ పరికరాలు మరియు బ్యాంక్ ఆటోమేటిక్ టెల్లర్ యంత్రాల ద్వారా బదిలీ చేయబడుతుంది. కమ్యూనికేషన్‌లోని డేటాను అడ్డగించినట్లు అనేక నివేదికలు వచ్చాయి. గత కొన్ని సంవత్సరాలుగా అనధికార వినియోగదారులచే నెట్‌వర్క్ ట్రాఫిక్ నుండి రక్షించడానికి నెట్‌వర్క్‌లలో ప్రసారం చేసినప్పుడు డేటాను కూడా గుప్తీకరిస్తున్నారు.

17.4 సమాచార సాంకేతిక చట్టం (INTRODUCTION TO INFORMATION TECHNOLOGY ACT)

21 వ శతాబ్దంలో కంప్యూటర్, ఇంటర్నెట్ మరియు ఐసిటి లేదా ఇ విప్లవం ప్రజల జీవన శైలిని మార్చాయి. నేడు కాగిత ఆధారిత కమ్యూనికేషన్ ఇ – కమ్యూనికేషన్ ద్వారా ప్రత్యామ్నాయం చేయబడింది. దీని ప్రకారం మనకు సైబర్ వరల్డ్ ఇ – లావాదేవీ, ఇ–బ్యాంకింగ్, ఇ–రిటర్న్ మరియు ఇ–కాంట్రాక్టులు వంటి కొత్త పరిభాషలు ఉన్నాయి. ఇ–విప్లవం యొక్క సానుకూల వైపు కాకుండా కంప్యూటర్ యొక్క ప్రతికూల వైపు కూడా ఉంది. అనగా నేరస్థుల చేతిలో ఇంటర్నెట్ మరియు ఐసిటి నేరాలు ఆయుధంగా మారాయి. సైబర్ చట్టం లేదా ఇన్ఫర్మేషన్ టెక్నాలీ చట్టం లేదా ఇంటర్నెట్ చట్టం సైబర్ ప్రదేశంలో సైబర్ నేరాల సమస్యలను పరష్కరించడానికి సభ్యుల కొత్త ప్యానెల్ ఉద్భవించింది.

భారతదేశంలో సైబర్ చట్టం మరియు ఐటి యాక్ట్ 2000 కంప్యూటర్ నేరాలను నివారించడానికి 2008 లో సవరింపబడినది. ఐటి యాక్ట్ 2000 అనేది ఎలక్ట్రానిక్ డేటా ఇంటర్ చేంజ్ (ఇడిఐ) మరియు ఎలక్ట్రానిక్ కమ్యూనికేషన్ యొక్క ఇతర మార్గాల ద్వారా జరిగే లావాదేవీలకు చట్టపరమైన గుర్తింపును అందించే చర్య. సైబర్ నేరము మరియు ఎలక్ట్రానిక్ కామర్స్ (ఇ–కామర్స్) తో వ్యవహరించే భారతదేశంలో ఇది ప్రాథమిక చట్టం. ఇ – కామర్స్ అంటే ఎలక్ట్రానిక్ డేటా మార్పిడి లేదా సమాచారం యొక్క ఎలక్ట్రానిక్ ఫైలింగ్.

11th Computer Science_Chapter-17 Telugu.indd 290 24-03-2023 16:51:36

మీకు తెలుసా?

సైబర్ చట్టం లేదా ఇంటర్నెట్ చట్టం అనేది ఇంటర్నెట్ వాడకానికి సంబంధించిన చట్టపరమైన సమస్యలను కలుపుతుంది.

నివారణ (PREVENTION)

25% సైబర్ నేరాలు పరిష్కరించబడలేదు. సమాచారాన్ని రక్షించడానికి క్రింది అంశాలను గమనించాలి.

- సంక్లిష్టమైన పాస్‌వర్డ్ సెట్టింగ్ మీ సర్ఫింగ్‌ను సురక్షితం చేస్తుంది
- ఇంటర్నెట్ ఉపయోగంలో లేనప్పుడు, దానిని డిస్‌కనెక్ట్ చేయండి
- తెలియని వారు పంపిన స్పామ్ మెయిల్ లేదా ఇ-మెయిల్‌లను తెరవవద్దు.
- యాంటీ వైరస్ సాఫ్ట్‌వేర్‌ను ఉపయోగిస్తున్నప్పుడు దాన్ని తాజాగా ఉంచండి.

మీకు తెలుసా?

- సమాచార భద్రత అనేది వ్యాపార సంస్థలో రోగనిరోధక వ్యవస్థ
- బౌన్స్ చేయని చెక్కుని భద్రతా తనిఖీ అంటారు. మీరు బయలుదేరే ముందు ప్రతిరోజూ చేయండి
- మీ వంతు చేయండి – సెక్యూరిటీ స్మార్ట్‌గా ఉండండి.
- క్లిక్ చేయడానికి తొందరపడకండి మీరు ఆన్‌లైన్‌లో షాపింగ్ చేసేటప్పుడు జాగ్రత్తగా ఉండండి
- పున:ప్రారంభం అనేది తెలివైన పని.
- పాస్‌వర్డ్‌లు టూత్ బ్రష్‌లు వంటివి. క్రొత్తగా ఉన్నప్పుడు అవి ఉత్తమమైనవి మరియు ఎప్పుడూ వాటిని ఇతరులతో పంచుకోకూడదు.
- మీరు మరియు మీ సిస్టమ్ విడిపోయినప్పుడు, మీ సిస్టమ్‌ను ముందుగానే ఆపివేయాలి.
- మీ మనస్సు సమాచార నిల్వ గది. ఆ తలుపుని లాక్ చేయండి
- _ a _ _ పదం రక్షించు సేవ్ మరియు సురక్షితం లేని పాస్‌వర్డ్ కాదు.
- లింక్ లింక్ నిర్లక్ష్యం ఆపండి కనెక్ట్ చేయడానికి ముందు ఆలోచించండి.

మూల్యాంకనం (Evaluation)

విభాగం - అ

సరైన సమాధానం ఎంచుకోండి

1. క్రింది వాటిలో విధానాలు అభ్యాసాలు మరియు విలువలతో వ్యవహరించునది ఏది?
   (అ) పైరసీ (ఆ) కార్యక్రమాలు (ఇ) వైరస్ (ఈ) కంప్యూటర్ నీతి

2. చట్టవిరుద్ధంగా ప్రజలకు అందుబాటులో ఉంచిన వాణిజ్య కార్యక్రమాలను _____ అంటారు.
   (అ) ఫివేర్ (ఆ) వారేజ్ (ఇ) ఉచిత సాఫ్ట్‌వేర్ (ఈ) సాఫ్ట్‌వేర్

3. క్రింది వాటిలో ఏది స్వీయ పునరావృతం మరియు కంప్యూటర్ ప్రోగ్రామ్ తమను తాము అటాచ్ చేయవలసిన అవసరం లేదు?
   (అ) వైరస్లు (ఆ) వామ్స్ (ఇ) స్పైవేర్ (ఈ) ట్రోజన్లు

4. వినియోగదారు వెబ్‌సైట్‌ను సందర్శించుటను కింది వాటిలో ఏది ట్రాక్ చేయును?
   (అ) స్పైవేర్ (ఆ) కుకీలు (ఇ) వామ్స్ (ఈ) ట్రోజన్లు

11th Computer Science_Chapter-17 Telugu.indd 291 24-03-2023 16:51:36

5. క్రింది వాటిలో ఏది కంప్యూటర్ సిస్టమ్స్‌తో హానికరమైన ప్రోగ్రామ్ కాదు?
(అ) వామ్స్ (ఆ) ట్రోజన్లు (ఇ) స్పైవేర్ (ఈ) కుకీలు

6. ఇన్కమింగ్ మరియు అవుట్గోయింగ్ ట్రాఫిక్ను పర్యవేక్షించే మరియు నియంత్రించే కంప్యూటర్ యొక్క నెట్‌వర్క్ భద్రత
(అ) కుకీలు (ఆ) వైరస్ (ఇ) ఫైర్‌వాల్ (ఈ) పామ్స్

7. సాంకేతిక లిపి వచనాన్ని సాధారణ వచనంగా మార్చే ప్రక్రియ
(అ) గుప్తీకరణ (ఆ) వ్యక్తీకరణ (ఇ) సి.కీ (ఈ) ప్రాక్సీసర్వర్

8. ఇ – కామర్స్ అంటే
(అ) ఎలక్ట్రానిక్ కామర్స్ (ఆ) ఎలక్ట్రానిక్ డేటా మార్పిడి
(సి) విద్యుత్ డేటా మార్పిడి (ఈ) ఎలక్ట్రానిక్ వాణిజ్యీకరణ

9. ఇతరులకు అవాంఛిత ఇ – మెయిల్ పంపిణీ చేయడం
(అ) స్కామ్ (ఆ) స్పామ్ (ఇ) మోసం (ఈ) స్ఫూఫింగ్

10. లావాదేవీలకు చట్టపరమైన గుర్తింపును నిర్వహిస్తారు
(అ) ఎలక్ట్రానిక్ డేటా ఇంటర్‌చేంజ్ (ఆ) ఎలక్ట్రానిక్ డేటా ఎక్సేంజ్
(ఇ) ఎలక్ట్రానిక్ డేటా బదిలీ (ఈ) ఎలక్ట్రికల్ డేటా ఇంటర్‌చేంజ్

విభాగం – ఆ

ఈ క్రింది ప్రశ్నలకు జవాబులు వ్రాయండి

1. హార్వెస్టింగ్ అంటే ఏమిటి?
2. వారేజ్ అంటే ఏమిటి?
3. క్రాకింగ్ గూర్చి చిన్న గమనిక వ్రాయండి.
4. రెండు రకాల సైబర్ దాడులను వ్రాయండి.
5. కుక్కీస్ అనగానేమి?

విభాగం – సి

లఘు ప్రశ్నలు

1. ఫైర్‌వాల్స్ పాత్ర ఏమిటి?
2. గుప్తీకరణ మరియు వ్యక్తీకరణ గురించి వ్రాయండి
3. ప్రాక్సీ సర్వర్ గురించి వివరించండి
4. ఏదైనా కంప్యూటర్ యూజర్ అనుసరించాల్సిన మార్గదర్శకాలు ఏమిటి?
5. నైతిక సమస్యలు అనగానేమి? కొన్ని పేర్కొనండి.

విభాగం – డి

ఈ క్రింది ప్రశ్నలను వివరించండి

1. కంప్యూటర్ ఉపయోగించి చేసే నేరాలు ఏమిటి?
2. పైరసీ అంటే ఏమిటి ? పైరసీ రకాలను ప్రస్తావించుము. వీటిని ఎలా నివారించగలరు?
3. వివిధ సైబర్ దాడులను గూర్చి వ్రాయుము.

సూచన పుస్తకాలు :

- కంప్యూటర్ నెట్‌వర్క్ సెక్యూరిటీ అండ్ సైబర్ ఎథిక్స్ బై జోసెఫ్ మిగ్గాకిజ్జా
- “ఇన్వెస్టిగేటింగ్ సైబర్ లా అండ్ సైబర్ ఎథిక్స్: ఇష్యూస్, ఇంపాక్ట్స్ అండ్ ప్రాక్టీసెస్: 1” అల్ఫ్రెడా డ్యూడ్లీ మరియు జేమ్స్ బ్రామన్.

11th Computer Science_Chapter-17 Telugu.indd 292 24-03-2023 16:51:36

యూనిట్ V

కంప్యూటర్ ఎథిక్స్ మరియు సైబర్ భద్రతా (Computer Ethics And Cyber Security)

అధ్యాయము 18

తమిళ కంప్యూటరింగ్ (Tamil Computing)



18.1 పరిచయము (Introduction)

“ பிறநாட்டு நல்லறிஞர் சாத்திரங்கள்
தமிழ்மொழியிற் பெயர்த்தல் வேண்டும்;
இறவாத புகழுடைய புதுநூல்கள்
தமிழ்மொழியில் இயற்றல் வேண்டும்;
மறைவாக நமக்குள்ளே பழங்கதைகள்
சொல்வதிலோர் மகிமை இல்லை;
திறமான புலமையெனில் வெளிநாட்டோர்;
அதை வணக்கஞ் செய்தல் வேண்டும்.”

- மகாகவி பாரதி

20 వ శతాబ్దంలో కంప్యూటర్ ఆవిష్కరణతో మానవ నాగరికత అభివృద్ధి చెందింది. కంప్యూటర్ అభివృద్ధి ప్రారంభంలో గణన సాధనంగా ప్రారంభమైంది మరియు ఇప్పుడు కంప్యూటర్లు లేకుండా మానవ జీవితం లేనంతగా ఈ వృద్ధి మారింది.

మారుతున్న సాంకేతిక పరిజ్ఞానాలకు అనుగుణంగా స్వీకరించే సామర్థ్యం లేనప్పుడు ఏ భాష అయిన పాతది అవుతుందనేది నిజం. తమిళం వేలాది సంవత్సరాలుగా ఉనికిలో గల భాష. ఆధునిక సాంకేతిక పరిజ్ఞానం అభివృద్ధి, పెరుగుతున్న సాంకేతిక మార్పులను అవలంభించడానికి సిద్ధంగా ఉన్నందున శాస్త్రీయ తమిళ వృద్ధిని ప్రభావితం చేయదు. తమిళం కేవలం భాష కాదు. అది మన గుర్తింపు, మన జీవితం ఇది మన భావన.

“எங்கள் வாழ்வும், எங்கள் வளமும் மங்காத தமிழென்று சங்கே முழங்கு” – புரட்சி கவி.

18.2 ఇంటర్నెట్‌లో తమిళం (Tamil in Internet)

ప్రతి మనిషి జీవితంలో ఇంటర్నెట్ నేడు కీలక పాత్ర పోషిస్తుందని మనకు తెలుసు. ఇంటర్నెట్ అనేది ఉత్తమ సమాచార సాంకేతిక పరికరం. దీని ద్వారా మనకు సమాచారం లభిస్తుంది.

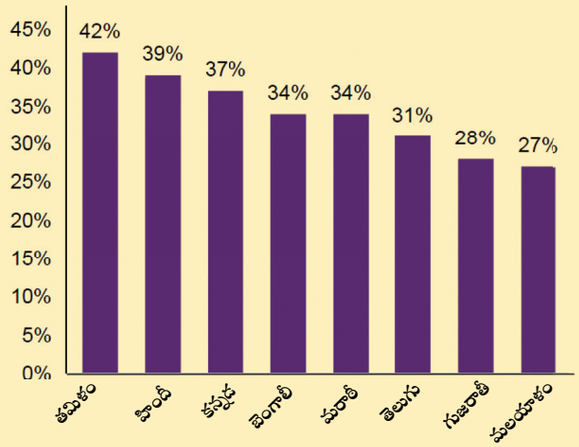
2017 లో గూగుల్‌తో పాటు సింగపూర్ ఆధారిత సంస్థ KPMG నిర్వహించిన ఒక అధ్యయనం ప్రకారం భారతదేశంలో ఎక్కువగా ఉపయోగించబడుతున్న భాషలలో తమిళం అగ్రస్థానంలో ఉంది. ఇక్కడ 42% మంది తమిళంలో ఇంటర్నెట్‌ను ఉపయోగిస్తున్నారు.

11th Computer Science_Chapter-18 Telugu.indd 293 24-03-2023 16:51:46

68%[9] ఇంటర్నెట్ వినియోగదారులు స్థానిక భాష డిజిటల్ కంటెంట్‌ను ఇంగ్లీష్ కంటే నమ్మదగినదిగా భావిస్తారు.

ప్రస్తుతం తమిళం 42%[9] అత్యధిక ఇంటర్నెట్ స్వీకరణ స్థాయిలను కలిగి ఉంది. భారతీయ భాషా వినియోగదారులలో హిందీ మరియు కన్నడ తరువాతి స్థానంలో ఉన్నాయి.

భారతీయ భాషా వినియోగదారుల కొరకు అందింపబడు భాషల వారీగా ఇంటర్నెట్ ప్రాప్యతా స్థాయిలు – 2016 [12]



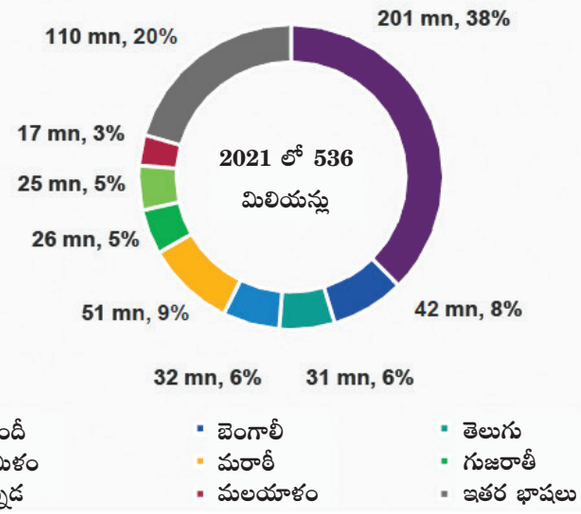
అధ్యయనాల ప్రకారం, భారతదేశంలో 2021 నాటికి 74% మంది ప్రజలు తమిళాన్ని ఉపయోగించి ఇంటర్నెట్‌ను యాక్సెస్ చేస్తారు మరియు భారతదేశంలో ఇంటర్నెట్ యొక్క అత్యధిక వినియోగంలో ఇది ఉంటుంది.

2021 సం॥ నాటికి ఇంటర్నెట్‌లో అంగ్లభాషా వీక్షకుల కంటే హిందీ భాషా వీక్షకుల సంఖ్య అధికం.

భారతీయ భాషల ఇంటర్నెట్ వీక్షకులలో మరాఠీ, బెంగాలీ, తమిళం మరియు తెలుగు భాషల్లోని మొత్తం ఇంటర్నెట్ వీక్షకుల సంఖ్య దాదాపు 30%.

భారతీయ భాషా పరిసరాల్లో భవిష్యత్తుకాలంలో తమిళం మరియు కన్నడ భాషలను మాట్లాడే ప్రజలు ఇంటర్నెట్ సేవలను వినియోగించుకోవడంలో ఉన్నత స్థానంలో ఉండగలరు.

భాషవారీగా ఇంటర్నెట్ వినియోగదారులు 2021[12]



భాష	స్వీకార ప్రాప్యత
హిందీ	54%
బెంగాలీ	46%
తెలుగు	60%
తమిళం	74%
మరాఠీ	43%
గుజరాతీ	43%
కన్నడ	74%
మలయాళం	44%
ఇతర భాషలు	NA*

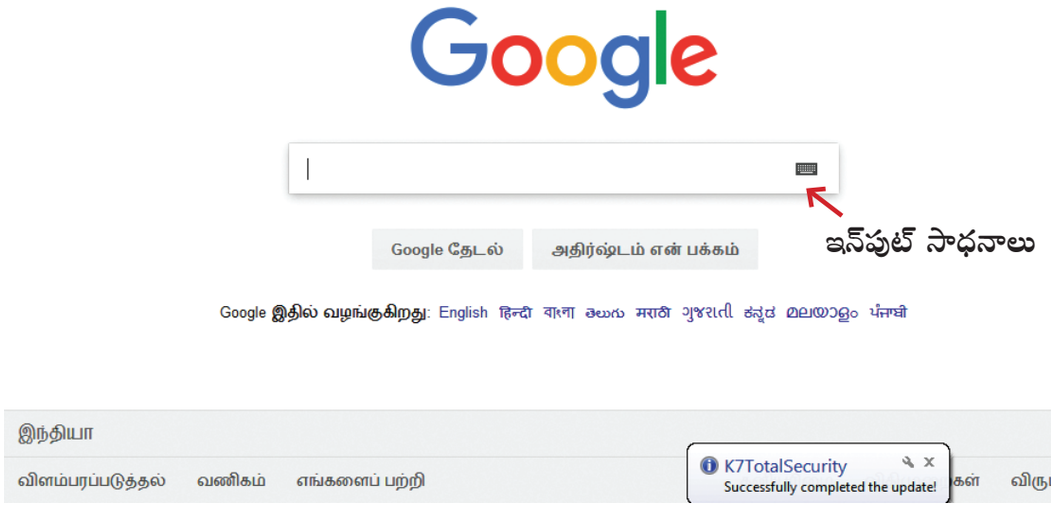
*పైన పేర్కొనిన 8 భారతీయ భాషల్లో ఇంటర్నెట్ వినియోగం భవిష్యత్తులో మరింత అధికరించునని అంచనా.

11th Computer Science_Chapter-18 Telugu.indd 294 24-03-2023 16:51:47

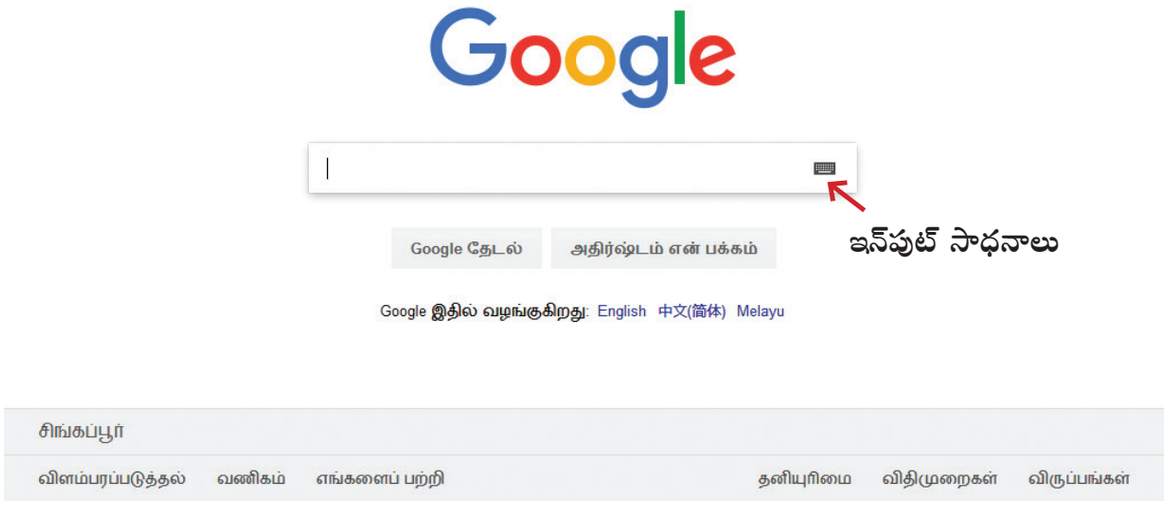
ఈ గణాంకాల డేటా తమిళంలో ఇంటర్నెట్ సేవలను మెరుగుపరచడానికి ఉపయోగపడుతుంది.

18.3 తమిళంలో సెర్చ్ ఇంజన్లు (Search Engines in Tamil)

సైబర్ స్థలం నుండి ఏదైనా సమాచారాన్ని శోధించడానికి "సెర్చ్ ఇంజన్లు" ఉపయోగించబడతాయి. చాలా సెర్చ్ ఇంజన్లు ఉన్నప్పటికీ, వాటిలో కొన్ని మాత్రమే తరచుగా వాడుకలో ఉన్నాయి. మొదటి పది సెర్చ్ ఇంజన్లలో గూగుల్ బింగ్ మరియు యాహూ వరుసగా మొదటి మూడు స్థానాలను దక్కించుకున్నాయి. గూగుల్ మరియు బింగ్ తమిళంలో శోధన సదుపాయాలను కల్పిస్తాయి. అంటే మీరు తమిళం ద్వారా ప్రతిదీ శోధించవచ్చు. గూగుల్ సెర్చ్ ఇంజన్ మీకు అంతర్నిర్మిత తమిళ వర్చువల్ కీబోర్డ్ ఇస్తుంది.

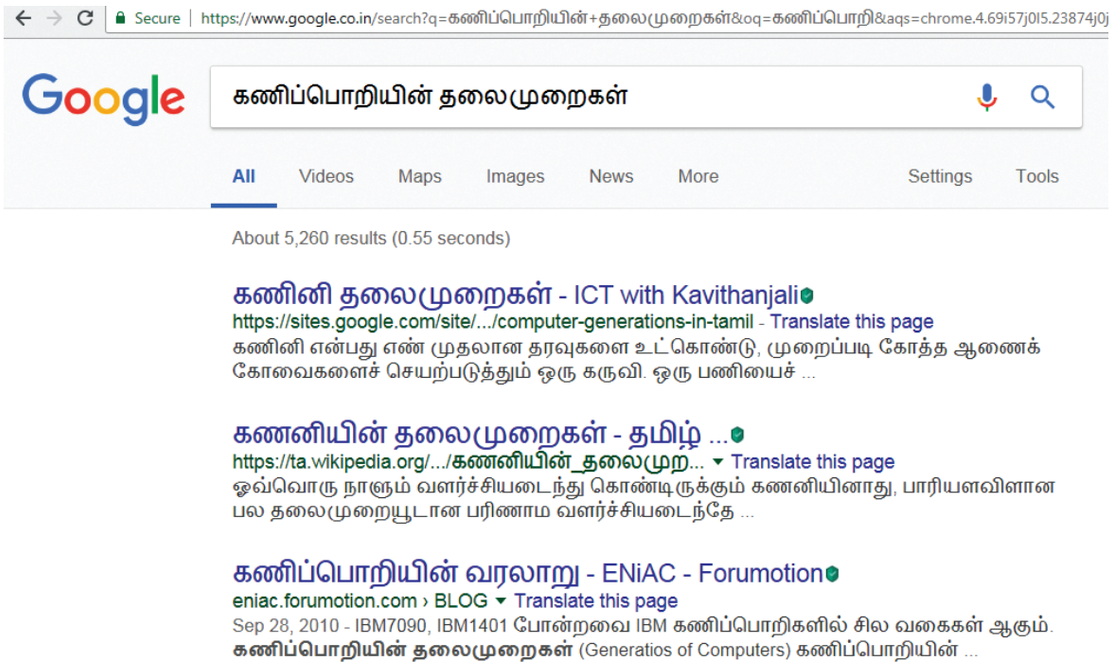


படము 18.1 గూగుల్ సెర్చ్ ఇంజన్లు



படము 18.1(b) (గూగుల్ సెర్చ్ ఇంజన్ (సింగపూర్)

11th Computer Science_Chapter-18 Telugu.indd 295 24-03-2023 16:51:48



படము 18.2 తమిళంలో శోధిస్తోంది

18.4 ఇ – పాలన (e – Governance)

ఇంటర్నెట్ ద్వారా ప్రభుత్వ సేవలను పొందడంను ఇ – గవర్నెన్స్ అంటారు. తమిళనాడు ప్రభుత్వం ఇంటర్నెట్ ద్వారా తన సేవలను అందిస్తుంది. ప్రభుత్వంతో సంభాషించవచ్చు. ప్రపంచంలోని ఏ మూల నుండి అయినా తమిళనాడు ప్రభుత్వ వెబ్‌పోర్టర్ నుండి ముఖ్యమైన ప్రకటనలు ప్రభుత్వ ఉత్తర్వులు మరియు ప్రభుత్వ సంక్షేమ పథకాలను పొందవచ్చు.



11th Computer Science_Chapter-18 Telugu.indd 296 24-03-2023 16:51:48

పటము 18.3 తమిళనాడు ప్రభుత్వ అధికారిక వెబ్‌సైటు

తమిళం ద్వారా ఇ - గవర్నెర్స్	వెబ్ అడ్రస్
తమిళనాడు ప్రభుత్వ అధికారిక వెబ్‌సైట్	http://www.tn.gov.in/ta
వ్యవసాయ ఇంజనీరింగ్ విభాగం	http://www.aed.tn.gov.in/
పర్యావరణ శాఖ	http://www.environment.tn.nic.in/
ప్రభుత్వ పరీక్షల డైరెక్టరేట్	http://www.dge.tn.nic.in/
తమిళనాడు ఆరోగ్య శాఖ	http://www.tnhealth.org/
తమిళనాడు సూక్ష్మ, చిన్న మరియు మధ్యతరహా పరిశ్రమల విభాగం	http://www.msmeonline.tn.gov.in/
గ్రామీణాభివృద్ధి పంచాయతీ రాజ్ విభాగం	http://www.tnrd.gov.in/
వెనుకబడిన చాలా వెనుకబడిన మరియు మైనారిటీల సంక్షేమ శాఖ	http://www.bcmbcmw.tn.gov.in/
తమిళనాడు అటవీ శాఖ	https://www.forests.tn.gov.in/
హిందూ మత మరియు స్వచ్ఛంద ఎన్డోమెంట్స్ శాఖ	http://www.tnhrce.org/
తమిళనాడు ప్రజాసేవా కమీషన్ (TNPSC)	http://www.tnpsc.gov.in/tamilversion/index.html
శ్రీలంక ప్రభుత్వ అధికారిక వెబ్‌సైట్	https://www.gov.lk/index.php

భారతదేశం వెలుపల శ్రీలంక ప్రభుత్వం వారి అన్ని సేవలను తమిళంలో అధికారిక వెబ్‌సైట్ ద్వారా అందిస్తుంది. సింగపూర్ ప్రభుత్వం తన అధికారిక వెబ్‌సైట్‌ను తమిళంలో నిర్వహించినప్పటికీ ఇది ఎల్లప్పుడూ తమిళంలో ప్రభుత్వ సేవలను అందిస్తోంది.

18.5 ఇ - గ్రంథాలయము (e-Library)

ఇ - గ్రంథాలయ పోర్టల్ లేదా ఇ - పుస్తకాల సేకరణ యొక్క వెబ్‌సైట్ తమిళ ఇ - లైబ్రరీ సేవలు వేలాది తమిళ పుస్తకాలను ఎక్కువగా ఉచితంగా ఇ - బుక్‌లుగా అందించాయి. మాతృభూమికి దూరంగా ఉన్న తమిళ ప్రజానికానికి ఇది ఒక ఉపయోగకరమైన సేవ.

తమిళ ఇ - గ్రంథాలయము

తమిళ ఇ - గ్రంథాలయము	వెబ్‌సైట్ చిరునామా
తమిళనాడు పాఠశాల విద్య మరియు ఉపాధ్యాయ విద్య శిక్షణ పాఠ్య పుస్తకాలు మరియు వనరుల పుస్తకాలు.	http://www.textbooksonline.tn.nic.in/
తమిళనాడు వర్చువల్ అకాడమి.	http://www.tamilvu.org/library/libindex.htm
కన్నిమర పబ్లిక్ - గ్రంథాలయము	http://connemarapubliclibrarychennai.com/Veettukku_oru_noolagam/index.html
తమిళ డిజిటల్ లైబ్రరీ	http://tamildigitallibrary.in/
చెన్నై గ్రంథాలయము	http://www.chennailibrary.com/
తమిళగమ్	http://www.thamizhagam.net/parithi/parithi.html

11th Computer Science_Chapter-18 Telugu.indd 297 24-03-2023 16:51:49

ప్రాజెక్ట్ మధురై	http://www.projectmadurai.org/pmworks.html
పురాతన పుస్తకాలు మరియు రాతపత్రులు	http://www.tamilheritage.org/old/text/ebook/ebook.html
నూలగమ్	http://www.noolaham.org/wiki/index.php/
అన్నా శతాబ్ధి గ్రంథాలయం	http://www.annacentenarylibrary.org/

18.6 ఇ – తమిళ ముద్రణ మరియు ఇంటర్ఫేస్ సాఫ్ట్వేర్ (Tamil Typing and Interface software)

వర్డ్ప్రాసెసర్లలో పత్రాలను ముద్రణ చేయడానికి మరియు ఇంటర్నెట్లో సమాచారాన్ని శోధించడానికి తమిళం ఎక్కువగా ఉపయోగపడుతుంది. తమిళ ఇంటర్ఫేస్ సాఫ్ట్వేర్ను ఉపయోగించి తమిళాన్ని ముద్రించడం అనేది ముద్రణ చేసే వివిధ పద్ధతులలో సుపరిచితం. కంప్యూటర్ మరియు ఇతర స్మార్ట్ఫోన్లలో తమిళాన్ని ముద్రణ చేసే సరళమైన పద్ధతి ఇది.

18.6.1 సుపరిచితమైన తమిళ కీబోర్డ్ ఇంటర్ఫేస్ (Familiar Tamil Keyboard Interface):

- NHM రచయిత ఇ – కలప్పై మరియు లిప్పికార్ అనునది తమిళ కీబోర్డులో పేరొందిన ఇంటర్ఫేస్ల సాఫ్ట్వేర్. ఇది తమిళ టైపింగ్ కోసం ఉపయోగించబడుతుంది. ఇది ఫొనెటిక్స్ ఉపయోగించి తమిళ యూనికోడ్గ పనిచేస్తుంది.
- సెల్లినం మరియు పోన్మడల్ సుపరిచితమైన తమిళ కీబోర్డ్ లే అవుట్లు. ఇవి ఫోనెటిక్స్ను ఉపయోగించి స్మార్ట్ ఫోన్లో ఆండ్రో ఆపరేటింగ్ సిస్టమ్లో పనిచేస్తాయి.

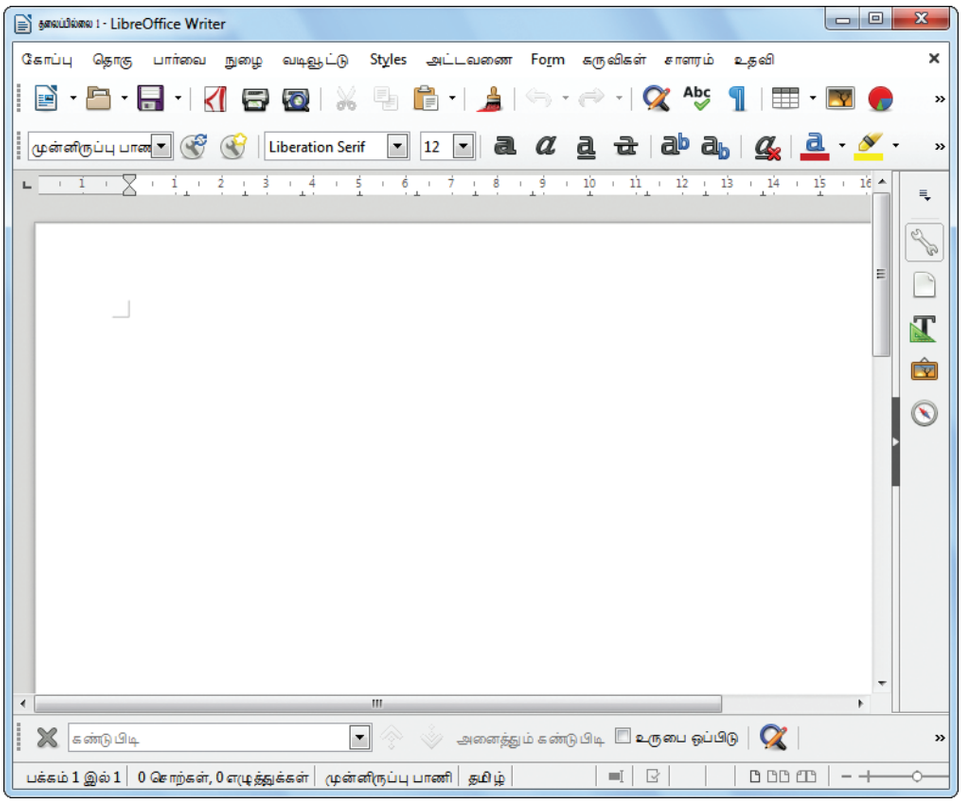


పటము 18.4 ఇ – కలప్పై ఓపనింగ్ స్క్రీన్

18.7 తమిళ కార్యాలయ ఆటోమేషన్ అనువర్తనాలు (Tamil Office Automation Applications)

మైక్రోసాఫ్ట్ ఆఫీస్, ఓపెన్ ఆఫీస్ అనునవి ప్రసిద్ధ ఆఫీస్ ఆటోమేషన్ సాఫ్ట్వేర్లు. ఇవి పూర్తి తమిళ ఇంటర్ఫేస్ సౌకర్యాన్ని అందిస్తుంది. ఈ సాఫ్ట్వేర్లు డౌన్లోడ్ చేయబడి, కంప్యూటర్లలో ఇన్స్టాల్ చేయబడతాయి. (Installation) తరువాత మీ ఆఫీస్ ఆటోమేషన్ సాఫ్ట్వేర్ వాతావరణం పూర్తిగా తమిళంలో మారుతుంది. మెనూబార్లు, చిహ్నాల పేర్లు, డైలాగ్ బాక్స్లు తమిళంలో చూపబడతాయి. అంతేకాక మీరు తమిళ పేర్లతో పైళ్ళను సేవ్ చేయవచ్చు మరియు తమిళ పేర్లతో ఫోల్డర్లను సృష్టించవచ్చు.

11th Computer Science_Chapter-18 Telugu.indd 298 24-03-2023 16:51:49



படము 18.5 తమిళంలో లిబ్ర కార్యాలయ రచయిత పరిసరాలు

తమిళ లిబ్ర ఆఫీస్ కాకుండా, తమిళ ఓపెన్ అజాగి యూనికోడ్ ఎడిటర్, పొన్మోజి మెంథామిజ్ కంబన్ వాని వంటివి తమిళం కోసం ప్రత్యేకంగా పనిచేసే కార్యాలయ ఆటోమేషన్ సాఫ్ట్‌వేర్ మీరు పూర్తిగా తమిళంలో పనిచేయడానికి ఈ అనువర్తనాలను ఉపయోగించవచ్చు.

18.8 తమిళ అనువాద అనువర్తనాలు (Tamil Translation Applications)

తమిళ్ పొరి [Thamizpori (தமிழ்பொறி)] అనేది ఆంగ్ల పదాలకు సమానమైన 30000 కంటే ఎక్కువ తమిళ పదాలను కలిగి వున్న తమిళ అనువాద అనువర్తనం. ఈ అనువర్తనాన్ని ఉపయోగించి మనము చిన్న ఆంగ్ల వాక్యాలను తమిళంలోకి అనువదించవచ్చు. గూగుల్ ఆన్‌లైన్ అనువాద సదుపాయాన్ని కూడా ఇస్తుంది. ఈ ఆన్‌లైన్ సదుపాయాన్ని ఉపయోగించి మనం తమిళం నుండి మరే ఇతర భాషకు అనువదించవచ్చు.

18.9 తమిళ్ సాఫ్ట్‌వేర్ ప్రోగ్రామ్ (Tamil Programming Language)

కంప్యూటర్లు మరియు స్మార్ట్‌ఫోన్లలో అభివృద్ధి చెందిన సాఫ్ట్‌వేర్ ప్రోగ్రామ్ లాంగ్వేజ్ ఆంగ్లంలో మాత్రమే అందుబాటులో ఉన్నది. ఇప్పుడు, తమిళంలో (programming languages) అభివృద్ధి చేయడానికి ప్రయత్నాలు జరుగుతున్నాయి. పెథాన్ ప్రోగ్రామింగ్ లాంగ్వేజ్‌ను ఆదారంగా చేసుకొని మొదటి తమిళ్ సాఫ్ట్‌వేర్ ప్రోగ్రామ్ రూపొందించబడింది. [Ezhil (எழில்)] సహాయంతో మీరు తమిళంలో సాధారణ ప్రోగ్రామ్‌ను వ్రాయవచ్చు.

18.10 తమిళ సమాచార మార్పిడి కోడింగ్ వ్యవస్థలు (Tamil Information Interchange Coding Systems)

11th Computer Science_Chapter-18 Telugu.indd 299 24-03-2023 16:51:49

TSCII (Tamil Script Code for Information Interchange)

కంప్యూటర్లు డేటా మరియు సమాచారాన్ని బైనరీ వ్యవస్థగా నిర్వహిస్తాయి. కంప్యూటర్‌కు ఇచ్చే ప్రతి సమాచారము బైనరీగా మార్చుతుంది. మీరు ఈ పాఠ్య పుస్తకం యొక్క మొదటి అధ్యాయంలో ఈ విషయాల గురించి తెలుసుకొన్నారు. కంప్యూటర్లు డేటా మరియు సమాచారాన్ని నిర్వహించడానికి ASCII వ్యవస్థను ఉపయోగిస్తాయి. ASCII ఎన్కోడింగ్ వ్యవస్థ ఆంగ్లభాషను నిర్వహించడానికి మాత్రమే వర్తిస్తుంది. అందువల్ల ఆంగ్లేతర కంప్యూటర్లతో సహా ఎలక్ట్రానిక్ పరికరాల్లో సులభంగా నిర్వహించబడే ఎన్కోడింగ్ పథకం యొక్క విశ్లేషణలో మన తమిళ భాషను నిర్వహించే మొదటి కోడింగ్ వ్యవస్థ TSCII (Tamil Script Code for Information Interchange) ఈ ఎన్కోడింగ్ పథకం (IANA & ICANN) యొక్క యూనిట్‌లో నమోదు చేయబడింది.

ISCII (Indian Script Code for Information Interchange)

తమిళం సహా ఇతర భారతీయ భాషల కోసం ప్రత్యేకంగా రూపొందించిన ఎన్‌కోడింగ్ పథకాల్లో ఇది ఒకటి. ఇది యూనికోడ్‌తో ఏకీకృతం చేయబడింది.

యూనికోడ్ (Unicode)

యూనికోడ్ అనేది ఎన్కోడింగ్ వ్యవస్థ. ఇది తమిళంతో సహా వివిధ ప్రపంచ భాషలను నిర్వహించడానికి రూపొందించబడింది. దీని మొదటి వెర్షన్ 1.0.0 అక్టోబర్ 1991 లో ప్రవేశపెట్టబడింది. యూనికోడ్ ప్రవేశపెట్టినప్పుడు ఇది తమిళంలో సహా దాదాపు 23 భాషలను నిర్వహించగలదు. వివిధ ఎన్కోడింగ్ పథకాలలో యూనికోడ్ తమిళాన్ని నిర్వహించడానికి ఉత్తమమైనది.

18.11 తమిళ ఆపరేటింగ్ వ్యవస్థ (Tamil Operating System)

కంప్యూటర్ మరియు స్మార్ట్‌ఫోన్ వంటి ఎలక్ట్రానిక్ సిస్టమ్‌లను యాక్సెస్ చేయడానికి ఆపరేటింగ్ వ్యవస్థ అవసరం. మైక్రోసాఫ్ట్ విండోస్ వ్యక్తిగత కంప్యూటర్లకు చాలా ప్రాచుర్యం పొందిన ఆపరేటింగ్ వ్యవస్థ.లైనక్స్ అనునది మరొక ప్రసిద్ధ ఓపెన్ సోర్స్ ఆపరేటింగ్ వ్యవస్థ. కంప్యూటర్‌ను సులభంగా యాక్సెస్ చేయడానికి ఆపరేటింగ్ సిస్టమ్ యొక్క వాతావరణం పనిచేయడానికి ఆపరేటింగ్ సిస్టమ్‌ను ఉపయోగిస్తారు. మరియు అర్థం చేసుకోవడానికి ఇది సులభమైన రూపం. అందువల్ల కంప్యూటర్లు మరియు స్మార్ట్‌ఫోన్‌లలో ఉపయోగించే అన్ని ఆపరేటింగ్ సిస్టమ్ తమిళ వాతావరణంలో అందించబడుతుంది.

విండోస్ తమిళ ఎన్విరాన్మెంట్ ఇంటర్ఫేసును ఇంటర్నెట్ నుండి డౌన్‌లోడ్ చేసి ఇన్‌స్టాల్ చేయాలి. ఇది టాస్క్ బార్ డెస్క్‌టాప్ ఎలిమెంట్స్, ఐకాన్ల పేర్లు మొదలగు ఎలిమెంట్లను తమిళంలో ప్రదర్శిస్తాయి.

18.12 తమిళాన్ని అభివృద్ధి చేయడానికి గల సంస్థలు మరియు ప్రాజెక్టులు (Organisation and projects to develop Tamil)

తమిళ వర్చువల్ అకాడమీ (Tamil Virtual Academy):

ఇంటర్నెట్ ద్వారా తమిళాన్ని ప్రపంచానికి వ్యాప్తి చేయాలనే లక్ష్యంతో ఫిబ్రవరి 17, 2001 న తమిళనాడు ప్రభుత్వం వర్చువల్ విశ్వవిద్యాలయాన్ని స్థాపించింది. ఇప్పుడు ఈ సంస్థ "తమిళ వర్చువల్ అకాడమీ" పేరుతో పనిచేస్తుంది. ఇది కిండర్ గార్డెన్ నుండి గ్రాడ్యుయేషన్ స్థాయి వరకు తమిళ భాష సంస్కృతికి సంబంధించి వివిధ కోర్సులను అందిస్తుంది.

11th Computer Science_Chapter-18 Telugu.indd 300 24-03-2023 16:51:49

Website: http://www.tamilvu.org/index.php

తమిళ భాషా మండలి సింగపూర్ (Tamil Language Council, Singapore)

సింగపూర్ వాసులలో తమిళం యొక్క అవగాహన మరియు వాడకాన్ని ప్రోత్సహించే లక్ష్యాలతో 2001 లో సింగపూర్ ప్రభుత్వం సమాచార మరియు కళల మంత్రిత్వ శాఖ, కౌన్సిల్ ఆఫ్ తమిళ భాషా మండలిని ఏర్పాటు చేసింది. ఈ కౌన్సిల్‌ను (வளர்தமிழ் இயக்கம்) అని పిలుస్తారు.



Website: http://tamil.org.sg/ta

మధురై ప్రాజెక్ట్ (Madurai Project):

మధురై ప్రాజెక్ట్ పురాతన తమిళ సాహిత్యం శాస్త్రీయం యొక్క ఉచిత ఎలక్ట్రానిక్ సంచికలను సేకరించి ప్రచురించడానికి చేయబడిన బహిరంగ మరియు స్వచ్ఛంద ప్రయత్నం. దీని అర్థం పాత పుస్తకాలను టైప్ చేయడం లేదా స్కాన్ చేయడం మరియు ప్రసిద్ధి చెందిన కంప్యూటర్ ప్లాట్‌ఫామ్‌లో ఉన్నటువంటి టెక్స్ట్‌లను స్వీకరించడం.

TSCII ఎన్కోడింగ్ ప్రకారం 1998 లో విడుదలైన మధురై (E-Texts Project) తమిళ లిపి రూపంలో ఉన్నాయి. 2004 నుండి మేము తమిళ యూనికోడ్‌లో కూడా (E-Texts) విడుదల చేయడం ప్రారంభించాము.

వెబ్‌సైట్: http://www.projectmadurai.org/

తమిళ వికీపీడియా(Tamil Wikipedia):

వికీపీడియా అనేది (Open Source Encyclopedia) ఇక్కడ ఏ వ్యక్తి అయినా ఏదైనా విషయం గురించి వ్యాసం వ్రాయవచ్చు. (Tamil) వికీపీడియాలో లక్షకు పైగా వ్యాసాలు ఉన్నాయి.

Web Site: https://ta.wikipedia.org/

తమిళాన్ని సజీవ భాషగా మార్చడానికి టెక్నాలజీ అభివృద్ధిలో తమిళాన్ని చురుకుగా ఉపయోగించడం ప్రతి తమిళుడి కర్తవ్యం.

11th Computer Science_Chapter-18 Telugu.indd 301 24-03-2023 16:51:49

గుర్తుంచుకోవలసిన విషయములు

- 2016 చివరి నాటికి భారతదేశంలో ఎక్కువగా ఉపయోగించే ప్రాంతీయ భాషల జాబితాలో తమిళం అగ్రస్థానంలో ఉంది. 42% మంది తమిళ భాషలో ఇంటర్నెట్ ఉపయోగిస్తున్నారు.
- గూగుల్ మరియు బింగ్ తమిళంలో శోధన సదుపాయాలను కల్పిస్తాయి.
- ఇంటర్నెట్ ద్వారా ప్రభుత్వ సేవలు పొందడంను ఇ – గవర్నెన్స్ అంటారు.
- తమిళ ఇ – లైబ్రరీ సేవలు వేలాది తమిళ పుస్తకాలను ఉచితంగా ఇ – పుస్తకాలుగా అందిస్తాయి.
- Thamizpori (தமிழ்பொறி) తమిళ్‌పోరి అనేది ఆంగ్ల పదాలకు సమానమైన 30000 కంటే ఎక్కువ తమిళ పదాలను కలిగి వున్న తమిళ అనువాద అనువర్తనం.
- [Ezhil (எழில்)] అనునది తమిళంలోని మొదటి ప్రోగ్రామింగ్ భాష
- యూనికోడ్ అనేది ఎన్కోడింగ్ వ్యవస్థ. ఇది తమిళంతో సహా వివిధ ప్రపంచ భాషలను నిర్వహించడానికి రూపొందించబడింది.
- విండోస్ తమిళ ఎన్విరాన్మెంట్ ఇంటర్ఫేసును ఇంటర్నెట్ నుండి డౌన్‌లోడ్ చేసి ఇన్‌స్టాల్ చేయాలి.

మూల్యాంకనం

క్రింది లఘు ప్రశ్నలకు జవాబులు వ్రాయండి

1. తమిళ భాష మద్ధతు వున్న సెర్చ్ ఇంజన్లను జాబితా పరచండి.
2. Android లో ఉపయోగించే కీబోర్డ్ లే అవుట్లు ఏమిటి?
3. తమిళ ప్రోగ్రామింగ్ భాష గురించి క్లుప్తముగా వ్రాయండి.
4. TSCII అనగానేమి?
5. తమిళ వర్చువల్ అకాడమీ గురించి క్లుప్తంగా వ్రాయండి.

WZ4U0

****

11th Computer Science_Chapter-18 Telugu.indd 302 24-03-2023 16:51:49

పదకోశం

పదాలు	విషయం
Paradigm	ప్రోగ్రాం యొక్క వ్యవస్థాపక సూత్రాలు
Abstraction	నేపథ్య వివరాలు లేకుండా అవసరమైన సమాచారం / అంశాలను అందించుటయే అబ్స్ట్రాక్షన్.
Modularity	కొన్ని ఫంక్షనల్ యూనిట్లు (నేమ్డ్ మాడ్యూల్స్)గా విభజించెడి విధంగా సిస్టంను రూపొందించుట. ఇది ఒక లార్జర్ అప్లికేషన్‌గా కంపోజ్ చేయబడును.
Base class	క్రొత్తగా సృష్టించబడిన న్యూ క్లాసుల సంక్రమణ, ధర్మాలను పొందునది 'క్లాస్' దీనికి "పేరెంట్ క్లాస్" అని కూడా పేరు.
Derived class	బేస్‌క్లాస్ యొక్క సంక్రమణ ధర్మాలను ప్రదర్శించెడి క్లాసును డిరైవుడ్ క్లాస్ అంటారు. దీనికి చైల్డ్ క్లాస్ లేదా సబ్‌క్లాస్ అంటారు.
Class	సమానధర్మాలను కల్గిన ఒకేరకమైన ఆబ్జెక్ట్‌ల సమూహాన్ని క్లాస్ అంటారు.
Object	గుర్తించిన కొన్ని లక్షణాలను, ప్రవర్తనను ప్రదర్శించునది ఆబ్జెక్ట్
Encapsulation	ఒక సింగల్ యూనిట్ నందు డేటా మరియు ఫంక్షన్లను పరస్పరం అందించుకొనెడి వ్యవస్థయే ఎన్‌కాప్సులేషన్.
Inheritance	మూలాధార లేదా బేస్ క్లాసుల నుండి డిరైవుడ్ క్లాసులను సృజించెడి ప్రక్రియ.
Signature	ఆర్గుమెంట్‌ల సంఖ్య మరియు రకం
Polymorphism	బహుళ నమూనా రూపాలు
Default argument	ఒక విలువతో ఆర్గుమెంట్‌ను ప్రారంభించడం
Base Class:	మరొక క్లాసును సంక్రమింపబేయు క్లాసు. (సూపర్ క్లాస్ లేదా పేరెంట్ క్లాస్ అని కూడా పేరు)
Derived Class:	మరో క్లాసు యొక్క సంక్రమణ ధర్మాలను ప్రదర్శించునది (సబ్‌క్లాస్)
Inheritance	ఒక క్లాసు యొక్క ధర్మాలను మరో క్లాసుకు అందించుట.
Inheritance Hierarchy (Inheritance Path):	ఒక బేస్ క్లాస్ మరియు డిరైవుడ్ క్లాస్‌ల మధ్య సంబంధాలను చిత్రీకరించెడి చక్రము. (డిరైవేషన్ హిమరార్కి అని కూడా పేరు)
Visibility mode	ఒక క్లాసులోని ఒక మెంబరును ప్రదర్శించెడి, నిలిపి ఉంచి పర్యవేక్షించెడి పబ్లిక్, ప్రైవేట్ లేదా ప్రొటెక్టడ్ స్పెషిఫయ్యర్.
Vulnerability	డాడి, విధ్వంసం చేయగల అవకాశం.

11th Computer Science_Chapter-18 Telugu.indd 303 24-03-2023 16:51:49

Ethics	నైతికత అనేది ఒక వ్యక్తి యొక్క ప్రవర్తనను నియంత్రించే నైతికసూత్రం
Cyber	కంప్యూటర్ సంస్కృతి సమాచార సాంకేతిక మరియు వర్చువల్ రియాలిటీ యొక్క స్వభావం
Computer Crime	కంప్యూటర్ నేరం యొక్క సైబర్ నేరం
Authenticity	నిజమైన లేదా నమ్మదగిన నాణ్యత
Sabotage	నాశనం ప్రమాదం లేదా ఆటంకం సృష్టించుట.
Perpetrator	హానికరమైన చట్ట విరుద్ధమైన లేదా అనైతిక చర్యకు పాల్పడేవాడు
Software Piracy	ఒక వ్యక్తి రూపొందించిన సాఫ్ట్‌వేర్‌ను మరోవ్యక్తి అనధికారికంగా ఉపయోగించి, కాపీరైట్ హక్కులను ఉల్లంఘించుటయే సాఫ్ట్‌వేర్ ప్రైవసీ.
Hacking	యజమాని అనుమతి లేకుండా సిస్టమ్ ద్వారా అనధికార ప్రాప్యత
Cracking	వినియోగదారులకు సేవను చట్టబద్ధంగా తిరస్కరించడానికి మరియు కాపీ చేసిన సురక్షిత ప్రోగ్రామ్‌ను సృష్టించడానికి కోడ్‌ను దొంగలించడం
Malicious	నిజాయితీ లేని, మోసం, అవినీతి, క్రిమినల్, చట్ట విరుద్ధం
Freeware	ఫ్రీవేర్ అనేది ఉచితంగా లభించే సాఫ్ట్‌వేర్
Shareware	షేర్‌వేర్ అనేది ఒక సాఫ్ట్‌వేర్ ఇది పరిమిత సమయం వరకు ట్రయల్ ప్రాతిపదికను ఉచితంగా పంపిణీ చేయబడుతుంది.
Phishing	పిషింగ్ అనేది హానికరమైన వ్యక్తి లేదా ఇ మెయిల్‌లను పంపడం ద్వారా వినియోగదారులను స్కామ్ చేసే వ్యక్తుల సమూహాన్ని వివరించడానికి ఉపయోగించే పదం, సేకరించడానికి రూపొందించబడిన వెబ్ పేజీలను మరియు వ్యక్తి యొక్క ఆన్‌లైన్ బ్యాంక్, క్రెడిట్ కార్డ్ లేదా ఇతర లాగిన్ సమాచారాన్ని సృష్టిస్తుంది.
Fraudulent	అనైతికత్వం, మోసం, ద్రోహం, అవనీతి, కుట్ర, చట్టవిరుద్ధం.
Anonymous	పేరులేని, గుర్తించబడని పేర్కొనబడని
Cookies	కుకీలు మీరు ఇంటర్నెట్ సైట్‌లను సందర్శించినప్పుడు వెబ్ సర్వర్‌లు మీ వెబ్ బ్రౌజర్‌కు పంపే సందేశాలు
Immune	ఒక నిర్దిష్ట ఇన్ఫెక్షన్ లేదా టాక్సిన్ను నిరోధకత
Firewall	ఫైర్‌వాల్ అనేది నెట్‌వర్క్ భద్రతా వ్యవస్థ, ఇది ముందుగా నిర్ణయించిన భద్రతా నియమాల ఆధారంగా ఇన్‌కమింగ్ మరియు అవుట్‌గోయింగ్ నెట్‌వర్క్ ట్రాఫికన్ పర్యవేక్షిస్తుంది. మరియు నియంత్రిస్తుంది.
Proxy server	ప్రాక్సీ సర్వర్ అనేది స్థానిక నెట్‌వర్క్ మరియు ఇంటర్నెట్ వంటి పెద్ద స్థాయి నెట్‌వర్క్ మధ్య గేట్‌వే, ప్రాక్సీ సర్వర్లు పెరిగిన పనితీరు మరియు భద్రతను అందిస్తాయి.
Encryption	ఒక మేసేజ్ లేదా సమాచారాన్ని ఇతరులకు అర్థంకాని సంకేతరీతిలో భద్రపరచుట. అధికారిక సభ్యుడు మాత్రమే దీనిని వివరించగలరు.
Decryption	సంకేత భాషలో భద్రపరచిన (ఎన్‌క్రిప్షన్) సమాచారాన్ని సాధారణ అక్షర రూపంలోకి మార్చి తిరిగి మార్చుటయే డిక్రిప్షన్

11th Computer Science_Chapter-18 Telugu.indd 304 24-03-2023 16:51:49

పరిచయము:

1. C++ నుండి పది ప్రయోగాలు ప్రాక్టికల్ తరగతుల్లో సాధన చేయబడతాయి.

2. అంతర్గత ఎంపికతో C++ నుండి ఒక ప్రశ్న

3. మార్కుల పంపిణీ

I. ఇంటర్నల్ అసెస్‌మెంటు	5 మార్కులు
రికార్డ్ బుక్	5 మార్కులు

II. ఎక్స్‌టర్నల్ అసెస్‌మెంట్:	15 మార్కులు
(a) C++ ప్రోగ్రామ్ కోడింగ్	10 మార్కులు
(b) నిర్వహించడం & అవుట్‌పుట్	5 మార్కులు
మొత్తం	**20 మార్కులు**

Computer Science Practical Page 298-321.indd 305 24-03-2023 16:51:56

శీర్షిక

వ. సంఖ్య	ప్రశ్న సంఖ్య	ప్రోగ్రాం పేరు	పేజీ నంబరు
1	CS1	మొత్తం జీతం లెక్కింపు	307
2	CS2	శాతం	308
3	CS3	పాలిన్‌డ్రోమ్	309
4	CS4	సంఖ్య సంభాషణ	310
5	CS5	ఫైబొనాక్సీ ప్రైమ్ సిరీస్	312
6	CS6	శ్రేణిలోని అంశాలను ప్రవేశపెట్టడం/ తొలగించడం	314
7	CS7	మాత్రిక యొక్క సరిహద్దు మూలకం	317
8	CS8	ABC ప్రచురణ కర్తలు	319
9	CS9	క్లాస్ ఉపయోగించి ఉద్యోగి వివరాలు	323
10	CS10	విద్యార్థుల వివరాలు	326

Computer Science Practical Page 298-321.indd 306 24-03-2023 16:51:57

CS-1 **ఉద్యోగి యొక్క ప్రాథమిక జీతం ఇన్పుట్ చేయడానికి C++ ప్రోగ్రామ్ వ్రాయండి మరియు క్రింది ప్రకారం దాని స్థూల జీతం లెక్కించండి:**

Basic Salary <25000 : HRA = 20%, DA = 80%
Basic Salary >= 25000 : HRA = 25%, DA = 90%
Basic Salary >= 40000 : HRA = 30%, DA = 95%

కోడింగ్

```
#include <iostream>
using namespace std;
int main()
{
float basic, gross, da,hra;
cout<<"Enter basic salary of an employee: ";
cin>>basic;
if (basic <25000)
{
da = basic *80/100;
hra= basic *20/100;
}
else if (basic >=25000 && basic<40000)
{
da = basic *90/100;
hra= basic *25/100;
}
else if (basic>=40000)
{
da = basic *95/100;
hra= basic *30/100;
}
gross= basic +hra+ da;
cout<< "\n\t Basic Pay ....."<< basic<<endl;
cout<< "\t Dearness Allowance ....." << da <<endl;
cout<< "\t House Rent Allowance....."<< hra <<endl;
  cout<< "\t ----------------------------------------"<<endl;
  cout<< "\t Gross Salary....."<<gross <<endl;
```


Computer Science Practical Page 298-321.indd 307 24-03-2023 16:51:57

```
cout<< "\t ----------------------------------------"<<endl;
return 0;
}
```

అవుట్‌పుట్

```
Enter basic salary of an employee: 25000
                Basic Pay       : 25000
        Dearness Allowance      : 22500
    House Rent Allowance        :   6250
                        ------------------------
            Gross Salary        : 53750
                        ------------------------
```

CS2 - శాతం

CS-2 **ఒక విద్యార్థి పొందిన మార్కులశాతం మరియు అతను పొందిన ఉత్తీర్ణతా స్థాయిలు ప్రదర్శించు విధముగా C++ ప్రోగ్రామ్ వ్రాయుము(distinction, first, second, third or fail)**

Percentage	Division
>=80	Distinction
>=60 and <80	First division
>=50 and <60	Second Division
>=40 and <50	Third Division
<40	Fail

కోడింగ్

```
#include <iostream>
using namespace std;
int main()
{
float percent;
int x;
cout<<"Enter your percentage: ";
cin>>percent;
cout<<"You scored "<<percent<<"%"<<endl;
   x = percent/10;
switch (x)
{
case 10:
case 9:
```


Computer Science Practical Page 298-321.indd 308 24-03-2023 16:51:57

```
case 8:
cout<<"You have passed with Distinction";
break;
case 7:
case 6:
cout<<"You have passed with First division";
break;
case 5:
cout<<"You have passed with Second division";
break;
case 4:
cout<<"You have passed with Third division";
break;
default:
cout<<"Sorry: You have failed";
}
return 0;
}
```

అవుట్‌పుట్ 1

```
Enter your percentage: 79
You scored 79%
You have passed with First division
```

అవుట్‌పుట్ 2

```
Enter your percentage: 39
You scored 39%
Sorry: You have failed
```

CS3 - PALINDROME

CS-3 **Write a C++ program to enter any number and check whether the number is palindrome or not using while loop.**

కోడింగ్

```
#include <iostream>
using namespace std;
int main()
{
int n,num, digit, rev =0;
cout<<"Enter a positive number: ";
```


Computer Science Practical Page 298-321.indd 309 24-03-2023 16:51:57

```
cin>>num;
    n =num;
while (num)
{
digit=num%10;
rev=(rev *10)+ digit;
num=num/10;
}
cout<<" The reverse of the number is: "<< rev <<endl;
if (n == rev)
cout<<" The number is a palindrome";
else
cout<<" The number is not a palindrome";
return 0;
}
```

అవుట్‌పుట్ 1

```
Enter a positive number to reverse: 1234
 The reverse of the number is: 4321
 The number is not a palindrome
```

అవుట్‌పుట్ 2

```
Enter a positive number to reverse: 1221
The reverse of the number is: 1221
The number is a palindrome
```

CS4 - సంఖ్య సంభాషణ

CS-4 **do while లూప్ ఉపయోగించి క్రింది మెను ఆధారిత C++ ప్రోగ్రామ్‌ను వ్రాయండి**

1. దశాంశాన్ని బైనరీ సంఖ్యకు మార్చండి
2. బైనరీ సంఖ్యను దశాంశంగా మార్చండి
3. ఎగ్జిట్

ఎంపికను బట్టి విలువను అంగీకరించి ఫలితాన్ని ప్రదర్శిస్తుంది. మూడవ ఎంపికను వినియోగదారు ఎంచుకునే వరకు ప్రోగ్రామ్ కొనసాగాలి.

కోడింగ్

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
```


Computer Science Practical Page 298-321.indd 310 24-03-2023 16:51:57

```
    int dec,d,i,temp,ch;
    long int bin;
do
{
    dec=bin=d=i=0;
    cout<<"\n\n\t\tMENU\n1. Decimal to Binary number\n2.Binary to Decimal number\
    n3.Exit\n";
    cout <<"Enter your choice(1/2/3)";
    cin>>ch;
    switch (ch)
    {
            case 1:  cout << "Enter a decimal number: ";  cin >> dec;
                    temp=dec;
    while (dec!=0)
    {
    d = dec%2;
    bin += d * pow(10,i);
    dec /= 2;
    i++;
}
    cout << temp << " in decimal = " << bin << " in binary" << endl ;break;
    case 2: cout << "Enter a binary number: ";    cin >> bin;
            temp=bin;
            while (bin!=0)
    {
    d = bin%10;
    dec += d*pow(2,i);
    bin /= 10;
    i++;
}
    cout << temp << " in binary = " <<dec << " in decimal";
    break;
    case 3: break;
    default : cout<<"Invalid choice";
 }
} while (ch!=3);
return 0;
}
```

అవుట్‌పుట్ 1

```
    MENU
1.Decimal to Binary number
```


Computer Science Practical Page 298-321.indd 311 24-03-2023 16:51:57

2.Binary to Decimal number
3.Exit
Enter your choice(1/2/3)1
Enter a decimal number: 23
23 in decimal = 10111 in binary
MENU
1.Decimal to Binary number
2.Binary to Decimal number
3.Exit
Enter your choice(1/2/3)2
Enter a binary number: 11001
11001 in binary = 25 in decimal
MENU
1.Decimal to Binary number
2.Binary to Decimal number
3.Exit
Enter your choice(1/2/3)3

అవుట్‌పుట్ 2

MENU
1.Decimal to Binary number
2.Binary to Decimal number
3.Exit
Enter your choice(1/2/3)4
Invalid choice
MENU
1.Decimal to Binary number
2.Binary to Decimal number
3.Exit
Enter your choice(1/2/3)3

CS5 - ఫైబొనాక్సీ ప్రైమ్ సిరిస్

CS-5 **n టర్మ్స్ వరకు ఫైబొనాక్సీ సిరీస్‌ను రూపొందించెడి యూజర్ డిఫైన్డ్ ఫంక్షన్‌కు ఒక C++ ప్రోగ్రాం వ్రాయుము. ఇవి prime లేదా Composite అనుటకు నిర్ధారించుటకు ప్రింట్ చేయుము.**

కోడింగ్

```
#include <iostream>
#include <stdlib.h>
```


Computer Science Practical Page 298-321.indd 312 24-03-2023 16:51:57

```
using namespace std;
void Primechk (int a )
{ int j;
    if ( a == 0 || a == 1 )
   { cout<< "  NEITHER PRIME NOR COMPOSITE ";}
    else
{
   for (j = 2 ; j<a; j++)
   {         if (a%j==0)
   { cout<< "\tCOMPOSITE" ;
   break ;
}
   }
    if ( a==j )
    cout<< "\tPRIME" ;
   }
}
   void  fibo ( int n )
   { int a = -1 , b = 1 ,c=0 ;
   for ( int i = 1 ; i <= n ; i++)
   {
            cout<<endl;
   c = a + b ;
   cout<<c;
   Primechk(c);
   a = b;
   b = c ;
    }
   }
 int main ()
 {
   int n ;
   cout << " ENTER THE NUMBER OF REQUIRED FIBO TERMS " ;
   cin >> n ;
   cout<< "\n\tFIBONACCI SERIES\n " ;
   fibo (n) ;
return 0;
  }
```


Computer Science Practical Page 298-321.indd 313 24-03-2023 16:51:57

అవుట్‌పుట్

```
ENTER THE NUMBER OF TERMS 10
   FIBONACCI SERIES
0  NEITHER PRIME NOR COMPOSITE
1  NEITHER PRIME NOR COMPOSITE
1  NEITHER PRIME NOR COMPOSITE
2       PRIME
3       PRIME
5       PRIME
8       COMPOSITE
13      PRIME
21      COMPOSITE
34      COMPOSITE
```

CS6 - శ్రేణిలోని అంశాలను ప్రవేశపెట్టుట / తొలగించడం

CS-6 పూర్ణాంకాల యొక్క ఒకే డైమెన్షన్ శ్రేణిలో మూలకాలను చొప్పించడానికి మరియు తొలగించడానికి మెను డైమెన్ C++ ప్రోగ్రామ్‌ను వ్రాసి చొప్పించడం లేదా తొలగించిన తర్వాత శ్రేణిని మద్రించండి.

కోడింగ్

```
#include<iostream>
using namespace std;
int a[20],b[20],c[40];
int m,n,p,val,i,j,key,pos,temp;
/*Function Prototype*/
void display();
void insert();
void del();
int main()
{
int choice;
cout<<"\nEnter the size of the array elements:\t";
cin>>n;
cout<<"\nEnter the elements for the array:\n";
for (i=0;i<n;i++)
```


Computer Science Practical Page 298-321.indd 314 24-03-2023 16:51:57

```
{
cin>>a[i];
}
do {
cout<<"\n\n--------Menu----------\n";
cout<<"1.Insert\n";
cout<<"2.Delete\n";
cout<<"3.Exit\n";
cout<<"----------------------";
cout<<"\nEnter your choice:\t";
cin>>choice;
switch (choice)
{
   case 1: insert();
   break;
   case 2: del();
   break;
   case 3:break;
   default :cout<<"\nInvalid choice:\n";
}
} while (choice!=3);
return 0;
}
void display()//displaying an array elements
{
   int i;
   cout<<"\nThe array elements are:\n";
   for(i=0;i<n;i++)
{
   cout<<a[i]<<" ";
}
}//end of display()
   void insert()//inserting an element in to an array
{
   cout<<"\nEnter the position for the new element:\t";
   cin>>pos;
```


Computer Science Practical Page 298-321.indd 315 24-03-2023 16:51:57

```
    cout<<"\nEnter the element to be inserted :\t";
    cin>>val;
    for (i=n; i>=pos-1; i--)
{
    a[i+1]=a[i];
}
a[pos-1]=val;
       n=n+1;
display();
}//end of insert()
void del()//deleting an array element
{
    cout<<"\n Enter the position of the element to be deleted:\t";
    cin>> pos;
    val= a [pos];
    for (i= pos;i<n-1;i++)
{
    a[i]=a[i+1];
}
    n=n-1;
cout<<"\nThe deleted element is = "<<val;
display();
}//end of delete()
```

అవుట్‌పుట్

```
Enter the size of the array elements:   5
Enter the elements for the array:
1
2
3
4
5
--------Menu-----------
1.Insert
2.Delete
3.Exit
----------------------
```


Computer Science Practical Page 298-321.indd 316 24-03-2023 16:51:57

```
Enter your choice:     1
Enter the position for the new element: 3
Enter the element to be inserted :     26
The array elements are:
1  2  26  3  4  5
--------Menu-----------
1.Insert
2.Delete
3.Exit
----------------------
Enter your choice:     2
Enter the position of the element to be deleted:      2
The deleted element is = 2
The array elements are:
1 3 26 4 5
--------Menu-----------
1.Insert
2.Delete
3.Exit
-------------------------
Enter your choice:     3
-------------------------
```


CS-7 మాత్రిక యొక్క సరిహద్దును ముద్రించడానికి C++ ప్రోగ్రామ్‌ను వ్రాయండి

కోడింగ్

```
#include <iostream>
using namespace std;
void printBoundary (int a[][10], int m, int n)
{
for (int i = 0; i < m; i++) {
for (int j = 0; j < n; j++)
{
if (i==0|| j==0||i==m-1||j==n-1)
```


Computer Science Practical Page 298-321.indd 317 24-03-2023 16:51:57

```
cout<<a[i][j]<<" ";
else
cout<<"  ";
}
cout <<endl ;
}
}
// Driver code
int main()
{
   int a[10][10] ,i,j,m,n;
   cout<<"Enter more than 3 number of rows and columns"<<endl;
   cin>>m>>n;
            for (i=0;i<m;i++)
   {
            for (j=0;j<n;j++)
   {
            cout<<"enter the value for array["<<i+1<<"]"<<"["<<j+1<<"] :";
            cin>>a[i][j];
    }
}
system("cls");
   cout<<"\n\nOriginal Array\n";
   for (i=0;i<m;i++)
{
   for (j=0;j<n;j++)
   {
            cout<<a[i][j]<<" ";
    }
   cout<<endl;
}
   cout<<"\n\n The Boundry element\n";
   printBoundary(a, m, n);
return 0;
}
```

అవుట్‌పుట్

```
Enter more than 3 number of rows and columns
4 4
```


Computer Science Practical Page 298-321.indd 318 24-03-2023 16:51:57

enter the value for array[1][1] :1
enter the value for array[1][2] :2
enter the value for array[1][3] :3
enter the value for array[1][4] :4
enter the value for array[2][1] :5
enter the value for array[2][2] :6
enter the value for array[2][3] :7
enter the value for array[2][4] :8
enter the value for array[3][1] :9
enter the value for array[3][2] :0
enter the value for array[3][3] :1
enter the value for array[3][4] :2
enter the value for array[4][1] :3
enter the value for array[4][2] :4
enter the value for array[4][3] :5
enter the value for array[4][4] :6

Original Array

1 2 3 4
5 6 7 8
9 0 1 2
3 4 5 6

The Boundary element

1 2 3 4
5 8
9 2
3 4 5 6

CS-8 క్రింది వివరణలతో C++ లో ప్రచురణకర్త అనే తరగతిని నిర్వచించండి

ప్రైవేట్ సభ్యులు

పుస్తకం పూర్ణాంకం

శీర్షిక 20 అక్షరాలు

Computer Science Practical Page 298-321.indd 319 24-03-2023 16:51:57

రచయిత 10 అక్షరాలు

ధర ఫ్లోట్

టొటమ్ట్ ఫ్లోట్

calculate () అనే .... ని నిర్వచించి దాని ద్వారా ఎన్ని కాపీల సంఖ్య మరియు ధరను లెక్కించి మొత్తాన్ని తిరిగి పంపించడం.

పబ్లిక్ సభ్యులు

- అన్ని డేటా సభ్యులను ప్రారంభించడానికి డిఫాల్ట్ కన్స్ట్రక్టర్ ఫంక్షన్. పుస్తక సంఖ్య 1001 నుండి స్వయంచాలకంగా ఉత్పత్తి చేయబడాలి.
- శీర్షిక, రచయిత మరియు ధర విలువలకు Readdata() ఫంక్షన్‌ను ఆమోదించుము. యూజర్ నుండి పలు కాపీలను గ్రహించి, లెక్కించుటకు () అనుమతించుము.
- Display data () ఫంక్షన్ క్రింది నమూనా రూపంలో అన్ని డేటా మెంబర్లను ప్రదర్శించును.

```
                    ABC PUBLISHERS
                    ~~~~~~~~~~~~~~~~~~
                        INVOICE
                        ~~~~~~~~~
==================================
Book Number         :
Title          :
Author Name         :
Price Per Book:
Total Amount        :
==================================
```

కోడింగ్

```
#include<iostream>
#include<stdlib.h>
using namespace std;
int id=1001;
class Publisher
{
int Bookno;
char Title[20];
char Author [10];
float Price;
float Totamt;
float calculate (int);
```


Computer Science Practical Page 298-321.indd 320 24-03-2023 16:51:57

```
public:
Publisher()
{Bookno=id;
Title[0]='\0';
Author[0]='\0';
    Price=0;
Totamt=0;
id++;
}
void Readdata();
void Displaydata();
};
void Publisher::Readdata()
{
int nocopies;
cout<<"Enter the Title name ";cin>>Title;
cout<<"Enter the Author name ";cin>>Author;
cout<<"Enter the Price ";cin>>Price;
cout<<"Enter the Number of copies ";cin>>nocopies;
Totamt=calculate(nocopies);
}
float Publisher::calculate(int x)
{
   return x*Price;
}
void Publisher::Displaydata()
{
cout<<"\n\t\tABC PUBLISHERS\n";
cout<<"\t\t~~~~~~~~~~~~~~~\n";
cout<<"\t\t  INVOICE\n";
cout<<"\t\t  ~~~~~~~\n";
cout<<"\n==================================\n";
cout<<" Book Number : "<<Bookno<<endl;
cout<<"Title                    : "<<Title<<endl;
cout<<"Author Name   : "<<Author<<endl;
cout<<"Price Per Book : "<<Price<<endl;
```


Computer Science Practical Page 298-321.indd 321 24-03-2023 16:51:57

```
cout<<"Total Amount  : "<<Totamt<<endl;
cout<<"\n==================================\n";
}
int main()
{
   int n,i;
   Publisher p[10];
   cout<<"Enter the number of object to be created";cin>>n;
   for (i=0;i<n;i++)
   p[i].Readdata();
   for (i=0;i<n;i++)
   p[i].Displaydata();
   return 0;
}
```

అవుట్‌పుట్

```
Enter the number of object to be created2
Enter the Title name C++Programming
Enter the Author name Balaguru
Enter the Price 500
Enter the Number of copies 3
Enter the Title name CoreJava
Enter the Author name Xavier
Enter the Price 250
Enter the Number of copies 5
                                ABC PUBLISHERS
                                ~~~~~~~~~~~~~~~
                                    INVOICE
                                    ~~~~~~~
==================================
Book Number   : 1001
Title                 : C++Programming
Author Name           : Balaguru
Price Per Book  : 500
Total Amount   : 1500
==================================
          ABC PUBLISHERS
```


Computer Science Practical Page 298-321.indd 322 24-03-2023 16:51:57

~~~~~~~~~~~~~~~
INVOICE
~~~~~~~

==========

Book Number : 1002

Title : CoreJava

Author Name : Xavier

Price Per Book : 250

Total Amount : 1250

==========

CS9 - ను ఉపయోగించి ఉద్యోగి వివరాలు

CS-9 **పబ్లిక్ నందు క్రింద సూచించిన పబ్లిక్ (అనుచరుల) సంఖ్యను కల్గియున్న క్లాస్ ఎంప్లాయి కొరకు C++ ప్రోగ్రాం రచన.**

Public members

eno integer

name 20 characters

des 20 characters

member Function

void get() to accept values for all data members

Declare the derived class called Salary which contain the following details.

Public members

bp, hra, da, pf, np float

member Function

void get1() to accept values for bp,hra,da and pf and invoke calculate()

calculate() calculate the np by adding bp,hra,da subtracting pf

display() Display all the details

Create the derived class object and read the number of employees.Call the function get(),get1() for each employee and display the details

కోడింగ్

```
#include<iostream>
using namespace std;
class emp{
```


Computer Science Practical Page 298-321.indd 323 24-03-2023 16:51:57

```
public:
int eno;
char name[20], des[20];
void get(){
cout<<"Enter the employee number:";
cin>>eno;
cout<<"Enter the employee name:";
cin>>name;
cout<<"Enter the designation:";
cin>>des;
}
};
class salary :public emp
{
float bp,hra, da,pf,np;
public:
void get1()
{
cout<<"Enter the basic pay:";
cin>>bp;
cout<<"Enter the HouseRent Allowance:";
cin>>hra;
cout<<"Enter the Dearness Allowance :";
cin>>da;
cout<<"Enter the Provident Fund:";
cin>>pf;
}
void calculate()
{
np=bp+hra+ da -pf;
}
void display()
   {
cout<<eno<<"\t"<<name<<"\t"<<des<<"\t"<<bp<<"\t"<<hra<<"\t"<<da<<"\t"<<pf<<"\
t"<<np<<"\n";
}
};
int main(){
int i, n;
char ch;
```


Computer Science Practical Page 298-321.indd 324 24-03-2023 16:51:57

```
salary s[10];
cout<<"Enter the number of employee:";
cin>>n;
for (i =0; i < n; i++){
s[i].get();
s[i].get1();
s[i].calculate();
}
   cout<<"\n\t\t\tEmployee Details\n";
   cout<<"\ne_no \t e_name\t des \t bp \t hra \t da \t pf \t np \n";
   for (i =0; i < n; i++){
   s[i].display();
}
return 0;
}
```

అవుట్‌పుట్

```
Enter the number of employee:2
Enter the employee number:1201
Enter the employee name:Ramkumar
Enter the designation:Engineer
Enter the basic pay:50000
Enter the House Rent Allowance:10000
Enter the Dearness Allowance :5000
Enter the Provident Fund:1000
Enter the employee number:1202
Enter the employee name:Viswanathan
Enter the designation:Engineer-Tech
Enter the basic pay:40000
Enter the House Rent Allowance:9000
Enter the Dearness Allowance :4500
Enter the Provident  Fund:1000
                    Employee Details
e_noe_name                  des                 bp      hra         da      pf      np
1201    Ramkumar    Engineer        50000   10000       5000    1000    64000
1202    Viswanathan Engineer-Tech       40000   9000        4500    1000    52500
```


Computer Science Practical Page 298-321.indd 325 24-03-2023 16:51:57

CS-10 క్రింది వివరాలతో స్టూడెంట్ అనుబడు క్లాసును సృష్టించెడి C++ ప్రోగ్రాంను రూపొందించుము.

Protected member

Rno integer

Public members

void Readno(int); to accept roll number and assign to Rno

void Writeno(); To display Rno.

The class Test is derived Publically from the Student class contains the following details

Protected member

Mark1 float

Mark2 float

Public members

void Readmark(float,float); To accept mark1 and mark2

void Writemark(); To display the marks

Create a class called Sports with the following detail

Protected members

score integer

Public members

void Readscore(int); To accept the score

void Writescore(); To display the score

The class Result is derived Publically from Test and Sports class contains the following details

Private member

Total float

Public member

void display() assign the sum of mark1 ,mark2,score in total.

invokeWriteno(),Writemark() and Writescore() .Display the total also.

కోడింగ్

```
#include<iostream>
using namespace std;
class Student
{
   protected:
   int Rno;
   public:
   void Readno(int r)
```


Computer Science Practical Page 298-321.indd 326 24-03-2023 16:51:57

```
    {
    Rno=r;
    }
    void Writeno()
    {
            cout<<"\nRoll no : "<<Rno;
    }
};
class Test :public Student
{
    protected:
    float Mark1,Mark2;
    public:
    void Readmark (float m1,float m2)
    {
            Mark1=m1;
            Mark2=m2;
    }
void Writemark()
{
cout<<"\n\n\tMarks Obtained\n ";
cout<<"\n Mark1       : "<<Mark1;
cout<<"\n Mark2       : "<<Mark2;
}
};
class Sports
{
protected:
int score;// score = Sports mark
public:
void  Readscore (int s)
{
score=s;
}
void  Writescore()
    {
cout<<"\n Sports Score : "<<score;
    }
};
class Result :public Test,public Sports
```


Computer Science Practical Page 298-321.indd 327 24-03-2023 16:51:57

```
{
int Total;
public:
void display()
{
   Total = Mark1 + Mark2 + score;
Writeno();
Writemark();
Writescore();
cout<<"\n\n Total Marks Obtained    : "<< Total<<endl;
}
};
int main()
{
   Result stud1;
stud1.Readno(1201);
stud1.Readmark(93.5,95);
stud1.Readscore(80);
cout<<"\n\t\t\t HYBRID INHERITANCE PROGRAM\n";
stud1.display();
return 0;
}
```

అవుట్‌పుట్

```
HYBRID INHERITANCE PROGRAM
Roll no : 1201
      Marks Obtained
 Mark1            : 93.5
 Mark2            : 95
 Sports Score     : 80
 Total Marks Obtained    : 268
```


Computer Science Practical Page 298-321.indd 328 24-03-2023 16:51:57

Higher Secondary - Class XI - Computer Science

List of Reviewers & Translators

TELUGU MEDIUM

Academic Coordinator

Dr. M. Ethirajulu
Senior Lecturer,
SCERT,
Chennai-6.

Reviewer

M.Sudhakar
Computer Instructor Gr-1
R.V.Govt. Boys Hr.Sec. School
Hosur – 635109

G.N. Manjula
Computer Instructor
GHSS , Peddabelagonda Palli
Denkani Kottai - 635114

Velam Nagaraju
B.T.Asst.
Govt. High School
Muthali - 635109

Translators

S. Mouli
P.G.Asst.
G.H.S.School, Athimanjeripet
Tiruvallur Dist

T. Venkatachalapathi
P.G.Asst.
G.H.S.School, K.G.Kandigai
Tiruvallur Dist

M.Devarajulu
B.T.Asst.
G.H.S.School, Athimanjeripet
Tiruvallur Dist

M. Ludhiya
B.T.Asst.
G.G.H.S.School, A.J.Pet
Tiruvallur Dist

B.Lokaiah
B.T.Asst.
G.H.School,Nediyam
Tiruvallur Dist

P.N.Venkatesulu
B.T.Asst.
G.H.S.School, Pallipat
Tiruvallur Dist

T. Parthasarathi
B.T.Asst.
G.H.S.School, K.G.Kandigai
Tiruvallur Dist

J.Srinivasulu
B.T.Asst.
G.H.School,Veligaram
Tiruvallur Dist

G.B.Geetha
B.T.Asst.
G.H.S.School, K.G.Kandigai
Tiruvallur Dist

J.Ashok
B.T.Asst.
G.H.School, Nochili
Tiruvallur Dist

V. Radhika
B.T.Asst.
G.H.School, Nochili
Tiruvallur Dist

Sunethra
B.T.Asst.
G.G.H.S.School, Tiruttani
Tiruvallur Dist

Kadiri Ravi
S.G.Asst.
G.H.School, Nochili
Tiruvallur Dist

Proof Work

P. Thulasiram
BT Asst.
PGHSS, Egmore
Chennai-8

V. Iyappa
S.G. Asst.
P.U.P.S, Medumuthukotta
Thally Block, Krishnagiri Dist,

ICT

D. Vasuraj
P.GT. & H.O.D., (Maths),
KRM Public School, Chennai.

EMIS Technology Team

R.M. Satheesh
State Coordinator Technical,
TN EMIS, Samagra Shiksha.

K.P. Sathya Narayana
IT Consultant,
TN EMIS, Samagra Shikaha

R. Arun Maruthi Selvan
Technical Project Consultant,
TN EMIS, Samagra Shiksha

Lay out

Yogesh.B
Balasubramaniyam
Ashok
Gopinath
Rajesh T

In-House QC

Daniel.A

Co-ordinator

Ramesh Munisamy

This book has been printed on 80 G.S.M. Elegant Maplitho paper.
Printed by offset at:

Computer Science Practical Page 298-321.indd 329 24-03-2023 16:51:57

Note

Computer Science Practical Page 298-321.indd 330 24-03-2023 16:51:57