GOVERNMENT OF TAMILNADU

పన్నెండవ తరగతి

HIGHER SECONDARY SECOND YEAR

కంప్యూటర్ సైన్స్

COMPUTER SCIENCE

TELUGU MEDIUM

Department of School Education

Untouchability is Inhuman and a Crime

12th_CS_Telugu_Medium_Introduction.indd 1 12-04-2023 10:48:10

Government of Tamil Nadu

First Edition - 2021

Revised Edition - 2022, 2023

Reprint - 2024

(Published under New Syllabus)

Content Creation

State Council of Educational
Research and Training
© SCERT 2021

Printing & Publishing

Tamil NaduTextbook and Educational
Services Corporation

www.textbooksonline.tn.nic.in

12th_CS_Telugu_Medium_Introduction.indd 2 12-04-2023 10:48:10

PREFACE

The tremendous effect of the computer and computing technology is in shaping the modern society for the betterment of mankind. Human civilization achieved the highest peak with the development of computer known as "**Internet Era**".

Python being a high level language is good for beginners to learn due to its easy syntax and powerful memory management. For any internet applications, it is good to learn as python being the better choice.

The user of this textbook being acquainted with the knowledge of C++ in std XI will have no difficulty in studying python language. No substantial knowledge nor experience, is required as the examples illustrated in the book are easy to follow

HOW TO USE THE BOOK?

- 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

How to get connected to QR Code?

o Download the QR code scanner from the google play store/ apple app store into your smartphone

o Open the QR code scanner application

o Once the scanner button in the application is clicked, camera opens and then bring it closer to the QR code in the textbook.

o Once the camera detects the QR code, a URL appears in the screen. Click the URL and go to the content page.

12th_CS_Telugu_Medium_Introduction.indd 3 12-04-2023 10:48:10

12th_CS_Telugu_Medium_Introduction.indd 4 12-04-2023 10:48:10

CAREER GUIDANCE AFTER 12TH



Computer Knowledge Hub

- Centre for Development of Advanced Computing https://www.cdac.in/
- Indian Computing Olympiad https://www.iarcs.org.in/inoi
- International Olympiad of Informatics https://ioinformatics.org
- Microsoft certification exams https://www.microsoft.com/en-in/learning/exam-list.aspx
- National Cyber Olympiad http://www.sofworld.org/nco
- National Institute of Electronics & Information Technology http://www.nielit.gov.in/
- National Programme on Technology Enhanced Learning https://nptel.ac.in/

After Completing +2

Arts & Science BSc Courses | Animation & Multimedia
Computer Science | Geography
Journalism / Mass Media Communication | Library Science
Maths / Physics / Chemistry / Statistics | Psychology/ Sociology
Social Work | Visual Communication | B.C.A.

Technical Diploma

Diploma in Engineering

Professional Degree & Entrance Exams

- Hotel & Catering Institute http://www.ihmchennai.org/ihmchennai/
- AIEEE- All India Engineering Entrance Exam
- B.E/B.Tech/ B.Arch (JEE, AIEEE in IITs and NITs)
- Fashion Technology & Design http://nift.ac.in
- GATE-Graduate Aptitude Test in Engineering www gate.iitm.ac.in
- Indian Navy – 10+2 BTech Entry Scheme
- Institute of Chartered Accountants of India https://icai.org
- Institute of Company Secretary https://www.icsi.edu
- Institute of Cost Accountants of India http://icmai.in
- Institute of Banking Personal Selection IBPS https://ibps.in
- JEE-Joint Entrance Examination https://jeemain.nic.in/
- JEST- Joint Entrance Screening Test https://www.jest.org.in
- Law 5-Year Integrated Course http://tndalu.ac.in
- National Defence Academy https://www.nda.nic.in
- NET- National Eligibility Test (CSIR and UGC) http://cbsenet.nic.in
- TamilNadu Dr. Ambedkar Law University http://tndalu.ac.in
- Technical Entry Scheme – Army/Navy/Airforce
- TIFR GS - Tata Institute of Fundamental Research Graduate School http://univ.tifr.res.in
- Visual Arts Degree http://www.tndipr.gov.in/television-traininginstitute.aspx

PG in Computer Science

Artificial Intelligence
Information Management and Data Analytics
Algorithms
Applied Computer Science
Big Data Analytics
Bio Informatics
Cloud Computing
Compiler Design & Construction
Computer and Network Security
Computer Networks
Computer Security
Computer Simulation
Cyber Security
Data Analytics
Data Mining
Database Management Systems
Fundamentals of Algorithms
Graphics
Machine Learning
Mobile and Web Computing
Mobile Device Programming
Modern Programming Practices
Modern Web Applications
Operating Systems
Parallel Programming
Robotics
Software Engineering
Software Engineering
Software Testing
Systems Analysis and Design
Web Application Architecture
Web Application Programming

Scholarships for graduate and post graduate courses

- DST – INSPIRE Fellowships (for Ph.D)
- DST – INSPIRE Scholarships (for UG and PG)
- In addition various fellowships for SC/ST/PWD,
- Indira Gandhi Fellowship for Single girl child (for UG and PG)
- Moulana Azad Fellowship for minorities (for Ph.D)
- UGC National Fellowship (for Ph.D)
- International Olympiad: for getting stipend for Higher Education in Science and Mathematics
- OBC etc are available.
- Visit website of University Grants Commission

12th_CS_Telugu_Medium_Introduction.indd 5

After PG courses

MPhil –Computer Science | PhD – Computer Science

Following are the latest topics for PhD in computer science:

BioInformatics and Computational Biology

Capturing and Visualizing Persona Through Faces

Comprehensive analysis of RNA sequencing experiments

Designing and Evaluating Information Gathering Robots

Digital Pathology: Diagnostic Errors, Viewing Behavior and Image Characteristics

Embedded Systems | Game Theory | Graph Theory

Graphics and Visualization | Human Computer Interaction – HCI

Improving Fault Tolerance and Performance of Data Center Networks

Increasing Access to Computer Science for Blind Students

In-situ Semantic 3D Modeling

Intelligent Crowdsourcing for Natural Language Learning and Other AI Applications

Learning Robust Tractable Models for Vision

Manipulators And Manipulation In High Dimensional Spaces

New Algorithmic Tools for Distributed Similarity Search

Reproducible measurements of web security and privacy

The Security and Privacy of Web and Mobile Advertising

Towards More Practical Reinforcement Learning, with Applications to Educational Games

Topics for Research after PG in Computer Studies

Architectures, Compiler Optimization, and Embedded Systems.

Bioinformatics and Computational Biology | Cloud Computing

Data Mining, Databases, and Geographical Information Systems.

Graphics and Visualization | High Performance Computing.

Human Computer Interaction | Internet of Things

Natural Language Processing | Networks, Distributed Systems, and Security.

Competitive Exams for Govt. Jobs

Airforce Common Admission Test – AFCAT | Army Education Officer Entry- AEC

Combined Defence Services –CDS | Defence Service Staff College, Nilgiris

Indian Defence Services | Indian Military Academy

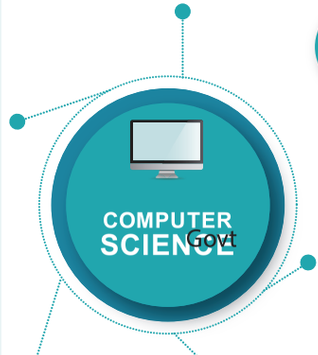
Judge Advocate General Department-JAG | NCC Entry | Railway Board Examination

SSC NAVY (Pilot/Observer) | SSC Tech Entry – Officers Training School

Staff Selection Examination | Tamil Nadu Public Service Commission

Teacher Recruitment Board | Technical Graduate Course –TGC

Territorial Army | Union Public Service Examination | Women Special Entry Scheme



Computer Related Jobs

Applications Software Developer | Big Data Analysts

Cloud Computing Programmer | College/ University Faculty

Computer Programmer | Computer Teacher

Computer Vocational Instructor | Computer Information Research Scientist

Computer Information Systems Manager | Computer Network Architect

Computer Support Specialist | Computer System Analysts

Data Mining Specialist | Database Administrator

Information Security Analysts | Market Research Analysts

Network & Computer System Administrator | Research Assistant

Systems Software Developer | User Interface Designer | Web Developer

Research Institutions in various areas of science

- Bhaba Atomic Research centre (BARC) Mumbai www.barc.gov.in
- BITS Pilani, https://www.bits-pilani.ac.in
- Central Universities www.ugc.ac.in
- Chennai Mathematical Institute (CMI) Chennai www.cmi.ac.in
- Delhi University, Delhi www.du.ac.in
- Hyderabad central university, Hyderabad www.uohyd.ac.in
- IISER Educational Institutions www.iiseradmission.in
- Indian Institute of Technology in various places (IIT's) www.iitm.ac.in
- Institute of Mathematical Sciences (IMSc) Chennai www.imsc.res.in
- Jawaharlal Nehru University (JNU) www.jnu.ac.in
- Mumbai University, Mumbai www.mu.ac.in
- National Institute of Science Education and Research (NISER), www.niser.ac.in
- National Institute of Technology (NITs) www.nitt.edu
- SavithiribaiPhule Pune university, Pune www.unipune.ac.in
- State Universities https://www.ugc.ac.in
- Tata Institute of Fundamental Research (TIFR) Mumbai www.tifr.res.in

12-04-2023 10:48:11

విషయసూచిక

యూనిట్ సంఖ్య	అధ్యాయం	కంప్యూటర్ సైన్స్	పుట సంఖ్య	నెల
యూనిట్- I సమస్య పరిష్కార పద్ధతులు	1	ఫంక్షన్	1	జూన్
	2	డేటా సంగ్రహణ	11	జూన్
	3	పరిధిలు	21	జూన్
	4	అల్గోరిథమిక్ యుక్తులు	32	జూన్
యూనిట్- II కోర్ పైథాన్	5	పైథాన్ వేరియబుల్స్ మరియు అపరేటర్లు	49	జూలై
	6	కంట్రోల్ స్ట్రక్చర్	69	జూలై
	7	పైథాన్ పంక్షన్స్	91	జూలై
	8	స్ట్రింగ్ మరియు స్ట్రింగ్ నిర్వహణ	116	ఆగస్టు
యూనిట్-III మాడ్యులారిటి మరియు OOPS	9	Lists, Tuples, Sets మరియు నిఘంటువు	134	ఆగస్టు
	10	పైథాన్ క్లాసెస్ మరియు ఆబ్జెక్ట్స్	172	ఆగస్టు
యూనిట్-IV డేటాబేసు భావనలు మరియు MySql	11	డేటాబేస్ భావనలు	185	అక్టోబరు
	12	నిర్మాణాత్మక ప్రశ్నాభాష (SQL)	201	అక్టోబరు
	13	పైథాన్ మరియు CSV ఫైళ్ళు	230	అక్టోబరు
యూనిట్-V MySql మరియు C++ తో పైథాన్ ను సమగ్రపరచడం	14	పైథాన్‌లో C++ ప్రోగ్రామ్‌ను ఉపయోగించుట	265	అక్టోబరు
	15	SQL ద్వారా డేటాను నిర్వహించుట	285	నవంబరు
	16	డేటా విజువలైజేషన్: ఉపయోగించి రేఖా చిత్రము వృత్త రేఖాపటము మరియు బార్ చిత్రము	314	నవంబరు
		పదకోశం	328	
అనుబంధం		పైథాన్ జాబితాలోని విధులు	334	
		ప్రయోగాభ్యాసములు	339	

N0YQB

అంతర్జాల పుస్తకము

QL2SM

మూల్యాంకనము

12th_CS_Telugu_Medium_Introduction.indd 6 12-04-2023 10:48:11

యూనిట్ I

అధ్యాయం 1

ఫంక్షన్

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశము అభ్యసించిన తరువాత విద్యార్థులు తెలుసుకొనునవి

- ఫంక్షన్ నిర్వచనం
- పారామితులు (మరియు వాదనలు)
- ఇంటర్ఫేస్ మరియు అమలు పరచడం
- Pure ఫంక్షన్‌లు
- దుష్ప్రభావాలు

1.1 పరిచయం

అల్గోరిధం రాయడం మరియు మూల్యాంకనం చేయడంలో ముఖ్యమైన ప్రమాణం ఒక పనిని పూర్తి చేయడానికి అది తీసుకొను సమయం. ప్రోగ్రామింగ్ భాష, కంపైలర్ మరియు ఉపయోగించిన (కంప్యూటర్) భాష యొక్క గణన సమయం యొక్క వ్యవధి స్వతంత్రంగా ఉండవలెను స్టేట్‌మెంట్‌లను ఉపయోగించి ప్రోగ్రామ్ అల్గోరిధంలు వ్యక్తమవుతాయని మీకు తెలుసు పెద్ద మొత్తంలో మొత్తం స్టేట్మెంట్ చాలాసార్లు పునరావృతం కావాలంటే సబ్‌రొటిన్‌లను (subroutines) ఉపయోగిస్తారు.

కంప్యూటర్ ప్రోగ్రామ్‌ల యొక్క ప్రాథమిక బిల్డింగ్ బ్లాక్స్ సబ్‌రొటిన్‌లు. సబ్‌రొటిన్‌లు కోడ్ యొక్క చిన్న విభాగాలు, ఇవి పదేపదే ఉపయోగించబడే ఒక నిర్దిష్ట పనిని నిర్వహించడానికి ఉపయోగిస్తారు. ప్రోగ్రామింగ్ భాషలలో ఈ సబ్‌రొటిన్‌లను ఫంక్షన్‌లు అంటారు.

1.2 ప్రోగ్రామింగ్ భాషకు సంబంధించిన ఫంక్షన్

ఒక ఫంక్షన్ అనేది కోడ్ యొక్క యూనిట్, ఇది తరచుగా ఎక్కువ కోడ్ నిర్మాణంలో నిర్వచింపబడుతుంది. ప్రత్యేకంగా ఒక ఫంక్షన్‌లో వైవిధ్యాలు, వ్యక్తీకరణలు వంటి అనేక రకాల ఇన్‌పుట్‌లపై పనిచేసే కోడ్ సమితి ఉంటుంది ఇది స్థిరమైన అవుట్‌పుట్‌ను ఉత్పత్తి చేస్తుంది.

1.2.1 ఫంక్షన్ స్పెసిఫికేషన్ (Function Specification)

a:= (24) అనే ఉదాహరణను పరిశీలించండి *a:= (24)* కు వ్యక్తీకరణ కలదు. కాని ఇందులో (24) మాత్రమే వ్యక్తీకరణ కాదు. బదులుగా, ఇది ఫంక్షన్ నిర్వచనం. నిర్వాచనాలు విలువలను పేర్లతో బంధిస్తాయి, ఈ సందర్భంలో విలువ 24 పేరు *'a'* కు కట్టుబడి ఉంటుంది. నిర్వచనాలు వ్యక్తీకరణలు కావు, అదే సమయంలో వ్యక్తీకరణలు కూడా నిర్వచనాలుగా పరిగణించబడవు. నిర్వచనాలు ప్రత్యేకమైన వాక్యనిర్మాణ బ్లాక్స్. నిర్వచనాలు వాటి లోపల బ్లాక్స్ కలిగి ఉంటాయి నిర్వచనాలతో వ్యక్తీకరణలు ఉండవచ్చు మరియు దీనికి విరుద్ధంగా కూడా ఉండవచ్చు.

1.2.2 పారామితులు (మరియు వాదనలు) (Parameters (and arguments))

పారామితులు ఫంక్షన్ నిర్వచనంలోని వేరియబుల్స్ మరియు వాదనలు ఫంక్షన్ నిర్వచనానికి పంపబడిన విలువలు అగును.

12th_CS_Telugu_Medium_Chapter_1.indd 1 27-03-2023 10:36:27

1. డేటా టైప్ లేని పారామితులు (Parameter without Type)

ఫంక్షన్ నిర్వచనముకు ఒక ఉదాహరణ చూద్దాం.

```
(requires: b>=0 )
(returns: a to the power of b)
let rec pow a b:=
        if b=0 then 1
        else a * pow a (b-1)
```

పై ఫంక్షన్ నిర్వచనంలో వేరియబుల్ 'b' అనునది పారామితి మరియు వేరియబుల్ 'b' కి పంపబడిన విలువ వాదన అగును. ఫంక్షన్ ముందస్తు నిబంధన ***(requires)*** మరియు తర్వాత నిబంధన ***(returns)*** ఇవ్వబడ్డాయి. గమనిక మనము ఏ రకాలను ***(data type)*** ప్రస్తావించలేదు. కొన్ని భాషా కంపైలర్‌లు ఈ రకమైన డేటా రకం అనుమతి సమస్యను అల్గోరిథమిక్‌గా పరిష్కరిస్తుంది, అయితే కొన్ని కంపైలర్‌లకి డేటా టైపును పేర్కొనడం అవసరం.

పై ఫంక్షన్ నిర్వచించడం ద్వారా 'if' స్టేట్‌మెంట్ ఒక శాఖకు ***'1'*** విలువను 'then' ద్వారా తిరిగి ఇస్తే, డేటాటైపు నియమం ప్రకారం 'if' అన్నింటిని ***'int'*** డేటా రకాన్ని కలిగి ఉంటుంది. 'if' యొక్క డేటా రకం ***'int'*** గా ఉండడం వలన ఫంక్షన్ యొక్క తిరిగి పంపు విలువ కూడా ***'int'*** గా ఉండును. ***'b'*** యొక్క విలువ సున్నతో '='ఆపరేటర్‌తో పోల్చబడుతుంది. కాబట్టి ***'b'*** డేటా రకం కూడా ***'int'*** అగును. * ఆపరేటర్‌తో ***'a'*** యొక్క విలువ మరొక వేరియబుల్ గుణించుట వలన ***'a'*** యొక్క డేటా రకం ***'int'*** అగును.

2. డేటాటైపుతోకూడినపారామితి(Parameter with Type)

కొన్ని కారణాల వల్ల ఇప్పుడు మనం డేటా టైపుతో అదే నిర్వచనాన్ని వ్రాద్దాం.

```
(requires: b>  0 )
(returns: a to the power of b )
let rec pow (a: int) (b: int) : int :=
        if b=0 then 1
        else a * pow a (b-1)
```

'a' మరియు ***'b'*** డేటా టైపు సూచన (type annotations) వ్రాసేటప్పుడు కుండలీకరణాలు () అవసరం. సాధారణంగా మనము ఈ సూచనలను వదిలివేయవచ్చు. ఎందుకంటే కంపైలర్ వీటిని ఊహించడం చాలా సులభం. మునుపు మనము డేటా టైపును స్పష్టంగా వ్రాస్తాము. మనకు కంపైలర్ నుండి టైప్ లోపం వచ్చినప్పుడు అర్థవంతం కాని సమయాల్లో ఇది ఉపయోగపడుతుంది. డేటా టైపును స్పష్టంగా సూచించడం, అటువంటి దోష సందేశాన్ని డీబగ్ చేయడంలో సహాయపడుతుంది.

ఫంక్షన్‌లను నిర్వచించే వాక్యనిర్మాణం గణిత వినియోగానికి దగ్గరగా ఉంటుంది. నిర్వచనం ***let*** అను ***keyword*** ద్వారా పరిచయం చేయబడినది. తదుపరి ఫంక్షన్ పేరు మరియు దాని వాదనలు; ఉంటాయి. కాబట్టి వాదన యొక్క చిత్రాన్ని లెక్కించే సూత్రం = గుర్తు తర్వాత వ్రాయబడుతుంది. మీరు పునరావృత ఫంక్షన్‌ను నిర్వచించాలనుకుంటే ***"let"*** కు బదులుగా ***"let rec"*** లు ఉపయోగించండి.

ఫంక్షన్: ఫంక్షన్ నిర్వచనానికి వాక్య నిర్మాణం

```
let rec fn a1 a2 ... an := k
```

ఇక్కడ ***'fn'*** అనునది ఒక వేరియబుల్. ఇది ఫంక్షన్ పేరును సూచిస్తుంది. పారామితులుగా ఉపయోగించబడిన వేరియబుల్స్ ***'a1'*** నుండి ***'an'*** ***'fn'*** పునరావృత ఫంక్షన్ కావాలంటే ***'rec'*** అనే కీవర్డ్ అవసరం. లేకపోతే దానిని వదిలి వేయవచ్చు.

12th_CS_Telugu_Medium_Chapter_1.indd 2 27-03-2023 10:36:27

గమనిక

తనకు తానుగానే ఫంక్షన్ నిర్వచనము పిలుచుటను పునరావృత ఫంక్షన్ అంటారు.

ఉదాహరణకు: ఎంటర్ చేసిన సంఖ్య సరి లేదా బేసి సంఖ్య అని తనిఖీ చేయడానికి ఒక ఉదాహరణ చూద్దాం.

```
(requires: x>= 0)
        let rec even x :=
        x=0 || odd (x-1)
        return 'even'
(requires: x>= 0)
        let odd x :=
        x<>0 && even (x-1)
        return 'odd'
```

ఫంక్షన్ రకాలకు వాక్యనిర్మాణము :

$x \rightarrow y$

$x1 \rightarrow x2 \rightarrow y$

$x1 \rightarrow ... \rightarrow xn \rightarrow y$

'x' మరియు *'y'* వేరియబుల్స్ ఫంక్షన్ టైపును సూచించును. $x \rightarrow y$ అను ఫంక్షన్ టైపు x టైపును ఇన్పుట్‌గా పొంది మరియు y టైపును అవుట్పుట్టుగా తిరిగి పంపును. $x1 \rightarrow x2 \rightarrow y$ అనే ఫంక్షన్ టైపు రెండు ఇన్పుట్లను తీసుకునే ఒక ఫంక్షన్ టైపు, మొదటి ఇన్పుట్ టైప్ *'x1'* మరియు రెండవ ఇన్పుట్ టైపు *'x2'* మరియు y తిరిగి పంపే అవుట్‌పుట్ టైపు అగును. యొక్క అదే విధంగా $x1 \rightarrow ... \rightarrow xn \rightarrow y$ లో పారామితి యొక్క *'x'* ఇన్పుట్ టైపు అగును. అవుట్ పుట్ టైపు *'y'* అగును.

గమనిక

అన్ని ఫంక్షన్‌లు static నిర్వచనములు అగును. డైనమిక్ ఫంక్షనలకు నిర్వచనం లేవు.

1.3 ఇంటర్ఫేస్ మరియు అమలుపరచడం

ఇంటర్ఫేస్ అనునది ఒక కార్యచరణ సమితి ఆబ్జెక్ట్ (object) చేయగల స్థితి. ఉదాహరణకు మీరు లైట్ స్విచ్ వేసినపుడు లైట్ వెలుగుతుంది. ఇది ఎలా వెలుగుతుంది అనే విషయం అవసరం లేదు. ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామింగ్ భాషలో, ఇంటర్ఫేస్ అనేది అన్ని ఫంక్షన్ల వివరణ (description) అగును. మన ఉదాహరణలో లైట్ లాగా పనిచేసే ఏదైనా ఒకటి turn_on () మరియు turn_off () లు యొక్క ఫంక్షన్ నిర్వచనాన్ని కలిగిఉండాలి. ఇంటర్ఫేస్ యొక్క ఉద్దేశ్యం కంప్యూటర్ క్లాస్ ప్రాపర్టీలను దృవపరచడం.

ఇంటర్ఫేస్ మరియు ఇంప్లిమెంటేషన్ మధ్య గల భేదాలు

ఇంటర్ఫేస్	ఇంప్లిమెంటేషన్ (అమలుపరచడం)
ఇంటర్ఫేస్ కేవలం ఒక ఆబ్జెక్ట్ ఏమి చేయగలదో నిర్వచిస్తుంది. కాని వాస్తవానికి అది చేయదు.	ఇంటర్ఫేస్‌నందు నిర్వచించబడిన సూచనలను అమలు చేస్తుంది.

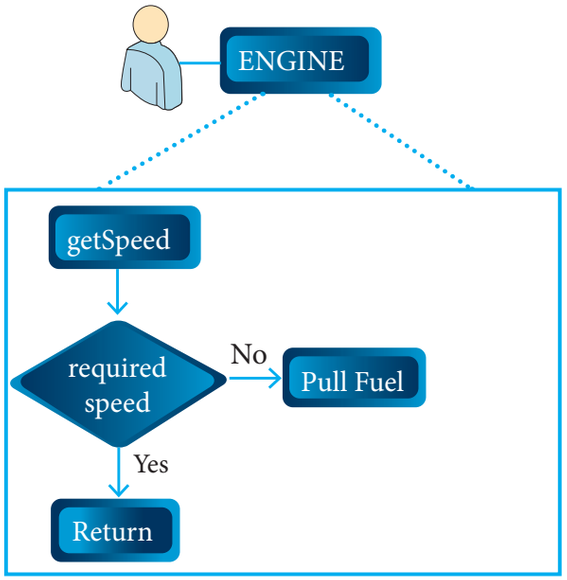
ఆబ్జెక్టివ్ ఓరియెంటెడ్ ప్రోగ్రామింగ్ లాంగ్వేజ్‌లో క్లాస్ అనేది ఇంటర్‌ఫేస్ ఆబ్జెక్ట్ ఎలా అమలు పరచి నెరవేర్చబడుతుందో అనే దాన్ని తెలియజేయునది ఇంప్లిమెంటేషన్.

1.3.1 ఇంటర్ఫేస్ యొక్క లక్షణాలు

- క్లాస్ టెంప్లేట్ ఒక ఆబ్జెక్ట్‌ను సృష్టించడానికి మరియు సరిగ్గా ఆపరేట్ చేయడానికి ఇంటర్ఫేస్‌లను నిర్దేశిస్తుంది.

- ఒక ఆబ్జెక్ట్ యొక్క విధులు మరియు behaviour ఆబ్జెక్ట్‌కు ఫంక్షన్లలను పంపడం ద్వారా నియంత్రించబడుతుంది.

ఉదాహరణకు కారు వేగాన్ని పెంచే ఉదాహరణను తీసుకుందాం.



కారు నడుపుతున్న వ్యక్తి కారు యొక్క అంతర్గత పని తీరు గురించి తెలుసుకోవలసిన అవసరం లేదు. కారు వేగాన్ని పెంచడానికి, అతను కారు యొక్క యాక్సిలరేటర్‌ను నొక్కడం ద్వారా కావలసిన దాని తీరును పొందుతాడు. ఇక్కడ యాక్సిలరేటర్ అంటే కార్ డ్రైవర్ (కాలింగ్ ఆబ్జెక్ట్) మరియు ఇంజిన్ (called object) మధ్య ఇంటర్ఫేస్.

ఈ సందర్భంలో, కాలింగ్ ఫంక్షన్ వేగం (70)అయి ఉండాలి. ఇది ఇంటర్ఫేస్.

అంతర్గతంగా, కారు యొక్క ఇంజన్ అన్ని పనులను చేస్తుంది. ఇంధనం, గాలి, పీడనం మరియు విద్యుత్ ఇక్కడ ఒక్కటిగా కలిసి, శక్తిని ఉత్తత్పి చేసి వాహనాన్ని కదిలిస్తాయి. ఈ కార్యకలాపాలన్నీ కారు డ్రైవర్ నుండి వేరు చేయబడతాయి. అతను వేగంగా నడపడానికి మాత్రమే ఆసక్తి చూపుతాడు. అందువలన, ఇంటర్ఫేస్, ఇంప్లిమెంటేషన్ నుండి మనము వేరు చేస్తాము.

ఒక సులభమైన ఉదాహరణ చూద్దాం. ఇవ్వబడిన మూడు ఫంక్షన్‌లలో అతి తక్కువ విలువను చూపు ఫంక్షన్ అమలును గమనించండి.

```
let min 3 x y z :=
        if x < y then
                if x < z then x else z
        else
                if y < z then y else z
```

1.4 Pure ఫంక్షన్

Pure ఫంక్షన్ అనునవి ఒకే రకమైన పారామితిని పంపించినపుడు ఖచ్చితమైన ఫలితాన్ని ఇస్తుంది. ఉదాహరణకు గణిత ఫంక్షన్ sin (0) ఎల్లప్పుడు 0 ఫలితాన్ని ఇస్తుంది. దీని అర్థం మీరు ఫంక్షన్ ఒకే పారామితిని పిలిచిన ప్రతిసారి ఒకే ఫలితాన్ని పొందుతారు. ఒక ఫంక్షన్ స్వచ్ఛమైన ఫంక్షన్ కావచ్చు, అది ఏదైనా బాహ్య వేరియబుల్స్ కలిగి ఉండకూడదు. అది ఆ వేరియబుల్స్ యొక్క ప్రవర్తనను మారుస్తుంది.

ఒక ఉదాహరణ చూద్దాం.

```
let square x
        return: x * x
```

పై దానిలో square ఫంక్షన్ pure ఫంక్షన్ అగును. ఎందుకంటే ఇది ఒక input కు విభిన్న ఫలితాలను ఇవ్వదు.

Pure ఫంక్షన్లను కలిగియుండడానికి పలు సిద్ధాంతపరమైన ప్రయోజనాలు ఉన్నాయి. ఒక ప్రయోజనం ఏమిటంటే ఒక ఫంక్షన్ Pure ఫంక్షన్‌గా ఉన్నప్పుడు పారామితితో ఫంక్షన్‌లను చాలా సార్లు

12th_CS_Telugu_Medium_Chapter_1.indd 4 27-03-2023 10:36:27

పిలువబడితే కంపైలర్ వాస్తవానికి ఒక్కసారి మాత్రమే ఫంక్షన్‌ను పిలుస్తుంది. ఒక ఉదాహరణ చూద్దాం.

```
let length s:=
        i: = 0
        if i <strlen (s) then
        -- Do something which doesn't affect s
        ++i
```

ఇది కంపైల్ చేయబడితే ***strlen (s)*** ను ప్రతిసారి పిలుస్తారు మరియు strlen మొత్తం ***'s'*** గా మార్చవలసిన అవసరం ఉంది. కంపైలర్ స్మార్ట్ strlenను అనునది pure ఫంక్షన్ తగినంత క్రియాత్మకంగా ఉంటే, తీసుకోబడును మరియు లూప్‌లో ***'s'*** ను మార్చబడదు. అప్పుడు అది అనవసరమైన అదనపు కాల్‌లను తొలగించి లూప్‌ను ఒక్కసారి మాత్రమే అమలు చేస్తుంది. వీటి నుండి మనం అర్థం చేసుకోగలిగేది ఏమనగా strlen లో అనేది pure ఫంక్షన్ ఎందుకంటే ఫంక్షన్ ఒక వేరియబుల్‌ను పారామితిగా తీసుకుంటుంది మరియు దాని పొడవును కనుగొనటానికి దాని access చేస్తుంది. ఈ ఫంక్షన్ బాహ్య మెమరీ నుండి input తీసుకుంటుంది. కాని దానిని మార్చదు మరియు తిరిగి వచ్చిన విలువ యాక్సెస్ చేసిన బాహ్య మెమరీ నుండి తీసుకోబడును.

గమనిక

Pure ఫంక్షన్ల మూల్యాంకనం దాని output లో ఎటువంటి దుష్ప్రభావాలను కలిగించదు.

1.4.1 Impure ఫంక్షన్లు (Impure functions)

ఫంక్షన్లలలో ఉపయోగించబడు చరరాశులు ఎటువంటి ఆర్గ్యుమెంట్లు ఎజ్ చేయబడిన ఫంక్షన్ల మూలముగా దుష్ప్రబావాలు కలిగించవచ్చు. అటువంటి సందర్భాలలో ఫంక్షన్లను impure ఫంక్షన్ అంటారు. ఒక ఫంక్షన్ దాని డెఫినిషన్ బ్లాక్ వెలుపల వేరియబుల్స్ లేదా ఫంక్షన్లపై ఆధారపడి ఉన్నప్పుడు, ఫంక్షన్ పిలువబడే ప్రతిసారీ అదే విధంగా ప్రవర్తిస్తుందని మీరు అనుకోలేరు. ఉదాహరణకు, గణిత ఫంక్షన్ రాండమ్ () ఒకే ఫంక్షన్ కాల్ కోసం వేర్వేరు ఫలితాలను ఇస్తుంది.

```
let Random number
let a := random()
    if a > 10 then
       return: a
    else
       return: 10
```

ఇక్కడ ఫంక్షన్ రాండమ్ impure ఫంక్షన్ ఎందుకంటే మనం ఫంక్షన్‌ను ఉపయోగించినపుడు అని పిలిచినప్పుడు ఫలితం ఎలా ఉంటుందో ఖచ్చితంగా తెలియదు.

1.4.2 దుష్ప్రభావాలు (Side-effects of Impure functions)

ఫంక్షన్లు బాహ్యప్రపంచంతో సంబంధం కలిగియుండునపుడు దుష్ప్రభావాలను కూడా ఏర్పరచునని మనకు తెలుసు మన ఫంక్షన్లను pure చేయడమే మన లక్ష్యం అయినప్పటికీ విధులు impure గా కొన్ని సందర్భాల్లో మారే పరిస్థితులు ఉన్నాయి. స్పష్టం చేయడానికి సైడ్ ఎఫెక్ట్ అవసరమైన చెడు విషయం కాదని గుర్తించుకోండి. కొన్నిసార్లు అవి ఉపయోగపడతాయి. (ముఖ్యంగా ఫంక్షనల్ ప్రోగ్రామింగ్ ఉదాహరణకి వెలుపల)

ఫంక్షన్ వెలుపల వేరియబుల్‌ను సవరించండి (Modify variable outside a function)

సైడ్ ఎఫెక్ట్స్ యొక్క అత్యంత ప్రాచుర్యం పొందిన సమూహాలలో ఒకటి ఫంక్షన్ వెలుపల వేరియబుల్‌ను సవరించడం.

ఉదాహరణకి

```
let y: = 0
(int) inc (int) x:
        y: = y + x;
        return (y)
```

పై ఉదాహరణలో, ఫంక్షన్ నిర్వచనం లోపల y యొక్క విలువ మారుతుంది, దీని ఫలితంగా ప్రతిసారీ ఫలితం మారుతుంది. inc () ఫంక్షన్ యొక్క దుష్ప్రభావం అది బాహ్య కనిపించే వేరియబుల్ y యొక్క డేటాను మారుస్తుంది. మీరు చూడగలిగినట్లుగా కొన్ని దుష్ప్రభావాలను గుర్తించడం చాలా సులభం మరియు వాటిలో కొన్ని గమ్మత్తైనవి. మా ఫంక్షన్ impure (సైడ్ ఎఫెక్ట్) ఒక మంచి సంకేతం ఏమిటంటే అది ఎటువంటి వాదనలు తీసుకోదు మరియు అది ఏ విలువను ఇవ్వదు.

pure మరియు impure ఫంక్షన్ల మధ్య ప్రధాన తేడాల గురించి మనం అర్థం చేసుకోగల ఈ ఉదాహరణలు మరియు నిర్వచనాల నుండి

Pure ఫంక్షన్	Impure ఫంక్షన్
Pure ఫంక్షన్ యొక్క తిరిగి వచ్చే విలువ దాని పారామితులపై ఆధారపడి ఉంటుంది. అందువల్ల, మీరు pure ఫంక్షన్లను అదే పారామితితో కూడిన సమితి అని పిలిస్తే, ఎల్లప్పుడూ తిరిగి అదే విలువలను పొందుతారు. వాటికి ఎటువంటి దుష్ప్రభావాలు లేవు.	Impure ఫంక్షన్ల యొక్క తిరిగి విలువ దాని ఆమోదించిన పారామితులపై మాత్రమే ఆధారపడి ఉండదు. అందువల్ల, మీరు **Impure** ఫంక్షన్లను ఒకే పారామితితో పిలిస్తే, ఉత్తీర్ణత ఉదాహరణ, యాదృచ్ఛిక () తేదీ ().
ఈ ఫంక్షన్ దానికి పంపిన పారామితులను సవరించదు.	ఈ ఫంక్షన్ తమకు పంపిన పారామితులను సవరించవచ్చు.

ఇప్పుడు రెండు సానుకూల పూర్ణాంక సంఖ్యల యొక్క గొప్ప సాధారణ విభజన (జిసిడి) ను నిర్ణయించడానికి *Pure* ఫంక్షన్ యొక్క ఉదాహరణను చూద్దాం.

```
let rec gcd a b :=
        if b <> 0 then gcd b (a mod b) else return a;

output
        gcd 13 27;;
        - : int = 1
        gcd 20536 7826;;
        - : int = 2
```

పై ఉదాహరణలో ప్రోగ్రామ్ *'gcd'* అనేది ఫంక్షన్ name అగును. వేరియబుల్ *'b'* అనునది *'0'* అయ్యే వరకు పునరావృతమయ్యే ఫంక్షన్ పేరు. *b* మరియు *(a mod b) gcd* ఫంక్షన్ యొక్క *'a'* మరియు *'b'* లకు పంపబడిన రెండు వాదనలను గుర్తించుకోండి.

1.4.3 ఫంక్షన్ ఉపయోగించి Chameleons లో ఊసరవెల్లిలు అను సమస్య సాధించుట (Chameleons of Chromeland problem using function)

XI వ తరగతిలో చదివిన క్రోమిలేన్ ఊసరవెల్లిలు అను అధ్యాయమును గుర్తు చేసుకోండి. రెండు రకాల ఊసరవెల్లి సమాన *C* సంఖ్య అయితే ఈ రెండు రకాలు కలిసి మూడవ రకం యొక్క రంగును మార్చడానికి ప్రోగ్రామ్ ప్రోటోకాల్ను సృష్టించాలి మరియు చివరికి రకం అన్ని ఒకే రంగును చూపించాలి.

ప్రతి ఊసరవెల్లి సంఖ్యను *a, b, c* వేరియబుల్స్గా తీసుకుందాం. దీని ప్రారంభ విలువ వరుసగా *A, B* మరియు *C* అనునది ఇన్పుట్ నియమము a = b అనునది ఇన్పుట్ నియమము.

దాని (input) ఇన్పుట్ అవుట్పుట్ యొక్క పరస్పర సంబంధం a = b = 0 మరియు c = A +

12th_CS_Telugu_Medium_Chapter_1.indd 6 27-03-2023 10:36:27

B + C దీని ప్రోగ్రామ్ ప్రోటోకాల్‌కు మోనోక్రోమెటైజ్ (monochromatize) అని పేరు పెట్టవచ్చు.

దీనిని monochromatize (a, b, c) అని గుర్తించవచ్చును.

-- ఇన్పుట్	:	a = A, b = B, c = C, a = b
-- అవుట్పుట్	:	a = b = 0, c = A+B+C

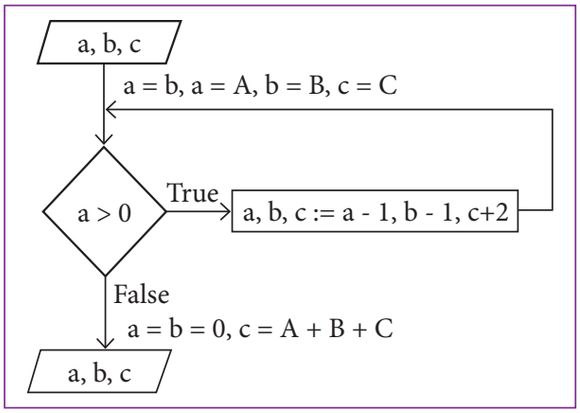
ప్రతి పునురుత్పాదక దశలో, రెండు రకాల ఊసరవెల్లులు (సంఖ్యకు సమానం) కలుస్తాయి మరియు వాటి రంగులను మూడవ దానికి మారుస్తాయి. ఉదాహరణకు A, B, C = 4, 4, 6, అయితే సమావేశం యొక్క శ్రేణి దారి తీస్తుంది.

iteration	a	b	c
0	4	4	4
1	3	3	8
2	2	2	10
3	1	1	12
4	0	0	14

ప్రతి సమావేశంలో ***a*** మరియు ***b*** తో ఒక్కొక్కటి ***1*** తగ్గుతాయి, ***c*** మరియు ***2*** పెరుగుతుంది. పరిష్కారం పునరుక్తి అల్గోరిథం 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
```

ఈ ప్రోటోకాల్ ప్రవాహ రేఖాచిత్రంగా క్రింద చూపబడింది.



ఫంక్షన్‌నుపయోగించి ప్రోగ్రామ్ ప్రోటోకాల్ను వ్రాయవచ్చును.

```
let rec monochromatize a  b  c :=
      if a > 0 then
        a, b, c := a-1, b-1, c+2
      else
        a:=0, b:=0, c:= a + b + c
        return c
```

గుర్తుంచుకోవలసిన విషయములు

- ప్రోగ్రామింగ్ భాష యొక్క స్టేట్‌మెంట్‌లను ఉపయోగించి అల్గోరిథంలు వ్యక్తీకరించబడతాయి.
- సబ్‌రొటిన్‌లు కోడ్ యొక్క చిన్న విభాగాలు, వీటిని ఒక నిర్దిష్ట పనిని చేయుట కొరకు పదేపదే ఉపయోగించుకోవచ్చు.
- ఒక ఫంక్షన్ అనేది కోడ్ యొక్క యూనిట్, ఇది తరచుగా గొప్ప కోడ్ నిర్మాణంలో నిర్వచించబడుతుంది.
- అనేక రకాల ఇన్పుట్‌లపై పనిచేసి మరియు కాంక్రీట్ అవుట్పుట్‌లను ఉత్పత్తి చేసే కోడ్ యొక్క సమితిని ఫంక్షన్ కలిగియుంటుంది.
- నిర్వచనాలు అనునవి విభిన్న సింటెక్స్‌లు.
- పారామితులు అనునవి ఒక ఫంక్షన్ నిర్వచనంలో వెరియబుల్స్ మరియు వాదనలు ఫంక్షన్ నిర్వచనం ద్వారా నిర్వచనానికి పంపబడే విలువలు.
- డెటా టైప్‌కి సూచనలు ఇచ్చినపుడు ఫంక్షన్ నిర్వచనంలో కుండలీకరణాలు () తప్పనిసరి.
- ఇంటర్‌ఫేస్ అనేది ఒక object చేయగల చర్య యొక్క సమితి అగును.
- ఇంటర్ఫేస్ ఒక object ఏమి చేయగలదో నిర్వచిస్తుంది కాని వాస్తవానికి అది చేయదు.
- ఇంటర్ఫేస్‌లో నిర్వచించిన సూచనలను ఇంప్లిమెంటేషన్ (Implementation) చేస్తుంది.
- Pure ఫంక్షన్లు అదే పారామితులను దాటినప్పుడు ఖచ్చితమైన ఫలితాలను ఇచ్చు ఫంక్షన్లు.
- ఫంక్షన్లలలో ఉపయోగించబడు చరరాశులు ఎటువంటి ఆర్గ్యుమెంట్లు ఎజ్ చేయబడిన ఫంక్షన్ల మూలముగా దుష్ప్రభావాలు కలిగించవచ్చు.

ప్రయోగ అభ్యాసం

1. 3 సంఖ్యలలో కనిష్ఠ సంఖ్యను కనుగొనడానికి అల్గోరిథమిక్ ఫంక్షన్ నిర్వచనాన్ని రాయండి.
2. సహజ సంఖ్యల మొత్తాన్ని కనుగొనడానికి గాను అల్గోరిథమిక్ పునరావృత ఫంక్షన్ యొక్క నిర్వచనం వ్రాయండి.

S2LUO

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. ఒక నిర్దిష్ట పనిని నిర్వహించడానికి ఉపయోగించే కోడ్ యొక్క చిన్న విభాగాలు _____ అందరు.

   అ) సబ్‌రొటిన్లు ఆ) ఫైల్స్ ఇ) Pseudo కోడ్ ఈ) మాడ్యూల్స్

2. క్రింది వాటిలో ఏది ఎక్కువ కోడ్ నిర్మాణంలో నిర్వచించబడిన కోడ్ యొక్క యూనిట్?

   అ) సబ్‌రొటిన్లు ఆ) ఫంక్షన్ ఇ) ఫైల్స్ ఈ) మాడ్యూల్స్

12th_CS_Telugu_Medium_Chapter_1.indd 8 27-03-2023 10:36:27

3. క్రింది వాటిలో ఏది సింటాక్టిక్ బ్లాక్?

అ) సబ్‌రొటిన్లు ఆ) ఫంక్షన్ ఇ) నిర్వచనం ఈ) మాడ్యూల్స్

4. ఫంక్షన్ నిర్వచనంలోని వేరియబుల్స్‌ను ఏమంటారు?

అ) సబ్‌రొటిన్లు ఆ) ఫంక్షన్ ఇ) నిర్వచనం ఈ) పారామితులు

5. ఫంక్షన్ నిర్వచనంకు పంపబడిన విలువలను ఏమంటారు?

అ) పారామితులు ఆ) సబ్‌రొటిన్లు ఇ) ఫంక్షన్ ఈ) నిర్వచనం

6. ఫంక్షన్ నిర్వచనంలో టైప్ సూచనలను వ్రాయడానికి క్రింది వాటిలో ఏది తప్పనిసరి?

అ) {} ఆ) () ఇ) [] ఈ) <>

7. క్రిందివాటిలో ఏది ఒక object ఏమి చేయగలదో నిర్వచించునది ఏది?

అ) ఆపరేటింగ్ సిస్టమ్ ఆ) కంపైలర్ ఇ) ఇంటర్ఫేస్ ఈ) ఇంటర్ప్రిటర్

8. క్రింది వాటిలో ఏది ఇంటర్ఫేస్ నందు నిర్వచించిన సూచనలను నిర్వహిస్తుంది?

అ) ఆపరేటింగ్ సిస్టమ్ ఆ) కంపైలర్ ఇ) అమలు పరచుట ఈ) ఇంటర్ప్రిటర్

9. ఒకే పారామితులను ఆమోదించినప్పుడు ఖచ్చితమైన ఫలితాన్ని ఇచ్చే విధులను _____ అంటారు.

అ) Impure ఫంక్షన్ ఆ) పాక్షిక ఫంక్షన్ ఇ) డైనమిక్ ఫంక్షన్ ఈ) Pure ఫంక్షన్

10. ఆమోదించిన పారామితులకు దుష్ప్రభావాలను కలిగించే ఫంక్షన్‌లను ఏమంటారు?

అ) impure ఫంక్షన్ ఆ) Partial ఫంక్షన్ ఇ) డైనమిక్ ఫంక్షన్ ఈ) Pure ఫంక్షన్

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. సబ్‌రొటిన్ అంటే ఏమిటి?
2. ప్రోగ్రామింగ్ భాషకు సంబంధించి ఫంక్షన్‌ను నిర్వచించండి.
3. X:=(78) నుండి ఏమి తెలియుచున్నది?
4. ఇంటర్ఫేస్ మరియు అమలుపరచుట(Implementation)కు మధ్య గల భేదాలు తెలపండి.
5. క్రింది వాటిలో ఏది సాధారణ ఫంక్షన్ నిర్వచనం మరియు ఏది పునరావృత ఫంక్షన్ నిర్వచనం?

i) let rec sum x y:
```
    return x + y
```
ii) let disp :
```
    print 'welcome'
```
iii) let rec sum num:
```
    if (num!=0) then return num + sum (num-1)
    else
    return num
```


12th_CS_Telugu_Medium_Chapter_1.indd 9 27-03-2023 10:36:27

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. ఇంటర్‌ఫేస్ యొక్క లక్షణాలను పేర్కొనుము.
2. స్ట్రెలిన్‌ను pure ఫంక్షన్ అని ఎందుకు పిలుస్తారు?
3. impure ఫంక్షన్ యొక్క దుష్ప్రభావం ఏమిటి? ఉదాహరణ ఇవ్వండి.
4. Pure మరియు impure ఫంక్షన్‌ల బేధాలను తెలుపుము?
5. మీరు ఫంక్షన్ వెలుపల వేరియబుల్‌లో సవరించినట్లయితే ఏమి జరుగుతుంది? ఒక ఉదాహరణ ఇవ్వండి.

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. పారామితులు అనగానేమి? మరియు క్రింది వాటి గురించి రాయండి.

   1. డేటా టైపు లేని పారామితి 2. డేటా టైపు గల పారామితి

2. క్రింది ప్రోగ్రామ్‌ను గుర్తించండి.

```
let rec gcd a b :=
    if b <> 0 then gcd b (a mod b) else return a
```

   i. ఫంక్షన్ పేరు

   ii. ఇది పునరావృత ఫంక్షన్ అని చెప్పే స్టేట్‌మెంట్‌ను గుర్తించండి.

   iii. పారామితులు ఉన్న వేరియబుల్ పేరు

   iv. ఫంక్షన్‌ను పునరావృతంగా ప్రారంభించే స్టేట్‌మెంట్.

   v. పునరావృతాన్ని ముగించే స్టేట్‌మెంట్.

3. Pure మరియు impure ఫంక్షన్‌లను ఉదాహరణతో వివరించండి.
4. ఇంటర్‌ఫేస్ మరియు అమలుపరచడం(Implementation)ను ఉదాహరణలతో వివరించండి.

REFERENCES

1. *Data Structures and Algorithms in Python By Michael T.Goodrich, RobertoTamassia and Michael H. Goldwasser.*
2. *Data Structure and Algorithmic Thinking in Python By Narasimha Karumanchi*
3. *https://www.python.org*

12th_CS_Telugu_Medium_Chapter_1.indd 10 27-03-2023 10:36:27

యూనిట్ I

అధ్యాయం 2

డేటా సంగ్రహణ

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశము అభ్యసించిన తరువాత విద్యార్థి తెలుసుకొనునవి

- సంగ్రహణ డేటా నిర్మాణాలు అనగానేమి?
- సంగ్రహణ డేటా రకము
- సంయుక్త మరియు సంగ్రహణ అమలుపరచుటలో గల బేధాలు
- జతలు
- డేటా సంగ్రహణలో నిర్మాణం

2.1 డేటా సంగ్రహణ – పరిచయం

డేటా సంగ్రహణ అనునది కంప్యూటర్ సైన్స్‌లో ఒక శక్తివంతమైన భావన. ఇది ప్రోగ్రామర్స్‌కి ప్రోగామ్ కోడ్ ఒక object గా పరిగణించడానికి అనుమతిస్తుంది. ఉదాహరణకు కారు, పెన్సిల్ మరియు ప్రజలు మొదలగు వాటిని ప్రోగ్రామ్ object గా పరిగణిస్తుంది. కోడ్ ఎలా అమలు చేయబడుతుందో గురించి ప్రోగ్రామర్‌లు ఆందోళన చెందాల్సిన అవసరం లేదు. ప్రోగ్రామర్‌లు objects ఏమి చేస్తున్నది వారికి మాత్రమే తెలుసును.

చాలామంది ప్రోగ్రామర్‌లు ఒక ప్రాజెక్టును చేస్తున్నపుడు ఇది చాలా ముఖ్యమైన. ఇక్కడ ప్రాజెక్టు ప్రోగామింగ్‌ను సూచిస్తుంది. డేటా సంగ్రహణతో మీ సమూహ సభ్యులు అర్థం చేసుకోవడానికి మీ కోడ్ యొక్క ప్రతి వాక్యమును చదవాల్సిన అవసరం లేదు. ఇది పనిచేస్తుందని ఊహించవచ్చు.

సంగ్రహణ మాడ్యులారిటిని అందిస్తుంది. (మాడ్యులారిటి అంటే ఒక ప్రోగ్రామ్స్‌ను అనేక మాడ్యులుగా విభజించడం) Classes (నిర్మాణాలు) అనునవి సంగ్రహణ డేటా రకమును (ADT) సూచిస్తుంది.

2.2 సంగ్రహణ డేటా – రకాలు

సంగ్రహణ డేటా రకము Abstract Data Type ADT అనునది object type కొరకు డేటా రకము (లేదా Class) అగును. దీని ప్రవర్తన సమూహ విలువలు మరియు కార్యకలాపాలను నిర్వచించబడుతుంది.

ADT యొక్క నిర్వచనం ఏ కార్యకలాపాలను నిర్వహించాలో మాత్రమే సూచిస్తుంది. కాని ఈ కార్యకలాపాలు ఎలా అమలు చేయడానికి ఏ అల్గోరిధమ్‌లు ఉపయోగించబడుతుందో ఇది పేర్కొనదు. ఇది అమలుకు స్వతంత్ర వీక్షణను ఇస్తుంది. కాబట్టి దీనిని "సంగ్రహణ" అని పిలుస్తారు. అవసరమైన వాటిని మాత్రమే అందించడం మరియు వివరాలను దాచడం అనే ప్రక్రియను "సంగ్రహణ" అంటారు.

ఈ ADT లు ఎలా ప్రాతినిధ్యం వహిస్తాయో మరియు కార్యకలాపాలు నిర్వహించబడుతున్నాయో, ఈ నిర్వచనాలు పేర్కొనలేదని మీరు చూడవచ్చు. ADT ని అమలుచేయడానికి వివిధ మార్గాలు ఉండవచ్చును. ఉదాహరణకు List ADT లు ఒకే లింక్డ్ లిస్ట్ లేదా రెట్టింపు లింక్డ్ లిస్ట్‌ను ఉపయోగించి అమలు చేయవచ్చు. అదేవిధంగా స్టేక్ (stack) ADT మరియు క్యూ (Queue) ADT జాబితాలను ఉపయోగించి అమలు చేయవచ్చును.

12th_CS_Telugu_Medium_Chapter_2.indd 11 12-04-2023 11:14:34

డేటా సంగ్రహణ మనం ప్రపంచం గురించి ఎలా ఆలోచిస్తామో ప్రతిబింబిస్తుంది. ఉదాహరణకు మీరు కారు నడపాలనుకున్నపుడు ఇంజిన్ ఎలా నిర్మించబడిందో లేదా టైర్లు ఎలాంటి పదార్థంతో తయారుచేయబడిందో మీరు తెలుసుకోవలసిన అవసరం లేదు. మీరు కారు ఎలా నడపాలి అని మాత్రమే తెలుసుకోవాలి. డేటా సంగ్రహణను సులభతరం చేయడానికి మీరు రెండు రకాల ఫంక్షన్‌లను సృష్టించాలి. కన్స్ట్రక్టర్లు ***(constructors)*** మరియు సెలెక్టర్లు ***(selectors)***.

2.3 కన్స్ట్రక్టర్లు (constructors) మరియు సెలక్టర్లు (selectors)

కన్స్ట్రక్టర్లు సంగ్రహణ డేటా రకాన్ని నిర్మించే ఫంక్షన్లు. సెలెక్టర్లు డేటా రకం నుండి సమాచారాన్ని తిరిగి పొందే ఫంక్షన్స్.

ఉదాహరణకు city అని పిలువబడే సంగ్రహణ డేటా రకము వుందని అనుకొనుము. ఈ city అనే object నగరం యొక్క పేరు మరియు దాని అక్షాంశం మరియు రేఖాంశాన్ని కలిగి వుంటుంది. city అను object ను సృష్టించడానికి ఇటువంటి ఫంక్షన్‌ను ఉపయోగించవచ్చును.

city = makecity (name, lat, lon)

city అనే object యొక్క సమాచారాన్ని సేకరించడానికి మీరు క్రింది ఫంక్షన్‌లను ఉపయోగిస్తారు.

- getname(city)
- getlat(city)
- getlon(city)

క్రింది స్యూడో కోడ్ (pseudo code) రెండు city objects మధ్య దూరాన్ని లెక్కిస్తుంది.

```
distance(city1, city2):
    lt1, lg1 := getlat(city1), getlon(city1)
    lt2, lg2 := getlat(city2), getlon(city2)
    return ((lt1 - lt2)**2 + (lg1 - lg2)**2))^1/2
```

పై కోడ్‌లో distance () మరియు getlon () మొదలగు వాటిని ఫంక్షన్‌లుగా చదవండి. మరియు lt అక్షాంశం మరియు lg రేఖాంశంగా చదవండి. := ను “assigned as” లేదా “becomes” గా చదవండి.

lt1, lg1 := getlat(city1), getlon(city1)

గమనించుము lt1 అనునది (city1) యొక్క విలువ అవుతుంది. మరియు lg1 అనునది (city1) యొక్క విలువ అవుతుంది.

ఈ ఫంక్షన్‌లు ఎలా అమలుచేయబడ్డాయో తెలుసుకోవలసిన అవసరం లేదని గమనించండి. వేరొకరు మనకోసం వాటిని నిర్వహించారని ఊహించవలెను.

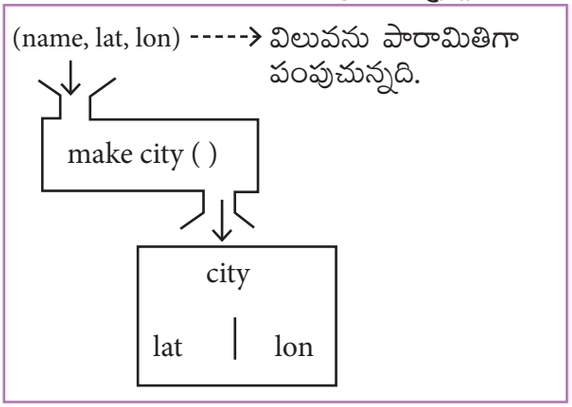
వినియోగదారులకు ఫంక్షన్‌లు ఎలా అమలు చేయబడిందో తెలియకపోతే పరవాలేదు, అయినప్పటికీ ఫంక్షన్‌లను ఇప్పటికీ ఎవరి చేతనైన నిర్వహించబడి ఉండాలి

పై కోడ్‌లో కన్స్ట్రక్టర్లను మరియు సెలెక్టర్‌లను గుర్తిద్దాం.

కన్స్ట్రక్టర్లు సంగ్రహణ డేటా రకాన్ని నిర్మించే ఫంక్షన్ అని ఇప్పటికే తెలుసు. పై pseudo code city యొక్క object ను సృష్టించే ఫంక్షన్ కన్స్ట్రక్టర్

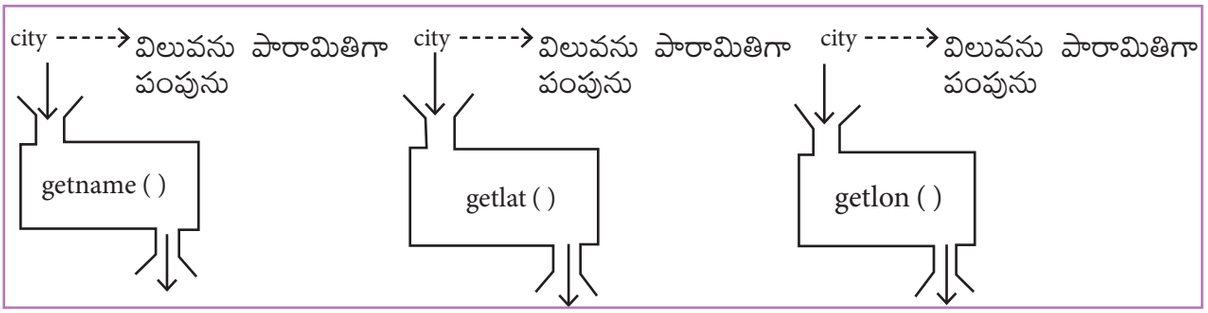
city = makecity (name, lat, lon)

ఇచ్చట makecity (name, lat, lon) అనునది city అనే object సృష్టించే కన్స్ట్రక్టర్లు.



పటం 1 కన్స్ట్రక్టర్

12th_CS_Telugu_Medium_Chapter_2.indd 12 12-04-2023 11:14:34



డేటా రకం నుండి సమాచారాన్ని తిరిగి పొందే ఫంక్షన్ సెలెక్టర్ తప్ప ఇంకేమి కాదు. అందువల్ల పై కోడ్‌లో

- getname(city)
- getlat(city)
- getlon(city)

అనునవి సెలెక్టర్లు, ఎందుకంటే ఈ ఫంక్షన్‌లు city, object యొక్క సమాచారాన్ని సంగ్రహిస్తుంది.

ఇపుడు మనం కన్స్‌ట్రక్టర్‌లు మరియు సెలెక్టర్‌లను గుర్తించడానికి మరొక ఉదాహరణకు పరిశీలిద్దాం. ఈ గుర్తును వ్యాఖ్యలు (comments) గా చదవాలి.

```
- - constructor
makepoint(x, y):
    return x, y
- - selector
 xcoord(point):
    return point[0]
- -selector
ycoord(point):
    return point[1]
```

గమనిక

ఒక సంగ్రహణ డేటా రకము (ADT) ను నిర్వచించడం ద్వారా డేటా సంగ్రహణకు మద్ధతు వుంది ఇది కనెక్టర్లు మరియు సెలెక్టర్ల సమూదాయం. కనెక్టర్లు ఒక object ను సృష్టిస్తుంది. వేర్వేరు సమాచార భాగాలను కలుపుతుంది. అయితే సెలెక్టర్లు object నుండి వ్యక్తిగత సమాచారాన్ని సేకరిస్తుంది.

2.4 అకరణీయ సంఖ్యలనుపయోగించి సంగ్రహణ డేటా రకము యొక్క ప్రాతినిధ్యం

సంగ్రహణ డేటా యొక్క ప్రాథమిక ఆలోచన ప్రోగ్రామ్‌లను రూపొందించడం తద్వారా అవి సంగ్రహణ డేటాపై పనిచేస్తాయి. అంటే ప్రోగ్రామ్‌లో డేటాను ఉపయోగిస్తున్నపుడు దాన్ని గురించి కొన్ని అంచనాలు చేయడం. ఆ సమయంలో కాంక్రీటు డేటా ఉత్పత్తి చేసేటపుడు అది ప్రోగ్రామ్ యొక్క వ్యక్తిగత భాగంగా నిర్వహిస్తుంది.

గమనిక

కాంక్రీటు డేటా ప్రాతినిధ్యంలో ప్రతి ఫంక్షన్ యొక్క నిర్వచనం తెలిసి ఉండాలి.

అన్ని ప్రోగ్రామ్‌లు రెండు భాగాలను కలిగి వుంటుంది. ఒక ప్రోగ్రామ్ యొక్క రెండు భాగాలు సంగ్రహణ డేటాపై పనిచేసే భాగం మరియు కాంక్రీటు ప్రాతినిధ్యంను నిర్వహించే భాగం, కాంక్రీటు ప్రాతినిధ్య పరంగా సంగ్రహణ డేటాను అమలుచేసే చిన్న ఫంక్షన్‌ల ద్వారా అనుసంధానించబడి వుంటుంది. ఈ పద్ధతిని వివరించడానికి, అకరణీయ సంఖ్యలుగా మార్చడానికి ఫంక్షన్‌ల సమితిని రూపొందించడానికిఒక ఉదాహరణను పరిశీలిద్దాం.

ఉదాహరణ

ఒక అకరణీయ సంఖ్య అనునది పూర్ణ సంఖ్యల నిష్పత్తి అగును, మరియు అకరణీయ

12th_CS_Telugu_Medium_Chapter_2.indd 13 12-04-2023 11:14:34

సంఖ్యలు వాస్తవ సంఖ్యల యొక్క ముఖ్యమైన ఉపతరగతి 8/3 లేక 19/23 వంటి అకరణీయ సంఖ్యలను సాధారణంగా ఇలా రాయబడుతుంది.

<లవము>/<హారము>

ఇక్కడ <లవము> మరియు <హారము> రెండూ పూర్ణసంఖ్యల విలువలను కలిగివుండును. వాస్తవానికి పూర్ణ సంఖ్యలను విభజించి ఉజ్జాయింపును ఉత్పత్తి చేస్తుంది. పూర్ణసంఖ్యల యొక్క ఖచ్చితమైన ఖచ్చితత్వాన్ని కోల్పోతుంది.

8/3 =2.66666666666666665

అయితే మీరు లవము మరిము హారములను కలపడం ద్వారా అకరణీయ సంఖ్యల ఖచ్చితమైన ప్రాతినిధ్యాన్ని సృష్టించవచ్చు.

ఫంక్షన్ సంగ్రహణలను ఉపయోగించడం నుండి మనకు తెలిసినట్లుగా ప్రోగ్రామ్ యొక్క కొన్ని భాగాలను ఉత్పాదకంగా అమలుచేయడంద్వారా ప్రోగ్రామింగ్‌ను ప్రారంభించవచ్చు. మీకు ఇప్పటికే ఒక లవము మరియు హారము నుండి అకరణీయ సంఖ్యను నిర్మించే మార్గాన్ని తెలిసియున్నారని ఊహించడం ద్వారా ప్రారంభిద్దాం. అకరణీయ సంఖ్యను ఇచ్చినట్లయితే దాని లవము మరియు హారమును ఎన్నుకొనే మార్గం వున్నదని అనుకుందాం.

ఇప్పుడు అకరణీయ సంఖ్యల యొక్క నిర్వహణ సెలెక్టర్ ఫంక్షన్‌లు లవము మరియు హారము; మరియు కన్స్‌స్ట్రక్టర్ ఫంక్షన్‌లు హేతుబద్ధతతో నిర్వహించబడుతాయి. కాని ఈ ఫంక్షన్‌లు నిర్వహించబడలేదు. లవము మరియు హారమును కలపడం ద్వారా వచ్చు మిశ్రమ విలువ మనకు అవసరం.

గమనిక

Visual thinking అనేది వాస్తవికత కంటే నమ్మకం ఆధారంగా నిర్ణయాలు తీసుకునే చర్య.

ఉదాహరణ : అకరణీయ సంఖ్య యొక్క ADT

- - constructor
- - constructs a rational number with numerator n, denominator d

rational(n, d)

- - selector

numer(x) → returns the numerator of rational number x

denom(y) → returns the denominator of rational number y

పై సూడోకోడ్‌ను ఉపయోగించి కన్స్‌స్ట్రక్టర్ల భాగాలు కరణీయ సంఖ్యలను సూచించును.

కన్స్‌స్ట్రక్టర్లు మరియు సెలెక్టర్లు కూడా అందుబాటులో వున్నాయని అనుకుందాం.

ప్రోగ్రామ్‌ల రూపకల్పన కోసం మనం ఇక్కడ ఒక శక్తివంతమైన వ్యూహాన్ని ఉపయోగిస్తున్నాం. “Visual thinking” మనం ఇంకా అకరణీయ సంఖ్యను ఏ విధంగా సూచించాలో చెప్పలేదు, లేక కన్స్‌స్ట్రక్టర్ మరియు సెలెక్టర్లును ఎలా అమలుచేయాలో చెప్పలేదు.

x,y:=8,3

rational(n,d)

numer(x)/denom(y)

- - output : 2.66666666666666665

12th_CS_Telugu_Medium_Chapter_2.indd 14 12-04-2023 11:14:34

2.5 Lists,Tuples (లిస్ట్స్, టుపుల్స్)

డేటా సంగ్రహణ యొక్క కాంక్రీటు స్థాయిని అమలుచేయడానికి పైథాన్ వంటి కొన్ని భాషలు ***Pairs*** అని పిలువబడే సమ్మేళ నిర్మాణాన్ని అందిస్తుంది. ***Pairs*** అనునది లిస్ట్ మరియు టుపుల్‌తో రూపొందించబడినది. పెయిర్స్‌లను అమలుచేయడానికి సులభం.

2.5.1 List (లిస్ట్)

"లిస్ట్ అనునది కామాలతో వేరుచేయబడిన చదరపు కుండలీకరణం *(square brackets)* లోపల సమాచారమును ఉంచడం ద్వారా నిర్మించబడింది." ఇటువంటి సమాచారమును "లిస్ట్ లిటరల్" అంటారు. లిస్ట్ అనునది బహుళ విలువలను నిల్వచేస్తుంది. ప్రతివిలువ ఏ రకమైనదైనా కావచ్చు, మరియు మరొక లిస్ట్‌కు కూడా చెందవచ్చును.

ఉదాహరణకు లిస్ట్ అనునది (10,20)

లిస్ట్‌లోని మూలకాలను రెండు విధాలుగా సంప్రదించవచ్చు. మొదటి మార్గం మనకు తెలిసిన బహుళ అసైనమెంట్ పద్ధతి, ఇది లిస్ట్‌ను దాని మూలకాలను (Unlike) చేస్తుంది, మరియు ప్రతిమూలకాన్ని వేరే పేరుతో బందిస్తుంది.

```
lst := [10, 20]
x, y := lst
```

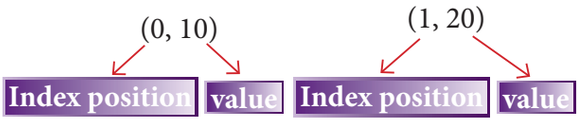
పై ఉదాహరణలో ***x*** విలువ '10' అవుతుంది. ***y*** విలువ '20' అవుతుంది.

లిస్ట్‌లోని మూలకాలను యాక్సెస్ చేయడానికి రెండవ పద్ధతి ఎలిమెంట్ సెలెక్సెన్ ఆపరేటర్, చదరపు కుండలీకరణగుర్తును ఉపయోగించి కూడా వ్యక్తీకరించబడుతుంది. లిస్ట్ అక్షరాలుగా కాకుండా మరొక వ్యక్తీకరణను నేరుగా అనుసరించే చదరపు కుండలీకరణాలు వ్యక్తీకరణ లిస్ట్ విలువను మూల్యాంకనం చేయదు. బదులుగా మునుపటి వ్యక్తీకరణ విలువ నుండి ఒక మూలకాన్ని ఎంచుకుంటుంది.

```
lst[0]
10
lst[1]
20
```

పైన పేర్కొన్న రెండు ఉదాహరణల యందు గణితశాస్త్రపరంగా మనం సమితికి సమానమైన జాబితాను (లిస్ట్‌ను) సూచించవచ్చు.

lst[(0, 10), (1, 20)] - where



ఏదైనా రెండు విలువలను ఒకదానితో ఒకటి కలుపుకునే మార్గాన్ని ఒక ***pair*** (జత)గా పరిగణించవచ్చును. లిస్ట్‌లు అలా చేయడం ఒక సాధారణ పద్ధతి అందువల్ల లిస్ట్‌ను (జత) ***Pair*** అని పిలుస్తారు.

లిస్ట్‌నుపయోగించి అకరణీయ సంఖ్యల యొక్క ప్రాతినిధ్యం:

మీరిపుడు అకరణీయ సంఖ్యను స్యుడోకోడ్‌లో (pseudo code) రెండు పూర్ణసంఖ్యల జతగా సూచించవచ్చును. అది ఒక లవము మరియు ఒక హారము.

```
rational(n, d):
    return [n, d]
numer(x):
    return x[0]
denom(x):
    return x[1]
```

2.5.2 టుపుల్ (Tuple)

ఒక pair అనేది రెండు ఇతర డేటాను కలిగివున్న సమ్మేళన డేటా టైపు అని గుర్తించండి.

12th_CS_Telugu_Medium_Chapter_2.indd 15 12-04-2023 11:14:34

ఇప్పటి వరకు pair డేటా టైపు సూచించే రెండు మార్గాలను మేము మీకు అందించాం. మొదటి మార్గం జాబితా (లిస్ట్) నిర్మాణాన్ని ఉపయోగించడం మరియు రెండవ మార్గం టుపుల్ నిర్మాణంతో pair ను అమలు చేయడం.

ఒక టుపుల్ అనేది కుండలీకరణాలతో చుట్టుముట్టబడిన విలువను కమాలతో వేరుచేయబడిన క్రమం. టుపుల్ లిస్ట్‌కు సమానంగా వుంటుంది. రెండింటికి మధ్య వ్యత్యాసం ఏమిటంటే ఒక టుపుల్ యొక్క మూలకాలను కేటాయించిన తర్వాత దాన్ని మార్చలేరు, అయితే లిస్ట్‌లో మూలకాలను మార్చవచ్చును.

ఉదాహరణకు రంగు = ('red', 'blue', 'Green')

టుపుల్‌లను Pair గా సూచించుట :

```
nums := (1, 2)
nums[0]
1
nums[1]
2
```

గమనించండి మీరు pair లో నిల్వచేసిన డేటాను యాక్సస్ చేయడానికి చదరపు కుండలీకరణాలను ఉపయోగించబడును. డేటా సున్నీ ఇండెక్స్ చేయబడింది, అంటే మీరు మొదటి మూలకాన్ని nums[0] తో మరియు రెండవదాన్ని nums[1] తో యాక్సస్ చేస్తారు.

2.6 నిర్మాణంలో డేటా సంగ్రహణ

లిస్ట్ డేటా సంగ్రహణను అనుమతిస్తుందని మీకు ఇప్పటికే తెలుసు, కాబట్టి మెమొరీ కణాల సమితికి పేరుపెట్టవచ్చు. ఉదాహరణకు మాస్టర్‌మైండ్ ఆటలో ఆటగాడు ఊహించే నాలుగు రంగుల లిస్ట్‌ను, ఆటగాడును ఊహించే నాలుగు రంగుల pair ను మీరు ట్రాక్ చేయాలి. నాలుగు వేర్వేరు వేరియబుల్స్ (కలర్1, కలర్2, కలర్ 3, కలర్ 4) ను ఉపయోగించడానికి బదులుగా ఒక వేరియబుల్ Predict. ఉదాహరణకు

Predict =['red', 'blue', 'green', 'green']

లిస్ట్ అనునది వస్తువు యొక్క వివిధ భాగాలకు పేరుపెట్టడానికి అనుమతించదు. mode లో మీరు భాగాలకు పేరుపెట్టవలసిన అవసరం లేదు. మీరు ప్రతి కాలంను టేబుల్‌తో పొందవచ్చు.

కాని ఒక వ్యక్తిలాగే మరింత సంక్లిష్టమైన విషయాలలో మనకు బహుళ అంశాల వస్తువు వుంది. ఇక్కడ ప్రతి అంశం పేరుపెట్టబడినది. మొదటి పేరు, చివరిపేరు, ఐడి (గుర్తింపు) మరియు ఇ-మెయిలు మొదలైనవి అంశాలగును. ఒక వ్యక్తిని సూచించడానికి ఒక లిస్ట్‌ను ఉపయోగించవచ్చు.

person=['Padmashri', 'Baskar', '994-222-1234', 'compsci@gmail.com']

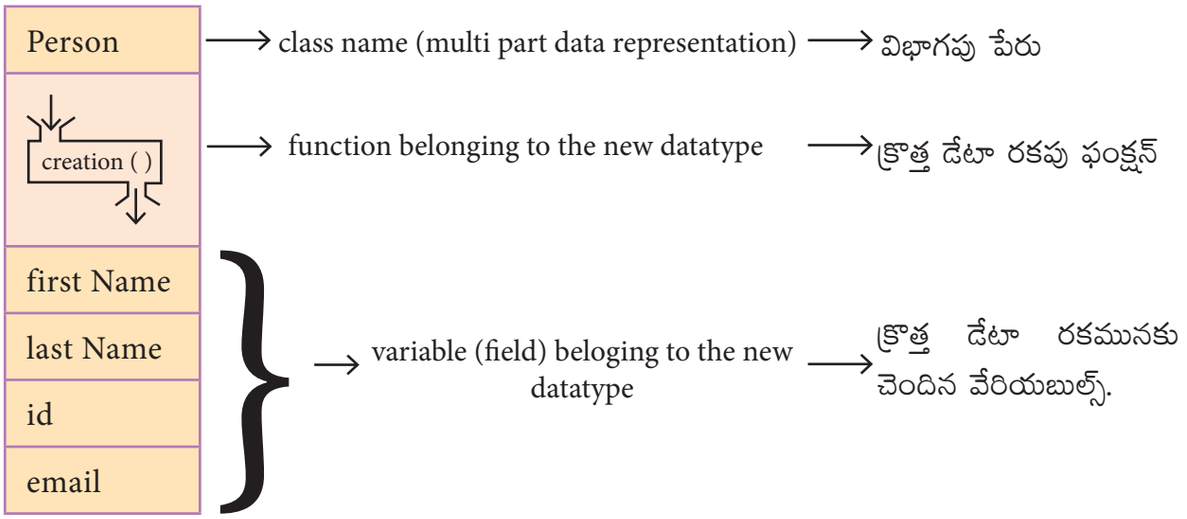
కాని అలాంటి ప్రాతినిధ్యం ప్రతిభాగం దేనిని సూచిస్తుందో స్పష్టంగా పేర్కొనలేదు.

ఈ సమస్య నందు లిస్ట్‌ను ఉపయోగించడానికి బదులుగా ప్రతిభాగం పేరుపెట్టబడిన వస్తువులను సూచించడానికి మీరు structure (OOP భాషలలో దీనిని క్లాస్ కన్స్ట్రక్టర్ అని పిలుస్తారు.) ఉపయోగించవచ్చు. (పేరు ఇవ్వబడింది) క్రింది స్యుడో కోడ్‌ను పరిగణించండి.

```
class Person:
    creation( )
    firstName := " "
    lastName := " "
    id := " "
    email := " "
```


12th_CS_Telugu_Medium_Chapter_2.indd 16 12-04-2023 11:14:34

Person అను కొత్త డేటా రకము పటం రూపంలో సూచించవచ్చును.



main() క్రిందివాటిని కలిగి వుండును.	
p1:=Person()	object సృష్టించు ప్రవచనము
firstName := " Padmashri "	firstName అను field కు పద్మశ్రీ అను విలువ ఇవ్వబడుతుంది.
lastName :="Baskar"	lastName అను field కు Baskar అను విలువ ఇవ్వబడుతుంది.
id :="994-222-1234"	id అను field కు విలువ 994-222-1234 ఇవ్వబడింది.
email="compsci@gmail.com"	email అను field కు compsci@gmail.com అను విలువ ఇవ్వబడింది.
- - output of firstName : Padmashri	

class (structure) నిర్మాణం, ఒక వ్యక్తిని సూచించే బహుళభాగాల వస్తువుల రూపాన్ని నిర్వహిస్తుంది. దీన్ని నిర్వచనం క్రొత్త డేటా రకాన్ని జోడిస్తుంది. ఈ సందర్భంలో వ్యక్తి అనే రకం ఒక సారి నిర్వచించగా మనం డేటారకం యొక్క క్రొత్త వేరియబుల్స్ ఉదాంతాలు సృష్టించవచ్చు. ఈ ఉదాహరణలో వ్యక్తిని class లేదా టైపుగా సూచిస్తారు. అయితే ***p1*** ను ఒక objects లేదా ఉదాంతాలుగా సూచిస్తారు. వేరే person ను కుకీ కట్టర్‌గా మరియు p1 ను ప్రత్యేకమైన కుకీగా భావించవచ్చు. కుకీ కట్టర్‌ను ఉపయోగించి మీరు చాలా కుకీలను తయారు చేయవచ్చు. class ఉపయోగించి అదేవిధంగా మీరు ఆ రకమైన అనేక objects ను సృష్టించవచ్చు.

ఇప్పటివరకు మీరు సంబంధిత డేటా అంశాలను సమూహపరచడం ద్వారా ఒక class డేటా సంగ్రహణను ఎలా నిర్వచిస్తుందో చూశారు. ఒక class కేవలం డేటా మాత్రమే కాదు దానిలో నిర్వచించిన ఫంక్షన్‌లు ఉన్నాయి. అటువంటి ఫంక్షన్‌ను class ఆధీనంలో వున్నాయని చెప్పవచ్చు. ఎందుకనగా class డేటాతో పనులు చేయడం వాటి యొక్క పని. ఉదాహరణకు వ్యక్తి object యొక్క డేటాను సవరించడానికి లేదా విశ్లేషించడానికి.

అందువల్ల మనము ఒక class ని బండిల్డ్ డేటా మరియు ఆ డేటాపై పనిచేసే ఫంక్షన్‌గా నిర్వచించవచ్చు. పైన పేర్కొన్న ఉదాహరణ మరియు వివరణ నుండి డేటా సంగ్రహణ యొక్క అందం ఏమిటంటే సంక్లిష్ట డేటాను మనం చాలా సరళమైన రీతిలో నిర్వహించడానికి తోడ్పడుతుంది.

12th_CS_Telugu_Medium_Chapter_2.indd 17 12-04-2023 11:14:34

గుర్తించుకోవలసిన విషయాలు

- సంగ్రహణ డేటా టైపు (ADT) అనేది object కోసం లేదా class యొక్క సమితి మరియు కార్యాకలాపాల సమితి ద్వారా నిర్వచించబడుతుంది.
- (ADT) యొక్క నిర్వచనం ఏ కార్యకలాపాలను నిర్వహించాలో మాత్రమే సూచిస్తుంది. కానీ ఈ కార్యకలాపాలు ఎలా అమలుచేయబడతాయో కాదు.
- మెమొరిలో డేటా ఎలా నిర్వహించబడుతుందో మరియు కార్యకలాపాలను అమలుచేయడానికి ఏ ఆల్గారిథమ్‌ను ఉపయోగించబడుతాయో (ADT) పేర్కొనలేదు.
- కన్స్ట్రక్టర్లు ఫంక్షన్ సంగ్రహ డేటా టైపును నిర్మిస్తుంది. (ఫంక్షన్స్)
- సెలెక్టర్లు ఫంక్షన్స్ డేటా టైపు నుండి సమాచారాన్ని తిరిగి పొందడానికి ఉపయోగపడుతుంది.
- కాంక్రీటు డేటా టైపు లేదా structures (CDT's) సాపేక్ష సాధారణ భావన యొక్క ప్రత్యక్ష అమలగును.
- సంగ్రహణ డేటా రకాలు (ADT's) దాని అమలు నుండి స్వతంత్ర్యంగా ఒక భావన యొక్క ఉన్నత స్థాయి వీక్షణను (మరియు ఉపయోగం) అందిస్తాయి.
- కాంక్రీటు డేటా రకం డేటా టైపు యొక్క ప్రాతినిధ్యం తెలిసినది మరియు సంగ్రహణ డేటా టైపులో డేటా టైపు యొక్క ప్రాతినిధ్యం తెలియదు.
- Pair (జత)లు అనేది ఒక సమ్మేళన నిర్ణయం. ఇది లిస్ట్‌తో లేదా 'టుపుల్' తో తయారుచేయబడుతుంది.
- లిస్ట్ అనునది కమాలతో వేరుచేయబడిన చదరపు కుండలీకరణం లోపల ఉంచడం ద్వారా నిర్మించబడింది.
- లిస్ట్‌లోని మూలకాలను రెండు విధాలుగా యాక్సెస్ చేయవచ్చు. మొదటి మార్గం బహుళ అసైనమెంట్ ద్వారా మరియు రెండవ పద్ధతి ఎలిమెంట్ సెలెక్షన్ ఆపరేటర్ ద్వారా చేయవచ్చు.
- రెండు విలువలను ఒక దానితో ఒకటి కలుపుతూ ఒక Pair గా పరిగణించవచ్చు.
- లిస్ట్ బహుళ అంశాల Objects యొక్క వివిధ భాగాలకు పేరుపెట్టడానికి అనుమతించదు.

NK14H

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. క్రిందివాటిలో ఏ ఫంక్షన్ సంగ్రహణ డేటా టైపును నిర్మించును?

ఎ) కన్స్ట్రెక్టర్స్ బి) డిస్ట్రక్టర్స్ సి) పునరావృత డి) నెస్టెడ్

2. క్రింది వాటిలో ఏ ఫంక్షన్ డేటా టైపు నుండి సమాచారాన్ని తిరిగి పొందుతుంది?

ఎ) కన్స్ట్రెక్టర్స్ బి) సెలెక్టర్స్ సి) పునరావృత డి) నెస్టెడ్

12th_CS_Telugu_Medium_Chapter_2.indd 18 12-04-2023 11:14:34

3. క్రమబద్ధీకరించిన వరుస మూలకాలను భర్తీచేయగల డేటా నిర్మాణాన్ని ఏమంటారు?

ఎ) Built in　　బి) List　　సి) Tuple　　డి) Derived data

4. మార్పు లేని Object క్రమాన్ని ఏమంటారు?

ఎ) Built in　　బి) List　　సి) Tuple　　డి) Derived data

5. ప్రాతినిధ్యం తెలిసిన డేటా టైపును ఏమంటారు?

ఎ) Built in datatype　　బి) Derived datatype

సి) Concrete datatype　　డి) Abstract datatype

6. ప్రాతినిధ్యం తెలియని డేటా టైపును ఏమంటారు?

ఎ) Built in datatype　　బి) Derived datatype

సి) Concrete datatype　　డి) Abstract datatype

7. క్రింది వాటిలో ఏది సమ్మేళన నిర్మాణం?

ఎ) Pair　　బి) Triplet　　సి) single　　డి) quadrat

8. రెండు విలువలు ఒక దానితో ఒకటి కలుపుకోవడాన్ని ఏ విధంగా పరిగణించవచ్చు.

ఎ) Pair　　బి) Triplet　　సి) single　　డి) quadrat

9. క్రిందివాటిలో బహుళ అంశాల object యొక్క వేర్వేరు భాగాలు ఏవి?

ఎ) Tuples　　బి) Lists　　సి) Classes　　డి) quadrats

10. చదరపు కుండలీకరణంలో సమాసాలను ఉంచడం ద్వారా క్రింది వాటిలో ఏది నిర్మించబడింది.

ఎ) Tuples　　బి) Lists　　సి) Classes　　డి) quadrats

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము.　　**(2 మార్కులు)**

1. సంగ్రహణ డేటా టైపు అనగానేమి?
2. కన్స్ట్రక్టర్లు మరియు సెలెక్టర్లు మధ్య బేధాన్ని తెల్పండి?
3. Pair అనగానేమి? ఒక ఉదాహరణను తెల్పండి?
4. List అనగానేమి? ఒక ఉదాహరణను తెల్పండి?
5. Tuple అనగానేమి? ఒక ఉదాహరణను తెల్పండి?

12th_CS_Telugu_Medium_Chapter_2.indd 19　　12-04-2023 11:14:34

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము. **(3 మార్కులు)**

1. కాంక్రీటు డేటా టైపు మరయు సంగ్రహణ డేటాటైపు మధ్య తేడాలను తెల్పండి?
2. ప్రోగ్రామ్ డిజైనింగ్ కోసం ఏ వ్యూహం ఉపయోగిస్తారు? ఆ వ్యూహాన్ని నిర్వచించండి?
3. క్రింది వాటిలో కన్స్ట్రక్టర్లు మరియు సెలెక్టర్లలను గుర్తించండి.

   (a) N1=number() (b) accetnum(n1) (c) displaynum(n1)

   (d) eval(a/b) (e) x,y= makeslope (m), makeslope(n)

   (f) display()
4. లిస్ట్‌లోని మూలకాలను యాక్సస్ చేయడానికి వివిధ మార్గాలు ఏమిటి? ఉదాహరణ ఇవ్వండి?
5. క్రింది వాటిలో లిస్ట్, టుపుల్ మరియు class (క్లాస్) లను గుర్తించండి?

   (a) arr [1, 2, 34] (b) arr (1, 2, 34) (c) student [rno, name, mark]

   (d) day= ('sun', 'mon', 'tue', 'wed') (e) x= [2, 5, 6.5, [5, 6], 8.2]

   (f) employee [eno, ename, esal, eaddress]

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము. **(5 మార్కులు)**

1. మీరు డేటా సంగ్రహణను ఎలా సులభతరం చేస్తారు? తగిన ఉదాహరణతో వివరించండి?
2. List అనగానేమి? లిస్ట్‌ను Pairs అని ఎందుకు పిలుస్తారు? తగిన ఉదాహరణతో వివరింపుము?
3. మీరు బహుళ అంశాన్ని ఎలా యాక్సస్ చేస్తారు? ఉదాహరణతో వివరించండి?

Reference Books

1. *Data structure and algorithmic thinking with python by narasimha karumanchi*
2. *sign and analysis of algorithms by s sridhar*
3. *Data Structures and Algorithms in Python by Goodrich, Tamassia & Goldwasser*
4. *https://www.tutorialspoint.com*

12th_CS_Telugu_Medium_Chapter_2.indd 20 12-04-2023 11:14:34

యూనిట్ I

అధ్యాయం 3

పరిధిలు

I7SCG

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశం పూర్తి అయిన తరువాత విద్యార్థులు తెలుసుకొనునవి.

- పరిధిని (Scope) అవగాహన చేసుకొనుట.
- LEGB నియమాన్ని అమలుచేయుట.
- మాడ్యుల్స్ అనగానేమి అని అవగాహన చేసుకొనుట.
- ప్రోగ్రామింగ్ భాషలో యాక్సస్ నియంత్రణ అమలును అవగాహన చేసుకొనుట.

3.1 పరిధి యొక్క పరిచయం

పరిధి అనునది వేరియబుల్స్, పారామితులు మరియు ఫంక్షన్‌లు అనునవి ఒక ప్రోగ్రామ్ యొక్క ఒక భాగంలో నుంచి అదే ప్రోగ్రామ్ యొక్క మరొక భాగానికి దృశ్యమానతను సూచిస్తుంది.

3.2 వేరియబుల్ పరిధి

ప్రోగ్రామింగ్ భాషలో వేరియబుల్స్ పరిధిని అర్థం చేసుకోనడానికి వేరియబుల్స్ నిజంగా ఏమిటో తెలుసుకోవడం చాలా ముఖ్యం. ముఖ్యంగా అవి మెమరీలోని ఒక object కు చిరునామాలు (సూచనలు లేదా పాయింటర్లు) వేరియబుల్‌ను := గుర్తుతో కేటాయిస్తారు. ఉదాహరణకి object ఇక్కడ object ను వేరియబుల్‌తో బంధిస్తున్నారు. (లేదా మ్యాపింగ్ చేస్తున్నారు) బహుళ వేరియబుల్స్ ఒకే ఉదాహరణానికి మ్యాప్ చేయవచ్చును.

గమనిక

ఒక object తో వేరియబుల్ పేరును బంధించే ప్రక్రియను మేపింగ్ అంటారు. = (సమానం సంకేతం) ప్రోగ్రామింగ్ భాషలలో వేరియబుల్ మరియు ఆబ్జెక్ట్‌ను మ్యాప్ చేయడానికి ఉపయోగపడుతున్నది.

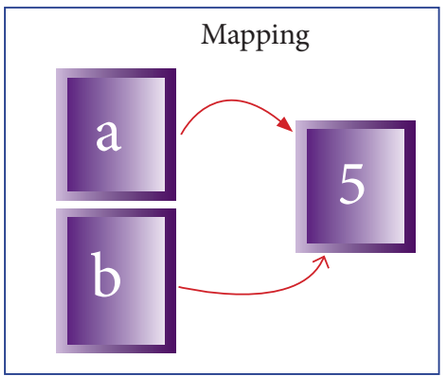
ప్రోగ్రామింగ్ భాషలు namespaces తో ఈ మ్యాపింగ్‌లన్నింటినీ ట్రాక్ చేస్తాయి. ***Namespaces*** వేరియబుల్స్ పేర్లను ***objects***కు మ్యాపింగ్ చేయడానికి కంటైనర్లు అగును. మీరు వాటిని dictionariesగా భావించవచ్చు. పదాల జాబితా మరియు దాని అర్థాలను కలిగి వుండును. పదాలు దాని అర్థంతో dictionary లలో మ్యాప్ చేయబడతాయి, అయితే పేర్లు ప్రోగ్రామింగ్ భాషలో objectతో (name = object) మ్యాప్ చేయబడతాయి. ఇది objectను మీరు ఎంచుకున్న పేర్ల ద్వారా కేటాయించడానికి అనుమతిస్తుంది.

క్రింది ఉదాహరణలో '***a***' మొదట పూర్ణ సంఖ్య '5' కు మ్యాప్ చేయబడుతుంది. ఈ సందర్భంలో '***a***' అనునది వేరియబుల్ పేరు. పూర్ణ సంఖ్య విలువ '5' అయితే ఇది ఒక object.

అపుడు '***b***' ను '***a***' కు సమానంగా చేయబడితే వాస్తవానికి '***b***' విలువ అదే పూర్ణసంఖ్య విలువకి కట్టుబడి వుంటుంది.

1. a:=5
2. b:=a

12th_CS_Telugu_Medium_Chapter_3.indd 21 27-03-2023 10:37:23

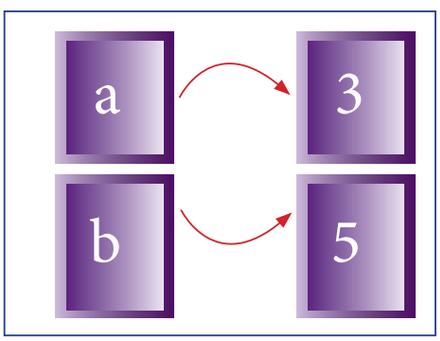


పైన ఇవ్వబడిన ప్రవచనాల యందు కొన్ని మార్పులను క్రింది విధంగా చూద్దాం.

1. a:=5
2. b:=a
3. a:=3

మీరు '***a***' ను '3'కి సమానంగా మార్చినట్లయితే వర్తమాన ప్రోగ్రామర్ '***b***' కూడా '3'కి సమానంగా వుంటుందని ఆశించవచ్చు. కాని అది అలా కాదు. '***b***' ఇప్పటికి '5' అను పూర్ణసంఖ్య విలువకు మ్యాప్ చేయబడింది. (లేదా సూచిస్తుంది) మార్చబడిన ఏకైక విషయం '***a***' ఇది ఇపుడు పూర్ణసంఖ్య విలువ '3' కు మ్యాప్ చేయబడింది.

'***a***' యొక్క విలువ మార్చిన తరువాత మ్యాపింగ్



వేరియబుల్స్ యొక్క పరిధి అది కనిపించే కోడ్ యొక్క భాగం. వాస్తవానికి దీనిని సూచించడానికి మీరు ఎటువంటి (Profixes) ను ఉపసర్గాలాను ఉపయోగించాల్సిన అవసరం లేదు. ఒక ఉదాహరణను తీసుకుందాం.

1. **Disp**():
2. a:=7

మీరు కోడ్ వెలుపల '***a***' విలువలను ప్రదర్శించడానికి ప్రయత్నించినపుడు ప్రోగ్రామ్ ***"name 'a' is not defined"*** అను దోష సమాచారాన్ని ఇస్తుంది. ఎందుకంటే వేరియబుల్ యొక్క జీవిత కాలం కోడ్ చివరి వరకు మాత్రమే వుంటుంది. వేరియబుల్ సజీవంగా వున్న కాలాన్ని దాని ***'life time'*** అని అంటారు.

3.3 LEGB నియమం

విలువను పొందడానికి వేరియబుల్స్ వస్తువుకు మ్యాప్ చేయబడిన క్రమాన్ని కూడా పరిధి నిర్వచిస్తుంది. క్రిందచూపిన. విధంగా ఒక ఉదాహరణను తీసుకుందాం

1. x:= 'outer x variable'
2. **display():**
3. x:= 'inner x variable'
4. print x
5. print x
6. display()

పై అమలుచేయబడినపుడు స్టేట్‌మెంట్ (4) మరియు (5) ఫలితాన్ని ఈ విధంగా ప్రదర్శించును.

అవుట్‌పుట్ (Output)

outer x variable

inner x variable

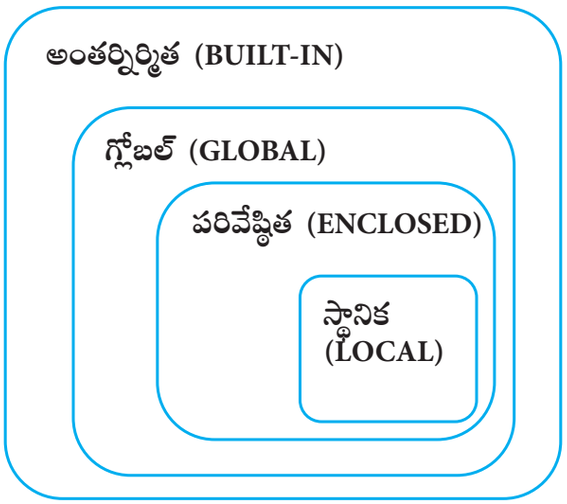
పైన పేర్కొన్న ప్రవచనాలు వేర్వేరు అవుట్‌పుట్‌లను ఇస్తాయి. ఎందుకుంటే ఒకే వేరియబుల్ పేరు ***x*** వేర్వేరు పరిధిలలో వుంటుంది. ఒకటి, ఫంక్షన్ display() లోపల మరియు మరొకటి పై స్థాయిలో వుంటుంది. ఫంక్షన్ నిర్వచనం వెలుపల ***x*** సూచించునపుడు ***'outer xvariable'*** విలువ ముద్రించబడుతుంది. display() అమలు అయినపుడు ***'inner x variable'*** ముద్రించబడుతుంది. ఇది ఫంక్షన్ నిర్వచనం లోపల

12th_CS_Telugu_Medium_Chapter_3.indd 22 27-03-2023 10:37:23

x విలువ. పై ఉదాహరణ నుండి వేరియబుల్ ఏ పరిధి నుండి ఎంచుకోవాలో నిర్ణయించడానికి ఒక నియమం వుందని మనము ఊహించవచ్చు.

పరిధి స్పష్టత (Scope resolution) కోసం పరిధులను శోధించవలసిన క్రమాన్ని నిర్ణయించడానికి ***LEGB*** నియమం ఉపయోగపడుతుంది. సోపాన క్రమం (అత్యధికం నుండి అత్యల్పం) పరంగా పరిధులు క్రింద ఇవ్వబడ్డాయి.

Local(**L**)	ఫంక్షన్ / క్లాస్ లోపల నిర్వచింపబడునవి.
Enclosed(**E**)	పరివేష్ఠిత (enclosing) ఫంక్షన్ లోపల నిర్వచింపబడునవి (నెస్టడ ఫంక్షన్ భావన)
Global(**G**)	ఉన్నతస్థాయి వద్ద నిర్వచింపబడునవి.
Built-in (**B**)	బిల్ట్ – ఇన్ ఫంక్షన్‌లలో రిజర్వు చేసిన పేర్లు (మాడ్యుల్స్)



3.4 వేరియబుల్ పరిధి రకాలు

నాలుగు రకాల వేరియబుల్ పరిధి (Scope) లు వున్నాయి. వాటిని ఒక్కొక్కటిగా చర్చిద్దాం.

3.4.1 స్థానిక పరిధి (Local Scope)

లోకల్ పరిధి ప్రస్తుత ఫంక్షన్‌లో వేరియబుల్‌ను సూచిస్తుంది. ఎల్లపుడు ఒక ఫంక్షన్ దాని స్థానిక పరిధిలో వేరియబుల్ పరిధి కోసం చూస్తుంది. అది అక్కడ కనబడకపోతే బాహ్య పరిధిలో తనిఖీ చేయబడుతుంది.

ఈ ఉదాహరణను చూడండి.

12th_CS_Telugu_Medium_Chapter_3.indd 23 27-03-2023 10:37:23

1. **Disp():** 2. a:=7 3. print a 4. Disp()	ప్రోగ్రామ్ అంతట Disp( ): a:=7 print a Disp ( )	ప్రోగ్రామ్ అవుట్‌పుట్ 7

పై కోడ్‌ను అమలుపరచునపుడు వేరియబుల్ ***a*** విలువ '7' ను ప్రదర్శించినది. ఎందుకనగా అది నిర్వచింపబడినది మరియు స్థానిక ఫరిధిలో లభ్యమయినది.

3.4.2 గ్లోబల్ పరిధి (Global Scope)

ఒక ప్రోగ్రామ్‌యొక్క అన్నీ ఫంక్షన్‌ల వెలుపల డిక్లేర్డ్ చేయబడిన వేరియబుల్‌ను గ్లోబల్ వేరియబుల్ అంటారు. దీని అర్థం ప్రోగ్రామ్‌లోని అన్నీ ఫంక్షన్‌ల లోపల లేదా వెలుపల గ్లోబల్ వేరియబుల్‌ను యాక్సస్ చేయవచ్చు.

ఈ ఉదాహరణను పరిగణిద్దాం.

1. a:=10 2. **Disp**(): 3. a:=7 4. print a 5. Disp() 6. print a	ప్రోగ్రామ్ అంతా a:=10 Disp( ) a:=7 print a Disp1( ): print a	ప్రోగ్రామ్ అవుట్‌పుట్ 7 10

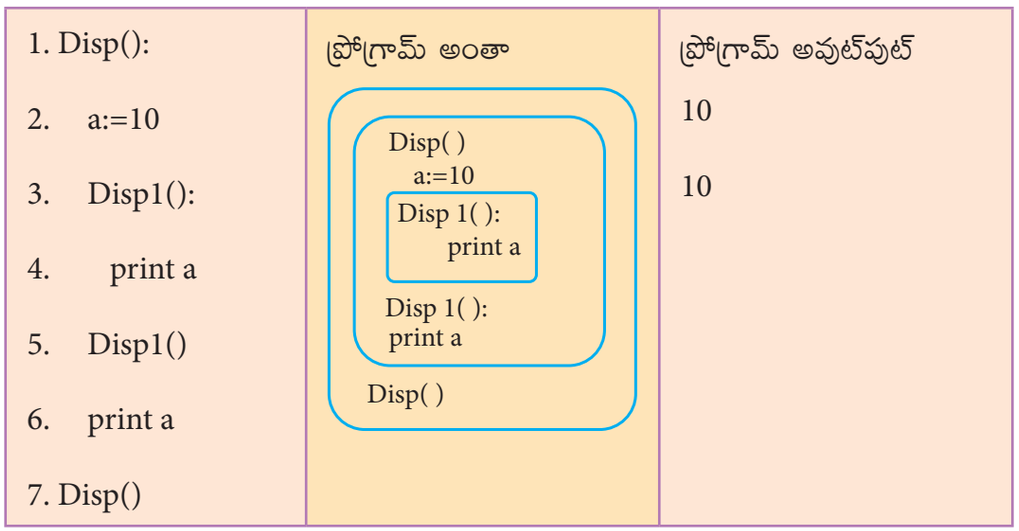
పై కోడ్‌ను అమలుచేసినపుడు ఫంక్షన్ లోపల నిర్వచింపబడిన, వేరియబుల్ ***a*** ఫంక్షన్ Disp() ను పిలిచినపుడు విలువ '7' ను ప్రదర్శించును. తరువాత అది '10' ను ప్రదర్శించును. ఎందుకంటే అనునవి ***a*** గ్లోబల్ పరిధిలో నిర్వచింపబడింది.

3.4.3 పరివేష్టిత పరిధి (Enclosed Scope)

అన్నీ ప్రోగ్రామింగ్ భాషలు ఫంక్షన్‌లను నెస్టడ్‌లో వుంచడానికి అనుమతిస్తాయి. ఫంక్షన్ (పద్ధతి)లో మరొక ఫంక్షన్‌ను నెస్టడ్ ఫంక్షన్ అంటారు. ఒక ఫంక్షన్ లోపల డిక్లర్ చేయబడిన వేరియబుల్, దానిలో మరొక ఫంక్షన్ డెఫనిషన్ కలిగివుండిన, లోపలి ఫంక్షన్, బాహ్య ఫంక్షన్ యొక్క వేరియబుల్‌ను కూడా యాక్సస్ చేస్తుంది. ఈ పరిధిని "పరివేష్టిత పరిధి" ***(enclosed scope)*** అంటారు.

ఒక ప్రోగ్రామ్‌లో కంపైలర్ లేదా ఇంటర్‌ప్రిటర్ వేరియబుల్ కోసం శోధిస్తే అది మొదట స్థానికను శోధించి తరువాత పరివేష్టిత పరిధిలను శోధిస్తుంది. క్రింది ఉదాహరణను పరిశీలిద్దాం.

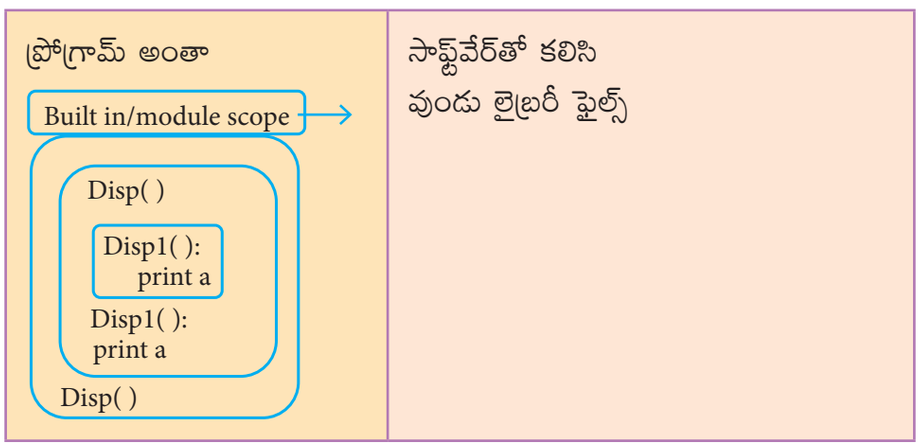
12th_CS_Telugu_Medium_Chapter_3.indd 24 27-03-2023 10:37:23



పై ఉదాహరణలో Disp1() ను Disp() లో నిర్వచింపబడినది. వేరియబుల్ 'a' Disp() అను ఫంక్షన్‌లో నిర్వచించబడి వుండినను Disp1() ఫంక్షన్ దానిని ఉపయోగించవచ్చును. ఎందుకనగా అది Disp() లో ఒక భాగం అగును.

3.4.4 అంతర్నిర్మిత పరిధి (Built-in Scope)

అంతర్నిర్మిత పరిధిలో మనం కంపైలర్ లేదా ఇంటర్‌ప్రెటర్‌ను ప్రారంభించినపుడు ప్రోగ్రామ్ పరిధిలోకి ముందే లోడ్ చేయబడిన అన్ని పేర్లను కలిగి ఉంటుంది. ప్రోగ్రామింగ్ భాష యొక్క మాడ్యూల్స్‌లో నిర్వచించబడిన ఏదైన వేరియబుల్ లేదా ఫంక్షన్, అంతర్నిర్మిత లేదా మాడ్యూల్ పరిధిని కలిగి ఉంటుంది. ప్రోగ్రామ్‌లు లైబ్రరీ ఫైల్ నుండి తీసుకోబడిన వెంటనే అవి లోడ్ అవుతాయి.



సాధారణంగా ఫంక్షన్‌ను లేదా మాడ్యూల్స్ సాఫ్ట్‌వేర్‌తో కలిసి Packages గా వచ్చును. కావున అవన్నియు అంతర్నిర్మిత పరిధి క్రింద వచ్చును.

3.5 మాడ్యుల్

మాడ్యుల్, ప్రోగ్రామ్‌లో ఒక భాగం. ఒకటి లేదా అంతకంటే ఎక్కువ స్వతంత్ర్యంగా అభివృద్ధి చేయబడిన మాడ్యులతో ప్రోగ్రామ్ కూడి వుంటుంది. ఒక మాడ్యుల్, ఒకదానితో ఒకటి దగ్గర సంబంధం వున్న ఒకటి లేదా

12th_CS_Telugu_Medium_Chapter_3.indd 25 27-03-2023 10:37:23

అనేక స్టేట్‌మెంట్‌ను కలిగివుంటుంది. మాడ్యుల్స్, అనునది వ్యక్తిగత స్థాయిలో సంపూర్ణంగా పనిచేస్తాయి, మరియు ఇతర మాడ్యులతో అనుసంధానం చేయబడి ఉండును ప్రోగ్రామని మరియు డిబగ్గింగ్ యొక్క పనిని సులభతరం చేయడానికి సాఫ్ట్‌వేర్ ప్రోగ్రామ్‌ను మాడ్యుల్స్‌గా విభజించవచ్చు. ప్రోగ్రామ్‌ను చిన్న ఫంక్షనల్ మాడ్యుల్స్‌గా విభజించవచ్చు. అవి అవుట్ పుట్ పొందడానికి కలిసి పనిచేస్తాయి. కంప్యూటర్ ప్రోగ్రామ్‌ను ప్రత్యేక ఉపప్రోగ్రామ్‌లుగా ఉపవిభజన ప్రక్రియలను మాడ్యులర్ ప్రోగ్రామింగ్ అంటారు. మాడ్యులర్ ప్రోగ్రామింగ్ ప్రోగ్రామర్ యొక్క ప్రోగ్రామ్ పనిని మరియు డిబగ్గ్ ముక్కలుగా విభజించుటను స్వతంత్ర్యంగా అనుమతిస్తుంది. మాడ్యుల్స్ యొక్క ఉదాహరణలు చేయు విధానాలు, subroutines మరియు ఫంక్షన్‌లు.

3.5.1 మాడ్యుల్స్ యొక్క లక్షణాలు :

ఈ క్రిందివి మాడ్యుల్స్ యొక్క నిర్దేశించబడిన లక్షణాలు:

1. సూచనలు, ప్రాససింగ్ లాజిక్ మరియు డేటాను మాడ్యుల్స్ కలిగివుంటాయి.
2. మాడ్యుళ్ళను విడిగా కంపైల్ చేసి లైబ్రరీలో నిల్వచేయవచ్చు.
3. మాడ్యుళ్ళను ప్రోగ్రామింగ్‌లో చేర్చవచ్చు.
4. పేరు మరియు కొన్ని పారామితులను ఉపయోగించి మాడ్యుల్ విభాగాలను అమలు చేయబడుతాయి.
5. మాడ్యుల్ విభాగాలను ఇతర మాడ్యుల్స్ ఉపయోగించవచ్చు.

3.5.2 మాడ్యుల్ ప్రోగ్రామింగ్‌ను ఉపయోగించడం వలన కలిగే ప్రయోజనాలు

- తక్కువ పంక్తులతో కోడ్‌ను రాయడం.
- పునర్వినియోగం కోసం ఒకే విధానాన్ని అభివృద్ధి చేయవచ్చు. కోడ్‌ను చాలాసార్లు టైప్ చేయవలసిన అవసరాన్ని తొలగిస్తుంది.
- ప్రోగ్రామ్‌లను మరింత సులభంగా రూపొందించవచ్చు. ఎందుకంటే చిన్న బృందంతో మొత్తం కోడ్‌లో చిన్న భాగంగా మాత్రమే వ్యవహరిస్తుంది.
- మాడ్యులర్ ప్రోగ్రామింగ్ ఒకే అప్లికేషన్‌ను చాలా మంది ప్రోగ్రామ్‌లకు అనుసందించడానికి అనుమతిస్తుంది.
- కోడ్ బహుళ ఫైల్‌లలో నిల్వచేయబడుతుంది.
- ఈ కోడ్ చిన్నది, సరళమైనది మరియు అర్థం చేసుకోదగినది.
- లోపాలను సులభంగా గుర్తించవచ్చు. ఎందుకంటే అవి subroutine లేదా ఫంక్షన్‌కు స్థానీకరించబడతాయి.
- ఒకే కోడ్‌ను చాలా అప్లికేషనలో ఉపయోగించవచ్చును.
- వేరియబుల్స్ యొక్క పరిధి సులభంగా నియంత్రించబడుతుంది.

3.5.3 యాక్సస్ కంట్రోల్ (Access Control)

యాక్సస్ కంట్రోల్ అనేది కంప్యూటర్ వాతావరణంలో ఎవరు లేదా ఏమి వనరులను చూడవచ్చు లేదా ఉపయోగించుటను నియంత్రించే భద్రతా సాంకేతికత. ఇది భద్రతలో దాని ప్రాథమిక భావన. Object యొక్క ప్రమాదాన్ని తగ్గిస్తుంది. మరోమాటలో చెప్పాలంటే యాక్సస్ కంట్రోల్ అనేది డేటా యాక్సస్ యొక్క ఎంపిక పరిమితిగా వుండును. ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామింగ్ languages ఇది యాక్సస్ మాడిఫైయర్లు ద్వారా అమలు చేయబడుతుంది. క్లాసికల్ ఆబ్జెక్ట్ ఓరియంటెడ్ languages C++ మరియు జావా వంటివి, public, private and protected keywords (కీలక పదాల) ద్వారా class సభ్యులను యాక్సస్ నియంత్రించబడును. ఒక class యొక్క ప్రవేటు సభ్యులకు class వెలుపల నుండి ప్రవేశం

12th_CS_Telugu_Medium_Chapter_3.indd 26 27-03-2023 10:37:23

నిరాకరించబడుతుంది. వాటిని class లో నుంచి మాత్రమే నిర్వహించవచ్చును.

Public సభ్యులు సాధారణంగా class లో ప్రకటించిన పద్ధతులు class వెలుపల నుండి అందుబాటులో వుంటారు. ఒకే class యొక్క object అమలు చేయడానికి public పద్ధతి అవసరం. ***private*** ఉదాహరణ వేరియబుల్స్ ***public*** పద్ధతుల అమరిక డేటా ఏన్క్యాప్సులేషన్ సూత్రాన్ని నిర్ధారిస్తుంది.

ఒక class యొక్క ప్రొటెక్టడ్ (Protected) సభ్యులు classes లోనే అందుబాటులో వుంటారు. దాని sub-classes కూడా అందుబాటులో వుంటారు. ఇతర ప్రక్రియను దీన్ని యాక్సస్ చేయడానికి అనుమతించవు. ఇది parent class యొక్క నిర్దిష్ట వనరులను child class వారసత్వంగా పొందడానికి అనుమతిస్తుంది.

పైథాన్‌కు ఏదైనా ఉదాహరణకు వేరియబుల్ లేదా పద్ధతికి యాక్సస్‌ను పరిమితం చేసే యంత్రంగం లేదు. (protected and private) యాక్సస్ స్పెసిపైయర్స్ ప్రవర్తనను అనుకరించడానికి పైథాన్ వేరియబుల్ లేదా పద్ధతి యొక్క పేరును సింగిల్ లేదా డబుల్ అండర్ స్కోర్‌తో ప్రిఫిక్స్ చేసే సమావేశాన్ని సూచిస్తుంది.

పైథాన్ classలోని సభ్యులందరు డిఫాల్ట్‌గా పబ్లిక్‌గా వుంటారు, అయితే C++ మరియు జావ (java) లో డిఫాల్ట్‌గా వారు ప్రైవేటుగా వుంటారు. పైథాన్‌లోని క్లాస్ ఎన్‌విరాన్‌మెంట్ లో (తరగతి వాతావరణం) వెలుపల నుండి ఏ సభ్యుడైన యాక్సస్ చేయవచ్చు. ఇది C++ మరియు (java)లలో సాధ్యం కాదు.

గుర్తించుకోవలసిన విషయాలు

- స్కోఫ్ ఒక ప్రోగ్రామ్ యొక్క ఒక భాగంలో వేరియబుల్స్ పారామితులు మరియు ఫంక్షన్‌లు యొక్క దృశ్యమానతను అదే ప్రోగ్రామ్ యొక్క మరొక భాగానికి సూచిస్తుంది.
- వేరియబుల్ పేరును ఒక Object లో బంధించే ప్రక్రియను మ్యాపింగ్ అంటారు.:= (colon and equal to sign) గుర్తు వేరియబుల్ మరియు Object మ్యాప్ చేయడానికి ప్రోగ్రామింగ్ భాషలలో ఉపయోగపడుతుంది.
- Namespaces లు Objectకు వేరియబుల్స్ పేర్లను మ్యాపింగ్ చేయడాన్ని కంటైనర్లు అగును.
- వేరియబుల్ పరిధి అది కనిపించే కోడ్ యొక్క భాగం.
- స్కోప్ రిజల్యూషన్‌లో (పరిధి స్పష్టత) పరిధులను శోధించాల్సిన క్రమాన్ని నిర్ణయించడానికి LEGB నియమం ఉపయోగపడుతుంది.
- స్థానిక పరిధి ప్రస్తుత ఫంక్షన్‌లో నిర్వచించిన వేరియబుల్స్‌ను సూచిస్తుంది.
- ప్రోగ్రామ్‌లోని అన్నీ ఫంక్షన్‌ల వెలుపల ప్రకటించబడిన వేరియబుల్‌ను “గ్లోబల్ వేరియబుల్” అంటారు.
- మరొక ఫంక్షన్‌లో వున్న ఫంక్షన్ (పద్ధతి)ను నెస్టెడ్ ఫంక్షన్ అంటారు.

12th_CS_Telugu_Medium_Chapter_3.indd 27 27-03-2023 10:37:23

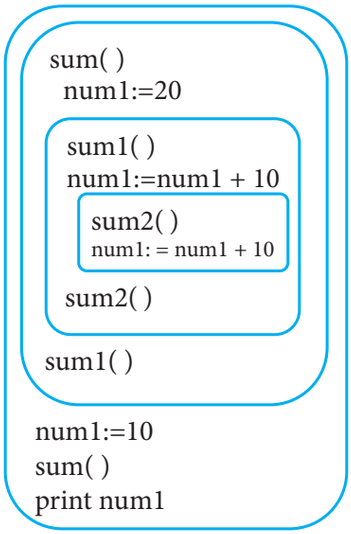
గుర్తించుకోవలసిన విషయాలు

- ఒక ఫంక్షన్ లోపల, డిక్లర్ చేయబడిన వేరియబుల్ దానిలో మరొక ఫంక్షన్ నిర్వచనాన్ని కలిగి వుంటుంది. లోపలి ఫంక్షన్, వెలుపల ఫంక్షన్ వేరియబుల్‌ను కూడా యాక్సస్ చేస్తుంది. ఈ పరిధిని "పరివేష్టిత పరిధి" (enclosed scope) అంటారు.
- అంతర్నిర్మిత పరిధిలో మనం కంపైలర్ లేదా ఇంటర్‌ప్రింటింగ్‌ను ప్రారంభించినపుడు ప్రోగ్రామ్ పరిధిలో ముందే నడిపించే అన్ని పేర్లులుంటాయి.
- మాడ్యుల్ ప్రోగ్రామ్‌లో ఒక భాగం. ప్రోగ్రామ్‌ను ఒకటి లేదా అంతకంటే ఎక్కువ స్వతంత్ర్యంగా అభివృద్ధి చేయబడిన మాడ్యులతో కూడి వుంటాయి.
- కంప్యూటర్ ప్రోగ్రామ్‌లో ప్రత్యేక ఉపప్రోగ్రామ్‌లుగా ఉపవిభజన చేసి ప్రక్రియను మాడ్యులర్ ప్రోగ్రామింగ్ అంటారు.
- యాక్సస్ కంట్రోల్ అనేది ఒక భద్రత సాంకేతికత ఇది కంప్యూటింగ్ వాతావరణంలో ఎవరు లేదా ఏమి వనరులను చూడవచ్చు లేదా ఉపయోగించుటను నియంత్రిస్తుంది. ఇది భద్రతలో ఒక ప్రాథమిక భావన, ఇది object యొక్క ప్రమాదాన్ని తగ్గిస్తుంది.
- పబ్లిక్ సభ్యులు (సాధారణంగా class ప్రకటించిన పద్ధతులు) class వెలుపల నుండి అందుబాటులో వుంటాయి.
- ఒక class యొక్క Protected సభ్యులు class లో అందుబాటులో వుంటారు, మరియు దాని Sub class కూడా అందుబాటులో వుంటారు.
- ఒక class యొక్క Private సభ్యులకు class వెలుపల నుండి ప్రవేశం నిరాకరించబడుతుంది. వాటిని class లో నుండి మాత్రమే నిర్వహించవచ్చు.
- Protected మరియు Private యాక్సస్ స్పెసిఫైయర్‌ల ప్రవర్తనను అనుకరించడానికి పైథాన్ వేరియబుల్ లేదా పద్ధతి యొక్క పేరును Single లేదా Double అండర్‌స్కోర్‌తో ప్రిఫిక్స్ చేసే సమావేశాన్ని సూచిస్తుంది.
- C++ మరియు Java, public, private మరియు protective కీలక పదాల ద్వారా class సభ్యులను యాక్సస్ నియంత్రించును.
- పైథాన్ class లోని సభ్యులందరు డిఫాల్ట్‌గా public అయితే C++ మరియు java లు సభ్యులందరు డిఫాల్ట్‌గా, ప్రైవేటుగా వుండును.

12th_CS_Telugu_Medium_Chapter_3.indd 28 27-03-2023 10:37:24

ప్రయోగ అభ్యాసం

1. క్రింద ఇవ్వబడిన పటానికి తగిన స్యూడోకోడ్ (pseudo code) ను రాయుము.



IPVLM

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. క్రింది వాటిలో ఏది వేరియబుల్ యొక్క దృశ్యమానతను ఒక ప్రోగ్రామ్ యొక్క చిన్న భాగం అదే ప్రోగ్రామ్ యొక్క మరొక భాగంను సూచించును.

   ఎ) పరిధి బి) మెమరి సి) చిరునామా డి) యాక్సెసిబిలిటి

2. వేరియబుల్ పేరు ఒక (object) తో బంధించే విధానాన్ని ఏమంటారు?

   ఎ) పరిధి బి) మ్యాపింగ్ సి) late బైడింగ్ డి) early బైడింగ్

3. క్రింది వాటిలో వేరియబుల్ మరియు object ను ఏది మ్యాప్ చేయడానికి ప్రోగ్రామింగ్ Language ఉపయోగించబడుతుంది.

   ఎ) :: బి) := సి) = డి) ==

12th_CS_Telugu_Medium_Chapter_3.indd 29 27-03-2023 10:37:24

4. వేరియబుల్స్ పేర్లను objects లో మ్యాపింగ్ చేసే కంటైనర్లను ఏమంటారు?

ఎ) పరిధి బి) మ్యాపింగ్ సి) బైడింగ్ డి) Namespaces

5. ప్రస్తుత ఫంక్షన్‌లో నిర్వచించిన వేరియబుల్స్‌ను ఏ పరిధి సూచిస్తుంది?

ఎ) స్థానిక పరిధి బి) గ్లోబల్ పరిధి సి) మాడ్యుల్ పరిధి డి) ఫంక్షన్ పరిధి

6. కంప్యూటర్ ప్రోగ్రామ్‌ను ప్రత్యేక ఉపప్రోగ్రామ్‌లుగా ఉప విభజన చేసే ప్రక్రియను ఏమంటారు?

ఎ) Procedural Programming బి) Modular programming

సి) Event Driven Programming డి) Object oriented Programming

7. క్రింది భద్రతా సాంకేతికత ఏది కంప్యూటింగ్ వాతావరణంలోని వనరులను ఉపయోగించుటను నియంత్రించునది?

ఎ) Password బి) Authentication సి) Access control డి) Certification

8. class ని ఈ క్రింది సభ్యులలో ఎవరు class లో మాత్రమే నిర్వహించగలరు?

ఎ) Public సభ్యులు బి) Protected సభ్యులు సి) Secured సభ్యులు డి) Private సభ్యులు

9. class వెలుపల నుండి ఏ సభ్యులు అందుబాటులో వుంటారు?

ఎ) Public సభ్యులు బి) Protected సభ్యులు సి) Secured సభ్యులు డి) Private సభ్యులు

10. class లో నుండి అందుబాటులో వున్న మరియు దాని sub-classes కూడా అందుబాటులో వున్న సభ్యులను ఏమంటారు?

ఎ) Public సభ్యులు బి) Protected సభ్యులు సి) Secured సభ్యులు డి) Private సభ్యులు

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. పరిధి అనగానేమి?

2. వేరియబుల్ కోసం పరిధిని ఎందుకు ఉపయోగించాలి? కారణమును తెల్పండి.

3. మ్యాపింగ్ అనగానేమి?

4. Namespacesపై మీ ఉద్దేశ్యాన్ని తెల్పండి?

5. పైథాన్, ప్రైవేట్ మరియు protected యాక్సస్ స్పెసిపైయరీలను ఎలా సూచిస్తుంది.

12th_CS_Telugu_Medium_Chapter_3.indd 30 27-03-2023 10:37:24

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. స్థానిక పరిధిని నిర్వచించి, ఉదాహరణను తెల్పండి?
2. గ్లోబల్ పరిధిని నిర్వచించి, ఉదాహరణను తెల్పండి?
3. పరివేష్టిత పరిధిని నిర్వచించి, ఉదాహరణను తెల్పండి?
4. యాక్సస్ కంట్రోల్ ఎందుకు అవసరం?
5. క్రింది స్యుడో కోడ్‌లోని వేరియబుల్స్ యొక్క పరిధిని గుర్తించండి? మరియు దాని అవుట్‌పుట్‌ను రాయండి?

```
output
color:= Red
mycolor():
  b:=Blue
  myfavcolor():
      g:=Green
      printcolor, b, g
  myfavcolor()
  printcolor, b
mycolor()
print color
```

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. వేరియబుల్ పరిధి రకాలు లేదా LEGB నియమము ఉదాహరణతో వివరించండి?
2. మాడ్యుల్స్ యొక్క ఏదైనా ఐదు లక్షణాలను రాయండి?
3. మాడ్యులర్ ప్రోగ్రామింగ్‌ని ఉపయోగించడంలో ఏదైనా ఐదు ప్రయోజనాలను రాయండి?

REFERENCES

1. *Data Structures and Algorithms in Python By Michael T.Goodrich, RobertoTamassia and Michael H. Goldwasser.*
2. *Data Structure and Algorithmic Thinking in Python By Narasimha Karumanchi*
3. *https://www.python.org*

12th_CS_Telugu_Medium_Chapter_3.indd 31 27-03-2023 10:37:24

యూనిట్ I

అధ్యాయం 4

అల్గోరిథమిక్ యుక్తులు

IGYL4

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశం పూర్తి అయిన తరువాత విద్యార్థులు తెలుసుకొనునవి.

- అల్గోరిథమ్‌ల యొక్క ప్రాథమిక అంశాలను మరియు సాంకేతిక దృక్పథాన్ని తెలుసుకుంటారు.
- అల్గోరిథమ్ యొక్క సమర్థతలను, కాలం మరియు స్థలం యొక్క సంక్లిష్టతను గురించి అవగాహన చేసుకుంటారు.
- శోధించడం మరియు క్రమబద్ధీకరించడం కోసం ఆల్గోరిథమ్‌లను అభివృద్ధి చేయడం మరియు విశ్లేషించడం.
- అల్గోరిథమ్ విధానం ద్వారా డైనమిక్ ప్రోగ్రామింగ్ గురించి తెలుసుకోవడం.

4.1 అల్గోరిథమిక్ యుక్తులు (స్ట్రాటజీల) - పరిచయం

అల్గోరిథమ్ అనేది ఒక నిర్ధిష్ట పనిని నేరవేర్చడానికి పరిమితమైన సూచనల సమితి. ఇచ్చిన సమస్యను పరిష్కరించడానికి ఇది దశలవారి విధానం. ప్రోగ్రామింగ్ లాంగ్వేజ్ దానికి తగినట్టుగా అల్గోరిథమ్‌ను అమలుచేయవచ్చు.

అల్గోరిథమ్‌లో ఇన్‌పుట్, అవుట్‌పుట్ కలిగి వుంటాయి. మరియు సరైనది, ప్రభావం వంటి లక్షణాల ఖచ్చితత్వాన్ని సంతృప్తిపరచును. Data Structures ద్వారా డేటా నిర్వహించబడుతుంది. మరియు సమర్థవంతంగా మార్చబడుతుంది. అటువంటి డేటా Structures నుండి డేటాను నిల్వచేయడానికి, మార్చటానికి మరియు తిరిగి పొందడానికి అల్గోరిథులు అభివృద్ధి చెందుతాయి. డేటా నిర్మాణాలకు ఉదాహరణ arrays (అమరిక) structures (నిర్మాణాలు) list, tuples మరియు డిక్షనరి.

Search	వరుస మరియు బైనరీ శోధనను ఉపయోగించి డేటా నిర్మాణంలో ఒక అంశాన్ని శోధించడానికి ఉపయోగపడును.
Sort	బబుల్ సార్ట్, ఇన్‌సెర్షన్ సార్ట్, సెలెక్షన్ సార్ట్ మొదలగు వాటిని ఉపయోగించి నిర్ధిష్ట క్రమంలో వస్తువులను క్రమబద్ధీకరించడానికి ఉపయోగపడుతుంది.
Insert	డేటా నిర్మాణంలో ఒక వస్తువును జోడించడానికి ఇన్‌సెర్ట్ ఉపయోగపడుతుంది.
Update	డేటా నార్మాణంలో ఇప్పటికే వున్న అంశాన్ని నవీనీకరించడానికి అప్‌డేట్ ఉపయోగపడుతుంది.
Delete	డేటా నిర్మాణంలో ఇప్పటికే వున్న అంశాన్ని తొలగించడానికి Delete ఉపయోగపడుతుంది.

అల్గోరిథమ్ నిర్వహించే మార్గాన్ని అల్గోరిథమ్ స్ట్రాటజీ అంటారు. ఉదాహరణకు ఇచ్చిన విలువ (**n**) కు క్రమగణితము (**n1**) లెక్కించడానికి, తరువాత పునరావృతం కోసం క్రమగణితము (**n1**) లెక్కించడానికి ఫంక్షన్‌ను నిర్వహించడం ద్వారా చేయవచ్చు. (iteration-1) తరువాత అవసరమైన పునరుక్తి సంఖ్య వచ్చేంతవరకు దీనిని పునరావృతంగా పిలుస్తారు.

12th_CS_Telugu_Medium_Chapter_4.indd 32 27-03-2023 10:38:00

మీకు తెలుసా?

అల్గోరిథమ్ అనే పదం గణితంపై పాఠ్యపుస్తకం రాసిన పర్షియన్ రచయిత అబూ జాఫర్ మొహమ్మద్ ఇబ్న్‌మూసా అల్ ఖోవారిజ్మి (c. 825 AD (CE)) పేరు నుండి వచ్చినది. అల్గోరిథమ్ అనే పదం సమస్యను పరిష్కరించడానికి ఒక పద్ధతిని సూచించడానికి వచ్చింది.

4.1.1 అల్గోరిథమ్ యొక్క లక్షణాలు

అల్గోరిథమ్ క్రింది లక్షణాలను కలిగి వుండవలెను.

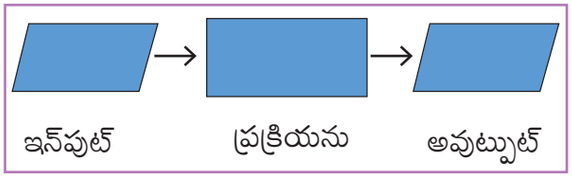
Input	సున్న లేక అంతకంటే ఎక్కువ పరిమాణాలను సరఫరా చేయవలెను.
Output	కనీసం ఒక పరిమాణంనైనా ఉత్పత్తి చేయును.
Finiteness	పరిమిత దశల సంఖ్యలు అల్గోరిథమ్‌లను ముగించాలి.
Definiteness	అన్ని కార్యకలాపాలను బాగుగా నిర్వహించాలి. ఉదాహరణకు సున్నతో భాగించుట లేదా ఋణ సంఖ్యకు వర్గమూలంను కనుగొనుట వంటివి కార్యకలాపాలు ఆమోదించుటకు వీలుకాదు.
Effectiveness	ప్రతి సూచన సమర్థవంతంగా నిర్వహించబడాలి.
Correctness	అల్గోరిథమ్‌లు లోపం లేకుండా వుండాలి.
Simplicity	అమలు చేయడం సులభం.
Unambiguous	అల్గోరిథమ్ స్పష్టంగా మరియు నిస్సందేహంగా వుండాలి. దాని ప్రతి దశ మరియు వాటి ఇన్ పుట్‌లు / అవుట్ పుట్‌లు స్పష్టంగా ఉండాలి. మరియు ఒకే ఒక అర్థానికి దారి తీయాలి.
Feasibility	అందుబాటులో వున్న వనరులతో సాధ్యమయ్యేలా వుండాలి.
Portable	ఒక అల్గోరిథమ్ అనేవి సాధారణమయినదిగానూ, ఏదైనా ప్రోగ్రామింగ్ భాష నుండి స్వతంత్ర్యంగానూ లేదా అన్ని రకాల ఇన్‌పుట్‌లను నిర్వహించగల ఆపరేటింగ్ సిస్టమ్‌గానూ వుండవలెను.
Independent	ఒక అల్గోరిథమ్ అనునది దశల వారి దిశలను కలిగి వుండాలి, ఇది ఏదైనా ప్రోగ్రామింగ్ కోడ్ నుండి స్వతంత్ర్యంగా వుండాలి.

4.1.2 అల్గోరిథమ్‌ను వ్రాయుట (Writing an Algorithm)

అల్గోరిథమ్‌లు సాధారణమైనవి మరియు కంప్యూటర్‌కు మాత్రమే పరిమితమైనవి కావు. ఇది వివిధ నిత్యకాలక్రమాలలో ఉపయోగించవచ్చు. తెలిసి, తెలియకుండా మన దైనందిన జీవితంలో పాఠశాల బ్యాగ్‌లో పుస్తకాలు ప్యాక్ చేయడం, స్థలాన్ని శోధించడానికి చిన్న మార్గాన్ని కనుగొనడం రోజువారీ కార్యకలాపాలను షెడ్యుల్ చేయడం, పరీక్షకు సిద్ధం చేయడం వంటి అనేక విషయాలలో మనం అల్గోరిథమ్‌ను తయారుచేస్తాము. అన్ని ప్రోగ్రామింగ్ భాషలకు ప్రాథమిక కోడ్ నిర్మాణాన్ని పంచుకుంటాయని మనకు తెలుసు అనగా నిబంధనలు మరియు పునరావృతులు వంటి

12th_CS_Telugu_Medium_Chapter_4.indd 33 27-03-2023 10:38:00

కోడ్ నిర్మాణాలు అల్గోరిథమ్‌ను వ్రాయడంలో ఉపయోగించవచ్చు. ఒక సాధారణ అల్గోరిథమ్‌ను క్రింది పటం 4.1 లో చూపబడింది.



అల్గోరిథమ్

ఉదాహరణ

కాఫీ తయారీని ఉదాహరణగా పరిగణించండి. కాఫీ (Coffee) తయారు చేయడానికి మనకు క్రింది పదార్థాలు వుండాలి. నీరు, పాలు, కాఫీ పౌడర్ మరియు చక్కెర. ఈ పదార్థాలు అన్ని అల్గోరిథమ్ యొక్క ఇన్‌పుట్‌లు కాఫీ తయారు చేయడం ప్రక్రియ అంటారు. కాఫీ ప్రక్రియ యొక్క అవుట్‌పుట్.

కాఫీ తయారు చేయుటకు గల విధానం క్రింది విధంగా

1. ఒక పాత్రలో కాఫీ పొడిని తీసుకొనుము
2. నీటిని మరిగించి మరియు దానిని కాఫీ పొడరు గల పాత్రలో పోయాలి.
3. దానిని వడకట్టాలి.
4. పాలను మరిగించాలి.
5. మరిగించిన పాలతో, వడబోసిన కాఫీని మరియు చక్కెరను కలపాలి.
6. కాఫీని కప్‌లో పోసుకోవాలి.

ఈ రకమైన పద్ధతి అల్గోరిథమ్‌ను ఉపయోగించడంలో ప్రాతినిధ్యం వహిస్తుంది. ఈ విధంగా అల్గోరిథమ్‌ను ఒక పనిని పూర్తి చేయడానికి అవసరమైన దశలవారి సూచనలను కలిగివుంటుంది. ప్రోగ్రామర్‌కు ప్రోగ్రామ్‌ను అభివృద్ధి చేయడానికి సహాయం చేయబడుతుంది.

సమస్య: ఇచ్చిన సంఖ్య యొక్క వర్గాన్ని కనుగొని ఫలితాన్ని ప్రదర్శించడానికి ఒక అల్గోరిథమ్‌ను రూపకల్పన చేయండి. అల్గోరిథమ్‌ను ఇలా రాయవచ్చు.

దశ –1 : ప్రక్రియను ప్రారంభించండి.

దశ –2 : ఇన్‌పుట్ (x) ను పొందండి.

దశ –3 : ఇన్‌పుట్ విలువను గుణించడం ద్వారా వర్గాన్ని లెక్కించవచ్చు. అనగా ← x* x

దశ –4 : ఫలితం వర్గాన్ని ప్రదర్శిస్తుంది.

దశ –5 : అల్గోరిథమ్‌ను ఆపండి.

అల్గోరిథమ్ ఇచ్చిన సమస్యకు పరిష్కారం పొందడానికి రూపొందించవచ్చును. ఒక సమస్యను అనేక విధాలుగా పరిష్కరించవచ్చును. అనేక అల్గోరిథమ్‌లలో ఏదో ఒకటి మాత్రమే అమలు చేయబడును.

మీకు తెలుసా?

చెల్లుబాటు అయ్యే ఇన్‌పుట్ కోసం ఊహించిన అవుట్‌పుట్‌ని ఇచ్చే అల్గోరిథంను అల్గోరిథమిక్ సొల్యూషన్ అంటారు.

అల్గోరిథం	ప్రోగ్రామ్
• అల్గోరిథం ఇచ్చిన సమస్యను తార్కికంగా పరిష్కరించడానికి సహాయపడుతుంది మరియు ఇది ప్రోగ్రామ్‌తో విభేదించవచ్చును.	• ప్రోగ్రామ్ అనేది ప్రోగ్రామింగ్ భాషలో అల్గోరిథం యొక్క వ్యక్తీకరణ.
• అల్గోరిథంను వాటి అమలు పద్ధతులు డిజైన్ పద్ధతులు మొదలైన వాటి ఆధారంగా వర్గీకరించవచ్చు.	• నిర్మాణాత్మక లేదా ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామింగ్ విధానం ద్వారా అల్గోరిథం అమలు చేయవచ్చు.

12th_CS_Telugu_Medium_Chapter_4.indd 34 27-03-2023 10:38:00

• అల్గోరిథంను రాయడానికి నిర్దిష్ట నియమాలు లేవు కాని కొన్ని మార్గదర్శకాలను పాటించాలి.	• నిర్దిష్ట వాక్యనిర్మాణంతో ఎంచుకున్న భాష కోసం ప్రోగ్రామ్ రాయాలి.
• అల్గోరిథం ఏ భాషలోనైన చేయగల ఒక సూడో కోడ్‌ను పోలి ఉంటుంది.	• ప్రోగ్రామింగ్ భాషకు ప్రోగ్రామ్ మరింత నిర్దిష్టంగా ఉంటుంది.

టేబుల్ 4.1 అల్గోరిథం *Vs* ప్రోగ్రామ్

4.1.3. అల్గోరిథం యొక్క విశ్లేషణ (Analysis of Algorithm):

కంప్యూటర్ వనరులు పరిమితం. అల్గోరిథం యొక్కసామర్థ్యం, సమయం మరియు స్థల సంక్లిష్టతను ఉపయోగించడం ద్వారా నిర్వచించబడుతుంది. ఒక అల్గోరిథం యొక్క విశ్లేషణ సాధారణంగా వివిధ కార్యకలాపాలు నడుపుతున్న మరియు అమలు చేసే సమయంతో వ్యవహరిస్తుంది. ఒక ప్రక్రియను నడుపు సమయం అనునది ప్రక్రియకు ఎన్ని ప్రోగ్రామ్ సూచనలను అమలు చేయబడింది అని లెక్కించబడుతుంది.

అల్గోరిథంలో విశ్లేషణ మరియు పనితీరు మూల్యాంకనాన్ని రెండు వేర్వేరు దశలుగా విభజించవచ్చు.

1. **ముందు అంచనా (A Priori estimates)**: ఇది అల్గోరిథం యొక్క సైద్ధాంతిక పనితీరు విశ్లేషణ. అల్గోరిథం యొక్క సామర్థ్యాన్ని భాహ్యకారకాలను ఊహించడం ద్వారా కొలుస్తారు.
2. **తరువాత పరిక్ష (A Posteriori testing)**: దీనిని పనితీరు కొలమాణం అని అంటారు. ఈ విశ్లేషణలో, నడుపుతున్న సమయం మరియు అల్గోరిథం అమలుకు అవసరమైన సంఖ్యలను సేకరించబడుతాయి.

మీకు తెలుసా?

విభిన్న (input) పరిమాణాల కోసం అల్గోరిథం యొక్క సమయం మరియు స్థల సంక్లిష్టతలను అంచనా వేయడం అల్గోరిథం విశ్లేషణ అంటారు.

4.2 అల్గోరిథం యొక్క సంక్లిష్టత

A అనునది ఒక అల్గోరిథం అని అనుకుందాం మరియు n అనేది ఇన్పుట్ డేటా యొక్క పరిమాణం, అల్గోరిథం A ఉపయోగించే సమయం మరియు స్థలం రెండు ప్రధాన కారకాలు, ఇవి A యొక్క సామర్థ్యాన్ని నిర్ణయిస్తాయి.

సమయ కారకం – సార్టింగ్ అల్గోరిథంలో పోలికలు వంటి కీలక కార్యకలాపాల సంఖ్యను లెక్కించడం ద్వారా సమయంను కొలుస్తారు.

స్థల కారకం – Space factorను అల్గోరిథంకు అవసరమైన గరిష్ఠ మెమరీ Space ద్వారా కొలుస్తారు.

అల్గోరిథం యొక్క సంక్లిష్టత f (n) అది తీసుకునే ఇన్‌పుట్ డేటా మొత్తం మరియు మెమరీని అమలుచేయడానికి ఎంత సమయం పడుతుంది అనే దానిపై ఆధారపడి వుంటుంది.

4.2.1 సమయ సంక్లిష్టత (Time Complexity):

అల్గోరిథమ్ ప్రక్రియను పూర్తిచేయడానికి అల్గోరిథమ్‌ను తీసుకున్న దశల సంఖ్య ద్వారా సమయ సంక్లిష్టత ఇవ్వబడుతుంది.

4.2.2. Space సంక్లిష్టత (Space Complexity)

Space సంక్లిష్టత అనునది అల్గోరిథమ్ పూర్తికి అవసరమైన మెమరీ మొత్తం అగును. ఒక అల్గోరిథమ్‌కు అవసరమైన Space ని క్రింది రెండు భాగాల మొత్తానికి సమానంగా వుండును.

12th_CS_Telugu_Medium_Chapter_4.indd 35 27-03-2023 10:38:00

స్థిరభాగం: ఒక అల్గోరిథమ్ కోసం నిర్దిష్ట డేటా మరియు వేరియబుల్స్ నిల్వచేయడానికి అవసరమైన మొత్తం స్థలంను స్థిర భాగంగా నిర్వచించబడుతుంది. ఉదాహరణకు సాధారణ వేరియబుల్స్ మరియు స్థిరాకాంలను అల్గోరిథమ్‌లో ఉపయోగించబడును.

వేరియబుల్ భాగం : వేరియబుల్స్‌కు అవసరమైన మొత్తం స్థలాన్ని వేరియబుల్ భాగంగా నిర్వహింపబడినది. దీని పరిమాణం అనునది సమస్య మరియు పునరావృతంపై ఆధారపడి వుంటుంది. ఉదాహరణకు పునరావృతాన్ని ఉపయోగించి ఇచ్చిన విలువ 'n'కు క్రమగుణిజము(factorial)ను కనుగొనవచ్చును.

4.3 అల్గోరిథమ్ యొక్క సమర్థత

కంప్యూటర్ వనరులు పరిమితం కావున వాటిని సమర్థవంతంగా ఉపయోగించవలెను. అల్గోరిథమ్ యొక్క సామర్థ్యాన్ని అల్గోరిథమ్ ఉపయోగించి గణన (computational) వనరుల సంఖ్యగా నిర్వచించవచ్చు. అల్గోరిథమ్ యొక్క విశ్లేషణ, అది ఉపయోగించు వనరులను బట్టి నిర్ణయించబడును. అల్గోరిథమ్ యొక్క సామర్థ్యాన్ని వివిధ వనరుల వినియోగం ఆధారంగా కొలువవచ్చును.

అల్గోరిథమ్ యొక్క గరిష్ఠ సామర్థ్యం కోసం మనం వనరుల వినియోగాన్ని తగ్గించాలి. సమయం మరియు స్థల సంక్లిష్టత వంటి ముఖ్యమైన వనరులను నేరుగా పోల్చలేము, కావున అల్గోరిథమ్ సమర్థత కొరకు సమయం మరియు స్థల సంక్లిష్టతలను పరిగణించవచ్చు.

4.3.1 అల్గోరిథమ్ సామర్థ్యాన్ని నిర్ణయించే పద్ధతి (Method for Determining Efficiencey)

ఒక అల్గోరిథమ్ యొక్క సామర్థ్యం, సమయం మరియు మెమరీ స్థలాన్ని ఎంత సమర్థవంతంగా ఉపయోగిస్తుంది దానిపై ఆధారపడి వుంటుంది.

అల్గోరిథమ్ యొక్క సమయ సామర్థ్యం వేరువేరు కారకాలచే కొలవబడును. ఉదాహరణకు నిర్వచించబడిన అల్గోరిథమ్ కోసం ఒక ప్రోగ్రామ్‌ను రాయుము. ఏదైనా ప్రోగ్రామింగ్ భాషను ఉపయోగించి అమలుచేయుము మరియు దానిని నడుపుటకు తీసుకొను మొత్తం సమాయాన్ని లెక్కింపుము. ఈ సందర్భంలో మీరు కొలిచే అమలు సమయం క్రింది అనేక కారకాలపై ఆధారపడి వుంటుంది.

- యంత్రం యొక్క వేగం
- కంపైలర్ మరియు ఇతర సిస్టమ్ సాఫ్ట్‌వేర్ సాధనాలు.
- ఆపరేటింగ్ సిస్టమ్
- ఉపయోగించిన ప్రోగ్రామింగ్ భాష
- అవసరమైన డేటా ఘనపరిమాణం

అయితే ఒక అల్గోరిథమ్ ఇచ్చిన సమస్యను ఎంత సమర్థవంతంగా పరిష్కరిస్తుందో తెలుసుకొనడానికి అల్గోరిథమ్ యొక్క స్వభావం అమలు అయ్యే సమయాన్ని ఎలా ప్రభావితం అవుతుందో తెలుసుకొనడం అవసరం. అందువల్ల అండర్‌లైయింగ్ అల్గోరిథమ్ యొక్క స్వభావం పరంగా ప్రోగ్రామ్ యొక్క సామర్థ్యాన్ని నిర్ణయించే ప్రాథమిక విధులను అభివృద్ధి చేయవలెను.

మీకు తెలుసా?

"అల్గోరిథమ్ రూపకల్పన యొక్క మార్గాన్ని అల్గోరిథమిక్ స్ట్రాటజీ అంటారు".

4.3.2 స్థలం – సమయ మార్పిడి (Space-Time tradeoff)

స్థలం – సమయ మార్పిడి లేదా సమయం మెమరీ మార్పిడి అనునది తక్కువ సమయంలో ఎక్కువ నిల్వ స్థలాన్ని ఉపయోగించి అల్గోరిథమ్‌ను పరిష్కరించే మార్గం.

12th_CS_Telugu_Medium_Chapter_4.indd 36 27-03-2023 10:38:00

ఇచ్చిన ప్రోగ్రామింగ్ సమస్యను పరిష్కరించడానికి అనేక విభిన్న అల్గోరిథమ్‌లను ఉపయోగించవచ్చు. ఈ అల్గోరిథమ్‌లో కొన్ని చాలా సందర్భాలలో సమయం సమర్థవంతంగా ఉండవచ్చు మరియు కొన్ని చాలా సందర్భాలలో స్థలం సమర్థవంతంగా ఉండవచ్చు.

సమయం – స్థలం మార్పిడి అనేది మీరు నెమ్మదిగా ప్రోగ్రామ్ అమలు ఖర్చుతో మెమరీ వాడకాన్ని తగ్గించగల పరిస్థితిని సూచిస్తుంది లేదా పెరిగిన మెమరీ వినియోగం ఖర్చుతో నడుస్తున్న సమయాన్ని తగ్గించవచ్చు.

మీకు తెలుసా?

ఇచ్చిన సమస్యను పరిష్కరించడానికి ఉత్తమ అల్గోరిథమ్, మెమరీలో తక్కువ స్థలం అవసరం మరియు అవుట్‌పుట్‌ను రూపొందించడానికి దాని సూచనలను అమలు చేయడానికి తక్కువ సమయం పడుతుంది.

4.3.3 అసింప్టిక్ నోటేషన్స్ (Asymptotic Notations)

(Asymptotic Notations) సంకేతాలు సమయం మరియు స్థల సంక్లిష్టత గురించి అర్థవంతమైన ప్రకటనలను ఉపయోగించే Languagesఅల్గోరిథమ్‌లయొక్కసమయసంక్లిష్టతను సూచించడానికి క్రింది మూడు (asymptotic) సంకేతాలు ఎక్కువగా ఉపయోగపడుతాయి.

(i) Big O

అల్గోరిథమ్ యొక్క చెత్త పరిస్థితిని వివరించడానికి (Big O) తరచుగా ఉపయోగించబడుతుంది.

(ii) Big Ω (బిగ్ ఒమేగా)

Big Ω, Big O యొక్క తలక్రిందులు అగును. (asymptotic function) యొక్క ఎగువ స్థాయి (చెత్త – పరిస్థితి) ని వివరించడానికి Big O ఉపయోగించబడితే (Big Omega) దిగవస్థాయి (ఉత్తమ–పరిస్థితి)ని వివరించడానికి ఉపయోగిస్తారు.

(iii) Big Θ

అల్గోరిథమ్ దిగువస్థాయి = ఎగువస్థాయితో సంక్లిష్టతను కలిగి వున్నపుడు, ఒక అల్గోరిథమ్ సంక్లిష్టత (O (n log n)) మరియు (Ω (n log n)) కలిగి వుందని చెప్పవచ్చు. వాస్తవానికి ఇది (Θ (n log n)) సంక్లిష్టతను కలిగి వుండును. అంటే ఆ అల్గోరిథమ్ యొక్క నడుస్తున్న సమయం ఎల్లపుడు ఉత్తమ పరిస్థితి చెత్త పరిస్థితులలో (n log n) లో వుండును.

4.3.4. ఉత్తమ, చెత్త, మరియు సరాసరి పరిస్థితుల సమర్థత

లిస్ట్‌ను అమరికలో (array) నిల్వచేయబడిన (n) సంఖ్యల ఊహించుకుందాం. ఈ లిస్ట్‌లోని ఒక నిర్దిష్ట మూలకాన్ని మనం శోధించాలనుకుంటే లిస్ట్‌లోని ప్రతిమూలకంతో కీ (key) మూలకాన్ని వరుసగా పోల్చడం ద్వారా(n) మూలకాల మధ్య లిస్ట్‌లోని ముఖ్యమూలకాన్ని శోధించేదే అల్గోరిథమ్.

లిస్ట్‌లోని మొదటి మూలకం శోధించవలసిన ముఖ్య మూలకంతో సరిపోల్చితే అది ఉత్తమ పరిస్థితి అగును. ఆ సందర్భంలో సామర్థ్యం O(1) గా వ్యక్తీకరించబడుతుది. ఎందుకంటే ఒక పోలిక మాత్రమే సరిపోతుంది.

అదే విధంగా ఈ సందర్భంలో చెత్త పరిస్థితి పూర్తి లిస్ట్‌ను శోధించి, మూలకం లిస్ట్ చివరలో మాత్రమే కనుగొనబడితే లేదా లిస్ట్‌లో కనుగొనలేకపోతే ఆ సందర్భంలో ఒక అల్గోరిథమ్ యొక్క సామర్థ్యం O(n) గా వ్యక్తీకరించబడుతుంది. ఎందుకంటే శోధనను పూర్తి చేయడానికి (n) పోలికలు అవసరం.

క్రింద ఇచ్చిన విధంగా సరాసరి పోలికల సంఖ్యను కనుగొనడం ద్వారా అల్గోరిథమ్ యొక్క

12th_CS_Telugu_Medium_Chapter_4.indd 37 27-03-2023 10:38:00

సరాసరి పరిస్థితి సామర్థ్యాన్ని పొందవచ్చు.

కనిష్ఠ పోలికల సంఖ్య = 1

గరిష్ఠ పోలికల సంఖ్య = n

మూలకం కనబడకపోతే గరిష్ఠ పోలికల సంఖ్య = n

కావున సరాసరి పోలికల సంఖ్య $(n + 1)/2$

అందువలన సరాసరి పరిస్థితి సామర్థ్యాన్ని O (n) గా వ్యక్తీకరించబడును.

4.4 అల్గోరిథంను శోధించుటకు గల సాంకేతికాలు

4.4.1 సరళ శోధన (Linear Search)

సరళ శోధన అనేది లిస్ట్‌లో ఒక నిర్దిష్ట విలువను, వరుసక్రమపద్ధతిలో కనుగొనుట అగును. కావున దీనిని వరుసక్రమ శోధన అని కూడా పిలువవచ్చును. ఈ పద్ధతి కావలసిన మూలకం కనుగొనబడే వరకు లేదా జాబితా అయిపోయే వరకు ప్రతిమూలకంతో శోధన మూలకాన్ని తనిఖీ చేస్తుంది.

ఈ శోధన అల్గోరిథంలో, జాబితాను ఆదేశించాల్సిన అవసరం లేదు.

సూడో కోడ్ (Pseudo code):

1. లూప్ కోసం అమరికను దాటండి.
2. ప్రతి పునరావృతంలో, లక్ష్యశోధన కీ విలువను జాబితా యొక్క ప్రస్తుత విలువతో పోల్చండి.
   - విలువలు సరిపోలితే, ప్రస్తుత సూచిక మరియు అమరిక యొక్క విలువను ప్రదర్శించును.
   - విలువలు సరిపోలకపోతే, తదుపరి అమరిక మూలకానికి వెళ్ళును.
3. సరిపోలిక కనబడకపోతే, శోధన మూలకం కనబడలేదు అని ప్రదర్శించును.

క్రింద ఇవ్వబడిన అమరికలో సంఖ్య '25'ను శోధించడానికి, ఇచ్చిన అమరికలోని మొదటి మూలకం నుండి ప్రారంభమై వరుస క్రమంలో సరళ శోధన దశల వారీగా వెళుతుంది. శోధన మూలకం కనుగొనబడితే సూచిక తిరిగి ఇవ్వబడును. లేకపోతే అమరిక యొక్క చివరి సూచిక వరకు శోధన కొనసాగుతుంది. ఈ ఉదాహరణలో సంఖ్య 25 సూచిక సంఖ్య 3 వద్ద కనుగొనబడింది.

సూచిక	0	1	2	3	4
విలువలు	10	12	20	25	30

ఉదాహరణ 1 :

Input: values[] = {5, 34, 65, 12, 77, 35}
target = 77
Output: 4

ఉదాహరణ 2 :

Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}
target = 200
Output: -1 (not found)

12th_CS_Telugu_Medium_Chapter_4.indd 38 27-03-2023 10:38:00

4.4.2. బైనరీ శోధన (Binary Search)

బైనరీ శోధనను సగం విరామ శోధన అల్గోరిథం అని కూడా పిలుస్తారు. ఇది క్రమబద్ధీకరించబడిన అమరికలోని శోధన మూలకం యొక్క స్థానాన్ని కనుగొంటుంది. బైనరీ శోధన అల్గోరిథం డివైడ్-అండ్-కాంక్వెర్ శోధన అల్గోరిథం వలె పనిచేయును మరియు లాగరిథమిక్ సమయంలో అమలు చేస్తుంది.

బైనరీ శోధనకు సూడోకోడ్ :

1. మధ్య మూలకంతో ప్రారంభించుము :
   - శోధన మూలకం అమరిక యొక్క మధ్య మూలకం సమానంగా ఉంటే, అనగా మధ్య విలువ = అమరికలోని మూలకాల సంఖ్య /2 ఆపై మధ్య మూలకం యొక్క సూచికను తిరిగి ఇవ్వబడును.
   - కాకపోతే మధ్య మూలకాన్ని శోధన విలువతో పోల్చండి.
   - మధ్య సూచికలోని సంఖ్య కంటే శోధన మూలకం ఎక్కువగా ఉంటే మధ్య సూచిక యొక్క కుడివైపున ఉన్న అంశాలను ఎంచుకొని దశ-1 కి వెళ్ళుము.
   - మధ్య సూచికలోని సంఖ్య కంటే శోధన మూలకం తక్కువగా ఉంటే, మధ్య సూచిక యొక్క ఎడమ వైపున ఉన్న మూలకాలను ఎంచుకొని దశ - 1 తో ప్రారంభించుము.
2. సరిపోలిక కనుగొనబడినప్పుడు, మూలకం యొక్క సూచికతో సందేశాన్ని విజయంగా ప్రదర్శించును.
3. అన్ని పోలికలకు సరిపోలిక లేనప్పుడు '-1' అనే విలువను తిరిగి పంపును (లేక) నిరాకరించబడినది అనే సందేశాన్ని ప్రదర్శించును.

బైనరీ శోధన పని సూత్రాలు (Binary Search Working principles)

మొదట అమరికలో వుండు మూలకాల లిస్ట్‌ను బైనరీ శోధన కోసం క్రమబద్ధీకరించాలి. క్రింది ఉదాహరణ బైనరీ శోధన యొక్క దశల వారీ కార్యకలాపాన్ని వివరిస్తుంది. క్రింది మూలకాల అమరికను పరిగణించండి, అమరిక క్రమబద్ధీకరించబడినది కాబట్టి బైనరీ శోధన అల్గోరిథం చేయడానికి వీలు కల్పిస్తుంది. శోధన మూలకం 60 అని అనుకుందాం మరియు బైనరీ శోధనను ఉపయోగించి శోధన మూలకం 60 యొక్క స్థానం లేదా సూచికను శోధించాలి.

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

మొదట మనం అమరికలోని మధ్య మూలకం యొక్క సూచికను క్రింది సూత్రంను ఉపయోగించి కనుగొనవలెను.

$$mid = low + (high - low) / 2$$

ఒక్కడ, 0 + (9 - 0) / 2 = 4 (దశాంశ విలువను వదిలివేయుము) కావున, అమరిక యొక్క మధ్య విలువ '4' అగును.

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

ఇప్పుడు శోధన మూలకాన్ని మధ్య విలువ స్థానం 4 వద్ద నిల్వచేసిన విలువతో పోల్చండి. స్థానం లేదా సూచిక 4 వద్ద నిల్వచేసిన విలువ 50, ఇది శోధన మూలకంతో సరిపోలేదు. శోధన విలువ 60, 50 కంటే ఎక్కువగా ఉన్నందున

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

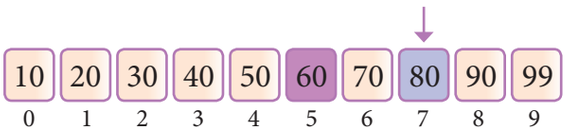
ఇప్పుడు తక్కువను (low) మధ్యకు (mid + 1) మార్చాము మరియు సూత్రాన్ని ఉపయోగించి మళ్ళీ కొత్త మధ్య విలువను కనుగొనవలెయును.

12th_CS_Telugu_Medium_Chapter_4.indd 39 27-03-2023 10:38:01

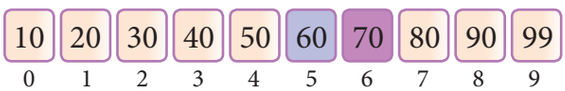
low = mid + 1

mid = low + (high - low) / 2

ఇప్పుడు మన కొత్త (mid) విలువ 7. 7వ స్థానంలో నిల్వ చేసిన విలువను లక్ష్య విలువ 60తో పోల్చుము.



స్థానం లేదా సూచిక 7 వద్ద నిల్వ చేసిన విలువ శోధన మూలకంతో సరిపోల్చడం లేదు. ఇది మనం వెతుకుతున్న దాని కంటే ఎక్కువ. కాబట్టి శోధన మూలకం తప్పనిసరిగా ప్రస్తుత మధ్య విలువ స్థానం నుండి దిగువ భాగంలో ఉండవలెను.

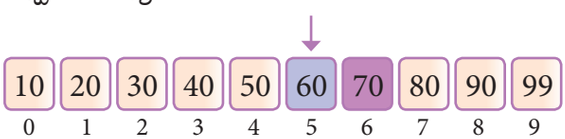


శోధన మూలకం ఇంకా కనుగొనబడలేదు. అందువల్ల మనం సూత్రాన్ని ఉపయోగించి మళ్ళీ mid ను లెక్కించుదాము.

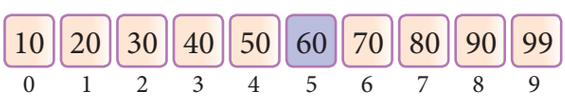
high = mid -1

mid = low + (high - low)/2

ఇప్పుడు మధ్య విలువ 5.



ఇప్పుడు మనం 5వ స్థానంలో నిల్వ చేసిన విలువను మన శోధన మూలకంతో పోల్చుము. రెండు సరిగ్గా పోలియుండుటను కనుగొనవచ్చును.



శోధన మూలకం 60 స్థానము లేదా సూచిక 5 వద్ద ఉన్నదని మనం నిర్ధారించగలము. ఉదాహరణకు, మనం శోధన మూలకాన్ని 95గా తీసుకుంటే, ఈ విలువకు బైనరీ శోధన అల్గోరిథం −1 లేక నిరాకరించబడినది అనే ఫలితాన్ని ఇస్తుంది.

4.5 సార్టింగ్ టెక్నిక్స్

4.5.1 బబుల్ సార్ట్ అల్గోరిథం (Bubble sort algorithm)

బబుల్ సార్ట్ అనేది ఒక సాధారణ సార్టింగ్ అల్గోరిథం. అమరికలో నిల్వచేయబడిన విలువల లిస్ట్ యొక్క ప్రారంభంతో అల్గోరిథం ప్రారంభమవుతుంది. ఇది ప్రతి జత ప్రక్కనే ఉన్న మూలకాలను పోల్చి, క్రమబద్ధీకరించని క్రమంలో ఉంటే వాటిని మార్పిడి చేస్తుంది. ఈ పోలిక మరియు మార్పిడులు అవసరం వరకు కొనసాగడానికి ఇది ఆమోదించబడుతుంది. ఇది అమరికలో నిల్వ చేయబడిన విలువల లిస్ట్‌ను క్రమబద్ధీకరించబడిందని సూచిస్తుంది. అల్గోరిథం ఒక పోలిక ఏ విధమైనది, లిస్ట్ యొక్క పై భాగములో గల చిన్న మూలకాలకు 'బబుల్' అనే పేరు పెట్టబడింది. అల్గోరిథం సరళమైనది అయినప్పటికీ, చొప్పించే క్రమబద్ధీకరణ insertion sort మరియు ఇతర సార్టింగ్ పద్ధతులతో పోల్చినప్పుడు ఇది చాలా నెమ్మదిగా మరియు తక్కువ సామర్థ్యంతో ఉంటుంది.

లిస్ట్ అనునది (n) మూలకాల అమరిక అని అనుకొనుము స్వాప్ ఫంక్షన్ ఇవ్వబడిన అమరిక మూలకాల విలువలను మార్పిడి చేస్తుంది.

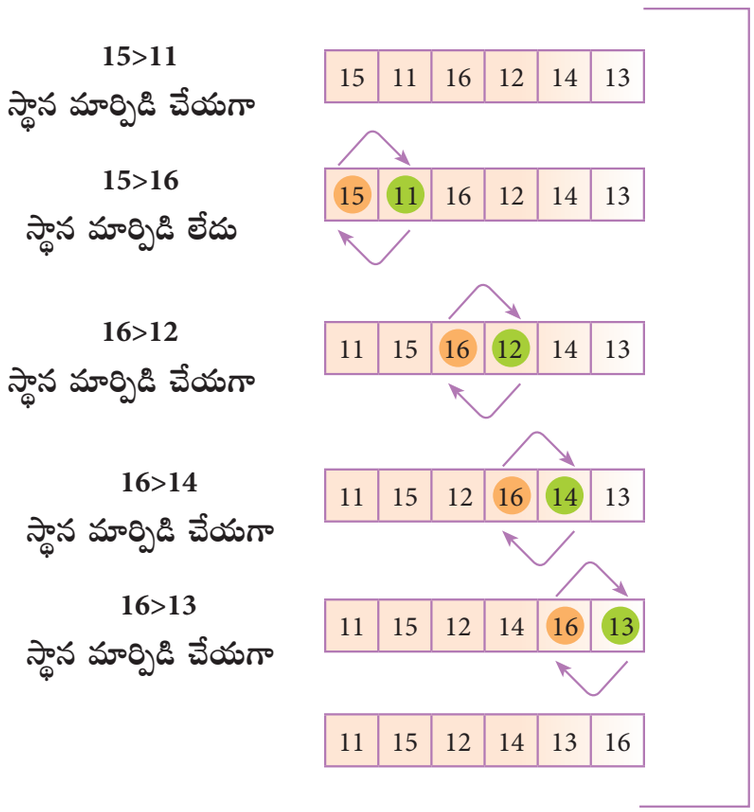
సూడో కోడ్ (Pseudo code)

1. మొదటి మూలకంతో ప్రారంభించి అనగా సూచిక = 0, ప్రస్తుత మూలకాన్ని అమరికలోని తదుపరి మూలకంతో పోల్చండి.
2. ప్రస్తుత మూలకం అమరిక యొక్క తదుపరి మూలకం కంటే ఎక్కువగా ఉంటే, వాటిని మార్చుకోండి.
3. ప్రస్తుత మూలకం, తదుపరి మూలకం లేదా కుడివైపు కంటే తక్కువగా ఉంటే, తదుపరి మూలకానికి వెళ్ళాలి. దశ−1 కి వెళ్ళి మరియు

12th_CS_Telugu_Medium_Chapter_4.indd 40 27-03-2023 10:38:01

సూచిక ముగింపుకు వచ్చే వరకు పునరావృతం చేయండి.

విలువలతో కూడిన అమరికను (15,11,16,12,14,13) బబుల్ సార్ట్ ఎలా క్రమబద్ధీకరిస్తుందో చిత్ర రూపంలో క్రింద ఇవ్వబడినదానిని పరిశీలిద్దాం.



పై చిత్రరూపం పునరావృతం −1 కొరకు ఉదాహరణ. అదేవిధంగా మిగిలిన పునరావృతములను పూర్తిచేయవలెను. చివరి పునరావృతము క్రమబద్ధీకరించిన అమరికను ఇచ్చును.

అన్ని పునరావృతాల చివరలో మనం క్రింద ఇచ్చిన విధంగా అమరికలో క్రమబద్ధీకరించిన విలువలను పొందుతాము.

11	12	13	14	15	16

4.5.2 సెలెక్షన్ సార్ట్ (ఎంపిక క్రమబద్ధీకరణ) (Selection sort)

ఎంపిక క్రమబద్ధీకరణ అనేది ఒక సాధారణ సార్టింగ్ అల్గోరిథం, ఇది లిస్ట్ ద్వారా ప్రతి ప్రవేశం కోసం ఒక మార్పిడిని మాత్రమే చేయడం వలన బబుల్ సార్ట్ యొక్క పనితీరును మెరుగుపరుస్తుంది. ఈ అల్గోరిథం అమరికలోని అతిచిన్న మూలకాలను కనుగొని, మొదటి స్థానంలో ఉన్న మూలకంతో దాన్ని మార్పిడి చేస్తుంది. తరువాత అది రెండవ చిన్న మూలకాన్ని కనుగొనును. మరియు ఆ మూలకాన్ని రెండవ స్థానంలో ఉన్న మూలకంతో మార్పిడి చేయిస్తుంది. మరియు ఇది మొత్తం అమరిక క్రమంలో క్రమబద్ధీకరించే వరకు కొనసాగుతుంది.

ఈ అల్గోరిథం పదేపదే తదుపరి చిన్న మూలకాన్ని ఎన్నుకుంటుంది. మరియు ప్రతి ప్రవేశంలో సరైన స్థలంలోకి మారుతుంది. అందువల్ల దానిని ఎంపిక క్రమబద్ధీకరణ అంటారు.

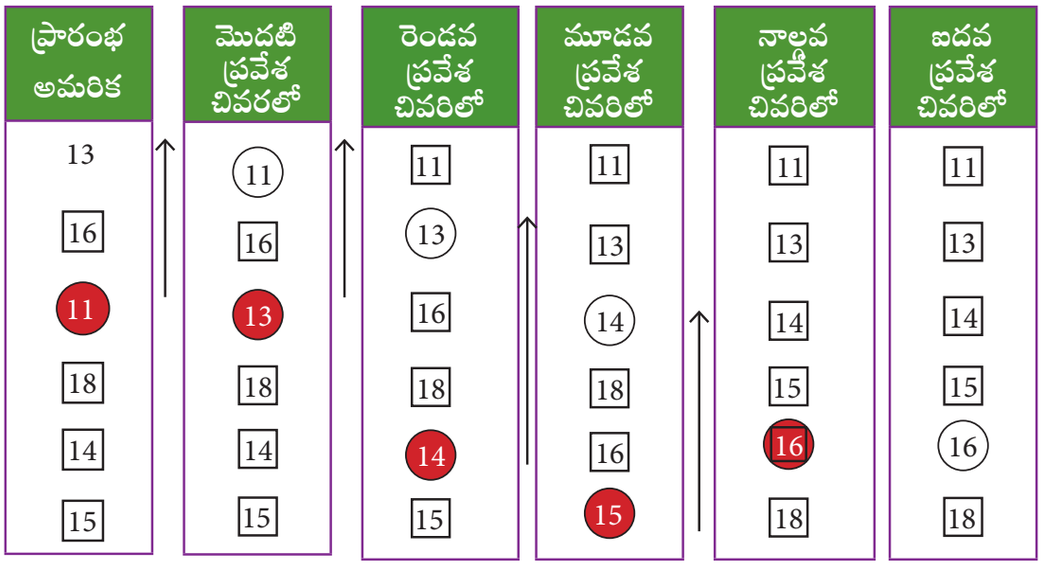
12th_CS_Telugu_Medium_Chapter_4.indd 41 27-03-2023 10:38:01

సూడో కోడ్ (Pseudo code):

1. మొదటి మూలకం నుండి ప్రారంభించండి, అనగా సూచిక – 0 మనం అమరికలోని అతి చిన్న మూలకాన్ని శోధిస్తాము మరియు దానిని మొదటి స్థానంలో ఉన్న మూలకంతో భర్తీచేయుము.
2. మనం ఇప్పుడు రెండవ మూలకం యొక్క స్థానానికి వెళ్తాము, మరియు ప్రారంభ సూచిక నుండి ఉప అమరిక యొక్క చివరి సూచిక వరకు గల ఉన్న అతిచిన్న మూలకమును చూద్దాం.
3. ఇప్పుడు దశ –2లో గుర్తించిన రెండవ చిన్నదాన్ని లేదా అసలు అమరికలోని రెండవ స్థానంలోకి మార్చండి లేదా ఉప అమరికలో మొదటి స్థానాన్ని మార్చవచ్చును.
4. అమరిక పూర్తిగా క్రమబద్ధీకరించబడే వరకు ఇది పునరావృతమవుతుంది.

(13,16,11,18,14,15) విలువలతో కూడిన అమరికను పరిశీలిద్దాం.

ఎంపిక సార్ట్ ఇచ్చిన అమరికను ఎలా క్రమబద్ధీకరిస్తుందో చిత్రరూపంలో క్రింద సూచించబడినది.



మొదటి ప్రవేశంలో అతి చిన్న మూలకం 11 కావున మొదటి స్థానంలో ఉంచబడుతుంది.

ఆ తరువాత, తదుపరి చిన్న మూలకం అమరికలో శోధించబడును. ఇప్పుడు మనకు చిన్నది 13 కనుక ఇది రెండవ స్థానంలో ఉంచబడుతుంది.

ఇప్పుడు మొదటి మూలకాన్ని వదిలివేస్తే, మిగిలిన మూలకాల నుండి తదుపరి చిన్న మూలకం శోధించబడుతుంది. మనం 13ను చిన్నదిగా పొందుతాము. కనుక ఇది రెండవ స్థానంలో ఉంచబడుతుంది.

ఇప్పుడు 11 మరియు 13 లను వదిలి వేయండి. అవి సరైన స్థితిలో ఉన్నందున, మిగిలిన మూలకాల నుండి తదుపరి చిన్న మూలకం కోసం శోధిస్తాము. దానిని మూడవ స్థానంలో ఉంచి మరియు అమరిక క్రమబద్ధీకరించబడే వరకు దీన్ని చేయవలెను.

చివరగా పై చూపిన రేఖాచిత్రం విధంగా ప్రవేశం యొక్క ముగింపులో క్రమబద్ధీకరించబడిన అమరకను పొందుతాము.

12th_CS_Telugu_Medium_Chapter_4.indd 42 27-03-2023 10:38:01

4.5.3 చొప్పించు క్రమబద్ధీకరణ (Insertion sort)

చొప్పించు క్రమబద్ధీకరణ ఒక సాధారణ సార్టింగ్ అల్గోరిథం. ఇది లిస్ట్ నుండి మూలకాలను ఒక్కొక్కటిగా తీసుకొని, వాటిని సరైన స్థానంలో చేర్చడం ద్వారా క్రొత్త క్రమబద్ధీకరణ లిస్ట్‌గా పనిచేస్తుంది. ఈ అల్గోరిథం చివరిలో క్రమబద్ధీకరించిన అమరికను నిర్మిస్తుంది. ఈ అల్డోరిథం మనం చర్చించినట్లుగా మునుపటి అల్గోరిథం ప్రకారం తుది క్రమబద్ధీకరించబడిన లిస్ట్‌ను పొందడానికి (n-1) సంఖ్య ప్రవేశాలను ఉపయోగిస్తుంది.

చొప్పించు క్రమబద్ధీకరణకు గల సూడో కోడ్

దశ –1 : ఇది మొదటి మూలకం అయితే, ఇది ఇప్పటికే క్రమబద్ధీకరించబడింది.

దశ –2 : తదుపరి మూలకాన్ని ఎంచుకొండి.

దశ –3 : క్రమబద్ధీకరించబడిన ఉప–లిస్ట్‌లోని అన్ని మూలకాలను పోల్చండి.

దశ –4 : క్రమబద్ధీకరించబడిన ఉప–లిస్ట్‌లోని అన్ని మూలకాలు క్రమబద్ధీకరించాల్సిన విలువ కంటే పెద్ద విలువైనవి అయితే, దాన్ని తప్పక భర్తీ చేయాలి.

దశ –5 : విలువను చొప్పించండి.

దశ –6 : లిస్ట్ క్రమబద్ధీకరించబడే వరకు ఈ విధానాన్ని పునరావృతం చేయండి.

44	16	83	07	67	21	34	45	10	Assume 44 is a sorted list of 1 item
16	44	83	07	67	21	34	45	10	inserted 16
16	44	83	07	67	21	34	45	10	inserted 83
07	16	44	83	67	21	34	45	10	inserted 07
07	16	44	67	83	21	34	45	10	inserted 67
07	16	21	44	67	83	34	45	10	inserted 21
07	16	21	34	44	67	83	45	10	inserted 34
07	16	21	34	44	45	67	83	10	inserted 45
07	10	16	21	34	44	45	67	83	inserted 10

ప్రవేశంలో చివరలో చొప్పించి క్రమబద్ధీకరణ అల్గోరిథం కమబద్ధీకరించిన ఔట్‌పుట్‌ను ఆరోహణ క్రమంలో క్రింది చూపిన విధంగా యిస్తుంది.

07	10	16	21	34	44	45	67	83

4.6. డైనమిక్ ప్రోగ్రామింగ్

డైనమిక్ ప్రోగ్రామింగ్ అనేది ప్రోగ్రామింగ్ అల్గోరిథమిక్ డిజైన్ పద్ధతి, ఇది ఒక సమస్యకు పరిష్కారాన్ని నిర్ణయాల క్రమం ఫలితంగా చూడటానికి ఉపయోగపడుతుంది. డైనమిక్ ప్రోగ్రామింగ్ విధానం విభజించి జయించడం వలె ఉంటుంది. ఇచ్చిన సమస్యను చిన్న మరియు ఇంకా చిన్న ఉపసమస్యలుగా విభజించబడుతుంది.

సమస్యలను సారూప్య ఉప–సమస్యలుగా విభజించినప్పుడల్లా డైనమిక్ ప్రోగ్రామింగ్ ఉపయోగించబడుతుంది. తద్వారా ప్రక్రియను పూర్తిచేయడానికి వాటి ఫలితాలను తిరిగి ఉపయోగించవచ్చు.

12th_CS_Telugu_Medium_Chapter_4.indd 43 27-03-2023 10:38:01

అనుకూల మార్గంలో పరిష్కారాన్ని కనుగొనడానికి డైనమిక్ ప్రోగ్రామింగ్ విధానాలు ఉపయోగించబడుతాయి. ప్రతి అంతర్గత ఉపసమస్యకు, డైనమిక్ అల్గోరిథం గతంలో పరిష్కరించిన ఉపసమస్యల ఫలితాలను తనిఖీ చేయడానికి ప్రయత్నిస్తుంది. మెరుగైన పరిష్కారం పొందడానికి పరస్పర ఉపసమస్యల పరిష్కారాలను కలుపవలెను. డైనమిక్ ప్రోగ్రామింగ్‌కు గల దశలు :

డైనమిక్ ప్రోగ్రామింగ్‌కు గల దశలు (Steps to do Dynamic programming)

- ఇచ్చిన సమస్య చిన్న పరస్పర ఉపసమస్యలుగా విభజించబడుతుంది.
- ఇచ్చిన సమస్యకు మెరుగైన పరిష్కారం చిన్న ఉపసమస్య యొక్క ఫలితాన్ని ఉపయోగించడం ద్వారా పొందవచ్చు.
- డైనమిక్ అల్గోరిథంలు మెమాయిజేషన్‌ను (Memoization) ఉపయోగించుచున్నవి.

గమనిక

మెమాయిజేషన్ అనేది ప్రధానంగా కంప్యూటర్ ప్రోగ్రామ్‌లను వేగవంతం చేయడానికి ఖరీదైన ఫంక్షన్ కాల ఫలితాలను నిల్వచేయడం ద్వారా మరియు అదే ఇన్‌పుట్ మళ్ళీ సంభవించినప్పుడు (cached) చేసిన ఫలితాన్ని తిరిగి ఇవ్వడానికి ఉపయోగించే ఆప్టిమైజేషన్ టెక్నిక్.

4.6.1 ఫైబొనాక్సీ సిరీస్ – ఉదాహరణ (Fibonacci Series – An example)

రెండు మునుపటి సంఖ్యలను జోడించడం ద్వారా ఫైబొనాక్సీ సిరీస్ తదుపరి సంఖ్యను ఉత్పత్తి చేస్తుంది. ఫైబొనాక్సీ సిరీస్ రెండు సంఖ్యలు – Fib 0 మరియు Fib 1 నుండి మొదలవుతుంది. Fib 0 మరియు Fib 1 యొక్క ప్రారంభ విలువలను 0 మరియు 1గా తీసుకొనవచ్చు.

ఫైబొనాక్సీ సిరీస్ ఈ క్రింది నిబంధనను తృప్తి పరుస్తుంది.

$Fibn = Fib_{n-1} + Fib_{n-2}$

'n' విలువ 8గా ఉండినా, ఫైబొనాక్సీ వరుస క్రింది విధంగా ఉండును.

Fib_8 = 0 1 1 2 3 5 8 13

4.6.2 డైనమిక్ ప్రోగ్రామింగ్ విధానంతో ఫైబొనాక్సీ పునరావృత అల్గోరిథం (Fibonacci Iterative Algorithm with Dynamic programming approach) :

క్రింది ఉదాహరణ ఫైబొనాక్సీ సిరీస్ ఉత్పత్తి కోసం సాధారణ డైనమిక్ ప్రోగ్రామింగ్ విధానాన్ని చూపిస్తుంది.

f0=0, f1 =1 అను ప్రారంభ విలువలను తీసుకొనవలెను.

దశ–1 : పై బొనాక్సి ప్రారంభ విలువలను f0 మరియు f1 లను ముద్రించవలెను. (Print the initial values of Fibonacci f0 and f1)

12th_CS_Telugu_Medium_Chapter_4.indd 44 27-03-2023 10:38:01

దశ –2 : ఫై బొనాక్సి fib ← f0 + f1 ను లెక్కించండి.

దశ –3 : f0← f1, f1← fib గా కేటాయించండి.

దశ –4 : ఫై బొనాక్సి fib యొక్క తదుపరి వరుసలో వున్న విలువను ముద్రించుము.

దశ –5 : నిర్దిష్ట పదాల సంఖ్య వరకు దశ 2 ను పునరావృతం చేయుము.

ఉదాహరణకు మనం ఫైబొనాక్సి సిరీస్ ను 10 అంకెలు వరకు ఉత్పత్తి చేయాలంటే అల్గోరిథం సిరీస్‌ను క్రింద చూపిన విధంగా ఉత్పత్తి చేస్తుంది.

ఫైబొనాక్సి సిరీస్ : : 0 1 1 2 3 5 8 13 21 34 55

గుర్తించుకోవలసిన విషయాలు

- అల్గోరిథం అనేది నిర్దిష్ట పనిని నెరవేర్చడానికి పరిమితమైన సూచనల సమితి.
- అల్గోరిథం ఒక పనిని పూర్తి చేయడానికి అవసరమైన దశల వారీ సూచనలను కలిగి ఉంటుంది మరియు ప్రోగ్రామర్‌ను ప్రోగ్రామ్‌ను అభివృద్ధి చేయడానికి సహాయపడుతుంది.
- ప్రోగ్రామింగ్ భాషలో ప్రోగ్రామ్ అనేది అల్గోరిథం యొక్క వ్యక్తీకరణ.
- అల్గోరిథం విశ్లేషణ వివిధ కార్యకలాపాల అమలు లేదా నడుస్తున్న సమయంతో వ్యవహరిస్తుంది.
- అల్గోరిథం యొక్క స్థల సంక్లిష్టత అనునది దాని పూర్తికి అవసరమైన మెమరీ మొత్తం అగును.
- అల్గోరిథం యొక్క చెత్త కేసును వివరించడానికి Big Oh తరచుగా ఉపయోగించబడుతుంది.
- Big Omega తక్కువ స్థాయిని వివరించడానికి ఉపయోగించబడుతుంది. ఇది స్థల సంక్లిష్టతను పరిష్కరించడానికి ఉత్తమ మార్గం.
- అల్గోరిథం యొక్క సమయ సంక్లిష్టిత అనునది అల్గోరిథం ప్రక్రియను పూర్తిచేయడానకి తీసుకొను దశల సంఖ్య ద్వారా ఇవ్వబడుతుంది.
- అల్గోరిథం యొక్క సామర్థ్యాన్ని అల్గోరిథం ఉపయోగించే గణన వనరుల సంఖ్యగా నిర్వచించారు.
- అల్గోరిథం రూపకల్పన యొక్క మార్గాన్ని అల్గోరిథమిక్ స్ట్రాటజీ అని అంటారు.
- స్థలం–సమయం మార్పిడి లేదా సమయం నిల్వ మార్పిడి అనునది తక్కువ సమయంలో ఎక్కువ నిల్వ స్థలాన్ని ఉపయోగించి అల్గోరిథంను పరిష్కరించే మార్గం.
- అసింప్టిక్ సంకేతాలు (Asymptotic Notations) సమయం మరియు స్థల సంక్లిష్టత గురించి అర్థవంతమైన ప్రకటనలను ఉపయోగించే భాషలు.
- బబుల్ సార్టింగ్ ఒక సాధారణ సార్టింగ్ అల్గోరిథం. ఇది ప్రతి జత ప్రక్కనే ఉన్న మూలకాలను పోల్చి, క్రమబద్ధీకరించిన క్రమంలో ఉంటే, వాటిని మార్పిడి చేస్తుంది.

12th_CS_Telugu_Medium_Chapter_4.indd 45 27-03-2023 10:38:01

గుర్తించుకోవలసిన విషయాలు

- ఎంపిక క్రమబద్ధీకరణ లిస్ట్ ద్వారా ప్రతిప్రవేశం కోసం ఒక మార్పిడిని మాత్రమే చేయడం వలన బబుల్‌సార్ట్ యొక్క పనితీరును మెరుగుపరుస్తుంది.
- చొప్పించు క్రమబద్ధీకరణ అనేది సాధారణ సార్టింగ్ అల్గోరిథం, ఇది తుది క్రమబద్ధీకరించబడిన అమరికని నిర్మిస్తుంది లేదా ఒక సమయంలో ఒక అంశాన్ని లిస్ట్ చేస్తుంది. ఇది ఎల్లప్పుడూ లిస్ట్ యొక్క దిగువ స్థానాల్లో క్రమబద్ధీకరించిన ఉప లిస్ట్‌ను నిర్వహిస్తుంది.
- నిర్ణయాల క్రమం ఫలితంగా సమస్యకు పరిష్కారాలను చూడగలిగినప్పుడు డైనమిక్ ప్రోగ్రామింగ్ ఉపయోగించబడుతుంది.

39Q05

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. పర్షియన్ గణిత శాస్త్రజ్ఞుడు అబూ జాఫర్ మహమ్మద్ ఇబ్నుమూసా అల్ ఖౌరిజ్మి పేరు నుండి పదం.

   ఎ) ఫ్లోచార్ట్ బి) ఫ్లో సి) అల్గోరిథం డి) సింటాక్స్

2. క్రింది సార్టింగ్ అల్గోరిథంలలో ఏ అల్గోరిథంకు కనీస సంఖ్య మార్పిడి అవసరం.

   ఎ) Bubble sort బి) Quick sort సి) Merge sort డి) Selection sort

3. అల్గోరిథం యొక్క సామర్థ్యానికి గల రెండు ప్రధాన గణనలు :

   ఎ) ప్రాసెసర్ మరియు మెమరీ బి) సంక్లిష్టత మరియు సామర్థ్యం

   సి) సమయం మరియు Space డి) డేటా మరియు Space

4. సరళ శోధన అల్గోరిథం యొక్క సంక్లిష్టత.

   ఎ) O(n) బి) O(log n) సి) O(n2) డి) O(n log n)

5. క్రింది సార్టింగ్ అల్గోరిథంలో దేనికి అతి తక్కువ చెత్త పరిస్థితి సంక్లిష్టతను కలిగివుండును.

   ఎ) Bubble sort బి) Quick sort సి) Merge sort డి) Selection sort

12th_CS_Telugu_Medium_Chapter_4.indd 46 27-03-2023 10:38:02

6. క్రింది వాటిలో ఏది స్థిర సార్టింగ్ అల్గోరిథం కాదు.

ఎ) Insertion sort బి) Selection sort సి) Bubble sort డి) Merge sort

7. ఉత్తమ పరిస్థితిలో బబుల్ సార్ట్ యొక్క సమయ సంక్లిష్టత.

ఎ) θ (n) బి) θ (nlogn) సి) θ (n2) డి) θ (n(logn) 2)

8. అసింప్టిక్ మూల్యాంకనంలో Θ సంజ్ఞామానం సూచించునది.

ఎ) ఆధార పరిస్థితి బి) సరాసరి పరిస్థితి సి) చెత్త పరిస్థితి డి) శూన్య పరిస్థితి

9. ఒక సమస్యను ఉపసమస్యలుగా విభజించి, వాటిని అనేక సార్లు తిరిగి ఉపయోగించబడితే, ఆ సమస్య ఏ నియమాన్ని కలిగివుంటుంది.

ఎ) పరస్పర ఉపసమస్యలు బి) ఆప్టిమల్ సబ్‌స్ట్రక్చర్

సి) మెయోయి జేషన్ డి) అత్యాశ

10. డైనమిక్ ప్రోగ్రామింగ్‌లో గతంలో లెక్కించిన విలువను నిల్వ చేసే సాంకేతికతను ఏమంటారు.

ఎ) విలువను నిల్వచేయు నియమం బి) విలువను సేకరించు నియమం

సి) మెమాయిజేషన్ డి) మ్యాపింగ్

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము. **(2 మార్కులు)**

1. అల్గోరిథం అంటే ఏమిటి?
2. సూడో కోడ్‌ను నిర్వచింపుము.
3. అల్గోరిస్ట్ ఎవరు?
4. క్రమబద్ధీకరించడం అంటే ఏమిటి?
5. శోధించడం అంటే ఏమిటి? దాని రకాలను వ్రాయండి.

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము. **(3 మార్కులు)**

1. అల్గోరిథం యొక్క లక్షణాలను జాబితా చేయండి.
2. అల్గోరిథమిక్ సంక్లిష్టత మరియు దాని రకాలను గురించి చర్చించండి.

12th_CS_Telugu_Medium_Chapter_4.indd 47 27-03-2023 10:38:02

3. సమయం మరియు స్థల సంక్లిష్టతలను ప్రభావితం చేయు కారకాలను తెలుపుము.

4. అసింప్టిక్ సంజ్ఞామానం పై గమనికను వ్రాయుము.

5. డైనమిక్ ప్రోగ్రామింగ్ ద్వారా మీరు అర్థం చేసుకొన్నది ఏమి?

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. అల్గోరిథం యొక్క లక్షణాలను వివరించండి.

2. సరళ శోధన అల్గోరిథం గురించి చర్చించండి.

3. బైనరీ శోధన అనగానేమి! ఉదాహరణతో వివరించుము.

4. బబుల్ సార్ట్ అల్గోరిథంను ఉదాహరణతో వివరించండి.

5. డైనమిక్ ప్రోగ్రామింగ్ భావనను తగిన ఉదాహరణతో వివరించండి.

Reference Books

1. *Fundamentals Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar, Rajasekaran, Second Edition, University press (India) Limited, 2013.*

2. *Design and Analysis of Algorithms, S. Sridhar, Oxford University Press, 2015*

Web References

www.wickipedia.org

విద్యార్థుల కృత్యము

1. మీ తరగతి విద్యార్థుల త్రైమాసిక పరీక్ష మార్కులు అన్ని పరిస్థితులను సంతృప్తిపరచు విధంగా గ్రేడింగ్ కోసం ఒక అల్గోరిథంను సృష్టించండి.

12th_CS_Telugu_Medium_Chapter_4.indd 48 27-03-2023 10:38:02

యూనిట్ II

అధ్యాయం 5

పైథాన్ వేరియబుల్స్ మరియు ఆపరేటర్లు

U86IT

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును చదివిన తరువాత విద్యార్థులు తెలుసుకొను విషయాలు

- పైథాన్ ప్రోగ్రాములను సృష్టించడానికి గ్రాఫికల్ యూసర్ ఇంటర్‌పేస్ మరియు ఇంటిగ్రేటడ్ డెవలప్‌మెంట్ ఎన్విరాన్‌మెంట్ వాడకాన్ని అబినందించడం.
- ప్రోగ్రామింగ్ కోసము ఇంటరాక్టివ్ మరియు స్క్రిప్ట్ మోడ్‌లో పనిచేయడము.
- వేరియబుల్స్ విలువలను సృష్టించడం మరియు కేటాయించడం.
- పైథాన్‌లో వేర్వేరు రకాల డేటా ఉపయోగాలు భావనను అవగాహన చేసుకొనుట.
- వివిధ రకాల ఆపరేటర్‌ల ప్రాముఖ్యత మరియు వినియోగాన్ని తెలుసుకొనుట.
- పైథాన్ వ్యక్తీకరణ మరియు ప్రకటనలను సృష్టించడము.

5.1 పరిచయము



పైథాన్ నెదర్లాండ్స్‌లోని గణితము మరియు కంప్యూటర్ సైన్స్ కోసం జాతీయ పరిశోధనా సంస్థ అయిన CWI (Centrum Wiskunde & Informatica) నుండి గైడో వాన్ పోసమ్ చేత సృష్టించబడిన ఒక సాధారణ ప్రయోజన ప్రోగ్రామింగ్. ఈ language 1991లో ప్రవేశపెట్టబడినది. డబ్బై దశకములో మాంటి పైథాన్ యొక్క ఫ్లయింగ్ సర్కస్ నుండి “Monty Python’s Flying Circus”. BBC కామిడీ సీరిస్ నుండి పైథాన్ పేరు వచ్చింది. పైథాన్ విధానపరమైన మరియు ఆబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రామింగ్ విధానాలకు మద్దత్తు ఇస్తుంది.

5.2 పైథాన్ యొక్క ముఖ్య లక్షణాలు

✓ ఇది సాధారణ ప్రయోజన programming language ఇది శాస్త్రీయ మరియు అశాస్త్రీయ ప్రోగ్రామింగ్ రెండింటికి ఉపయోగించబడుతుంది.

✓ ఇది ఒక స్వతంత్ర programming language ప్లాట్‌ఫామ్ అవుతుంది.

12th_CS_Telugu_Medium_Chapter_5.indd 49 27-03-2023 10:38:21

✓ పైథాన్‌లో వ్రాసిన ప్రోగ్రామ్‌లు సులభముగా చదవగలిగినవి మరియు అర్థమగునవి.

పైథాన్ 3.x వర్షన్‌కు **IDLE** (**I**ntegrated **D**evelopment **L**earning **E**nvironment)ఉపయోగించి పైథాన్ కోడ్ అభివృద్ధి చేయడానికి మరియు అమలు చేయుటకు వీలగును దీనికి. **www.python.org.** నుండి డౌన్‌లోడ్ చేసుకోవచ్చు.

5.3 పైథాన్‌లో ప్రోగ్రామింగ్

పైథాన్ ప్రోగ్రామ్‌లను ఇంటరాక్టివ్ మోడో మరియు స్క్రిప్ట్ మోడ్ అని రెండు విధాలుగా వ్రాయవచ్చు. ఇంటరాక్టివ్ మోడ్ పైథాన్ కమాండ్ ప్రాంప్ట్‌లో (>>>) కోడ్‌లను వ్రాయడానికి అనుమతిస్తుంది. అయితే స్క్రిప్ట్ మోడ్ ప్రోగ్రామ్‌లను ఎక్స్‌టెన్షన్‌లో ప్రత్యేక ఫైల్‌గా వ్రాసి నిల్వ చేయవచ్చు. పై మరియు అమలు పైథాన్ సోర్స్ ఫైల్‌ను సృష్టించడానికి మరియు సవరించడానికి స్క్రిప్ట్ మోడ్ ఉపయోగించబడుతుంది.

5.3.1 ఇంటరాక్టివ్ మోడ్ ప్రోగ్రామింగ్ (Interactive mode Programming)

ఇంటరాక్టివ్ మోడ్‌లు పైథాన్ కోడ్‌ను నేరుగా type చేయవచ్చు మరియు వ్యాఖ్యలా ఫలితాన్ని వెంటనే ప్రదర్శిస్తుంది. ఇంటరాక్టివ్ మోడ్‌ను సాధారణ కాలిక్యులేటర్‌గా ఉపయోగించవచ్చు.

(i) పైథాన్ IDLE ని ప్రారంభించడము (Invoking Python IDLE)

క్రింది ఆదేశముతో విండోస్ ఆపరేటింగ్ సిస్టమ్‌లో పైథాన్ IDLE ను ప్రారంభించవచ్చును.

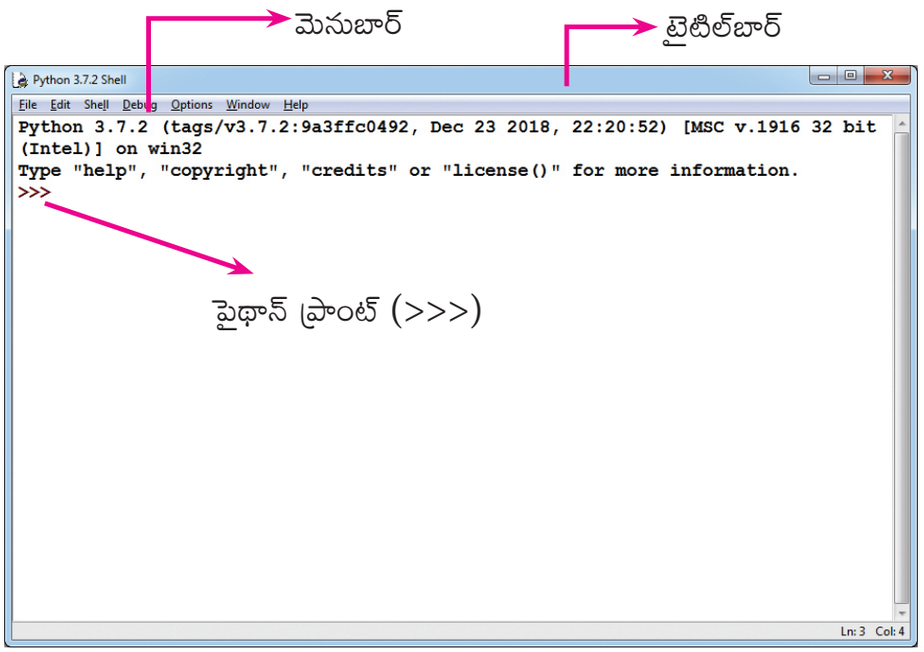
Start → All Programs → Python 3.x → IDLE (Python 3.x)

(లేదా)

హోమ్ స్క్రీన్‌లో 

ఒక చిత్రముంటే ప్రారంభించటానికి మీరు దానిపై క్లిక్ చేయవచ్చు.

ఇప్పుడు పటం 5.1 పైథాన్‌లో చూపిన విధంగా పైథాన్ **IDLE** స్క్రీన్ కనిపిస్తుంది.



పైథాన్ *IDLE* విండో

12th_CS_Telugu_Medium_Chapter_5.indd 50 27-03-2023 10:38:21

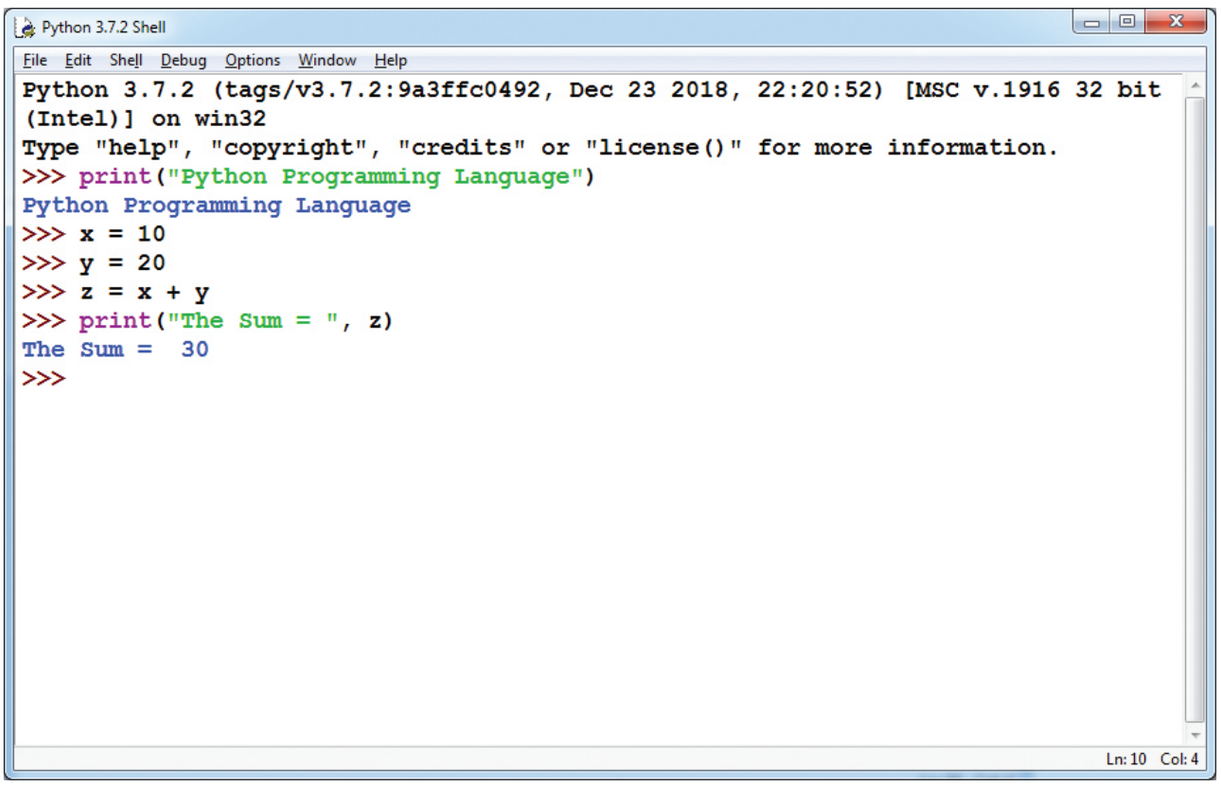
ప్రాంప్ట్ (>>>) సూచనలను అంగీకరించడానికి వ్యాఖ్యత (Interpreter) సిద్ధంగా ఉందని సూచిస్తుంది. అందువల్ల స్క్రీన్‌పై ప్రాంప్ట్ ఉంటే **IDLE** ఇంటరాక్టివ్ మోడ్‌లో పనిచేస్తుందని అర్థం. ఇపుడు సాధారణ గణిత వ్యక్తీకరణలను (mathematical expressions) ను ఉపయోగించి సాధారణ క్యాలికులేటర్‌గా ప్రయత్నిద్దాం.

ఉదాహరణ 1:

```
>>> 5 + 10
15
>>> 5 + 50 *10
505
>>> 5 ** 2
25
```

ఉదాహరణ 2:

```
>>>print ("Python Programming Language")
Python Programming Language
>>>x=10
>>>y=20
>>>z=x + y
>>>print ("The Sum", z)
The Sum = 30
```



పైథాన్ ఇంటరాక్టివ్ విండో

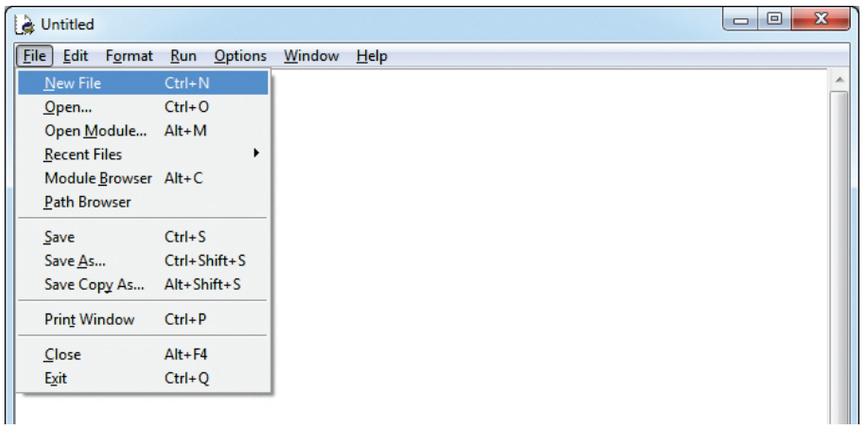
5.3.2 స్క్రిప్ట్ మోడ్ ప్రోగ్రామింగ్ (Script mode Programming)

ప్రాథమికంగా స్క్రిప్ట్ అనేది పైథాన్ స్టేట్‌మెంట్ ఉన్న టెక్స్ట్ ఫైల్. పైథాన్ స్క్రిప్ట్ పునర్వినియోగ కోడ్. స్క్రిప్ట్ సృష్టించబడినపుడు దాన్ని మళ్ళీ మళ్ళీ అమలు చేయడం ద్వారా అమలు చేయవచ్చు.

(i) పైథాన్ స్క్రిప్ట్‌ను నిర్మించుట:

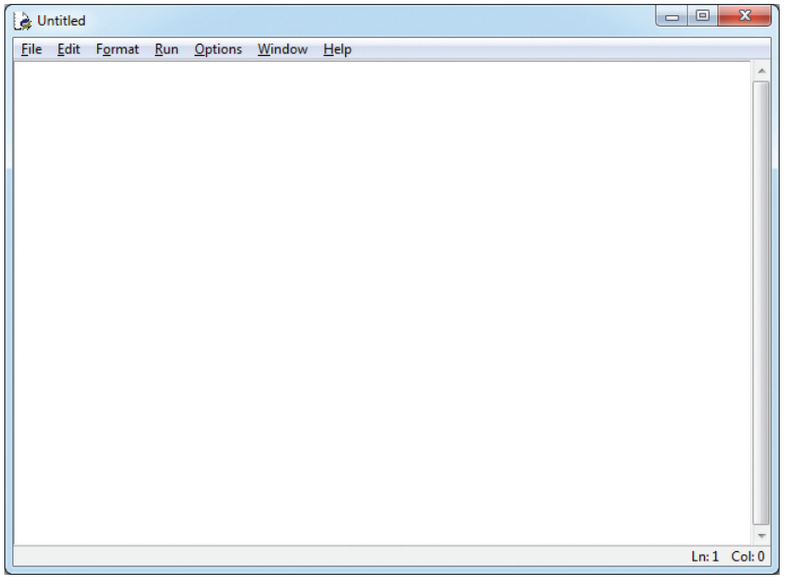
File → New File లేదా **Ctrl + N** నుప్రెస్ చేయడం.

12th_CS_Telugu_Medium_Chapter_5.indd 51 27-03-2023 10:38:21



పటం 5.3 – క్రొత్త ఫైల్‌ను నిర్మించుట

2. పేరు పెట్టని ఖాళీ స్క్రిప్ట్ టెక్స్ట్ ఎడిటర్ చూపిన విధముగా తెరపై ప్రదర్శించబడుతుంది.



పటం 5.3 పేరు పెట్టని ఖాళీ పైథాన్ స్క్రిప్ట్ ఎడిటర్

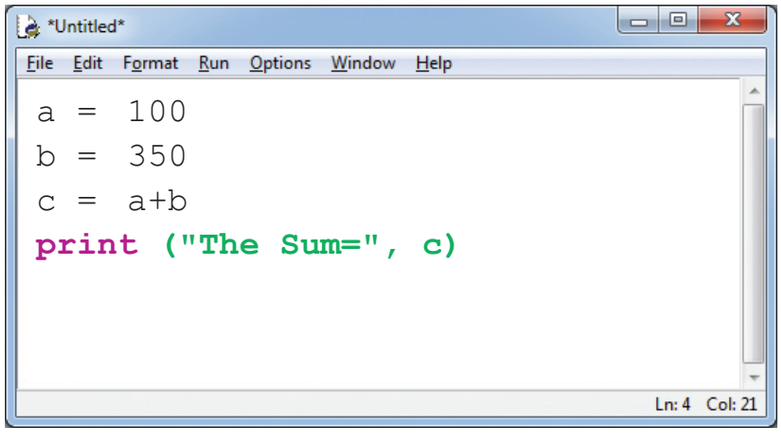
3. స్క్రిప్ట్ ఎడిటర్‌లో క్రింది కోడ్‌ను టైపు చేయండి.

```
a =100

b = 350

c = a+b

print ("The Sum=", c)
```

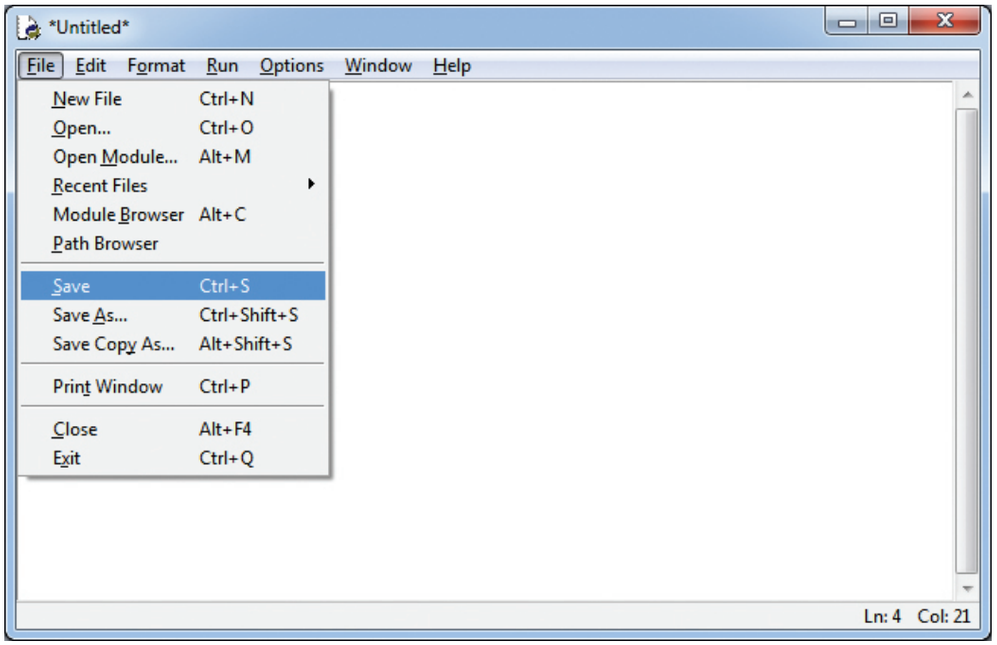


పటం 5.4 పైథాన్ నమూనా కోడ్

12th_CS_Telugu_Medium_Chapter_5.indd 52 27-03-2023 10:38:21

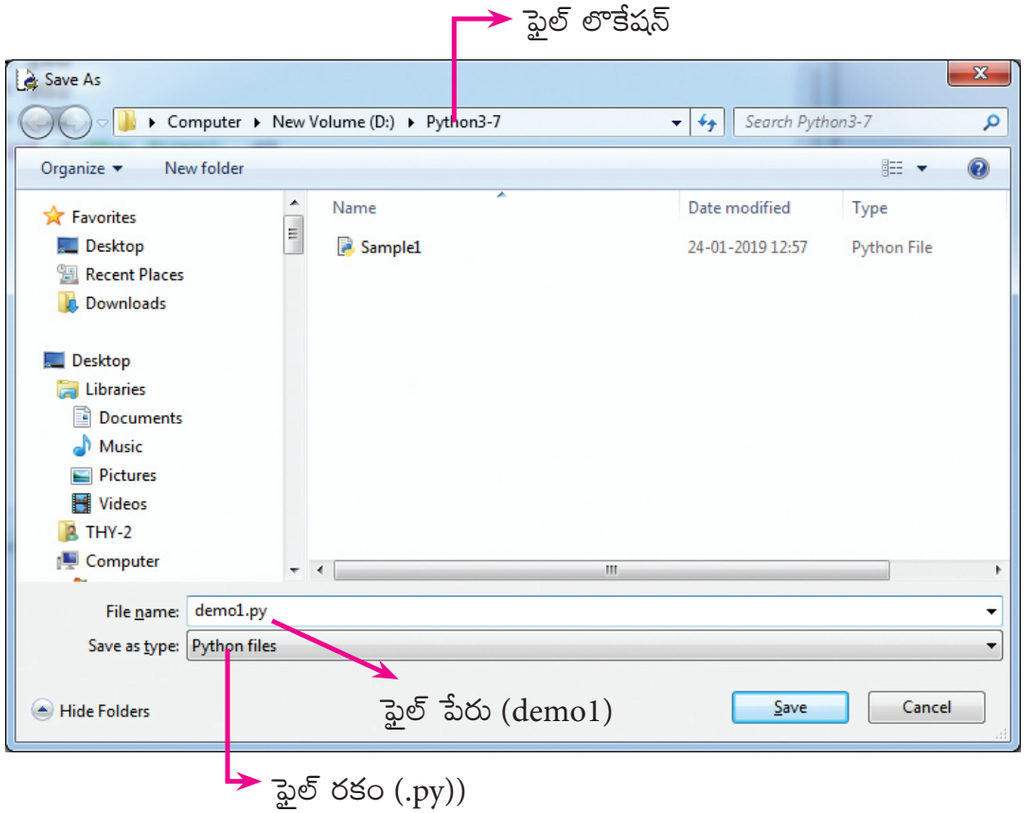
(ii) పైథాన్ స్క్రిప్ట్ ను సేవ్ చేయడం

(1) Choose **File** → **Save** లేదా Press **Ctrl + S**



పటం 5.5 file ను మొదటిసారి Save చేయడం.

2. ఇప్పుడు **Save As** డైలాగ్ బాక్స్ స్క్రీన్ పైన కనిపించుట



పటం 5.6 Save As డైలాగ్ Box

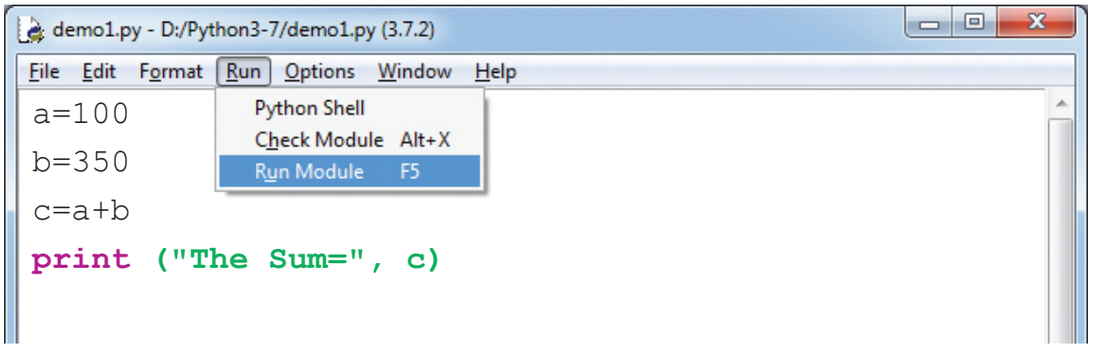
12th_CS_Telugu_Medium_Chapter_5.indd 53 27-03-2023 10:38:22

3. సేవ్ ఆస్ (**Save As**) డైలాగ్ బ్లాక్స్‌లో save చేయవలసిని ఫైల్‌ను ఎంచుకొనవలయును. అపుడు మీరు ఫైల్ పేరు డైలాగ్ బాక్స్‌లో పైథాన్ ప్రమేయంగా నమోదు చేయాలి. **py** పొడిగింపుతో ఫైల్‌ను save చేయబడతాయి. కాబట్టి పైథాన్ స్క్రిప్ట్ చేయబడిన ఫైళ్ళ కోసం వినియోగదారు పొడిగింపును. **py** పేర్కొనవలసిన అవసరం లేదు.

4. చివరగా Save బటన్‌ను ఎంచుకోవడం ద్వారా మీరు స్క్రిప్ట్‌ను ఫైల్‌లో Save చేయవచ్చు

(iii) పైథాన్ స్క్రిప్ట్‌ను అమలుచేయడం. (Executing Python Script)

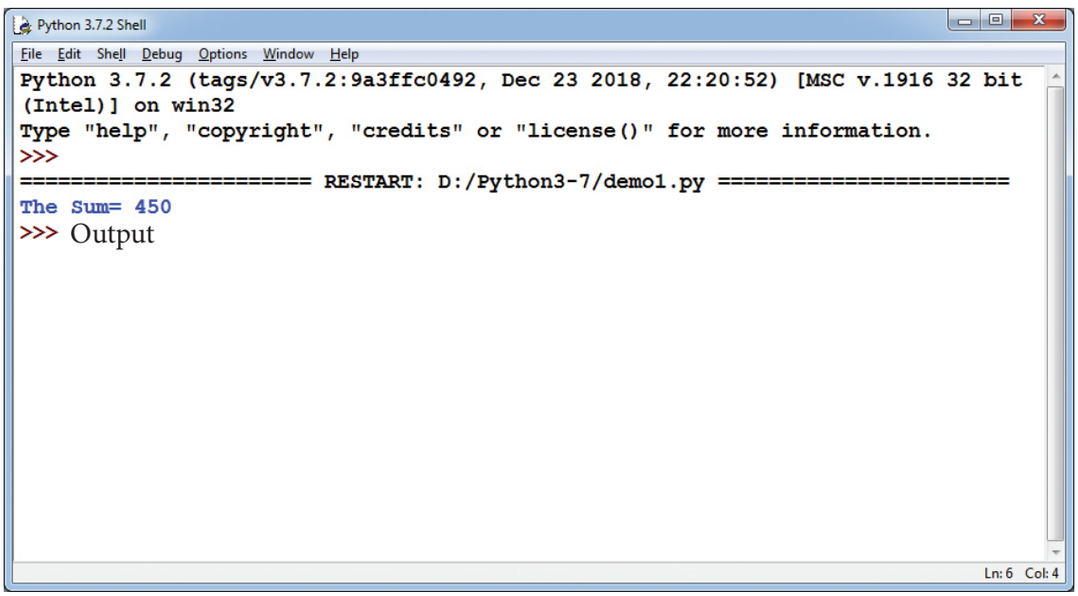
(1) **Run → Run Module** లేదా **F5** ను **Press** చేయడం



పటం 5.7 పైథాన్ స్క్రిప్ట్‌ను అమలుచేయడం

(2) కోడ్‌లో ఏదైన లోపాలు ఉంటే అవి IDLE స్క్రీన్‌లో ఎరుపు రంగులో ప్రదర్శించబడుతాయి. పైథాన్ లోపానికి గల కారణమును వివరిస్తుంది. లోపాలను సరిచేయడానికి స్క్రిప్ట్ ఎడిట్ (Script edit) లోకి వెళ్ళి లోపాలను సరిచేసి (**Ctrl + S**) లేదా (**File → Save**) ఆదేశమును ఉపయోగించండి. save చేసిన తరువాత మళ్ళి అమలు చేయండి.

(3) మీరు లోపం లేని పైథాన్ కోడ్‌ను అమలు చేసిన తరువాత దాని output ను పైథాన్ IDLE స్క్రీన్ 5.8లో ఉన్నట్లు కనిపిస్తుంది. (పటం *5.8*)



పటం 5.8 పైథాన్ స్క్రిప్ట్ అవుట్‌పుట్ విండో

12th_CS_Telugu_Medium_Chapter_5.indd 54 27-03-2023 10:38:22

5.4 Input మరియు Output Functions

వినియోగదారునికి కావలసిన పనిని నెరవేర్చడానికి ఒక program వినియోగదారునితో సంభాషించాల్సిన అవసరం ఉంది. ఇది **Input-Output functions** వినియోగిస్తుంది. **input()** (function) ఒక program ను అమలుచేస్తుంది డేటా Input చేసినపుడు తెరపై Output function **print** () పరిష్కారాలలో చూపిస్తుంది.

5.4.1 The print() function

పైథాన్‌లో program నడుపుతున్నపుడు డేటాను process చేయడానికి **print()** ప్రమేయము ఉపయోగపడుతుంది. print () functions యొక్క వాక్యనిర్మాణము క్రింద ఇవ్వబడినది.

ఉదాహరణ

```
print (“string to be displayed as output ” )
print (variable )
print (“String to be displayed as output ”, variable)
print (“String1 ”, variable, “String 2”, variable, “String 3” ……)
```

ఉదాహరణ

```
>>> print (“Welcome to Python Programming”)
        Welcome to Python Programming
>>> x = 5
>>> y = 6
>>> z = x + y
>>> print (z)
        11
>>> print (“The sum = ”, z)
        The sum = 11
>>> print (“The sum of ”, x, “ and ”, y, “ is ”, z)
        The sum of 5 and 6 is 11
```

తెరపై **print ( ) function** ప్రదర్శించడానికి ముందు ఫైల్ మూల్యాంకనము చేయబడుతుంది. print ( ) function ఇచ్చిన మొత్తం (statement) ని print () తో తెరపై ప్రదర్శించబడుతుంది. కమా **Comma (,)** ని ఉపయోగించి function లో ఒకటి కంటే ఎక్కువ అంశాలను వేరుచేయవచ్చు.

5.4.2 input() function

పైథాన్‌లో Programme ను నడుపుతున్నపుడు **input( )** function డేటా అందుకుంటుంది. **input( )** function యొక్క syntax

```
Variable = input (“prompt string”)
```


12th_CS_Telugu_Medium_Chapter_5.indd 55 27-03-2023 10:38:22

ఈ వాక్య నిర్మాణంలో **prompt string** అనేది యూజర్ ఇవ్వవలసిన input దాని అర్థమును తెలియజేసే ఒక ప్రకటన లేదా సందేశమును కలిగి ఉంటుంది.

input ( ) command prompt "prompt string" తెరపై కనిపిస్తుంది. ఇది వినియోగదారుని **input** పరికరం నుండి **data** ఊహించిన డేటాను అందించగలరు. ఇన్పుట్ () కీబోర్డ్ నుండి టైప్ చేసిన దాన్ని తీసుకుకొని, ఇచ్చిన వేరియబుల్‌లో ఎంటర్ చేసిన డేటాను నిల్వచేస్తుంది. **"prompt string"** ఇన్పుట్లో ఇవ్వకపోతే తెరపై సందేశం ప్రదర్శించబడదు. అందువల్ల ఇన్పుట్ టైప్ చేయవలసినది వినియోగదారునికి తెలియదు.

ఉదాహరణ 1:input( ) ప్రాంట్ స్ట్రింగ్‌తో

```
>>> city=input ("Enter Your City: ")
        Enter Your City: Madurai
>>> print ("I am from ", city)
        I am from Madurai
```

ఉదాహరణ 2:input( ) ప్రాంట్ స్ట్రింగ్ లేకుండా

```
>>> city=input()
        Rajarajan
>>> print ("I am from", city)
        I am from Rajarajan
```

పైన ఇవ్వబడిన ఉదాహరణ:2లో **input( )function** లో **prompt string** లేదు. అందువలన ఏమి చేర్చాలో వినియోగదారు అర్థము చేసుకోలేరు. (program string,) కు సంబంధము లేని డేటాను చేర్చడము వలన తప్పుడు పరిష్కారం వస్తుంది. కాబట్టి వినియోగదారు (program) ను యాక్సస్ చేయలేరు. సిస్టమ్‌ను అమలు చేయడానికి ప్రాంట్ స్ట్రింగ్ **input( )** ను అమలు చేయవలయును.

దత్తాంశము అన్నింటిని **strings** లేదా చిహ్నాలుగా ఆమోదించును సంఖ్యలను కాదు. వినియోగదారు సంఖ్యలను, సంఖ్యలుగా కాకుండా విలువలుగా నమోదు చేసినపుడు అవి స్పష్టంగా సంఖ్య డేటాగా మార్చాలి. **input ( ) function** డేటాను మిళితం చేస్తుంది. క్రింది విషయాలను స్పష్టంగా మార్చడానికి మరియు ఇతర సారూప్య విధులకు సహాయపడుతుంది.

ఉదాహరణ 3:

```
x = int (input("Enter Number 1: "))
y = int (input("Enter Number 2: "))
print ("The sum = ", x+y)
```

Output:

```
        Enter Number 1: 34
        Enter Number 2: 56
        The sum = 90
```


12th_CS_Telugu_Medium_Chapter_5.indd 56 27-03-2023 10:38:22

ఉదాహరణ:4 పై ప్రోగ్రామ్ కోసం ప్రత్యామ్నాయ పద్ధతి

```
x,y=int (input("Enter Number 1 :")),int(input("Enter Number 2:"))
print ("X = ",x," Y = ",y)
```

Output:

```
Enter Number 1 :30
Enter Number 2:50
X =  30  Y =  50
```

5.5 పైథాన్‌లో వ్యాఖ్యాలు

పైథాన్‌లో వ్యాఖ్యాలు హేష్ (#) గుర్తుతో ప్రారంభమవుతుంది. # గుర్తుతో ప్రారంభమగు పంక్తులు వాఖ్యలుగా పరిగణించబడి పైథాన్ interpreter చేత విస్మరించబడతాయి. వ్యాఖ్యలు ఒకే పంక్తి లేదా బహుళ పంక్తులుగా ఉండవచ్చును. బహుళ పంక్తులు క్రింది ఇవ్వబడిన """(triple quotes) మధ్యలో ఉండవలయును.

```
# It is Single line Comment
""" It is multiline comment
which contains more than one line """
```

5.6 ఇండెంటేషన్

ప్రోగ్రామింగ్ బ్లాక్‌లను నిర్వచించడానికి పైథాన్ **spaces** మరియు **tabs** వంటి వైట్‌స్పేస్‌ను ఉపయోగిస్తుంది. అయితే C, C++, java వంటి ఇతర Languageలు (curly braces) {} లను ఉపయోగిస్తాయి. class ఫంక్షన్ లేదా లూప్‌ల యొక్క బ్లాక్‌లను సూచించడానికి (block of selection command) ను వాడును. ఇండింటేషన్‌లోని వైట్ ల సంఖ్య spaces (Spaces and tabs) పరిష్కరించబడలేదు. కానీ బ్లాక్‌లోని అన్ని స్టేట్‌మెంట్‌లు ఒకే మొత్తంలో ఖాళీలతో ఇండెంట్ చేయాలి.

5.7 టోకెన్లు

పైథాన్ ప్రతి తార్కిక పంక్తిని టోకెన్స్ అని పిలువబడే ప్రాథమిక లెక్సికల్ భాగాల శ్రేణిగా విచ్ఛిన్నం చేస్తుంది. సాధారణ టోకెన్ రకాలు:

1) గుర్తింపులు,
2) కీఓడ్స్,
3) ఆపరేటర్లు,
4) డీలిమిటర్లు మరిము
5) లిటరల్స్

- టోకన్లు ఐడెన్టిఫైయర్స్ (లేదా) కీఓడ్స్ మధ్య వైట్ spaces విభజన అవసరము.

12th_CS_Telugu_Medium_Chapter_5.indd 57 27-03-2023 10:38:22

ఐడెన్‌టిఫైయర్స్ (Identifiers)

ఐడిన్‌ట్‌ఫైయర్ అంటె వేరియబుల్ ఫంక్షన్, క్లాస్, మాడ్యూల్ లేదా ఆబ్జెక్ట్‌ను గుర్తించడానికి ఉపయోగించే పేరు.

- ఐడింటిఫైయర్‌లు కచ్చితముగా వర్ణాలతో (A..Z or a..z) లేదా అండర్‌స్కోర్‌తో ప్రారంభించవలయును.
- ఐడింటిఫైయర్‌లలో అంకెలు (0 .. 9) ఉండవచ్చును.
- పైథాన్ ఐడింటిఫైయర్లు కేస్ సెన్సిటివ్ i.e. పెద్ద అక్షరాలు మరియు చిన్న అక్షరాలు విభిన్నంగా ఉంటాయి.
- ఐడింటిఫైయర్‌లు పైథాన్ కీఓర్డ్ కాకూడదు.
- ఐడింటిఫైయర్‌లు %,$, @ మొదలైన విరామ చిహ్నాలను పైథాన్ అనుమతించబడదు.

చెల్లుబాటు అయ్యే ఐడింటిఫైయర్ల ఉదాహరణలు:

Sum, total_marks, regno, num1

చెల్లు బాటు కాని ఐడెంటిఫైయర్ల ఉదాహరణలు:

12Name, name, total-mark, continue

5.7.2. కీవర్డ్స్ (Keywords)

కీవర్డ్స్ program నిర్మాణాన్ని గుర్తించడానికి పైథాన్ వ్యాఖ్యలలో ఉపయోగించే ప్రత్యేక పదాలు. ఈ పదాలకు వ్యాఖ్యాతకు నిర్ధిష్ట అర్థం ఉన్నందున వాటికి ఇతర ప్రయోజనాల కోసం ఉపయోగించలేరు.

పట్టిక 5.1 పైథాన్ కీవర్డ్స్

False	class	finally	is	return
none	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	None
break	except	in	raise	

5.7.3 ఆపరేటర్లు (Operators)

కంప్యూటర్ ప్రోగ్రామింగ్ Language లో ఆపరేటర్లు ప్రత్యేక చిహ్నాలు. ఇవి గణనలను షరతులతో కూడిన సరిపోలికలను సూచిస్తాయి. ఉపయోగించిన ఆపరేటర్ యొక్క విలువను ఒపెరాండ్స్ అంటారు. ఆపరేటర్లు అంకగణితం, రిలేషనల్, లాజికల్, అసైన్‌మెంట్ మొదలైనవిగా వర్గీకరించబడ్డాయి. ఆపరేటర్‌లో ఉపయోగించిన విలువను మరియు వేరియబుల్స్‌ను ఒపరాండ్స్ అంటారు.

12th_CS_Telugu_Medium_Chapter_5.indd 58 27-03-2023 10:38:22

(i) అంకగణిత ఆపరేటర్లు:

అంకగణిత ఆపరేటర్ ఒక గణిత ఆపరేటర్, ఇది రెండు ఒపెరాండ్లును తీసుకుంటుంది, మరియు వాటిపై గణన చేస్తుంది. ఒది సాధారణ అంక గణితంగా ఉపయోగించబడతాయి. చాలా కంప్యూటర్ భాషలలో వేర్వేరు రకాల వరుస గణనలను నిర్వహించడానికి సమీకరణలో ఉపయోగించ గల ఆపరేటర్ల సమితి ఉంటుంది.

పైథాన్ క్రింది అంకగణిత ఆపరేటర్లకు మద్దతు ఇస్తుంది.

ఆపరేటర్ - ఆపరేషన్స్	ఉదాహరణ	ఫలితము
a=100 and b=10. అయిన క్రింది వ్యవస్థీకరణను అంచనా వేయండి.		
+ (కూడిక)	>>> a + b	110
- (తీసివేత)	>>>a – b	90
* (గుణకారము)	>>> a*b	1000
/ (విభజన)	>>> a / b	10.0
% (మాడ్యులస్)	>>> a % 30	10
** (ఘాతాంకం)	>>> a ** 2	10000
// (నేల విభజన)	>>> a//30 (పూర్ణ సంఖ్య విభజన)	3

ప్రోగ్రామ్ 5.1 అంకగణితత ఆపరేటర్లను పరీక్షించుట

```
#Demo Program to test Arithmetic Operators
        a=100
        b=10
        print ("The Sum        = ",a+b)
        print ("The Difference = ",a-b)
        print ("The Product    = ",a*b)
        print ("The Quotient   = ",a/b)
        print ("The Remainder  = ",a%30)
        print ("The Exponent   = ",a**2)
        print ("The Floor Division =",a//30)
#Program End
```

Output:

```
        The Sum            = 110
        The Difference     = 90
        The Product        = 1000
        The Quotient       = 10.0
        The Remainder      = 10
        The Exponent       = 10000
        The Floor Division = 3
```


12th_CS_Telugu_Medium_Chapter_5.indd 59 27-03-2023 10:38:22

(ii) రిలేషనల్ మరియు తులనాత్మక ఆపరేటర్స్

రిలేషనల్ ఆపరేటర్‌ను తులనాత్మక ఆపరేటర్ అని కూడా పిలుస్తారు. ఇది రెండు ఒపెరాండ్ల మధ్య సంబంధాన్ని తనిఖీ చేస్తుంది. సంబంధం నిజమైతే అది **True** అని నిలుపుకుంటుంది. లేకపోతే **False** అని కలిగి ఉంటుంది.

పైథాన్ క్రింది రిలేషనల్ ఆపరేటర్స్‌కు మద్దతు ఇస్తుంది.

ఆపరేటర్ - ఆపరేషన్	ఉదాహరణ	ఫలితము
a=100 మరియు b=35. అనుకొనిన క్రింది వ్యవస్థీకరణను అంచనా వేయండి.		
== (సమానము)	>>> a==b	(false)
> (అంతకన్నా ఎక్కువ)	>>> a > b	(true)
< (అంతకన్నా తక్కువ)	>>> a < b	(false)
>= (అంతకన్నా ఎక్కువ లేదా తక్కువ)	>>> a >= b	(true)
<= (కన్నా తక్కువ లేక సమానము)	>>> a <= b	(false)
!= (సమానము కాదు)	>>> a != b	(true)

5.2 రిలేషనల్ ఆపరేటర్స్‌ను పరీక్షించుట

```
#Demo Program to test Relational Operators
    a=int (input("Enter a Value for A:"))
    b=int (input("Enter a Value for B:"))
    print ("A = ",a," and B = ",b)
    print ("The a==b = ",a==b)
    print ("The a > b = ",a>b)
    print ("The a < b = ",a<b)
    print ("The a >= b = ",a>=b)
    print ("The a <= b = ",a<=0)
    print ("The a != b = ",a!=b)
#Program End
```

అవుట్‌పుట్:

```
Enter a Value for A:35
Enter a Value for B:56
A =  35  and          B= 56
The a==b              = False
The a > b             = False
The a < b             = True
The a >= b            = False
The a <= b            = False
The a != b            = True
```


12th_CS_Telugu_Medium_Chapter_5.indd 60 27-03-2023 10:38:22

(iii) లాజికల్ ఆపరేటర్స్:

పైథాన్‌లో లాజికల్ ఆపరేటర్లు ఇచ్చిన రిలేషనల్ వ్యక్తీకరణపై తార్కిక కార్యకలాపాలను నిర్వహించడానికి ఉపయోగిస్తారు. తార్కిక ఆపరేటర్లు మూడు అవి and, or మరియు not

ఆపరేటర్	ఉదాహరణ	ఫలితము
a = 97 మరియు b = 35, క్రింది వ్యవస్థీకరణను అంచనా వేయండి		
or	>>> a>b or a==b	True
మరియు	>>> a>b and a==b	False
not	>>> not a>b	False i.e. Not True

ప్రోగ్రామ్ 5.3 తార్కిక ఆపరేటర్‌ను పరీక్షించుట:

ఉదాహరణ – కోడ్

```
#Demo Program to test Logical Operators
a=int (input("Enter a Value for A:"))
b=int (input("Enter a Value for B:"))
print ("A = ",a, " and b = ",b)
print ("The a > b or a == b  = ",a>b or a==b)
print ("The a > b and a == b = ",a>b and a==b)
print ("The not a > b       = ",not a>b)
#Program End
```

ఉదాహరణ – ఫలితము

```
Enter a Value for A:50
Enter a Value for B:40
A =  50  and b =  40
The a > b or a == b  =  True
The a > b and a == b =  False
The not a > b        =  False
```

(iv) అసైన్‌మెంట్ ఆపరేటర్స్ (Assignment operators)

పైథాన్లో = ఒక ధ్రువీకరణ పంక్షన్ ఇది వేరియబుల్స్‌కు విలువ ఇస్తుంది. **a** = 5 మరియు **b** = 10 అను కేటాయించబడిన **a** యొక్క విలువ 5 అని **b** విలువ 10 అని కలిగి ఉంది. ఈ రెండు విలువలను కేటాయించడంలో వేరే విధంగా **a,b=5,10** అని ఇవ్వవచ్చును. ఇది కుడివైపున ఉండు 5 మరియు 10 అను విలువలను **a,b** అను చలరాశుల విలువను తెలియపరచును. పైథాన్ +=, -=, *=, /=, %=, **= మరియు //= వంటి గుర్తులను అంగీకరిస్తుంది.

ఆపరేటర్స్ (Operator)	వివరణ (Description)	ఉదాహరణ (Example)
x=10 అయిన		
=	ఎడమవైపు వేరియబుల్‌కు కుడివైపు ఓపరాండ్స్‌ను కేటాయించును	>>> x=10 >>> b="Computer"
+=	ఎడమ ఆపరెండ్స్‌కు కూడికా చేసి తరువాత ఫలితాన్ని తిరిగి కేటాయిస్తుంది. అనగా i.e. x=30	>>> x+=20 # x=x+20
-=	ఎడమ ఆపరెండ్స్‌కు తీసివేత చేసి తరువాత ఫలితాన్ని తిరిగి కేటాయిస్తుంది. అనగా i.e. x=25	>>> x-=5 # x=x-5

12th_CS_Telugu_Medium_Chapter_5.indd 61 27-03-2023 10:38:22

*=	ఎడమ ఆపరెండ్స్‌కు గుణకారము చేసి తరువాత ఫలితాన్ని తిరిగి కేటాయిస్తుంది అనగా i.e. x=125	>>> x*=5 # x=x*5
/=	ఎడమ ఆపరెండ్స్‌కు మాడ్యులస్ చేసి ఫలితాన్ని తిరిగి కేటాయిస్తుంది. అనగా i.e. x=62.5	>>> x/=2 # x=x/2
%=	ఎడమ ఆపరెండ్స్‌కు ఘాతాంకము చేసి ఫలితాన్ని తిరిగి కేటాయిస్తుంది. అనగా i.e. x=2.5	>>> x%=3 # x=x%3
**=	ఎడమ ఆపరెండ్స్‌కు ఫ్లోర్ డివిజన్ చేసి ఫలితాన్ని తిరిగి కేటాయిస్తుంది అనగా i.e. x=6.25	>>> x**=2 # x=x**2
//=	ఎడమ ఆపరేటర్స్‌కు ఫ్లోర్ డివిజన్ చేసి ఫలితాన్ని తిరిగి కేటాయిస్తుంది. అనగా i.e. x=2.0	>>> x//=3

ప్రోగ్రామ్ 5.4 అసైన్‌మెంట్ ఆపరేటర్స్‌ను పరీక్షించుట

ప్రోగ్రామ్ కోడింగ్

```
#Demo Program to test Assignment Operators
x=int (input("Type a Value for X : "))
print ("X = ",x)
print ("The x is       =",x)
x+=20
print ("The x += 20 is  =",x)
x-=5
print ("The x -= 5 is   = ",x)
x*=5
print ("The x *= 5 is   = ",x)
x/=2
print ("The x /= 2 is   = ",x)
x%=3
print ("The x %= 3 is   = ",x)
x**=2
print ("The x **= 2 is   = ",x)
x//=3
print ("The x //= 3 is   = ",x)
#Program End
```

అవుట్‌పుట్

```
Type a Value for X : 10
X =  10
The x is       = 10
The x += 20 is  = 30
The x -= 5 is   =  25
The x *= 5 is   =  125
The x /= 2 is   =  62.5
The x %= 3 is   =  2.5
The x **= 2 is   =  6.25
The x //= 3 is   =  2.0
```

నిబంధిత ఆపరేటర్స్ (Conditional operator)

ట్రిపుల్ ఆపరేటర్స్ నిబంధనతో కూడిన ఆపరేటర్స్ అని కూడా అంటారు. ఇది సమీకరణలలోని నిబంధనలను true or false అని పరిక్షించి అమలుచేయును. ట్రిపుల్ ఫంక్షన్ బహుళ – పంక్తులు if-else స్టేట్మెంట్ కాకుండా ఒకే పంక్తిలో పరిస్థితులను పరీక్షించడానికి ట్రిపుల్ ఫంక్షన్ అనుమతిస్తుంది.

12th_CS_Telugu_Medium_Chapter_5.indd 62 27-03-2023 10:38:22

నిబంధిత ఆపరేటర్స్ యొక్క సింటాక్స్

Variable Name = [on_true] if [Test expression] else [on_false]

ఉదాహరణ:

```
min= 49 if 49<50 else 50 # min = 49
min= 50 if 49>50 else 49 # min = 49
```

ప్రోగ్రామ్ 5.5 నిబంధిత ఆపరేటర్స్‌ను పరీక్షించుట

```
# Program to demonstrate conditional operator
        a, b = 30, 20
# Copy value of a in min if a < b else copy b
        min = a if a < b else b
        print ("The Minimum of A and B is ",min)
# End of the Program
```

అవుట్‌పుట్:

```
The Minimum of A and B is  20
```

5.7.4 పరిమితులు (Delimiters):

సంకేతాలు మరియు సంకేతాల సమూహమును పైథాన్ పరిమితుల వ్యక్తీకరణ గాను జాబితా నిఘంటువు (strings) గా ఉపయోగించును. క్రింది కొన్ని పరిమితులు ఇవ్వబడినవి.

(	)	[	]	{	}
,	:	.	‘	=	;

5.7.5 లిటరల్స్ (Literals):

లిటరల్స్ అనునది వేరియబుల్ లేదా స్థిరాంకమునకు ఇచ్చిన ముడి దత్తాంశము పైథాన్‌లోని వివిధ రకాల లిటరల్స్ ఉన్నవి.

1) సంఖ్య (Numeric)
2) తీగలు (String)
3) బూలియన్ (Boolean)

(i) సంఖ్యా లిటరల్స్ (Numeric Literals):

సంఖ్యా లిటరల్స్ అనునదిఅంకెలను కలిగి ఉండును మరియు అని మార్పు లేనివి. సంఖ్యా లిటరల్స్ 3 వేరు వేరు అంకెలు అవి ఇంటీజర్, ఫ్లోట్, కాంప్లెక్స్.

12th_CS_Telugu_Medium_Chapter_5.indd 63 27-03-2023 10:38:22

ప్రోగ్రామ్ 5.6 సంఖ్య లిటరల్స్‌ను నిరూపించుట:

```
# Program to demonstrate Numeric Literals
        a = 0b1010                  #Binary Literals
        b = 100                     #Decimal Literal
        c = 0o310                   #Octal Literal
        d = 0x12c                   #Hexadecimal Literal
        print ("Integer Literals :",a,b,c,d)
        #Float Literal
        float_1 = 10.5
        float_2 = 1.5e2
        print ("Float Literals :",float_1,float_2)
        #Complex Literal
        x = 1 + 3.14 j
        print ("Complex Literals :", x)
        Print ("x = ", x , "Imaginary part of x = ", x.imag, "Real part of x = ", x.real)
#End of the Program
```

అవుట్‌పుట్:

```
Integer Literals : 10 100 200 300
Float Literals : 10.5 150.0
Complex Literals :
x = (1.3.14) Imaginary part of x = 3.14 Real part of x = 1.0
```

(ii) String లిటరల్స్ (String Literals)

పైథాన్‌లో string లిటరల్స్ అంటే కోట్స్‌తో చుట్టు ముట్ట బడిన అక్షరాల క్రమం పైథాన్ ఏక, ద్వి మరియు త్రి కోట్స్‌ను stringer సపోర్ట్ చేస్తుంది. string లిటరల్స్‌లో అక్షరాలు ఏక లేదా ద్వి కోట్స్‌లతో చుట్టుముట్టి ఉంటాయి. string లిటరల్స్‌లో విలువలను నిక్షిప్తించేయడానికి బహుళ పంక్తులను ఉపయోగిస్తారు.

ప్రోగ్రామ్ 5.7 తీగ లిటరల్స్‌ను పరీక్షించుట:

```
# Demo Program to test String Literals
        strings = "This is Python"
        char = "C"
        multiline_str = '''This is a multiline string with more than one line code.'''
        print (strings)
        print (char)
        print (multiline_str)
# End of the Program
```

అవుట్‌పుట్:

```
This is Python
C
This is a multiline string with more than one line code.
```


12th_CS_Telugu_Medium_Chapter_5.indd 64 27-03-2023 10:38:22

(iii) బూలియన్ లిటరల్స్ (Boolean Literals):

బూలియన్ లిటరల్స్‌లో True or False అను రెండు విలువలు ఉంటాయి.

ప్రోగ్రామ్ 5.8 బూలియన్ లిటరల్స్‌ను పరీక్షించుట:

```
# Demo Program to test String Literals
        boolean_1 = True
        boolean_2 = False
        print ("Demo Program for Boolean Literals")
        print ("Boolean Value1 :",boolean_1)
        print ("Boolean Value2 :",boolean_2)
# End of the Program
```

అవుట్‌పుట్:

```
        Demo Program for Boolean Literals
        Boolean Value1 : True
        Boolean Value2 : False
```

(iv) తప్పించుకొను వరుసలు (Escape Sequences):

పైథాన్ (strings) తప్పించుకొను వరుసలను బ్యాక్‌స్లాష్ ("\") అని కూడా పిలుస్తారు. ఇది కొన్ని వైట్ స్పేష్ అక్షరాలను సూచించడంలో ఉపయోగపడుతుంది. ("**\t**") ఒక ట్యాబ్ ("**\n**") ఒక కొత్త లైన్("**\r**") మరియు ఒక క్యారేజ్ రిటన్ (carriage returns) ఉదాహరణకు "ఇది వర్షం పడుతుంది" అనునది పైథాన్ ఆదేశం.

```
>>> print ("It\'s rainning")
It's rainning
```

పైథాన్ క్రింది తప్పించుకొను వరుసలకు మద్దతిస్తుంది.

తప్పించుకొను వరుస అక్షరాలు	వివరణ	ఉదాహరణ	ఫలితం
\\	బ్లాక్ స్లాష్	>>> print("\\test")	\test
\'	ఏక కోట్స్	>>> print("Doesn\'t")	Doesn't
\"	ద్వి కోట్స్	>>> print("\"Python\"")	"Python"
\n	కొత్త లైన్	print("Python","\n","Lang..")	Python Lang..
\t	ట్యాబ్	print("Python","\t","Lang..")	Python Lang..

12th_CS_Telugu_Medium_Chapter_5.indd 65 27-03-2023 10:38:22

5.8 పైథాన్ డేటా టైపులు

పైథాన్‌లోని అన్ని డేటా విలువలు ఆబ్జెక్ట్ (object) మరియు (else) ఆబ్జెక్ట్ లేదా విలువ రకాన్ని కలిగి ఉంటాయి. పైథాన్ (Built-in) లేదా ప్రాథమిక డేటా టైపు కొన్ని సంఖ్య (String) బూలియన్, టపుల్స్ List మరియు (dictionaries) నిఘంటువులు మొదలగునవి.

5.8.1 సంఖ్యా డేటా రకములు (Number Data type):

పైథాన్‌లో అంతర్నిర్మిత సంఖ్యా వస్తువులు, పూర్ణాంకాలు, ఫ్లోటింగ్ పాయింట్ సంఖ్యలు మరియు సంక్లిష్ట సంఖ్యలకు మద్దతు ఇస్తుంది.

పూర్ణాంక డేటా దశాంశ, (decimal) ఆక్టా (octal) లేదా హెక్సాడెసిమల్ (hexadecimal) కావచ్చు. ఆక్టా అంకెలను సూచించడానికి ఆక్టల్ పూర్ణాంకం 0 (సున్న) అంకె తరువాత 0 అక్షరం మరియు హెక్సాడెసిమల్ సూచించడానికి ('**0X**') మరియు (**L**) ను ఉపయోగిస్తారు.

ఉదాహరణకు:

```
102, 4567, 567          # Decimal integers
0o102, 0o876, 0o432     # Octal integers
0X102, 0X876, 0X432     # Hexadecimal integers
34L, 523L               # Long decimal integers
```

ఫ్లోటింగ్ పాయింట్ డేటా వీటిని కలిగి ఉన్న దశాంశ అంకెల క్రమం ద్వారా ఒక ఘాతాంక డేటా దశాంశ అంకె భాగం దశాంశ బిందువు ఘాతాంక భాగం కలిగి ఉంటుంది. అంతకంటే ఒకటి లేదా అంతకన్నా ఎక్కువ అంకెల తరువాత

ఉదాహరణకు:

```
123.34,  456.23,  156.23      # Floating point data
12.E04, 24.e04                # Exponent data
```

మిశ్రమ సంఖ్యలు రెండు ఫ్లోటింగ్ విలువలతో రూపొందించబడినవి. ప్రతి ఒక్కటి నిజమైనవి మరియు ఊహాత్మక భాగాలు.

5.8.2 బూలియన్ డేటా టైపు (Boolean Data type):

బూలియన్ డేటా (True or False) అను రెండు విలువలలో ఏదైన ఒక దానిని కలిగియుండును.

ఉదాహరణ:

```
Bool_var1=True
Bool_var2=False
```


12th_CS_Telugu_Medium_Chapter_5.indd 66 27-03-2023 10:38:22

5.8.3 String డేటా టైపు

String డేటాను సింగిల్ కోట్ లేదా డబుల్ కోట్ లేదా ట్రిపుల్ కోట్ జత చేయబడును.

ఉదాహరణకు:

```
Char_data = 'A'
    String_data= "Computer Science"
    Multiline_data= """String data can be enclosed in single quotes or
    double quotes or triple quotes."""
```

గుర్తించుకోవలసిన విషయాలు

- గైడోవాన్ రోసుమ్ చేత సృష్టించబడిన సాధారణ ప్రోగ్రామింగ్ (language) పైథాన్
- పైథాన్ షెల్‌ను ఇంటరాక్టివ్ మోడ్ మరియు స్క్రిప్ట్ మోడ్ అను రెండు విధాలుగా ఉపయోగింతురు.
- ప్రోగ్రామ్ బ్లాక్స్‌ను నిర్వచించడానికి పైథాన్ వైట్‌స్పేస్‌ను ఉపయోగిస్తారు.
- వినియోగదారునికి కావలసిన పనిని నెరవేర్చడానికి ఒక ప్రోగ్రామ్ తుది వినియోగదారునితో సంభాషించవలసిన అవసరం ఉంది. ఇది జరుగుటకు ఇన్‌పుట్ – అవుట్‌పుట్ సౌకర్యంను ఉపయోగిస్తారు.
- పైథాన్ ప్రతి తార్కిక రేఖను ఎలిమెంట్‌లో లెక్సికల్ భాగాల శ్రేణిగా విచ్ఛినం చేస్తుంది. దీనినే టోకన్స్ అందురు.
- ప్రోగ్రామ్ నిర్మాణాలను గుర్తించడానికి పైథాన్ interpreter కొన్ని ప్రత్యేక కీ వోర్డ్స్ (key work) ను ఉపయోగించెను.

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. పైథాన్‌ను అభివృద్ధి పరిచినది ఎవరు?
   అ) రిచ్చే ఆ) గైడోవాన్ రోసుమ్ ఇ) బిల్‌గేట్స్ ఈ) సుందర్ పిచై
2. పైథాన్ ప్రాంప్ట్ ఇంటర్పెటర్ సూచనలను Accept చేయడానికి సిద్ధంగా ఉందని సూచించునది.
   అ) >>> ఆ) <<< ఇ) # ఈ) <<
3. పైథాన్ ప్రోగ్రామ్‌ను క్రియోట్ చేయడానికి క్రింది వాటిలో ఏది సత్వర మార్గంలో ఉపయోగపడుతుంది.
   అ) Ctrl + C ఆ) Ctrl + F ఇ) Ctrl + B ఈ) Ctrl + N
4. పైథాన్ ప్రోగ్రామ్‌లో వాక్యాలు ఇవ్వబడిన క్రింది వాటిలో ఏది ఉపయోగపడును.
   అ) # ఆ) & ఇ) @ ఈ) $

12th_CS_Telugu_Medium_Chapter_5.indd 67 27-03-2023 10:38:22

5. ఈ గుర్తు ఒకటి కంటే ఎక్కువ అంశాలను ఒకే పంక్తితో ముద్రించడానికి ఉపయోగపడుతుంది.

అ) Semicolon (;) ఆ) Dollor ($) ఇ) comma (,) ఈ) Colon (:)

6. క్రింది వానిలో టోకన్ కానిది ఏది?

అ) వ్యాఖ్యాత ఆ) ఐడెంటిఫైర్స్ ఇ) కీవర్డ్ ఈ) ఆపరేటర్లు

7. పైథాన్‌లో ఈ క్రింది వాటిలో ఏది కీవోడ్ కాదు.

అ) విరామం (break) ఆ) అయితే (while)

ఇ) కొనసాగింపు (continue) ఈ) ఆపరేటర్స్ (operators)

8. ఏ ఆపరేటర్‌ను కంపారిటివ్ ఆపరేటివ్ అని పిలుస్తారు.

అ) అర్థిమెటిక్ ఆ) రిలేషనల్ ఇ) లాజికల్ ఈ) అసైన్‌మెంట్

9. క్రింది వాటిలో లాజికల్ ఆపరేటర్ కానిది?

అ) and ఆ) or ఇ) not ఈ) like

10. ఏ ఆపరేటర్‌ను షరతులతో కూడిన ఆపరేటర్ అని కూడ పిలుస్తారు?

అ) టెర్నరీ ఆ) రిలేషనల్ ఇ) లాజికల్ ఈ) అసైన్‌మెంట్

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. పైథాన్ ప్రోగ్రామ్‌ను పరీక్షించడానికి ఉపయోగించు వివిధ పద్ధతులు ఏవి?
2. టోకన్స్‌పై చిన్న గమనికను వ్రాయండి.
3. పైథాన్‌లో ఉపయోగించు వివిధ ఆపరేటర్లు ఏవి?
4. లిటరల్ అనగానేమి? వాటి రకాలును వివరించండి?
5. ఘాతాంక డేటా పైన చిన్న గమనికను వ్రాయండి.

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. అంకగణిత ఆపరేటర్లపై చిన్న గమనికను ఉదాహరణ సహాయంతో వ్రాయండి.
2. పైథాన్‌లో ఉపయోగించు అసైన్‌మెంట్ ఆపరేటర్లు ఏవి?
3. టెర్మినరీ ఆపరేటర్లను ఉదాహరణతో వివరించండి.
4. Escape sequences పై చిన్న గమనికలను ఉదాహరణతో వ్రాయండి.
5. String లిటరల్స్ అనగానేమి? వివరింపుము?

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. స్క్రిప్ట్ మోడ్ ప్రోగ్రామ్ గురించి విపులంగా వివరణ ఇవ్వండి.
2. input() మరియు print() functions ఉదాహరణతో వివరించండి.
3. పైథాన్‌లోని టోకన్స్ గురించి విపులంగా వివరించండి.

12th_CS_Telugu_Medium_Chapter_5.indd 68 27-03-2023 10:38:22

యూనిట్ II

అధ్యాయం 6

కంట్రోల్ స్ట్రక్షర్



అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును చదివిన పిమ్మట విద్యార్థులు తెలుసుకొను అంశాలు.

- పైథాన్ language వివిధ రకాల ఫ్లో ఆఫ్ కంట్రోల్స్ గూర్చి తెలుసుకోవడం.
- షరతులలో కూడిన నిర్మాణమును ఎలా ఉపయోగించాలో syntax ద్వారా తెలుసుకోవడాన్ని ప్రోగ్రామ్ ఫ్లో ద్వారా తెలుసుకోవడం.
- ప్రోగ్రామ్ విభాగాన్ని పునరావృతం చేయడానికి కోడ్‌ను అభివృద్ధి చేయడానికి మరల నిర్మాణాలను వర్తింపచేయడం లేదా పరిస్థితి సంతృప్తి చెందే వరకు కొన్ని సార్లు ప్రత్యేక నిర్మాణమును అమలుచేయడం.

6.1 పరిచయం

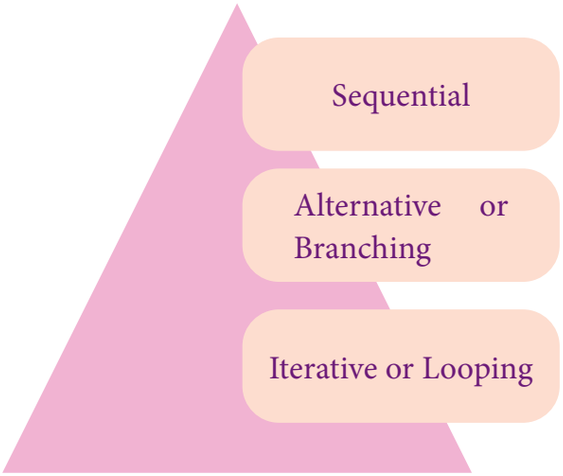
ప్రోగ్రామ్‌లలో నివేదికల సమితి ఉండవచ్చు. ఈ statements ఎగ్జిక్యూటబుల్‌కు విభాగాల ఫలితాన్ని అందిస్తుంది. సాధారణంగా statements వరుసగా ఒకదాని తరువాత ఒకటి అమలు చేయబడుతాయి. కొన్ని సందర్భాలలో మన నిజ జీవిత ప్రోగ్రామ్‌లు ఉండవచ్చు. ప్రోగ్రామ్ యొక్క కొంత భాగాన్ని లేదా statements సమితిని నెరవేర్చకుండా వదిలివేస్తుంది. ప్రోగ్రామ్ యొక్క మరొక భాగాన్ని షరతులలో ప్రత్యామ్నాయ లేదా బ్రౌంచింగ్ అని పిలుస్తారు. అలాగే statement సమితిని పలుమార్లు లేదా మరల అమలచేయవలసి వస్తుంది. దీనిని ఇటిరేషన్ (***iteration***) లేదా లూపింగ్ (***looping***) అందురు. ఈ అధ్యాయములో మనం పైథాన్‌లోని వివిధ కంట్రోల్ స్ట్రక్షర్‌ల పైన దృష్టిపెట్టి ప్రోగ్రామ్‌ను ఎలా అభివృద్ధి చేయాలో తెలుసుకోవాలి.

6.2 నియంత్రణ నిర్మాణాలు

నియంత్రణ నిర్మాణాలు నియంత్రణ ప్రోగ్రామ్ యొక్క ఒక భాగం నుండి మరొక భాగమునకు మారడానికి గల కారణాలను నియంత్రణ నిర్మాణాలు లేదా నియంత్రణ statements అని అంటారు. ఇప్పటికే నేర్చుకొన్నట్లుగా C++ ఈ నియంత్రణ statements మిశ్రమ statements ఆపరేషన్ మోడ్‌ను బట్టి ప్రక్రియ యొక్క నియంత్రణ ప్రవాహం లేదా ప్రోగ్రామ్ నియంత్రణ ప్రవాహంగా మారుతుంది.

12th_CS_Telugu_Medium_Chapter_6 .indd 69 27-03-2023 11:08:33

పైథాన్‌లో మూడు ప్రధాన రకాల నియంత్రణ నిర్మాణాలు ఉంటాయి.



6.2.1 నివేదికల క్రమం వరుసలు (Sequential Statement):

నివేదికలు క్రమంగా ఒకదాని తరువాత ఒకటిగా ఉంటుంది. ఉదాహరణ మీ పేరు చిరునామా మరియు ఫోన్ నంబర్‌ను ముద్రించినట్లు నివేదికల క్రమం ఉంటుంది.

ఉదాహరణ 6.1

```
# నీ పేరు మరియు చిరునామాను ముద్రించడం ప్రోగ్రామ్‌కు ఉదాహరణ
print ("Hello! This is Shyam")
print ("43, Second Lane, North Car Street, TN")
```

అవుట్‌పుట్

```
Hello! This is Shyam
43, Second Lane, North Car Street, TN
```

6.2.2 ప్రత్యామ్నాయ లేదా బ్రాంచింగ్ స్టేట్‌మెంట్ (Alternative or Branching Statement):

మన దైనందిన జీవితంలో లక్ష్యాలను సాధించడానికి వివిధ నిర్ణయాలు తీసుకోవాలి. మన లక్ష్యాలను సాధించడానికి ప్రత్యామ్నాయ మార్గమును ఎంచుకొంటాము. మనం రోజు వెళ్లె మార్గము అవరోదం ఏర్పడితే మనం వెల్లవలసిన ప్రదేశమునకు వెల్లడానికి ప్రత్యామ్నాయ మార్గమును ఎంచుకుంటాము. అలాగే ఈ నిర్ణయాన్ని ప్రత్యామ్నాయ లేదా శాఖ విభజన ద్వారా నేర్చుకోబోతున్నాము. ఇచ్చిన సంఖ్య సానుకూల సంఖ్య లేదా ప్రతికూల సంఖ్య లేదా భేసి సంఖ్య లేదా సరి సంఖ్య అని నిర్ణయించడానికి ప్రత్యామ్నాయ లేదా బ్రాంచింగ్ ఉపయోగించబడుతుంది.

పైథాన్ క్రింది ప్రత్యామ్నాయ లేదా బ్రాంచింగ్‌ను కలిగి ఉంటుంది.

- **Simple if statement**
- **if..else statement**
- **if..elif statement**

12th_CS_Telugu_Medium_Chapter_6 .indd 70 27-03-2023 11:08:33

(i) Simple if statement

Simple if statement అనునది అన్నిటికంటే సరళమైన నిర్ణయాత్మక స్టేట్‌మెంట్. నిబంధన యొక్క రూపము రిలేషనల్ లేదా లాజికల్ వ్యక్తీకరణగా ఉండాలి.

Syntax:

if <condition>:
statements-block1

పై వాక్య నిర్మాణంలో నిబంధన నిజం అయితే సరి స్టేట్‌మెంట్ బ్లాక్ – 1 అమలు అగును.

ఉదాహరణ 6.2

```
# వయస్సును తనిఖీ చేసి ఓటింగ్‌కు అర్హత ఉందా లేదా  ముద్రించు ప్రోగ్రామ్
x=int (input("Enter your age :"))
if x>=18:
        print ("You are eligible for voting")
```

అవుట్‌పుట్ 1:

```
Enter your age :34
You are eligible for voting
```

అవుట్‌పుట్ 2:

```
Enter your age :16
>>>
```

రెండవ అవుట్‌పుట్ మీరు చూడగలిగినట్లుగా అవుట్‌పుట్ ముద్రించబడదు. కేవలం పైథాన్ ప్రాంప్ట్ మాత్రమే ప్రదర్శించబడుతుంది. ఎందుకనగా నిబంధన విఫలమైనందున ప్రోగ్రామ్ ప్రత్యామ్నాయ ప్రక్రియను తనిఖీ చేయదు.

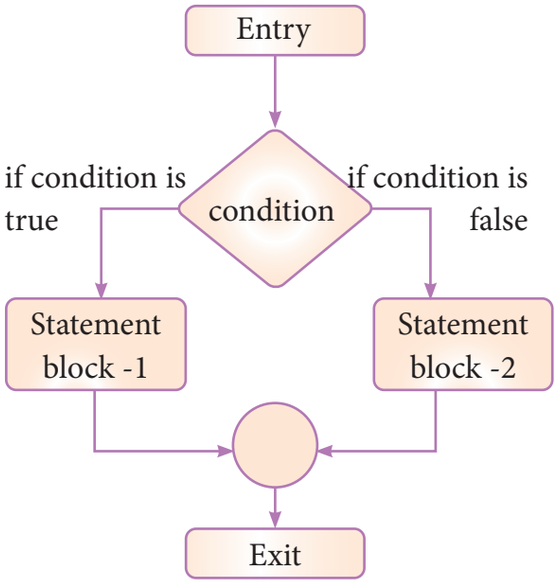
(ii) if..else స్టేట్‌మెంట్స్ (if..else statement)

'if..else' స్టేట్‌మెంట్స్ నిజమైన బ్లాక్స్‌తో పాటు తప్పుడు బ్లాక్‌ను తనిఖీ చేయడానికి నియంత్రణను అందిస్తుంది. (**'if..else'**) స్టేట్‌మెంట్ యొక్క వాక్య నిర్మాణం క్రింద ఇవ్వబడినది.

Syntax:

if <condition>:
statements-block 1
else:
statements-block 2

12th_CS_Telugu_Medium_Chapter_6 .indd 71 27-03-2023 11:08:33



పటం. 6.1 if..else స్టేట్‌మెంట్‌ను అమలుపరుచుట

(if..else) స్టేట్‌మెంట్ రెండు అవకాశాలను అందిస్తుంది మరియు నిబంధన అనునది ఏ బ్లాక్ అమలు కావాలో నిర్ణయిస్తుంది.

ఉదాహరణ 6.3 అంగీకరించిన సంఖ్య భేసి లేదా సరి సంఖ్య అని నిర్ణయించు ప్రోగ్రామ్

```
a = int(input("Enter any number :"))
if a%2==0:
        print (a, " is an even number")
else:
        print (a, " is an odd number")
```

అవుట్‌పుట్ 1:

```
Enter any number :56
56  is an even number
```

అవుట్‌పుట్ 2:

```
Enter any number :67
67  is an odd number
```

పై ప్రోగ్రామ్‌ను తిరిగి రాయడానికి ప్రత్యామ్నాయ పద్ధతి పైథాన్‌లో కూడా అందుబాటులో ఉంది. ఈ పూర్తి **(if..else)** ను ఇలా రాయవచ్చును.

Syntax:

variable = variable1 if condition else variable 2

12th_CS_Telugu_Medium_Chapter_6 .indd 72 27-03-2023 11:08:33

గమనిక

(if) లో పేర్కొన్న నిబంధన తనిఖీ చేయబడితే అది సరి అయితే వేరియబుల్ 1 లోని విలువ అసైన్‌మెంట్ యొక్క ఎడమవైపు ఉన్న వేరియబుల్‌లో నిల్వచేయబడుతుంది. లేకపోతే వేరియబుల్ 2లోని విలువను తీసుకోబడుతుంది.

ఉదాహరణ 6.4: అంగీకరించబడిన సంఖ్య భేసి సంఖ్య లేదా సరి సంఖ్య (odd or even) అని తనిఖీ చేయుట ప్రోగ్రామ్ (if...else యొక్క ప్రత్యామ్నాయ పద్ధతిని ఉపయోగించి)

```
a = int (input("Enter any number :"))
x="even" if a%2==0 else "odd"
print (a, " is ",x)
```

అవుట్‌పుట్ 1:

```
Enter any number :3
3 is odd
```

అవుట్‌పుట్ 2:

```
Enter any number :22
22 is even
```

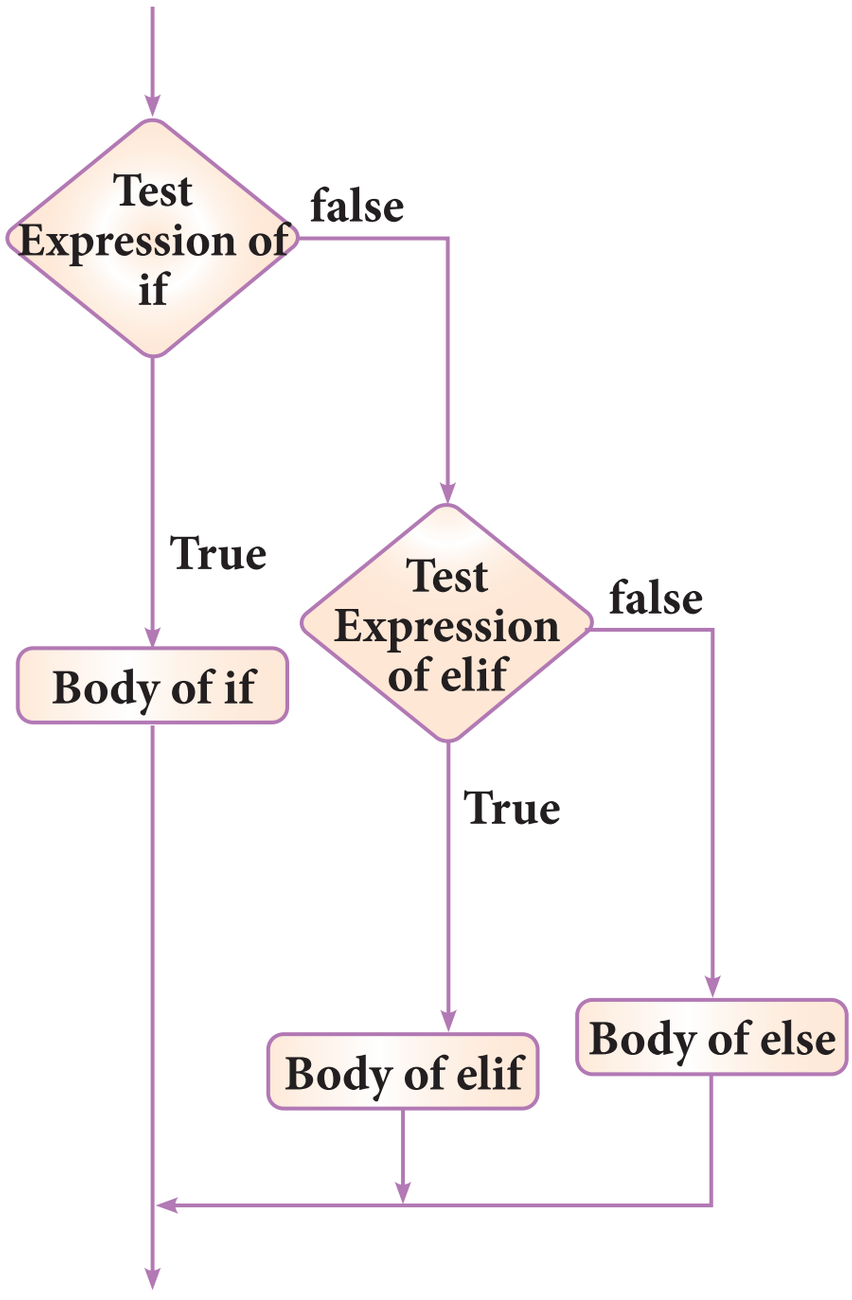
(iii) నెస్టెడ్ if..elif...else స్టేట్‌మెంట్ (Nested if..elif...else statement):

'if' స్టేట్‌మెంట్ గొలుసును నిర్మించవలసిన అవసరం వచ్చినపుడు **'else'** కు బదులుగా **'elif'** ను ఉపయోగించవచ్చు.

Syntax:

```
if <condition-1>:
    statements-block 1
elif <condition-2>:
    statements-block 2
else:
    statements-block n
```

పైన పేర్కొన్న **'if..elif..else'** యొక్క syntax condition-1 నిజమైతే పరిక్షించబడుతుంది. అపుడు statements-block1 అమలు అవుతుంది లేకపోతే కంట్రోల్ condition-2 తనిఖీ చేస్తుంది. నిజమైతే statements-block2 అమలు అవుతుంది. అన్ని statements విపలమైనప్పటికి statements-block n else భాగము అమలవుతుంది.



పటం 6.2 if..elif..else statement అమలుపరుచుట

elif భాగము **if..else-if..else** స్టేట్‌మెంట్‌లను ఒకే **if..elif…else. 'elif'** బందించును. **elif** యొక్క పొడగింపు **else if' if** స్టేట్‌మెంట్‌లో ఉపయోగించిన **elif** ప్రాంతానికి పరిమితం కాదు. కాని **else** భాగంను ఉపయోగిస్తున్నప్పుడు స్టేట్‌మెంట్‌ను చివరిలో ఉపయోగించాలి.

గమనిక

పైథాన్‌లోని స్టేట్‌మెంట్ **if..elif..else** (nested) C++ ని if స్టేట్‌మెంట్‌కు సమానము.

12th_CS_Telugu_Medium_Chapter_6 .indd 74 27-03-2023 11:08:33

ఉదాహరణ 6.5: నెస్టెడ్ (if) స్టేట్‌మెంట్ ప్రోగ్రామ్‌ను ఊహించుట.

సరాసరి	గ్రేడ్
>=80 and above	*A*
>=70 and <80	*B*
>=60 and <70	*C*
>=50 and <60	D
Otherwise	E

```
m1=int (input("Enter mark in first subject : "))
m2=int (input("Enter mark in second subject : "))
avg= (m1+m2)/2
if avg>=80:
        print ("Grade : A")
elif avg>=70 and avg<80:
        print ("Grade : B")
elif avg>=60 and avg<70:
        print ("Grade : C")
elif avg>=50 and avg<60:
        print ("Grade : D")
else:
        print ("Grade : E")
```

అవుట్‌పుట్ 1:

```
Enter mark in first subject : 34
Enter mark in second subject : 78
Grade : D
```

అవుట్‌పుట్ 2 :

```
Enter mark in first subject : 67
Enter mark in second subject : 73
Grade : B
```

గమనిక

if స్టేట్‌మెంట్ పై ఉదాహరణలోని కోడ్ యొక్క రెండు బ్లాక్స్ రెండూ ఇండెంట్ నాలుగు ఖాళీలు. ఇది పైథాన్ కోసం ఇండెంటెషన్ యొక్క సాధారణ మొత్తం. చాలా ఇతర ప్రోగ్రామింగ్ languages ఇండెంటెషన్ కోడ్‌ను అందంగా కనిపించడానికి మాత్రమే ఉపయోగించబడుతుంది. కాని పైథాన్‌లో అది ఏ స్టేట్‌మెంట్ యొక్క కోడ్ బ్లాక్‌కు చెందినదో సూచించడానికి అవసరం.

ఉదాహరణ6.5a: #'in' మరియు 'not in' if statement ఉపయోగించు ప్రోగ్రామ్

```
ch=input ("Enter a character :")
# to check if the letter is vowel
if ch in ('a', 'A', 'e', 'E', 'i', 'I', 'o', 'O', 'u', 'U'):
        print (ch,' is a vowel')
# to check if the letter typed is not 'a' or 'b' or 'c'
if ch not in ('a', 'b', 'c'):
        print (ch,'  the letter is not a/b/c')
```

అవుట్‌పుట్ 1:

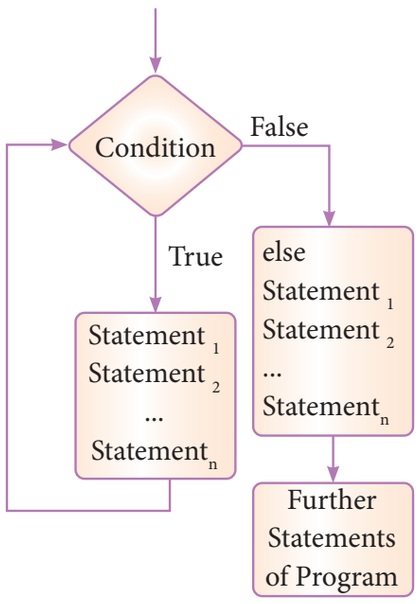
```
Enter a character :e
e is a vowel
```

అవుట్‌పుట్ 2:

```
Enter a character :x
x the letter is not a/b/c
```

6.2.3. పునరావృతం లేదా లూపింగ్ నిర్మాణాలు (Iteration or Looping constructs):

వినియోగదారు యొక్క బ్లాక్‌ను అమలుచేయాల్సిన పరిస్థితి వచ్చినపుడు పునరావృత లేదా లూపింగ్ నిర్మాణాలు ఉపయోగపడును. అనేక సార్లు లేదా నిబంధన తృప్తి చెందే వరకు లూపింగ్ స్టేట్‌మెంట్‌ను అమలు చేయడానికి అనుమతిస్తుంది.



పటం 6.3 పునరావృతం లేదా లూపింగ్ నిర్మాణాలు అమలు పరుచు విధం

పైథాన్ రెండు రకాలు లూపింగ్ నిర్మాణాలను ప్రతిపాదించుట:

- while లూప్
- for లూప్

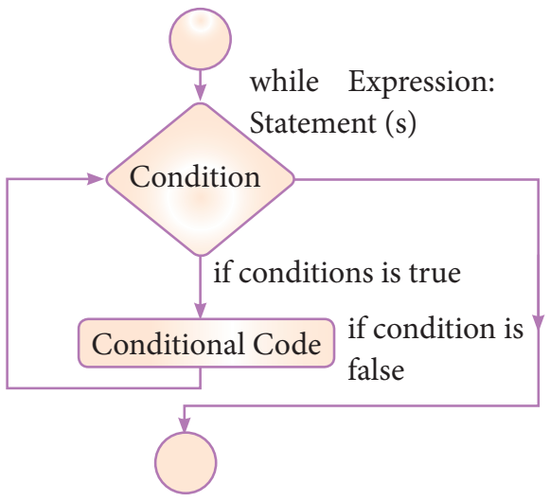
12th_CS_Telugu_Medium_Chapter_6 .indd 76 27-03-2023 11:08:34

(i) while లూప్

పైథాన్‌లో **while** లూప్‌కు **syntax** క్రింద ఇవ్వబడినది

Syntax:

while <condition>:
    statements block 1
[else:
    statements block2]



పటం 6.4 while లూపింగ్ అమలు

while లూప్‌లో నిబంధన ఏదైన చెల్లు బాటు అయ్యే బూలియన్ వ్యక్తీకరణలో True or False అని తిరిగి వస్తుంది. while of **else** part లో అనునది ఐచ్ఛిక భాగం నిబంధన True అయితే **statements block1** అమలు అవుతుంది. అమలుచేయబడు నిబంధన False అయితే **else** భాగం వ్రాయబడుతుంది. **while** లూప్ అనునది ఎంట్రీ check లూప్ రకానికి చెందినది అనుకొనిన నిబంధన విఫలమైనపుడు (తప్పు అయినపుడు) అది ఒక సారి కూడా అమలు చేయుటకు వీలుకాదు.

ఉదాహరణ 6.6: while లూప్ ఉపయోగించి సంఖ్యలు 10 నుండి 15 వరకు ముద్రించుట ప్రోగ్రామ్

```
i=10                        # intializing part of the control variable
while (i<=15):              # test condition
    print (i,end='\t')      # statements - block 1
    i=i+1                   # Updation of the control variable
```

అవుట్‌పుట్:

```
10  11  12  13  14  15
```


12th_CS_Telugu_Medium_Chapter_6 .indd 77 27-03-2023 11:08:34

గమనిక

నియంత్రణ వేరియబుల్ **(i)** అయిన దాని ప్రారంభ విలువ (10) **(i<=15,)** నిబంధన పరీక్షించినపుడు సరియైన విలువ ముద్రించబడుతుంది. తరువాత నియంత్రణ వేరియబుల్ నవీనకరించి **(i=i+1)** అవుతుంది. ఎప్పుడు అయితే నిబంధన విఫలం అయితే **(i)** అనునది 16 అవుతుంది. మరియు లూప్ నుండి వైతొలుగుతుంది.

గమనిక

print end మరియు sep అనే పారామితిని కలిగియుండును. '\t' ను ట్యాబ్ కొరకు '\n' ను క్రొత్త పంక్తి కొరకు అను తప్పించుకొను వరుసలను ఇవ్వడానికి end పారామితిని ఉపయోగింతురు. ప్రత్యేక చిహ్నాలు అయిన comma గుర్తించడానికి semicolon గుర్తించడానికి అను పారామితిని ఉపయోగింతురు. (పారామితులను గురించి మునుపటి అధ్యాయంలో మీరు నేర్చుకొన్న భావనను గుర్తుచేసుకోండి).

లూప్‌లో వేరే భాగాన్ని else పార్ట్ గుర్తించడానికి ప్రోగ్రామ్:

ఉదాహరణ 6.7: while లూప్‌లో else పార్ట్

```
i=10                                          # intializing part of the control variable
while (i<=15):                                # test condition
      print (i,end='\t')                      # statements - block 1
      i=i+1                                   # Updation of the control variable
else:
    print ("\nValue of i when the loop exit ",i)
```

అవుట్‌పుట్: 1

```
10   11   12   13   14   15
Value of i when the loop exit  16
```

(ii) for లూప్:

లూప్ for లూప్‌ను ఖచ్చితమైన లూప్‌గా పరిగణిస్తారు. ఎందుకంటే ఇందులో ఇందు ఎన్నిసార్లు లూప్‌ను ఉపయోగించాడో ప్రోగ్రామర్ తెలుసుకోవచ్చు.

Syntax:

```
for counter_variable in sequence:
      statements-block 1
[else:                        # optional block
      statements-block 2]
```

పైథాన్ నందు శ్రేణిలోని అంశాల పునరావృతం (Iterate) కొరకు నివేదికలో 'లూపింగ్' నివేదిక ఉపయోగించినపుడు ఇది శ్రేణిలోని ప్రతి అంశంను లెక్కలుకి తీసుకొంటుంది. ఫలితంలో శ్రేణిలో మనం కోరిన లేదా కోరని విలువలుండవచ్చు.

12th_CS_Telugu_Medium_Chapter_6 .indd 78 27-03-2023 11:08:34

ఇవి ప్రతి పునరావృతంలో (iteration). నియాంత్రిత అంశాలు (Conteoled variable) శ్రేణిలోని (Sequence) చివరి అంశం చేరే వరకు యాక్సిస్ చేస్తుంది.

ఉదాహరణ 6.8 (అ)

```
For x in                "హలు అనే పదం"

Print
(x, End =)
Out put

"హలు అనే పదం"
```

ఉదాహరణ 6.8 (ఆ)

```
For x in (1,2,3,4,5):
Print "హలు అనే పదం"
        "హలు అనే పదం"
        "హలు అనే పదం"
        "హలు అనే పదం"
        "హలు అనే పదం"
```

పై ఉదాహరణలో 6.8(అ) మనం శ్రేణిగా 'హలో అనే పదం' ను string గా సృష్టించడం జరిగింది.

ప్రారంభంలో x అనే విలువను string లో మొదటి అంశం element గా set చేయడం జరిగింది. తర్వాత నియాత్రిత అంశం x ని string లు తర్వాతి అంశంగా update చేసి మరలా నివేదికను రూపొందించడం జరుగుతుంది. ఈ విధంగా string చివరి అంశం యాక్సిస్ అయ్యేవరకు లూప్ loop run అవుతుంది.

ఇదేవిధంగా ఉదాహరణ 6.8(ఆ) లో ఇచ్చిన శ్రేణిలోని చివరి అంశం element ను చేరే వరకు ఐదుసార్లు "హలో అనే పదం " ను string print చేస్తుంది.

విలువల శ్రేణిని చేతిలో వ్రాయడానికి బదులుగా మనం rant () ఉపయోగించవచ్చు.

రెండు అంకెల మధ్య విలువల శ్రేణిని generate చెయ్యడానికి range () ని built చేస్తారు.

range() యొక్క syntaxక్రింది విధంగా ఉంటుంది.

range (start,stop,[step])

ఇక్కడ

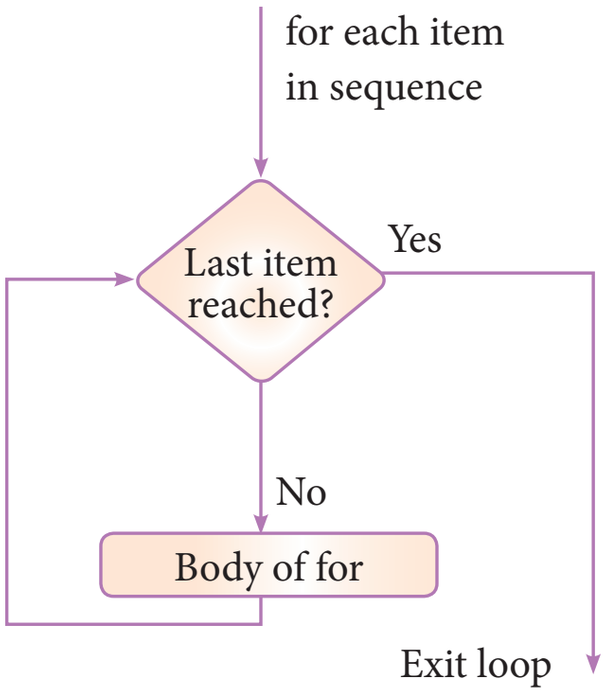
start – ప్రారంభ విలువను తెలియపరుచును.

stop – చివరి విలువను తెలియపరుచును

step – ఇంక్రిమెంట్ విలువలను తెలియపరుచును. ఇది ఐచ్ఛిక భాగము

ఉదాహరణ 6.8 range() కు ఉదాహరణ

```
range (1,30,1)      range యొక్క విలువ 1 నుండి 29 వరకు కలిగి ఉండును
range (2,30,2)      range () విలువలు 2 నుండి 28 వరకు
range (30,3,-3)     range () విలువలు 30 నుండి 6 వరకు
range (20)          20 అను విలువ చివరి విలువగా తీసుకుంటే (ఎగువహద్దు) 0 నుండి 19 వరకు
                    గల విలువలను ఇవ్వగలదు range1 ఎప్పుడును stop విలువ -1 వరకు విలువను
                    ఇవ్వగలదు అని తెలుసుకొనవచ్చును.
```



పటం 6.5 for లూప్ అమలుపరుచుట

ఉదాహరణ 6.9 for లూప్‌ను ఉపయోగించి ఒక అంకె సరిసంఖ్యను ముద్రించుట

```
for i in range (2,10,2):
        print (i, end=' ')
```

అవుట్‌పుట్:

2 4 6 8

for లూప్‌లో else భాగంను ఊహించుట

ఉదాహరణ 6.10: #for లూప్‌ను ఉపయోగించి else భాగంతో ఏక అంకె సరిసంఖ్యను ముద్రించుట.

```
for i in range(2,10,2):
        print (i,end=' ')
else:
        print ("\nEnd of the loop")
```

అవుట్‌పుట్:

```
2 4 6 8
End of the loop
```

గమనిక

పైథాన్‌లో లూప్ మరియు ఇతర నియంత్రణ స్టేట్‌మెంట్స్‌లో ఇండెంటెషన్ చాలా ముఖ్యం. C, C++ వంటి languages ఇండెంటెషన్ నిర్మాణ బ్లాక్‌లను మాత్రమే సృష్టిస్తుంది మరియు ఉప బ్లాక్‌లను కుండలీకరణ { } గుర్తులో సూచిస్తుంది.

12th_CS_Telugu_Medium_Chapter_6 .indd 80 27-03-2023 11:08:34

range() ను ఉపయోగించి 1 నుండి 100 వరకు గల సంఖ్యల మొత్తంను ఊహించు మరియొక్క ప్రోగ్రామ్

ఉదాహరణ 6.11: #1 నుండి 100 వరకు గల సంఖ్యల మొత్తంను గణించుట.

```
n = 100
sum = 0
for counter in range(1,n+1):
        sum = sum + counter
print("Sum of 1 until %d: %d" % (n,sum))
```

అవుట్‌పుట్:

```
Sum of 1 until 100: 5050
```

పై కోడ్‌లో *n* ప్రారంభ విలువ 100 మొత్తంకు ప్రారంభ విలువ 0, లూప్ ప్రారంభ విలువ 1 నుండి అమలు చేయబడుతుంది. ప్రతి పునరావృతానికి మొత్తం విలువను (counter) వేరియబుల్ మొత్తంతో నిల్వ చేయబడుతుంది. for లూప్ 1 నుండి ఎగువహద్దు 1 ని మళ్ళిస్తుందని గమనించండి. (ie. Value of n is set as 100, so this loop will iterate for values from 1 to 99 only, that is the reason why we have set the upper limit as n+1)

గమనిక

range (), string, list, dictionary మొదలగు వాటి నుండి కూడ విలువలను తీసుకుంటుంది. స్ట్రింగ్ వాడకాన్ని వివరించడానికి క్రింది ఉదాహరణ.

range నందు string ను ఉపయోగించడానికి ఉదాహరణలు

ఉదాహరణ 6.12: #range() లో స్ట్రింగ్‌ను వాడుట

```
for word in 'Computer':
   print (word,end=' ')
else:
   print ("\nEnd of the loop")
```

అవుట్‌పుట్

```
C o m p u t e r
End of the loop
```

(iii) నెస్టెడ్ లూప్ నిర్మాణము (Nested loop structure):

వేరొక లూప్‌లో ఉంచిన లూప్‌ను నెస్టెడ్ లూప్ నిర్మాణము అంటారు. ఒక దానిలో మరొకటి ఉంచవచ్చు **for** అటువంటి సమూహ ఉచ్చులను **while** నిర్మించడానికి కొంత కాలం లోపల మరియు లోపల క్రింద నమూనాను ముద్రించడానికి for లూప్‌ను వాడుదురు.

```
1
1   2
1   2   3
1   2   3   4
1   2   3   4   5
```

ఉదాహరణ 6.13: # while లూప్‌లో నెస్టెడ్ లూప్ యొక్క ఉపయోగం

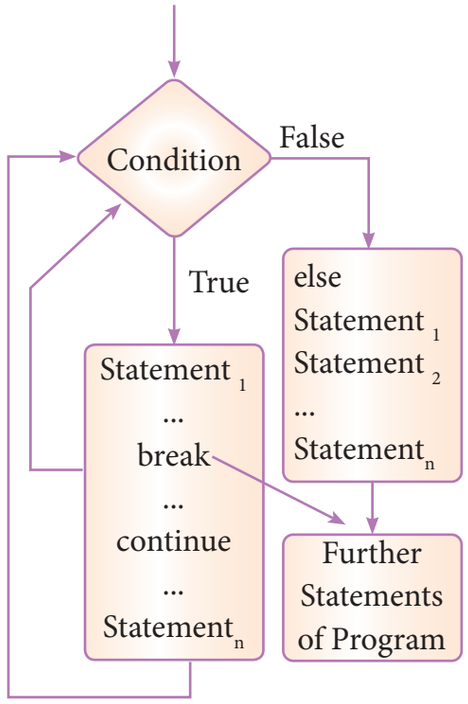
```
i=1
while (i<=6):
    for j in range (1,i):
        print (j,end='\t')
    print (end='\n')
    i +=1
```

అవుట్‌పుట్:

```
1
1    2
1    2   3
1    2    3     4
1    2    3     4      5
```

6.2.4 పైథాన్‌లో Jump స్టేట్‌మెంట్స్

పైథాన్‌లోని jump స్టేట్‌మెంట్స్ ఎటువంటి షరతులు లేకుండా ప్రోగ్రామ్ యొక్క ఒక భాగం నుండి మరొక భాగమునకు నియంత్రణను బదిలీచేయడానికి ఉపయోగించబడుతుంది. jump స్టేట్‌మెంట్‌ను ఉపయోగించడానికి పైథాన్‌లో మూడు ప్రత్యేక **keywords** ఉన్నాయి అవి **Break, Continue, Pass**.



పటము 6.6 break, continue statement ను లూప్స్ నిర్మాణంలో ఉపయోగించడం.

(i) Break స్టేట్‌మెంట్ (Break statement):

Break స్టేట్‌మెంట్ అది కలిగి ఉన్న లూప్స్‌కు ముగుస్తుంది. ప్రోగ్రామ్ యొక్క నియంత్రణ లూప్ యొక్క **body** ఎప్పుడు లేదా లూప్ కోసం తప్పుగా పరీక్షించబడే వరకు మళ్ళిస్తుంది.

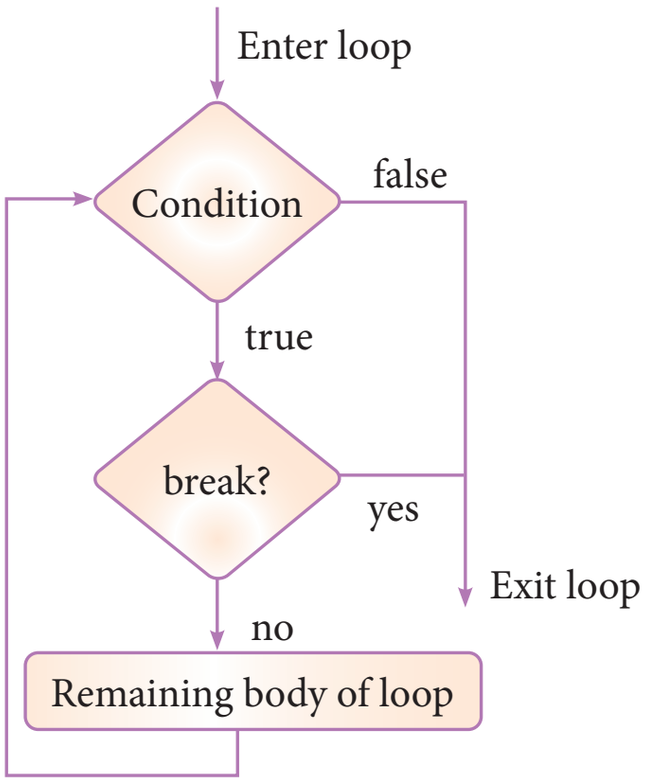
12th_CS_Telugu_Medium_Chapter_6 .indd 82 27-03-2023 11:08:34

కొంత కాలం లేదా లూప్ కోసం పరిస్థితి తప్పుగా పరీక్షించబడే వరకు మళ్ళించబడును. కాని ప్రతి ఒకరు కూడా లూప్‌ను **break** స్టేట్మెంట్ సహాయంతో నిబంధనను బదలాయించెదరు. ఎప్పుడయితే loop స్టేట్‌మెంట్ అమలు చేయబడుతుందో నియంత్రణ ప్రవాహం loop నుండి బయటకు వచ్చి loop నిర్మాణము తరువాత loop code అమలవుతుంది.

break స్టేట్మెంట్ ఒక సమూహ లూప్ లోపల ఉంటే (మరొక లూప్ లోపల లూప్) ఇన్‌నర్ లూప్‌ను బహిష్కరిస్తుంది.

Syntax:

break



పటము 6.7 *break* స్టేట్మెంట్ పనిచేయుట

for లూప్ మరియు **while** లూప్‌లో break స్టేట్మెంట్ పనిచేయు విధము

```
for var in sequence:
        if condition:
                break
                #code inside for loop
#code outside for loop
while test expression:
        #code inside while loop
        if condition:
           break
        #code inside while loop
#code outside while loop
```

ఉదాహరణ: 6.14: for లూప్ లోపల break స్టేట్మెంట్ ఉపయోగము

```
for word in “Jump Statement”:
    if word = = “e”:
        break
        print (word, end= ' ')
```

అవుట్‌పుట్:

```
Jump Stat
```

పైన ఇవ్వబడిన ప్రోగ్రామ్ string Jump స్టేట్మెంట్‌ను పునరావృతపరుస్తుంది. string సీక్వెన్స్‌లో ఇచ్చిన యొక్క ప్రతి అక్షరము ‘e’ అక్షరము ఎదురయ్యే వరకు పరీక్షించబడుతుంది. ఎదురైన నియంత్రణ లూప్ block వెలుపల బదిలీ చేయబడుతుంది, లేదా ముగుస్తుంది. అవుట్‌పుట్‌లో చూపిన విధముగా లూప్ ఆగిపోయిన తరువాత ‘e’ అక్షరము తనిఖీ అయ్యే వరకు ఇది ప్రదర్శించబడుతుంది.

ఇక్కడ ఒక ముఖ్యమైన విషయం గమనించాలి. విరామం ద్వారా ఒక లూప్ మిగిలి ఉంటే మిగతా భాగం అమలు చేయబడదు. దీన్ని వివరించడానికి మునుపటి ప్రోగ్రామ్‌ను వేరే భాగములో మెరుగు పరచడానికి అనుమతిస్తుంది. దీన్ని అవుట్‌పుట్ ఎలా ఉంటుందో చూడండి.

ఉదాహరణ: 6.15 # forలూప్ లోపల break స్టేట్మెంట్‌ను వాడకాన్ని వివరించే ప్రోగ్రామ్

```
for word in “Jump Statement”:
    if word = = “e”:
        break
        print (word, end=”)
else:
        print (“End of the loop”)
print (“\n End of the program”)
```

అవుట్‌పుట్:

```
Jump Stat
End of the program
```

break స్టేట్మెంట్ లూప్ మరియు else విభాగాన్ని కూడా దాటి వేసిందని గమనించాలి మరియు లూప్ block తరువాతి పంక్తిని బదిలీని కూడ నియంత్రించును.

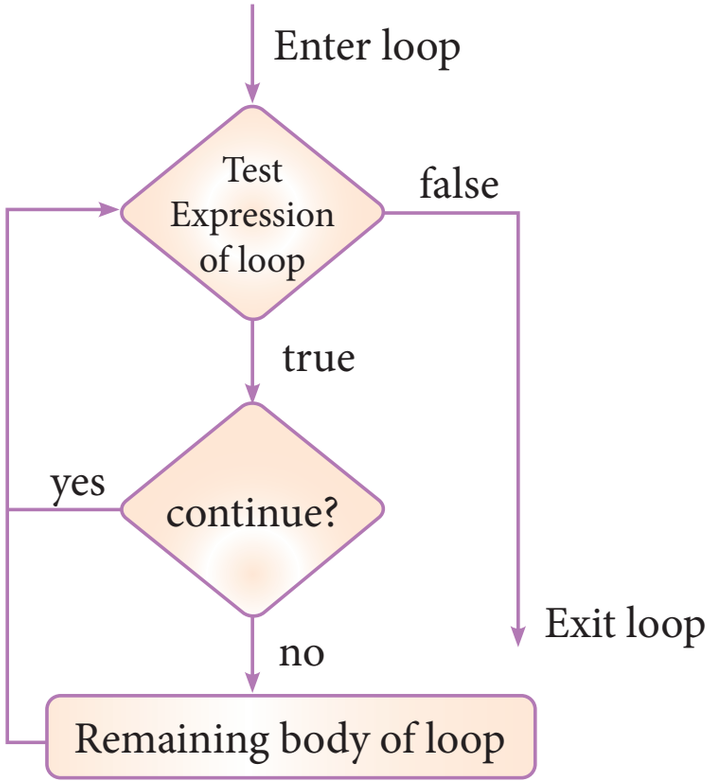
(ii) continue స్టేట్మెంట్:

Continue స్టేట్మెంట్ break స్టేట్మెంట్ మాదిరిగా కాకుండా మిగిలిన భాగాన్ని దాటవేయడానికి లూప్ మరియును పునరావృతపరుస్తుంది.

12th_CS_Telugu_Medium_Chapter_6 .indd 84 27-03-2023 11:08:34

Syntax:

continue



పటము 6.8 *Working of continue* స్టేట్మెంట్ పనిచేయుట

while లూప్ మరియు **for** లూప్‌లో continue స్టెట్మెంట్ పనిచేయు విధము క్రింద ఇవ్వబడినది.

```
for var in sequence:
        # code inside for loop
        if condition:
                continue
                #code inside for loop
#code outside for loop
while test expression:
        #code inside while loop
        if condition:
                continue
                #code inside while loop
#code outside while loop
```


12th_CS_Telugu_Medium_Chapter_6 .indd 85 27-03-2023 11:08:34

ఉదాహరణ: 6.16: for లూప్‌లో continue స్టేట్‌మెంట్ యొక్క ఉపయోగము

```
for word in “Jump Statement”:
    if word = = “e”:
        continue
    print (word, end = ' ')
print (“\n End of the program”)
```

Output:

```
Jump Statmnt
End of the program
```

పై ప్రోగ్రామ్ break స్టేట్‌మెంట్ కోసం మేము వ్రాసిన ప్రోగ్రామ్ మాదిరిగానే ఉంటుంది. మేము దీనిని continue లో భర్తీ చేసాము. అక్షరం ‘e’ మినహా దీనిని అవుట్‌పుట్‌లో చూడవచ్చు.

(iii) pass స్టేట్‌మెంట్ (pass statement):

పైథాన్ ప్రోగ్రామ్‌లోని **pass** స్టేట్‌మెంట్ శూన్యమైన స్టేట్‌మెంట్ వ్యాఖ్యత చేత అమలు చేయబడినపుడు (pass) స్టేట్‌మెంట్ పూర్తిగా విస్మరించబడుతుంది. pass అమలు అయినపుడు ఏమి జరగదు. అది ఆపరేషన్ చేయదు.

(if) నిబంధనలు మరియు లూప్ నిర్మాణములు ఆ బ్లాక్‌లోని ఏవైన స్టేట్‌మెంట్ లేదా ఆదేశాలను అమలుచేయకూడదు అనుకుంటే pass స్టేట్‌మెంట్‌ను ఉపయోగించవచ్చు

Syntax:

pass

ఉదాహరణ: 6.17: స్టేట్‌మెంట్ వాడకాన్ని వివరించే ప్రోగ్రామ్

```
a=int (input(“Enter any number :”))
if (a==0):
    pass
else:
    print (“non zero value is accepted”)
```

అవుట్‌పుట్:

```
Enter any number :3
non zero value is accepted
```

When the above code is executed if the input value is 0 (zero) then no action will be performed, for all the other input values the output will be as follows:

12th_CS_Telugu_Medium_Chapter_6 .indd 86 27-03-2023 11:08:34

గమనిక

pass స్టేట్‌మెంట్‌ను సాధారణంగా placeholder ఉపయోగపడుతుంది. భవిష్యత్‌లో అమలుచేయవలసిన లూప్ లేదా పంక్షన్ మనకు ఉన్నప్పుడు, ఇప్పుడు కాదు, ఖాళీ body సెగ్మంట్‌తో అలాంటి ఫంక్షన్‌లు లేదా లూప్‌లను అభివృద్ధి చేయలేము. ఎందుకనగా వ్యాఖ్యత లోపం పెరుగుతుంది. కాబట్టి దీనిని నివారించడానికి pass స్టేట్‌మెంట్‌ను ఉపయోగించవచ్చు.

ఉదాహరణ 6.18: for లూప్ లో pass స్టేట్‌మెంట్ వాడకాన్ని వివరించు program

```
for val in “Computer”:
        pass
print (“End of the loop, loop structure will be built in future”)
```

అవుట్‌పుట్:

```
End of the loop, loop structure will be built in future.
```

గుర్తించుకోవలసిన విషయాలు

- ప్రోగ్రామ్‌లో స్టేట్‌మెంట్‌ను కలిగి ఉంటుంది. వీటిని వరుసగా అమలు చేయడానికి ప్రవాహ నియంత్రణ స్టేట్‌మెంట్‌ను ఉపయోగిస్తారు.
- ప్రోగ్రామ్ స్టేట్‌మెంట్‌లోని నిబంధనను ఒక భాగం నుండి మరొక భాగమునకు బదిలీ చేయడాన్ని నిబందిత నిర్మాణము లేదా నిబందిత స్టేట్‌మెంట్ అంటారు.
- నియంత్రణ ప్రవాహాలు మూడు రకాలు:
    1. Sequencing
    2. Branching లేదా ప్రత్యామ్నాయం
    3. పునరావృతం
- పైథాన్‌లో if స్ట్రక్షర్ యొక్క వివిధ రూపాలను ఉపయోగించి ప్రత్యామ్నాయాలను తయారుచేస్తారు.
- పైథాన్ ప్రోగ్రామింగ్‌లో ఇండెంటేషన్ కీలక పాత్ర పోషిస్తుంది. ఇండెంటేషన్‌లో సమూహ స్టేట్‌మెంట్‌ను {} ఉపయోగించాల్సిన అవసరం లేదు.
- పైథాన్ ఇంటర్ ప్రెటరర్ అన్ని ఇండెంటేషన్ దోషాలకు లోపం వివరిస్తుంది.
- runtime input ను అంగీకరించడానికి పైథాన్ యొక్క మునుపటి సంస్కరణలు raw_input() కి తేజ సంస్కరణలకు input() కి మద్దతిస్తుంది.
- print() అవుట్‌పుట్‌ను యూజర్ ఫార్మేట్ చేయడానికి escape సీక్వెన్స్ వాడకానికి మద్దతిస్తుంది.
- range() లూప్ కోసం విలువల శ్రేణిని సరఫరా చేయడానికి ఉపయోగించబడుతుంది.
- పైథాన్‌లో break, continue, pass అనునవి jump స్టేట్‌మెంట్‌గా ఉంటాయి.
- pass స్టేట్‌మెంట్ అనునది ఒక శూన్యాంశ స్టేట్‌మెంట్ దీనిని సాధారణంగా place holder లో ఉపయోగిస్తారు

12th_CS_Telugu_Medium_Chapter_6 .indd 87 27-03-2023 11:08:34



ప్రయోగాత్మక పరీక్ష:

1. ఇవ్వబడిన అక్షరం అచ్చు (vowel) అగునా కాదా అని పరీక్షించుటకు ప్రోగ్రామ్ రాయండి.
2. if..else స్టేట్‌మెంట్‌ను ఉపయోగించి ఇవ్వబడిన మూడు సంఖ్యలలో సంఖ్యని పరీక్షించుటకు ప్రోగ్రామ్ రాయండి
3. ఇవ్వబడిన సంఖ్య ధనాత్మకము, ఋణాత్మకము లేదా zero (సున్న) పరీక్షించుటకు ప్రోగ్రామ్ రాయండి
4. ఫిబినాకి (Fibonacci) 0 1 1 2 3 4 5..... సీరిస్‌లో ప్రదర్శించుటకు ప్రోగ్రామ్ రాయండి.
5. సహజ సంఖ్యలు మొత్తానికి ప్రోగ్రామ్ రాయండి.
6. ఇవ్వబడిన సంఖ్య palindrome అవునా కాదా అని ప్రోగ్రామ్ రాయండి.
7. క్రింది వాటిని ముద్రించుటకు ప్రోగ్రామ్ రాయండి.

```
* * * * *
* * * *
* * *
* *
*
```

8. లీఫ్ సంవత్సరం అగునా కాదా అను పరీక్షించుటకు ఒక ప్రోగ్రామ్ రాయండి



మూల్యాంకనం

భాగము - I

F5IBP

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. పైథాన్‌లో ఎన్ని ముఖ్యమైన కంట్రోల్ స్ట్రక్చర్స్ ఉన్నవి?

   అ) 3 ఆ) 4 ఇ) 5 ఈ) 6

2. elif యొక్క విస్తరణ.

   అ) nested if ఆ) if..else ఇ) else if ఈ) if..elif

3. పైథాన్ ప్రోగ్రామింగ్‌లో ఏది ముఖ్య పాత్ర వహించును.

   అ) nested if ఆ) if..else ఇ) Structure ఈ) Indentation

4. సాధారణంగా placeholder గా ఉపయోగించునది ఏది?

   అ) continue ఆ) break ఇ) pass ఈ) goto

12th_CS_Telugu_Medium_Chapter_6 .indd 88 27-03-2023 11:08:34

5. if statement యొక్క నిబంధన క్రింది వాటిలో ఏ రూపంలో ఉండును.

అ) అంకగణిత లేదా సంబందాల వ్యక్తీకరణ

ఆ) అంకగణిత లేదా తార్కిక వ్యక్తీకరణ

ఇ) సంబంధ లేదా తార్కిక వ్యక్తీకరణ

ఈ) అంకగణిత

6. ఏది సులభతరమైన లూప్ అగును.

అ) do..while ఆ) while ఇ) for ఈ) if..elif

7. క్రింది snippet యొక్క output ఏది?

```
i=1
while True:
        if i%3 ==0:
                break
        print(i,end='')
        i +=1
```

అ) 12 ఆ) 123 ఇ) 1234 ఈ) 124

8. క్రింది snippet యొక్క output ఏది?

```
T=1
while T:
        print(True)
        break
```

అ) False ఆ) True ఇ) 0 ఈ) 1

9. jump statement కానిది ఏది?

అ) for ఆ) pass ఇ) continue ఈ) break

10. punctuation ఖాళీగా ఉపయోగిస్తారు.

```
if <condition>_
   statements-block 1
else:
   statements-block 2
```

అ) ; ఆ) : ఇ) :: ఈ) !

భాగము -II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. పైథాన్‌లోని control structures ఏవి?
2. break statement గురించి గమనికను రాయండి?
3. if..else statement యొక్క syntax ను రాయండి?
4. control structure ను వివరించండి.
5. లూప్‌లో range () గురించి రాయండి

భాగము -III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. క్రింది అవుట్‌పుట్‌కు ప్రోగ్రామ్ రాయండి.

   A

   A B

   A B C

   A B C D

   A B C D E

2. if..else structureపైన గమనిక రాయండి.
3. if..else..elif statement ను ఉపయోగించి అతి పెద్ద మూడెంకల సంఖ్యలను ప్రదర్శించుటకు సరియైన ప్రోగ్రామ్‌లను రాయండి.
4. while loop యొక్క syntax రాయండి.
5. break మరియు continue statements మధ్య గల భేదాలు తెల్పండి?

భాగము -IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. for loop గురించి విపులంగా వివరించండి.
2. if..else..elif statement ను ఉదాహరణాలతో సహా వివరించండి.
3. మూడు అంకెల బేసి సంఖ్యలను ప్రదర్శించుటకు ప్రోగ్రామ్ రాయండి.
4. ఇవ్వబడిన సంఖ్యకు గుణకార పట్టికను ప్రదర్శించుటకు ప్రోగ్రామ్ రాయండి.

12th_CS_Telugu_Medium_Chapter_6 .indd 90 27-03-2023 11:08:34

యూనిట్ II

అధ్యాయం 7

పైథాన్ ఫంక్షన్స్

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశమును నేర్చుకున్న తరువాత విద్యార్థులు:

- విధులు మరియు దాని రకాలను తెలుసుకొనుట.
- యూజర్ డిఫెండ్ (User defined) మరియు Built in ఫంక్షన్‌ల మధ్యగల తేడాలను తెలుసుకోగలరు.
- ఒక ఫంక్షన్‌ను (function) ఎలా పిలవాలో తెలుసుకొందురు.
- ఫంక్షన్ యొక్క పారామితులను అర్థం చేసుకోగలుగుట, అనాయక ఫంక్షన్‌ల గురించి తెలుసుకొనుట.
- గణిత మరియు కొన్ని (String) ఫంక్షన్‌ల గురించి తెలుసుకొనుట.

7.1 పరిచయం

ఒక నిర్ధిష్టమైన పనిని చేయడానికి రూపొందించిన కోడ్‌ను బ్లాక్స్ అని పిలుస్తారు. ఫంక్షన్ చేసే ఒక నిర్ధిష్ట పనిని దానికి భాద్యత వహించే దానిని function పిలుస్తారు. ఒక ప్రోగ్రామ్‌లో ఒక పనిని చాలా సార్లు చేయవలసి వస్తే ఒకే పనికోసం మనం అన్ని కోడ్‌లను మళ్ళీ మళ్ళీ టైపు చేయవలసిన అవసరం లేదు మరియు ఆ పనిని నిర్వహించడానికి అంకితమైన function పిలువబడును మరియు ఫంక్షన్ లోపల గల కోడ్‌ను అమలు చేయమని పైథాన్ చెబుతుంది. function లో ఉపయోగించడం వలన మనం ప్రోగ్రామ్‌లను రాయడం, చదవడం, పరీక్షించడం మరియు లోపాలను పరిష్కరించడం సులభమవుతుందని గమనించవచ్చు.

ఫంక్షన్‌ల యొక్క ప్రధాన ఉపయోగాలు:

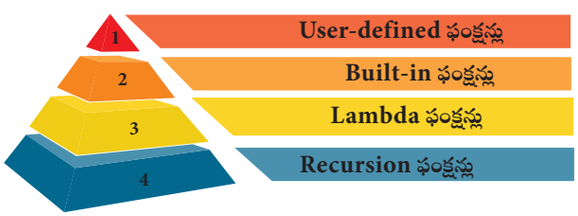
- ఇవి పునరావృతం (*repetition*) ను నివారిస్తుంది మరియు అధిక స్థాయిలో కోడ్‌ను పునర్వినియోగం చేస్తుంది.
- ఇది మీ అప్లికేషన్‌ను మెరుగైన మాడ్యులారిటిని అందిస్తుంది.

గమనిక

ఫంక్షన్ అనునది సమూహం తప్ప మరొకటి కాదు. ఇది ఒక ప్రత్యేకమైన పనిని తెలియజేస్తుంది.

7.1.1 ఫంక్షన్ రకాలు (Types of Functions)

సాధారణంగా ఫంక్షన్‌లను క్రింది విధముగా విభజించవచ్చును.



పటం 7.1 పైథాన్ ఫంక్షన్ యొక్క రకాలు

ఫంక్షన్	వివరణ
User-defined ఫంక్షన్స్	వినియోగదారులు స్వతహగా నిర్వచించని ఫంక్షన్లు

12th_CS_Telugu_Medium_Chapter_7.indd 91 29-06-2022 15:02:49

అంతర్నిత ఫంక్షన్లు	పైథాన్‌లో అంతనిర్మిత ఫంక్షన్లు
లేమిడా ఫంక్షన్లు	అనీమక పేరుతని ఫంక్షన్ అగును.
పునరావృత ఫంక్షన్లు	పునరావృతాన్ని పిలిచే ఫంక్షన్లు అంటారు.

పట్టిక 7.1 – పైథాన్ విధులు మరియు దాని యొక్క విధులు.

7.2 ఫంక్షన్‌ను నిర్వచించండి

ఒక ఫంక్షన్‌ను సృష్టించి దానిని ఉపయోగించడానికి మరొక ఫంక్షన్‌ను నిర్వచించవలయును. పైథాన్ లాంగ్వేజ్‌లో పలు అంతర్నిమిత ఫంక్షన్లు ఉన్నవి. (ఉదాహరణకు) అయినను వినియోగదారు తనకు అవసరమైన ఫంక్షన్‌లను సృష్టించుకొనుటకు వీలు అగును. ఫంక్షన్‌లను నిర్వచించేటపుడు గుర్తించుకోవలసిన విషయాలు లేదా అంశాలు.

- ఫంక్షన్ బ్లాక్స్ **def** అను కీవోర్డ్స్‌తో ప్రారంభమై ఫంక్షన్ పేట మరియు () తో అంతమవుతుంది.
- ఒక ఫంక్షన్ నిర్వచించేటప్పుడు ఏదైనా ఇన్‌పుట్ పారామితులు లేదా వాదనలు () గుర్తులు ఉండాలి.
- కోడ్ బ్లాక్ ఎల్లప్పుడు కోలన్ (:) మరియు ఇండెంటెడ్ తరువాత వస్తుంది.
- **return** స్టేట్‌మెంట్ ఫంక్షన్‌ను ముగిస్తుంది. కావాలి అనుకుంటే ఫంక్షన్ విలువని మరల పంపిస్తుంది. ఫంక్షన్ లేని **return return none** కు సమానమైనది.

గమనిక

ఫంక్షన్ పేరులో పైథాన్ keywords ను ఉపయోగించకూడదు.

7.2.1 వినియోగదారు నిర్వచించు ఫంక్షన్ యొక్క Syntax (Syntax for User defined function)

```
def <function_name ([parameter1, parameter2...] )> :
        <Block of Statements>
        return <expression / None>
```

గమనిక

ఈ (Syntax) [] కు విచ్ఛికము.

బ్లాక్:

బ్లాక్ అనునది ఒకటి లేదా అంతకన్న ఎక్కువ పంక్తులతో కోడ్‌ను సమూహంగా కలపడానికి దానిని అమలు చేయునపుడు ఒక పెద్ద స్టేట్‌మెంట్ పంక్తిగా తీసుకోబడుతుంది. పైథాన్‌లో బ్లాక్స్‌లోని స్టేట్‌మెంట్‌ను వ్రాయడానికి ఇండెంటేషను వాడుదురు. సాదారణంగా ఒక పంక్తిని 4 విరామాలు తో ప్రారంభించబడుతుంది. బ్లాక్స్‌లో ఉన్న అన్ని స్టేట్‌మెంట్స్ బక్& మాధిరి ఇండెంట్ లెవల్‌గా ఉండాలి.

నెస్టెడ్ బ్లాక్:

ఒక బ్లాక్ లోపల మరొక బ్లాక్ ఉన్నట్లు అయితే దానిని నెస్టెడ్ బ్లాక్ అంటారు. మొదటి బ్లాక్ స్టేట్‌మెంట్‌ని ఇండెంట్ చేయడానికి ఒక విరామ tab ను రెండవ బ్లాక్ స్టేట్‌మెంట్‌ని ఇండెంట్ చేయడానికి ద్వి విరామ tab ను వాడవలయును.

ఫంక్షన్ నిర్వచించడానికి ఇక్కడ ఒక ఉదాహరణ

```
def Do_Something( ):
value =1        #Assignment Statement
return value    #Return Statement
```


12th_CS_Telugu_Medium_Chapter_7.indd 92 29-06-2022 15:02:49

ఇప్పుడు ఫంక్షన్ పని తీరును పరీక్షించుట మూలముగా అది ప్రోగ్రామ్‌లో ఏ విధంగా ప్రదర్శించబడుతుంది అని చూడవచ్చు క్రింద ఇవ్వబడిన ఉదాహరణ ఒక సాధారణ ఫంక్షన్‌ను ప్రదర్శించును.

ఉదాహరణ:

```
def hello():
        print ("hello - Python")
        return
```

7.2.2 వినియోగదారు – నిర్వచించు ఫంక్షన్ యొక్క ఉపయోగాలు:

1. ఫంక్షన్ ప్రోగ్రామ్‌ని చిన్న మాడ్యుల్స్‌గా విభజించడానికి ఉపయోగపడును. ఇది కోడ్‌ని సులభంగా నిర్వహించగలదు.
2. కోడ్‌ను తిరిగి ఉపయోగించడంలో సహాయపడుతుంది. స్టేట్‌మెంట్ క్రమాన్ని అమలు చేసిన ప్రతిసారి ఆ ప్రమేయాన్ని పిలవడానికి సరిపోతుంది.
3. ఫంక్షన్ యొక్క విధులను మార్చడానికి సులభం అవుతుంది. వేరు వేరు ప్రోగ్రామ్‌లు వేరు వేరు ఫంక్షన్‌లో పనిచేయగలవు.

7.3 ఫంక్షన్‌ను పిలవడం

ఉదాహరణ 7.2.1 లో hello() ఫంక్షన్ పిలవడం కొరకు ఈ కోడ్ ఉపయోగించుట:

"hello()" ఫంక్షన్ పిలిచినపుడు క్రింది ప్రోగ్రామ్‌లో string అవుట్ పుట్‌గా ప్రదర్శించబడుతుంది.

అవుట్‌పుట్

hello – Python

ప్రత్యాన్మాయంగా "hello()" ఫంక్షన్ పిలవడానికి print() ఫంక్షన్ క్రింది ఉదాహరణలో కనబడుతుంది.

ఉదాహరణ:

```
def hello():
        print ("hello - Python")
        return
print (hello())
```

return స్టేట్‌మెంట్‌లో ఎటువంటి పిలుపు లేనపుడు చివరగా అవుట్ పుట్‌లో **"None"** ప్రదర్శించబడుతుంది.

పై ఫంక్షన్ యొక్క అవుట్ పుట్ క్రింది విధంగా ఉంటుంది.

అవుట్‌పుట్:

hello – Python
None

7.4 ఫంక్షన్‌లో Passing Parameters

పారామితులు లేదా arguments ను ఫంక్షన్‌లోకి పంపవచ్చును.

```
def function_name (parameter(s) separated by comma):
```

ఫంక్షన్‌లను నిర్వచించేటపుడు పారామితులు యొక్క ఉపయోగాన్ని చూద్దాం. పారామితులుని కుండలీకరణ ({}) గుర్తులతో ఫంక్షన్‌కి ఉపయోగిస్తారు. మీరు అన్ని రకాల డేటాను ఫంక్షన్‌లోకి పంపవచ్చును pass పారామితులను ఉపయోగించి ఫంక్షన్‌లో నిర్వచించే ఒక ప్రోగ్రామ్‌కు ఈ ఉదాహరణ.

ఉదాహరణ:

```
# assume w = 3 and h = 5
    def area(w,h):
        return w * h
    print (area (3,5))
```

పై కోడ్ వెడల్పు మరియు ఎత్తు విలువలను **w** మరియు **h** అనే పారామితులకు కేటాయిస్తుంది. వైశాల్యం అను ఫంక్షన్‌ను సృష్టించుటకు ఈ పారామితులు ఉపయోగపడును. పైన ఇవ్వబడిన ఫంక్షన్‌లని పిలిచినపుడు అది వెడల్పు మరియు ఎత్తుల లబ్దాన్ని అవుట్ పుట్‌గా తిరిగి ఇస్తుంది.

3 మరియు 5 అను విలువలను వరుసగా **w** మరియు **h** కు పంపినపుడు. 15ను output return ఫంక్షన్ చేయును.

మీకు తెలుసా!

మనం తరచుగా పారామితులు మరియు arguments అను పదాలను పరస్పరం మార్చి మార్చి ఉపయోగిస్తుంటాము. ఇది వాటి మధ్య స్వల్ప వ్యత్యాసము పారామితులు ఫంక్షన్‌లలో ఉపయోగించే వేరియబుల్స్ అయితే arguments అనునది ఫంక్షన్‌లకు పారామితి పంపే విలువలు.

7.5 ఫంక్షన్ ఆర్గుమెంట్స్

arguments ని ఫంక్షన్లు అని పిలుస్తారు మరియు అని 4 ఉపయోగించ తగిన పంక్షలు అవి:

1. అవసరమైన వాదనలు
2. keyword వాదనలు
3. డీఫాల్ట్ వాదనలు
4. వేరియబుల్ – లెంథ్ (length) arguments

12th_CS_Telugu_Medium_Chapter_7.indd 94 29-06-2022 15:02:49

ప్రమేయ వాధనలు

1. అవసరమైన వాదనలు
2. keyword వాదనలు
3. డీఫాల్ట్ వాదనలు
4. వేరియబుల్ – లెంథ్ (length) arguments

7.5.1 అవసరమైన వాధనలు (Required Arguments):

Arguments సరియైన క్రమములో ఫంక్షన్‌లో పంపుటను అవసరమైన arguments అందురు. ఇక్కడ ఫంక్షన్‌లో పిలువబడే వాదనల సంఖ్య ఫంక్షన్ నిర్వచనంలో ఖచ్చితంగా సరిపోవాలి. అవసరమైన వాటిని పొందడానికి, syntax లోపాలను నివారించడానికి ఒక పారామితి అవసరము.

ఉదాహరణ:

```
def printstring(str):
        print ("Example - Required  arguments ")
        print (str)
        return
# Now you can call printstring() function
printstring()
```

పై కోడ్‌ను అమలు చేస్తున్నపుడు క్రింది (error) ను చూపించును.

```
Traceback (most recent call last):
File "Req-arg.py", line 10, in <module>
printstring()
    TypeError: printstring() missing 1 required positional argument: 'str'
```

పై కోడ్‌ల **printstring()** బదులుగా **printstring (“Welcome”)** ఉపయోగిస్తే దాని అవుట్ పుట్

అవుట్‌పుట్:

Example - Required arguments
Welcome

7.5.2 Keyword Arguments

పారామితులు వాటి పారామితుల పేర్లు గుర్తించిన తరువాత Keyword arguments ఫంక్షన్‌లను ప్రారంభిస్తాయి. Keyword arguments యొక్క విలువ పారామితి యొక్క పేరుతో సరిపోతుంది. అందువల్ల arguments సరికాని క్రమంలో కూడా ఉండవచ్చు.

ఉదాహరణ:

```
def printdata (name):
        print (“Example-1 Keyword arguments”)
        print (“Name :”,name)
        return
# Now you can call printdata() function
printdata(name = “Gshan”)
```

పైన ఉన్న కోడ్ అమలు అయినట్లు అయితే క్రింది అవుట్‌పుట్ వెలువడును.

అవుట్‌పుట్:

Example-1 Keyword arguments
Name :Gshan

ఉదాహరణ:

```
def printdata (name):
        print (“Example-2 Keyword arguments”)
        print (“Name :”, name)
        return
# Now you can call printdata() function
printdata (name1 = “Gshan”)
```

పైన ఉన్న కోడ్ అమలు అయినట్లు అయితే క్రింది ఫలితము వెలువడును.

ఉదాహరణ:

```
def printdata (name, age):
        print ("Example-3 Keyword arguments")
        print ("Name :",name)
        print ("Age :",age)
        return
# Now you can call printdata()  function
printdata (age=25, name="Gshan")
```


12th_CS_Telugu_Medium_Chapter_7.indd 96 29-06-2022 15:02:49

పైన ఉన్న కోడ్ అమలు అయినట్లు అయితే క్రింది ఫలితము వెలువడుతుంది.

అవుట్‌పుట్:

```
Example-3 Keyword arguments
Name : Gshan
Age : 25
```

గమనిక

పైన ఉన్న ప్రోగ్రామ్‌లో పారామితుల వరుస మారియున్నది.

7.5.3 Arguments డీఫాల్ట్ (Default Arguments)

పైథాన్‌లో ఫంక్షన్ పిలవబడినపుడు ఎటువంటి విలువ ఇవ్వకపోయిన డీఫాల్ట్ తామే విలువలను తీసుకుంటాయి. దీనిని డీఫాల్ట్ ఆర్గుమెంట్స్ అని అంటారు. క్రింద ఇవ్వబడిన ఉదాహరణ argument ఇవ్వనపుడు డీఫాల్ట్ argument డీఫాల్ట్ salary ను ముద్రించును.

ఉదాహరణ:

```
def printinfo( name, salary = 3500):
        print ("Name: ", name)
        print ("Salary: ", salary)
        return
printinfo("Mani")
```

పైన ఉన్న కోడ్ అమలు అయితే క్రింది అవుట్ పుట్ వెలువడును.

అవుట్‌పుట్:

```
Name: Mani
Salary: 3500
```

పైన ఉన్న కోడ్‌లో print info ("Ram",2000) అని మార్చిన క్రింది అవుట్ పుట్ వెలువడును.

అవుట్‌పుట్:

```
Name: Ram
Salary: 2000
```

పైన ఉన్న విలువ 2000 ను salary argument పంపినపుడు ముందుగా ఉన్న డీఫాల్ట్ విలువ నిరాకరించబడును.

12th_CS_Telugu_Medium_Chapter_7.indd 97 29-06-2022 15:02:49

7.5.4 వేరియబుల్ – లెంథ్ ఆర్గుమెంట్స్ (Variable-Length Arguments):

కొన్ని సమయాలలో ముందుగా నిర్దేశించబడిన ఆర్గుమెంట్స్ కంటే అధికమైన ఆర్గుమెంట్స్ పంపవలసిన అవసరం ఉంటుంది. మరల ఫంక్షన్ పునర్నిర్మాణము చేయడం కష్టతరం. దానికి బదులుగా మారే వేరియబుల్ లెంథ్ arguments ఉపయోగించవచ్చు. ఫంక్షన్ నిర్వచనంలో ఇది అంతముఖ్యము కాదు మరియు asterisk (*) ను ఆర్గుమెంట్స్ నిర్వచించడానికి ఉపయోగిస్తారు.

3 కన్న ఎక్కువ arguments పంపినపుడు ఏమి జరుగును అని చూద్దాం.

ఉదాహరణ:

```
def sum(x,y,z):
        print("sum of three nos :",x+y+z)
sum(5,10,15,20,25)
```

పైన ఉన్న కోడ్ అమలు అయినపుడు క్రింది అవుట్ పుట్ వెలువడును.

TypeError: sum() takes 3 positional arguments but 5 were given

7.5.4.1 Syntax వేరియబుల్ Length Arguments (Syntax - Variable-Length Arguments)

```
def function_name(*args):
        function_body
        return_statement
```

ఉదాహరణ:

```
def printnos (*nos):
        for n in nos:
        print(n)
        return
# now invoking the printnos() function
print ('Printing two values')
printnos (1,2)
print ('Printing three values')
printnos (10,20,30)
```

అవుట్‌పుట్:

```
Printing two values
        1
        2
Printing three values
        10
        20
        30
```

నిన్ను నీవు పరీక్షించు

పైనున్న ప్రోగ్రామ్‌లో printnos ఫంక్షన్ పేరుకు బదులుగా printnames అని అన్నీ స్థలాలలో మార్చవలయును. ఉదాహరణకు printnos (10, 20, 30) ను printnames ('mala', 'kala', 'bala') అని ఇచ్చినపుడు వెలువడు అవుట్ పుట్

12th_CS_Telugu_Medium_Chapter_7.indd 98 29-06-2022 15:02:49

వేరియబుల్ లెంథ్ ఆర్గుమెంట్‌లలో మనం రెండు పద్ధతులను ఉపయోగించి వాదనలను పాస్ చేయవచ్చును.

1. కీ ఓర్డ్ కాని పేరియబుల్ ఆర్గుమెంట్స్

2. కీ ఓర్డ్ వేరియబల్ ఆర్గుమెంట్స్

కీ ఓర్డ్ కాని పేరియబుల్ ఆర్గుమెంట్స్‌ను. టుపుల్స్ **tuples** అని అందురు. ఈ అధ్యాయములో మనము టిపుల్స్ గురించి మరికొన్ని విషయాలు నేర్చుకొనెదము. ఇచ్చిన ప్రోగ్రామ్ కీ ఓర్డ్ కాని వేరియబుల్ keyword variable argument యొక్క ఉదాహరణ

గమనిక

కీ ఓర్డ్ వేరియబుల్ arguments ఈ పుస్తకము యొక్క పరిధికి మించినది.

మీకు తెలుసా!

పైథాన్ (ప్రింట్) print() ఫంక్షన్లు వేరియబుల్ లెన్త్ arguments ఆమోదించు ఫంక్షన్‌లకు ఒక ఉదాహరణ

7.6 పేరులేని ఫంక్షన్

పేరు లేని ఫంక్షన్లు అనగానేమి? (What is anonymous function?)

పైథాన్‌లో పేరు లేకుండా నిర్వచించబడిన ఫంక్షన్‌లను పేరులేని ఫంక్షన్లు అందురు. సాదారణ ఫంక్షన్ **def** కీ ఓర్డ్‌ను ఉపయోగించి నిర్వచించగా పైథాన్‌లో **lambda** కీ ఓర్డ్‌ను ఉపయోగించి నిర్వచించబడుతాయి. పేరు లేని ఫంక్షన్ లేదా **lambda** ఫంక్షన్లు అని అందురు.

lambda లేదా పేరులేని ఫంక్షన్ల ఉపయోగాలు:

- Lambda ఫంక్షన్లు ఎక్కువగా చిన్న మరియు ఒకసారి ఉపయోగించు పేరు లేని పంక్షనలుగా వాడుదురు.
- Lambda ఫంక్షన్ ప్రధానంగా filter(), map() మరియు reduce() వంటి ఫంక్షన్‌లో కలిపి ఉపయోగిస్తారు.

గమనిక

filter(), map() and reduce() ఫంక్షన్లు ఈ పుస్తకము యొక్క పరిధికి మించినది.

మీకు తెలుసా!

Lambda ఫంక్షన్ ఎన్ని విలువలను (వాదనలు) మరియు వ్యక్తీకరణ రూపములో ఒక విలువను తిరిగి ఇవ్వాలి. Lambda ఫంక్షన్ దాని పారామితి list లో గ్లోబల్ వేరియబుల్స్ మరియు వేరియబుల్స్‌ను మాత్రమే access చేయగలదు.

7.6.1 పేరులేని ఫంక్షన్ యొక్క సిన్‌టేక్స్ (Syntax of Anonymous Functions):

పేరులేని ఫంక్షన్ యొక్క సిన్‌టేక్స్ క్రింద ఇవ్వబడినది.

lambda [argument(s)] :expression

ఉదాహరణ:

```
sum = lambda arg1, arg2: arg1 + arg2
print ('The Sum is :', sum(30,40))
print ('The Sum is :', sum(-30,40))
```

అవుట్‌పుట్:

```
The Sum is : 70
The Sum is : 10
```

పైన పేర్కొనబడిన lambda ప్రమేయము ***arg1*** వాదనను ***arg2*** వాదనతో కూడి ఫలితమును sum() వేరియబుల్‌లో దాచును. దీని ఫలితము print() ఫంక్షన్ ఉపయోగించి ప్రదర్శించును.

7.7 return స్టేట్‌మెంట్స్

- రిటర్న్ (return) స్టేట్‌మెంట్ మీ ఫంక్షన్ నుండి నిష్క్రమించడానికి కారణమవుతుంది దాని caller విలువను తిరిగి ఇస్తుంది. సాధారణముగా ఫంక్షన్ point inputs ను తీసుకొని ఏదేని తిరిగి ఇవ్వడం.
- ఒక ఫంక్షన్ దాని caller కు విలువను తిరిగి ఇవ్వడానికి సిద్ధంగా ఉన్నప్పుడు రిటర్న్ స్టేట్‌మెంట్ ఉపయోగపడుతుంది.
- ఫంక్షన్‌లో ఒకటికి మించి రిటర్న్ స్టేట్‌మెంట్‌లు ఉండిన ఒకే ఒక రిటర్న్ స్టేట్‌మెంట్ మాత్రము రన్‌టైమ్‌లు అమలవుతుంది.

7.7.1 రిటర్న్ స్టేట్‌మెంట్ యొక్క సిన్‌టెక్స్ (Syntax of return):

return [expression list]

ఈ స్టేట్‌మెంట్ ఉన్న వ్యక్తీకరణలను మూల్యాంకణము చేసి విలువలను తిరిగి పంపుతుంది. ఫంక్షన్‌లో ఉన్న స్టేట్‌మెంట్ (expression) లేనిదిగా లేదా return స్టేట్‌మెంట్‌తో లేనిదిగా ఉన్నట్లు అయితే ఫంక్షన్ None విలువను తిరిగి పంపుతుంది.

12th_CS_Telugu_Medium_Chapter_7.indd 100 29-06-2022 15:02:50

ఉదాహరణ:

```
# return statment
def usr_abs (n):
        if n>=0:
                return n
        else:
                return –n
# Now invoking the function
x=int (input("Enter a number :")
print (usr_abs (x))
```

అవుట్‌పుట్ 1:

```
Enter a number : 25
        25
```

అవుట్‌పుట్ 2:

```
Enter a number : -25
        25
```

7.8 వేరియబుల్స్ యొక్క పరిధి

ఇది ప్రోగ్రామ్ యొక్కభాగాలను సూచిస్తుంది. అనగా ఏ భాగములో వేరియబుల్స్‌ను ఉపయోగించుతున్నాము దానిని తెలుపుతుంది. పరిధి దాని ప్రస్తుత వేరియబుల్ సమితిని మరియు వాటి విలువలను కలిగి ఉందని చెప్పగలము. రెండు రకాలైన పరిధులను **local scope** మరియు **global scope** లను అధ్యయనము చేస్తాము.

7.8.1 లోకల్ స్కోప్ (Local Scope)

ఫంక్షన్ సమితిలో చెప్పబడిన వేరియబుల్‌ను స్థానిక వేరియబుల్ అంటారు.

స్థానిక వేరియబుల్ యొక్క నియమాలు:

- స్థానిక పరిధి ఉన్న వేరియబుల్ అది సృష్టించబడిన ఫంక్షన్ బ్లాక్‌లో మాత్రమే access చేయవచ్చు.
- బ్లాక్ లోపల వేరియబుల్‌ను సృష్టించినపుడు వేరియబుల్ దానికి స్థానికంగా (local) మారుతుంది.
- formal arguments ఫంక్షన్ అమలు చేస్తునపుడు మాత్రమే స్థానిక వేరియబుల్ ఉంటుంది.
- పనిచేయడానికి కూడా స్థానికంగా ఉంటాయి.

ఉదాహరణ: స్థానిక వేరియబుల్‌ను సృష్టించుట

```
def loc():
        y=0 # local scope
        print(y)
loc()
```

అవుట్‌పుట్:

```
0
```

ఉదాహరణ: పరిధి వెలుపల స్థానిక వేరియబుల్‌ను Access చేయడం

```
def loc():
        y = "local"
loc()
print(y)
```

పై కోడ్‌ను అమలు చేసినపుడు (output) క్రింది లోపాన్ని చూపుతుంది.

NameError: name 'y' is not defined

పై లోపము గ్లోబల్ పరిధిలో లోకల్ వేరియబుల్ **'y'** అని ఇవ్వడం వలన ఏర్పడినది.

7.8.2 గ్లోబల్ పరిధి (Global Scope):

గ్లోబల్ స్కోప్‌తో ఉన్న వేరియబుల్‌ను ప్రోగ్రామ్‌లో ఎక్కడైనా ఉపయోగించవచ్చు ఎటువంటి ఫంక్షన్ పరిధినైనను వేరియబుల్‌కు బయట సృష్టించవచ్చు.

గ్లోబల్ కీ ఓర్డ్ యొక్క నియమాలు (Rules of global Keyword):

పైథాన్‌లో సాదారణగా ఉన్న గ్లోబల్ కీ ఓర్డ్స్ నియమాలు

- ఫంక్షన్‌ని బయట వేరియబుల్‌ను నిర్వచించునపుడు అది స్వతహాగా గ్లోబల్ అవుతుంది. గ్లోబల్ కీ ఓర్డ్‌ని మనం ఉపయోగించవలసిన అవసరం లేదు.
- ఫంక్షన్ లోపల గ్లోబల్ వేరియబుల్‌ను చదువుటకు మరియు వ్రాయుటకు గ్లోబల్ కీ ఓర్డ్స్‌ను ఉపయోగిస్తాము.
- ఫంక్షన్ వెలుపల గ్లోబల్ కీ ఓర్డ్ యొక్క వాడకం ఎటువంటి ప్రభావము లేదు.

గోబల్ కీ ఓర్డ్ యొక్క ఉపయోగాలు

ఉదాహరణ:ఫంక్షన్ లోపల గ్లోబల్ వేరియబుల్‌ను Access చేయడం

```
c = 1          # global variable
def add():
        print(c)
add()
```

అవుట్‌పుట్:

```
1
```


12th_CS_Telugu_Medium_Chapter_7.indd 102 29-06-2022 15:02:50

ఉదాహరణ: ఫంక్షన్ లోపల గ్లోబల్ వేరియబుల్ Access చేయుట.

```
c = 1          # global variable
def add():
        print(c)
        add()
```

అవుట్‌పుట్:

```
1
```

ఉదాహరణ : ఫంక్షన్ లోపల గ్లోబల్ వేరియబుల్‌ను మార్చుట

```
c = 1          # global variable
def add():
        c = c + 2 # increment c by 2
        print(c)
add()
```

అవుట్‌పుట్:

```
Unbound Local Error: local variable 'c' referenced before assignment
```

గమనిక

గ్లోబల్ అను ప్రత్యేక కీ ఓర్డ్‌ను ఉపయోగించకుండా ఫంక్షన్ లోపల గ్లోబల్ వేరియబుల్‌ను మార్చుటకు వీలుకాదు. కాని గ్లోబల్ వేరియబుల్‌ను (access) చేయవచ్చు.

ఉదాహరణ: గ్లోబల్ కీ ఓర్డ్‌ను ఉపయోగించి ఫంక్షన్ లోపల గ్లోబల్ వేరియబుల్‌ను మార్చుట. పై ప్రోగ్రామ్‌ను రన్ చేయునపుడు క్రింది **Output** వెలువడును.

```
x = 0                          # global variable
def add():
        global x
        x = x + 5              # increment by 2
        print ("Inside add() function  x value is :", x)
add()
print ("In main x value is :", x)
```

అవుట్‌పుట్:

```
Inside add() function  x value is : 5
In main x value is : 5
```

పై ప్రోగ్రామ్‌లో **x** అనునది గ్లోబల్ వేరియబుల్‌గా నిర్వచించబడినది. **add()** ఫంక్షన్ గ్లోబల్ కీ ఓర్డ్‌ను **x** వేరియబుల్‌కు ఉపయోగించి దానికి 5 అనే విలువను కూడబడుచున్నది. ఇప్పుడు మనం మార్చుచేసిన గ్లోబల్ వేరియబుల్ **x** ను ఫంక్షన్ వెలుపల చూడవచ్చు. అనగా **x** యొక్క విలువ 5.

7.8.3 గ్లోబల్ మరియు స్థానిక వేరియబుల్ (Global and local variables)

ఇక్కడ మనం ఒకే విధమైన కోడ్‌లో గ్లోబల్ వేరియబుల్ మరియు స్థానిక వేరియబుల్‌ను ఎలా ఉపయోగించాలో చూద్దాం.

ఉదాహరణ: ఒకే కోడ్‌లో లోకల్ మరియు గ్లోబల్ వేరియబుల్‌ను ఉపయోగించడం.

```
x=8                    # x is a global variable
def loc():
        global x
        y = "local"
        x = x * 2
        print(x)
        print(y)
loc()
```

అవుట్‌పుట్:

```
16
local
```

పై ప్రోగ్రామ్‌లో ***x*** అనునది గ్లోబల్ వేరియబుల్ గాను మరియు ***y*** ను లోకల్ వేరియబుల్‌గాను **loc()** ఫంక్షన్‌లో చెప్పబడినది.

loc() ఫంక్షన్‌ను పిలిచిన తర్వాత యొక్క విలువ 16 గా మారును. ఎందుకనగా **x=x * 2.** అని ఉపయోగించబడినది. దానికి తర్వాత స్థానిక వేరియబుల్ ***y*** విలువ ముద్రించబడుతున్నది.

ఉదాహరణ: ఒకే పేరు ఉన్న గ్లోబల్ వేరియబుల్ మరియు స్థానిక వేరియబుల్

```
x = 5
def loc():
        x = 10
        print ("local x:", x)
loc()
print ("global x:", x)
```

అవుట్‌పుట్:

```
local x: 10
global x: 5
```

పై కోడ్‌లో గ్లోబల్ వేరియబుల్ మరియు స్థానిక వేరియబుల్‌లో ఒకే విధమైన పేరు 'x' ను ఉపయోగించబడమైనది. రెండు పరిధులలో వేరియబుల్స్‌ను తెలుపడము వలన ముద్రించునపుడు వేర్వేరు ఫలితాలు లభించును. అనగా loc() ఫంక్షన్ లోపల ఉన్న స్థానిక పరిధి మరియు loc() ఫంక్షన్ బయట ఉన్న గ్లోబల్ పరిధి

The output :- local x: 10, is called local scope of variable.

The output:- global x: 5, is called global scope of variable.

12th_CS_Telugu_Medium_Chapter_7.indd 104 29-06-2022 15:02:50

7.9 లైబ్రరీలను ఉపయోగించి ఫంక్షన్లు

7.9.1 అంతర్ నిర్మిత మరియు గణిత విధులు ఫంక్షన్లు (Built-in and Mathematical functions)

ఫంక్షన్లు	వివరణ	సింటెక్స్	ఉదాహరణ
abs ( )	సంఖ్య యొక్క పూర్ణ సంఖ్యను త్రిప్పి పంపుతుంది. ఫంక్షన్ అనునది పూర్ణ సంఖ్యగాను లేదా దశాంశ సంఖ్యగాను ఉండవచ్చును.	abs (x)	x=20 y=-23.2 print('x = ', abs(x)) print('y = ', abs(y)) **అవుట్‌పుట్:** x = 20 y = 23.2
ord ( )	ఇవ్వబడిన యూనిక్ కోడ్ అక్షరానికి ASCII విలువను త్రిప్పి పంపుతుంది. ఇది chr() ఫంక్షన్‌కు విలోమము అగును.	ord (c)	c= 'a' d= 'A' print ('c = ',ord (c)) print ('A = ',ord (d)) **అవుట్‌పుట్:** c = 97 A = 65
chr ( )	ఇవ్వబడిన ASCII విలువకు యూనికోడ్ అక్షరము త్రిప్పి పంపును. ఇది ord() ఫంక్షన్ యొక్క విలోమము అగును.	chr (i)	c=65 d=43 print (chr (c)) prin t(chr (d)) **అవుట్‌పుట్:** A +
bin ( )	ఇవ్వబడిన పూర్ణసంఖ్యకు సమమైన బైనరీ స్ట్రింగ్ 0b ముందుగా ఉండి త్రిప్పి పంపించును గమనిక: దీనికి బదులుగా (format) ఫంక్షన్ ఉపయోగించవచ్చు. పంపించును.	bin (i)	x=15 y=101 print ('15 in binary : ',bin (x)) print ('101 in binary : ',bin (y)) **అవుట్‌పుట్:** 15 in binary : 0b1111 101 in binary : 0b1100101

12th_CS_Telugu_Medium_Chapter_7.indd 105 29-06-2022 15:02:50

type ( )	ఇవ్వబడిన single object. యొక్క రకమును వేరుచేసి త్రిప్పి పంపును. గమనిక: (single object) పారామితులకు ఈ ఫంక్షన్ వాడుదురు.	type (object)	x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s)) **అవుట్‌పుట్:** <class 'float'> <class 'str'> <class 'bool'>
id ( )	ఇవ్వబడిన (object) యొక్క గుర్తింపును దాని చిరునామాకు త్రిప్పి పంపును. గమనిక: x, y చిరునామాలు ప్రతి యొక్క సిస్టమ్‌కు వేర్వేరుగా నుండును.	id (object)	x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y)) **అవుట్‌పుట్:** address of x is : 1357486752 address of y is : 13480736
min ( )	ఇవ్వబడిన list నుండి అతి చిన్న విలువను త్రిప్పి పంపును.	min (list)	MyList = [21,76,98,23] print ('Minimum of MyList :', min(MyList)) **అవుట్‌పుట్:** Minimum of MyList : 21
max ( )	ఇవ్వబడిన list నుండి అతి పెద్ద విలువను త్రిప్పి పంపును.	max (list)	MyList = [21,76,98,23] print ('Maximum of MyList :', max(MyList)) **అవుట్‌పుట్:** Maximum of MyList : 98
sum ( )	ఇవ్వబడిన list లోని విలువల మొత్తాన్ని త్రిప్పిపంపును.	sum (list)	MyList = [21,76,98,23] print ('Sum of MyList :', sum(MyList)) **అవుట్‌పుట్:** Sum of MyList : 218

12th_CS_Telugu_Medium_Chapter_7.indd 106 29-06-2022 15:02:50

format ( )	ఇవ్వబడిన సంఖ్యను వేరు సంఖ్యా రూపములోనికి మార్చి త్రిప్పి పంపును. 1. ద్వి సంఖ్యా మానములో వెలువడిన Outputs ఆ సంఖ్య యొక్క ఆధారము 2 2. ఆక్టల్ ఫార్మేట్‌లో వెలువడిన Outputs ఆ సంఖ్య యొక్క ఆధారము 8 3. Fixed-point నంబరులో ప్రదర్శించు సంఖ్య fixed-point సంజ్ఞతో ఉండును.	format (value [, format_spec])	x= 14 y= 25 print ('x value in binary :',format(x,'b')) print ('y value in octal :',format(y,'o')) print('y value in Fixed-point no ',format(y,'f')) **అవుట్‌పుట్:** x value in binary : 1110 y value in octal : 31 y value in Fixed-point no : 25.000000
round ( )	ఇవ్వబడిన సంఖ్యను దగ్గరగా ఉన్న పూర్ణ సంఖ్యగా మార్చి త్రిప్పి పంపును. 1. మొదటి ప్రాముఖ్యత గల సంఖ్యను గుర్తించును.	round (number [,ndigits])	x= 17.9 y= 22.2 z= -18.3 print ('x value is rounded to', round (x)) print ('y value is rounded to', round (y)) print ('z value is rounded to', round (z))
	2. సంఖ్య తర్వాత ఉన్న దశాంశ స్థానాల సంఖ్యను గుర్తించి రెండవ argument గుర్తించును.		**అవుట్‌పుట్:1** x value is rounded to 18 y value is rounded to 22 z value is rounded to -18 n1=17.89 print (round (n1,0)) print (round (n1,1)) print (round (n1,2)) **అవుట్‌పుట్:2** 18.0 17.9 17.89

pow ( )	ఇవ్వబడిన సంఖ్య ab ఘాత లబ్ధమును త్రిప్పి పంపును. (a**b) యొక్క ఘాతము b	pow (a,b)	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) print (pow (a+b,3)) **అవుట్‌పుట్:** 25 125.0 343

గణిత విధులు (Mathematical Functions):

గమనిక

అన్ని గణిత విధులను ఉపయోగించే ముందు గణిత మాడ్యుల్ వివరణను పేర్కొనండి.

ప్రమేయము	వివరణ	Syntax	ఉదాహరణలు
floor ( )	x కన్న తక్కువ లేదా x కు సమమైన పెద్ద పూర్ణాంకమును త్రిప్పి పంపును.	math.floor (x)	import math x=26.7 y=-26.7 z=-23.2 print (math.floor (x)) print (math.floor (y)) print (math.floor (z)) **అవుట్‌పుట్:** 26 -27 -24
ceil ( )	x కన్న పెద్దది లేదా x కు సమమైన చిన్న పూర్ణాంకమును త్రిప్పి పంపును.	math.ceil (x)	import math x= 26.7 y= -26.7 z= -23.2 print (math.ceil (x)) print (math.ceil (y)) print (math.ceil (z)) **అవుట్‌పుట్:** 27 -26 -23

12th_CS_Telugu_Medium_Chapter_7.indd 108 29-06-2022 15:02:50

sqrt ( )	x యొక్క వర్గ మూలమును త్రిప్పి పంపును. గమనిక: x యొక్క విలువ 0 (zero) కన్న ఎక్కువగా ఉండవలెను.	sqrt (x)	import math a= 30 b= 49 c= 25.5 print (math.sqrt (a)) print (math.sqrt (b)) print (math.sqrt (c)) **అవుట్‌పుట్:** 5.477225575051661 7.0 5.049752469181039

7.9.2 ఫంక్షన్‌లలో కూర్పు (Composition in functions):

ఫంక్షన్‌తో కూర్పు అంటే ఏమిటి? (What is Composition in functions?)

ఫంక్షన్ ద్వారా తిరిగి ఇవ్వబడిన విలువ మరొక ఫంక్షన్ కోసం argument గా ఉపయోగించబడుతుంది. దీనిని కూర్పు పద్ధతి అంటారు. ఉదాహరణకు మనము సంఖ్య యొక్క విలువను లేదా వినియోగదారు నుండి ఇన్‌పుట్‌గా వ్యక్తీకరణను, తెలుసుకొనడానికి (ఇన్‌పుట్) **input()** మూలముగా వినియోగదారు నుండి **input string**ను తీసుకొంటాము మరియు దాని విలువను అంచనగా వేయడానికి **eval()** ఫంక్షన్‌ను అమలు చేస్తాము.

ఉదాహరణ: ఈ ప్రోగ్రామ్ కూర్పుఉ వివరిస్తుంది.

```
# This program explains composition
        >>> n1 = eval (input ("Enter a number: "))
Enter a number: 234
        >>> n1
        234
>>> n2 = eval (input ("Enter an arithmetic expression: "))
        Enter an arithmetic expression: 12.0+13.0 * 2
>>> n2
        38.0
```

7.10 పైథాన్ పునరావృత విధులు

ఒక ఫంక్షన్ తనను తాను పలవడాన్ని పునరావృతం అంటారు. పునరావృతం లూప్ లాగా పనిచేస్తుంది. కాని కొన్ని సార్లు లూప్ కన్న పునరావృతాన్ని ఉపయోగించడం మరింత అర్థమే మీరు ఏదైనా లూప్‌ను పునరావృతంగా మార్చవచ్చును.

పునరావృత ఫంక్షన్ తనను తాను పిలుస్తుంది. కొన్ని షరతులతో ఆపకపోతే ఒక ప్రక్రియ నిరవధికంగా మల్లిస్తుందని ఊహించుకోండి. ఇటువంటి ప్రక్రియను అనంత పునరుక్తి అంటారు. ఏదైన పునరావృత ఫంక్షన్‌లో వర్తించే నిబంధనను బేస్ నిబంధన అని అంటారు. ప్రతి పునరావృత ఫంక్షన్‌లో బేస్ నిబంధన ఉండాలి, లేకేంటే అది అనంతమైన లూప్ లాగ అమలు అగును.

12th_CS_Telugu_Medium_Chapter_7.indd 109 29-06-2022 15:02:50

పునరావృత ఫంక్షన్ ఏ విధంగా పనిచేస్తుంది.

1. కొన్ని పునరావృత ఫంక్షన్‌ను బాహ్య కోడ్స్ అంటారు.
2. బేస్ నిబంధన నెరవేరితే ప్రోగ్రామ్ అర్థవంతమైన అవుట్‌పుట్‌ను ఇస్తుంది మరియు నిష్క్రమిస్తుంది.
3. లేకపోతే ఫంక్షన్ కొన్ని అవసరమైన ప్రాససింగ్ చేస్తుంది మరియు తరువాత పునరావృతం కొనసాగించమని పిలుస్తుంది.

factorial ను లెక్కించడానికి ఉపయోగించే పునరావృత ఫంక్షన్ యొక్క ఉదాహరణ.

ఉదాహరణ:

```
def fact(n):
        if n == 0:
                return 1
        else:
                return n * fact (n-1)
print (fact (0))
print (fact (5))
```

అవుట్‌పుట్:

```
1
120
```

మీకు తెలుసా!

print (fact(2000)) కూర్పుతో గరిష్ట పునరావృత లోతు మించిపోయిన తరువాత Runtime Error లోపం ఇస్తుంది. ఎందుకనగా పైథాన్ అప్రమేయంగా 1000 కాల్స్ తరువాత పునరావృత పంక్షను పిలవడం వలన ఇది జరుగుతుంది. ఇది sys.setrecursionlimit ((limit_value)) ఉపయోగించి పరిమితిని మార్చడానికి కూడ మిమ్మల్ని అనుమతిస్తుంది.

ఉదాహరణ:

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
        if n == 0:
                return 1
         else:
                return n * fact(n-1)
print(fact (2000))
```


12th_CS_Telugu_Medium_Chapter_7.indd 110 29-06-2022 15:02:50

గుర్తించుకోవలసిన అంశాలు:

- ఫంక్షన్ గుర్తించిన పనిని చేయడానికి డిజైన్ చేయబడిన బ్లాక్స్ అని అంటారు.
- వినియోగదారు నిర్వచించిన Built-in, lambda మరియు recursion అనునవి ఫంక్షన్ రకాలు.
- ఫంక్షన్ బ్లాక్ "def" అను keyword తో ప్రారంభమవు ఫంక్షన్ పేరు మరియు పారామితి () తో అంతమవుతుంది.
- arguments లేని "return" ఏదికాదు. పైథాన్‌లో "return" స్టేట్‌మెంట్ ఐచ్ఛికం
- పైథాన్‌లో ఒక బ్లాక్‌లోని స్టేట్‌మెంట్‌లు ఇండెంటెషన్‌తో ప్రారంభం కావాలి.
- ఒక బ్లాక్‌లోని బ్లాక్‌ను నెస్టెడ్ బ్లాక్ అంటారు.
- ఒక ఫంక్షన్ పిలవడానికి arguments ఉపయోగపడుతాయి. మరియు ప్రధానంగా నాలుగు రకాల ఫంక్షన్లు ఉపయోగించవచ్చును. అవి అవసరమైన వాదనలు, Keyword వాదనలు, డీఫాల్ట్ వాదనలు, వేరియబుల్ లెంథ్ వాదనలు
- అవసరమైన arguments సరైన స్థాన క్రమంలో ఒక ఫంక్షన్‌కు పంపబడిన arguments
- పారామితులు తరువాత Keyword arguments ఫంక్షన్‌ను ప్రారంభిస్తాయి. ఫంక్షన్ నిర్వచనంలో పారామితులు కోసం డీఫాల్ట్ విలువను ఇవ్వడానికి పైథాన్ అనుమతిస్తుంది. మేము దీనిని డీఫాల్ట్ arguments అని పిలుస్తాము.
- ఫంక్షన్ యొక్క నిర్వచనంలో వేరియబుల్ లెంథ్ arguments పెర్కొన బడలేదు మరియు నిర్వచించడానికి asterisk (*) ఉపయోగించబడుతుంది.
- పేరు లేని ఫంక్షన్ అనునది పేరు లేకుండా నిర్వచించబడిన ఒక ఫంక్షన్
- వేరియబుల్ యొక్క పరిధి ప్రోగ్రామ్ యొక్క భాగమును చూపిస్తుంది. ఇక్కడ అది అందుబాటులో ఉంటుంది. అనగా దాని ప్రాంతాన్ని మీరు సూచించవచ్చు.
- ఒక ఫంక్షన్ ద్వారా తిరిగివచ్చే విలువ మరొక ఫంక్షన్ కోసం సమూహ పద్ధతిలో arguments ఉపయోగించబడుతుంది. దీనిని కూర్పు అంటారు.
- తనను తాను పిలిచే ఒక ఫంక్షన్‌ను పునరావృతం అంటారు. పునరావృతం లూప్ లాగ పనిచేస్తుంది. కాని కొన్ని సార్లు లూప్ కంటే పునరావృతిని ఉపయోగించడం ఎక్కువ అర్ధమే.

12th_CS_Telugu_Medium_Chapter_7.indd 111 29-06-2022 15:02:50



ప్రయోగాత్మక పరీక్ష:

ప్రయోగం:1 ఉదాహరణ 7.5.3 లో ఉన్న ప్రోగ్రామ్‌ని క్రింది కోడ్‌లో ప్రయత్నించి చూడండి.

Slno	కోడ్	ఫలితము
1	printinfo(“3500”)	
2	printinfo(“3500””Sri”)	
3	printinfo(name=”balu”)	
4	printinfo(“Jose”,1234)	
5	printinfo(“ ”,salary=1234)	

ప్రయోగం:2 క్రింది ఫంక్షన్‌ను మూల్యాంకనము చేసి దాని outputని రాయండి.

Slno	ప్రమేయం	అవుట్‌పుట్
1	eval(‘25*2-5*4')	
2	math.sqrt(abs(-81))	
3	math.ceil(3.5+4.6)	
4	math.floor(3.5+4.6)	

ప్రయోగం:3 క్రింది ఫంక్షన్‌ను మూల్యాంకనము చేసి అవుట్‌పుట్ రాయండి.

Slno	ప్రమేయము	అవుట్‌పుట్
1	1) abs(-25+12.0)) 2) abs(-3.2)	
2	1) ord('2') 2) ord('$')	
3	type('s')	
4	bin(16)	
5	1) chr(13) 2) print(chr(13))	
6	1) round(18.2,1) 2) round(18.2,0) 3) round(0.5100,3) 4) round(0.5120,3)	

12th_CS_Telugu_Medium_Chapter_7.indd 112 29-06-2022 15:02:50

7	1) format(66, 'c') 2) format(10, 'x') 3) format(10, 'X') 4) format(0b110, 'd') 5) format(0xa, 'd')	
8	1) pow(2,-3) 2) pow(2,3.0) 3) pow(2,0) 4) pow((1+2),2) 5) pow(-3,2) 6) pow(2*2,2)	



మూల్యాంకనం

N6J01

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. ఒక నిర్ధిష్ట పని చేయడానికి రూపొందింపబడిన కోడ్‌ను ఏ విధంగా తెలుపుతారు.

   అ) లూప్ ఆ) Branching ఇ) ఫంక్షన్ ఈ) బ్లాక్

2. తనను తాను పిలిచే ఒక ఫంక్షన్ అంటారు.

   అ) అంతర్నిర్మిత ఆ) పునరావృతం సి) లాంబ్డా ఈ) తిరిగి

3. ఏ ఫంక్షన్‌ను అనామక అన్‌నేమ్డ్ ఫంక్షన్ అంటారు.

   ఎ) లాంబ్డా బి) పునరావృతం సి) ఫంక్షన్ డి) నిర్వచించండి.

4. ఫంక్షన్ బ్లాక్ ప్రారంభించడానికి క్రింది వాటిలో ఏది కీవర్డ్ ఉపయోగించబడుతుంది.

   ఎ) నిర్వచించండి బి) కోసం సి) చివరకు డి) డెఫ్

5. ఫంక్షన్ బ్లాక్ నుండి నిష్క్రమించడానికి కింది వాటిలో ఏది కీవర్డ్ ఉపయోగించబడుతుంది.

   ఎ) నిర్వచించండి బి) తిరిగి సి) చివరకు డి) డెఫ్

6. ఒక ఫంక్షన్‌ను నిర్వచించేటప్పుడు క్రింది వాటిలో ఏది ఉపయోగపడుతుంది.

   ఎ) ; సెమికోలన్ బి) . చుక్క (డాట్) సి) : కోలాన్ డి) $ లు

12th_CS_Telugu_Medium_Chapter_7.indd 113 29-06-2022 15:02:50

7. ఏ వాదనలో సరైన స్థాన క్రమం ఒక ఫంక్షన్‌కు పంపబడుతుంది.

ఎ) అవసరం బి) కీ ఓర్డ్ సి) డిపాల్ట్ డి) వేరియబుల్ పొడవు

8. క్రింది స్టేట్‌మెంట్ చదవండి మరియు సరైన స్టేట్‌మెంట్‌లు ఎన్నుకోండి

(I) 1. పైథాన్‌లో నిర్వచించేటప్పుడు మీకు నిర్ధిష్ట డేటా రకాలను సూచించాల్సిన అవసరం లేదు.

(II) 2. పైథాన్ కీలక పదాలను ఫంక్షన్ పేరుగా ఉపయోగించవచ్చు

ఎ) మొదటి (I) సరి మరియు (II) తప్పు

బి) రెండునూ సరి

సి) (I) తప్పు (II) సరియైనది

డి) రెండు తప్పు

9. ఇచ్చిన స్టేట్‌మెంట్‌ను విజయవంతముగా అమలు చేయడానికి సరియైన దానిని ఎన్నుకొనండి.

if _____ : (పింట్ x లీఫ్ ఇయర్)

ఎ) x%2=0 బి) x%4==0 సి) x/4=0 డి) x%4=0

10. టెస్ట్ పైథాన్ 0 ఫంక్షన్‌ను నిర్వచించడానికి క్రింది వాటిలో ఏది Key Word ఉపయోగపడుతుంది.

ఎ) నిర్వచించండి బి) పాస్ సి) డెఫ్ డి) అయితే

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. ఫంక్షన్ అంటే ఏమిటి?
2. వివిధ రకములైన ఫంక్షన్‌లను వ్రాయండి?
3. ఫంక్షన్ యొక్క ప్రధాన ప్రయోజనములను వ్రాయండి?
4. వేరియబుల్ యొక్క పరిధి అంటే ఏమిటి? దాని రకాలను పేర్కొనండి.
5. గ్లోబల్ పరిధిని నిర్వచించండి?
6. పునరావృత ఫంక్షన్‌లో బేస్ కండిషన్ అంటే ఏమిటి?
7. పునరావృత ఫంక్షన్ కోసం పరిమితిని ఎలా నిర్మించాలి? ఒక ఉదాహరణ నివ్వండి.

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. స్థానిక వేరియబుల్ యొక్క నియమాలను వ్రాయండి.
2. పైథాన్‌లో గ్లోబల్ కీవర్డ్ కోసం ప్రాథమిక నియమాలను వ్రాయండి.
3. ఫంక్షన్ లోపల గ్లోబల్ వేరియబుల్‌ను సవరించినప్పుడు ఏమి జరుగుతుంది?

12th_CS_Telugu_Medium_Chapter_7.indd 114 29-06-2022 15:02:50

4. (సిల్) ceil() మరియు (ఫ్లోర్) floor() ఫంక్షన్‌ల వ్యత్యాసమును తెల్పుము.

5. ఇచ్చిన సంవత్సరం లీపు సంవత్సరము కాదా అని తనిఖీ చేయడానికి పైథాన్ కోడ్‌ను వ్రాయండి.

6. ఫంక్షన్‌లలో కూర్పు అంటే ఏమిటి?

7. పునరావృత ఫంక్షన్ ఎలా పనిచేస్తుంది?

8. ఒక ఫంక్షన్‌ను నిర్వచించేటప్పుడు గమనించవలసిన అంశములు ఏవి?

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. వివిధ రకములైన ఫంక్షన్‌లను తెలిపి వాటికి ఒక్కొక్క దానికి ఒక ఉదాహరణము ఇవ్వండి.

2. వేరియబుల్ యొక్క పరిధిని ఉదాహరణతో వివరించండి.

3. క్రింద ఇవ్వబడిన అంతర్నిర్మిత విధులను వివరించండి.
   (a) id()
   (b) chr()
   (c) round()
   (d) type()
   (e) pow()

4. ఇచ్చిన రెండు సంఖ్యల యొక్క క.సా.గు ను కనుగొనుటకు పైథాన్ కోడ్‌ను ఉపయోగించండి?

5. పునరావృత ఫంక్షన్‌ను ఉదాహరణలతో వివరించండి.

Reference Books

1. *Python Tutorial book from tutorialspoint.com*
2. *Python Programming: A modular approach by Pearson – Sheetal, Taneja*
3. *Fundamentals of Python –First Programs by Kenneth A. Lambert*

12th_CS_Telugu_Medium_Chapter_7.indd 115 29-06-2022 15:02:50

యూనిట్ II

అధ్యాయం 8

స్ట్రింగ్ మరియు స్ట్రింగ్ నిర్వహణ

అభ్యసన లక్ష్యాలు

ఈ పాఠ్యాంశం పూర్తి అయిన తరువాత విద్యార్థులు తెలుసుకొను అంశములు.

- వచనాన్ని ఎలా ఉపయోగించాలో తెలుసుకొంటారు.
- పైథాన్‌లో వివిధ స్ట్రింగ్ ప్రమేయాలను అర్థం చేసుకొనుట.
- స్ట్రింగ్‌ను ఎలా ఫార్మాట్ చేయాలో తెలుసుకోవడం
- స్ట్రింగ్ పద్ధతులను గురించి తెలుసుకోవడం
- వాస్తవ ప్రపంచంలో స్ట్రింగ్ ఉపయోగాలను గురించి తెలుసుకొనుట

C85YS

8.1 పరిచయం

పైథాన్‌లో స్ట్రింగ్ అనునది ఒక డేటా రకం. దీనిని అక్షరాల కలయిక కోసం ఉపయోగిస్తారు. అక్షరాల కలయిక సింగల్, డబుల్ లేదా ట్రిపుల్ కోట్స్‌లో "జత" చేయబడిన సంఖ్యల చిహ్నాలు యూనికోడ్ అక్షరాల శ్రేణిని నిర్వహించడానికి స్ట్రింగ్‌లను ఉపయోగిస్తారు.

ఉదాహరణ

‘Welcome to learning Python’

“Welcome to learning Python”

‘‘‘ “Welcome to learning Python” ’’’

పైథాన్‌లో, స్ట్రింగ్‌లు మార్చలేనివి. దీని అర్థం మీరు ఒక స్ట్రింగ్‌ను నిర్వహించిన తరువాత అమలు చేయు సమయంలో స్ట్రింగ్‌ను మార్చుటకు వీలుకాదు.

8.2 స్ట్రింగ్‌ను సృష్టించుట

మనం ఇంతకు ముందు నేర్చుకున్నట్లు పైథాన్‌లో ఒక స్ట్రింగ్‌ను సింగల్ లేదా డబుల్ లేదా ట్రిపుల్ కోట్స్ ఉపయోగించి సృష్టించవచ్చు. సింగల్ కోట్స్‌లోని స్ట్రింగ్ ఏ ఇతర కోటేడ్ అక్షరాలను కలిగిఉండదు, ఎందుకనగా స్ట్రింగ్‌ను ఎక్కడ ప్రారంభించాలో మరియు ముగించాలో కంపైలర్ గుర్తించదు. ఈ సమస్యను అధిగమించడానికి మీరు డబుల్ కోట్స్‌లను (“ ”) ఉపయోగించాలి. డబుల్ కోట్స్‌లో ఉన్న స్ట్రింగ్‌ను ట్రిపుల్ కోట్స్ (‘‘‘ ’’’) లో నిర్వహించాలి. ట్రిపుల్ కోట్స్‌లో స్ట్రింగ్‌లను నిర్వచించడం కూడా బహుల పంక్తి మల్టీలైన్ స్ట్రింగ్‌లను సృష్టించడానికి అనుమతిస్తుంది.

12th_CS_Telugu_Medium_Chapter_8.indd 116 20-02-2023 16:21:52

ఉదాహరణ

```
#సింగిల్ కోటా లోపల స్ట్రింగ్‌ను నిర్వచించడం
>>> print (‘Greater Chennai Corporation’)
        Greater Chennai Corporation

#సింగిల్ కోటా లోపల సింగిల్ కోటెడ్ స్ట్రింగ్‌ను నిర్వచించడం
>>> print ('Greater Chennai Corporation's student')
        SyntaxError: invalid syntax

#డబుల్ కోట్స్ లోపల స్ట్రింగ్‌ను నిర్వచించుట
>>>print (“Computer Science”)
        Computer Science

#డబుల్ కోట్స్ లోపల డబుల్ కోటెడ్ స్ట్రింగ్‌ను నిర్వచించుట
>>> print ('" "Computer Science" "')
        "Computer Science"

#ట్రిపుల్ కోట్స్ లోపల సింగిల్ మరియు డబుల్ కోటెడ్ బహుల పంక్తి స్ట్రింగ్‌ను
  నిర్వచించడం
>>> print ('" "Strings are immutable in 'Python',
                which means you can't make any changes
                once you declared" "')
        "Strings are immutable in 'Python',
        which means you can't make any changes once you declared"
```

8.3 స్ట్రింగ్‌లో అక్షరాలను ప్రయోగించడం

మనము ఒక స్ట్రింగ్‌ను నిర్వచించిన తరువాత పైథాన్ దాని ప్రతి అక్షరానికి సూచిక విలువను కేటాయిస్తుంది. ఈ సూచిక విలువలను పాదిక (subscript) అని పిలుస్తారు. వీటిని స్ట్రింగ్‌లను ప్రవేశపెట్టటకు మరియు మార్చుటకు ఉపయోగిస్తారు. ఈ పాదిక (subscript) అనుకూల లేదా ప్రతికూల పూర్ణాంక సంఖ్యలు కావచ్చును.

అనుకూల పాధిక (subscript) 0 ను మొదటి అక్షరానికి మరియు చివరి అక్షరానికి **n-1** కేటాయించబడుతుంది. ఇక్కడ n అనేది స్ట్రింగ్‌లోని అక్షరాల సంఖ్య. రివర్స్ క్రమములో చివరి అక్షరం నుండి మొదటి అక్షరానికి కేటాయించిన ప్రతికూల సూచిక −1లో ప్రారంభించబడుతుంది.

ఉదాహరణ

స్ట్రింగ్	S	C	H	O	O	L
అనుకూల పాదిక	0	1	2	3	4	5
ప్రతికూల పాదిక	-6	-5	-4	-3	-2	-1

12th_CS_Telugu_Medium_Chapter_8.indd 117 20-02-2023 16:21:52

ఉదాహరణ 1 ఇవ్వబడిన స్ట్రింగ్ యొక్క అనుకూల పాధికతో అక్షరాన్ని ప్రవేశపెట్టే ప్రోగ్రామ్

```
str1 = input ("Enter a string: ")
index=0
for i in str1:
        print ("Subscript[",index,"] : ", i)
        index + = 1
```

అవుట్‌పుట్

```
Enter a string: welcome
Subscript [ 0 ] :  w
Subscript [ 1 ] :  e
Subscript [ 2 ] :  l
Subscript [ 3 ] :  c
Subscript [ 4 ] :  o
Subscript [ 5 ] :  m
Subscript [ 6 ] :  e
```

ఉదాహరణ 8.2 ఇవ్వబడిన స్ట్రింగ్ యొక్క అనుకూల పాధికతో అక్షరాన్ని ప్రవేశపెట్టే ప్రోగ్రామ్

```
str1 = input ("Enter a string: ")
index=-1
while index >= -(len(str1)):
        print ("Subscript[",index,"] : " + str1[index])
        index += -1
```

Output

```
Enter a string: welcome
Subscript [ -1 ] : e
Subscript [ -2 ] : m
Subscript [ -3 ] : o
Subscript [ -4 ] : c
Subscript [ -5 ] : l
Subscript [ -6 ] : e
Subscript [ -7 ] : w
```

8.4 స్ట్రింగ్‌లను సవరించడం మరియు తొలగించడం (Modifying and Deleting Strings):

మనం ముందుగా నేర్చుకొన్నట్లుగా పైథాన్‌లోని స్ట్రింగ్‌లు మార్చలేనివి. అనగా మనము నిర్వచించిన స్ట్రింగ్ తరువాత మార్పులు లేదా తొలగింపునకు అనుమతించబడదు. మనము స్ట్రింగ్‌ను సవరించాలి అనుకుంటే, కొత్త స్ట్రింగ్ విలువను ఇప్పటికే ఉన్న స్ట్రింగ్ వేరియబుల్‌కు కేటాయించవచ్చు.

12th_CS_Telugu_Medium_Chapter_8.indd 118 20-02-2023 16:21:52

ఉదాహరణ

```
>>> str1="How are you"
>>> str1[0]="A"
Traceback (most recent call last):
        File "<pyshell#1>", line 1, in <module>
        str1[0]="A"
TypeError: 'str' object does not support item assignment
```

పై ఉదాహరణలో స్ట్రింగ్ వేరియబుల్ str1 అను స్టేట్‌మెంట్‌లో “How are you” అనే స్ట్రింగ్‌తో కేటాయించబడుతుంది. తరువాత స్టేట్‌మెంట్‌లో స్ట్రింగ్ యొక్క మొదటి అక్షరాన్ని ‘A’ అక్షరంతో నవనీకరించడానికి ప్రయత్నించినపుడు కాని పైథాన్ నవనీకరణను అనుమతించదు. మరియు Type Error అని చూపిస్తుంది.

ఈ సమస్యను అధిగమించడానికి ఇప్పటికే ఉన్న స్ట్రింగ్ వేరియబుల్‌కు కొత్త స్ట్రింగ్ విలువను నిర్వచించవచ్చును. పైథాన్ ఇప్పటికే ఉన్న స్ట్రింగ్‌లో కొత్త స్ట్రింగ్‌ను పూర్తిగా ఓవర్ రైట్ చేస్తుంది.

ఉదాహరణ

```
>>> str1="How are you"
>>> print (str1)
        How are you
>>> str1="How about you"
>>> print (str1)
        How about you
```

సాధారణంగా పైథాన్ దాని స్ట్రింగ్‌లో ఎటువంటి మార్పులకు మద్దతు ఇవ్వదు. కాని ఇది స్ట్రింగ్‌లోని ఒక నిర్ధిష్ట అక్షరము యొక్క అన్ని సంఘటనలను మార్చడానికి ఫంక్షన్ చేయబడిన మార్పులు ఒరిజినల్ స్ట్రింగ్‌ను భాదించవు replace() ఫంక్షన్‌చే అందిస్తుంది.

replace () యొక్క సాధారణ రూపం:

replace (“char1”, “char2”)

replace ఫంక్షన్ char1 లోని అన్ని సంఘటనలను char2 తో రిప్లేస్ చేయును.

ఉదాహరణ

```
>>> str1="How are you"
>>> print (str1)
        How are you
>>> print (str1.replace("o", "e"))
        Hew are yeu
```

సవరణ మాదిరిగానే పైథాన్ ఒక నిర్ధిష్ట అక్షరాన్ని స్ట్రింగ్‌లో తొలగించడానికి అనుమతించదు. కాని command ను ఉపయోగించి మొత్తం స్ట్రింగ్ వేరియబుల్‌ను తొలగించవచ్చు.

12th_CS_Telugu_Medium_Chapter_8.indd 119 20-02-2023 16:21:52

ఉదాహరణ 3 : స్ట్రింగ్‌లోని నిర్ధిష్ట అక్షరాన్ని తొలగించడానికి ప్రోగ్రాం.

```
>>> str1="How are you"
>>> del str1[2]
Traceback (most recent call last):
        File "<pyshell#7>", line 1, in <module>
            del str1[2]
        TypeError: 'str' object doesn't support item deletion
```

ఉదాహరణ 4: స్ట్రింగ్‌లోని నిర్ధిష్ట అక్షరాన్ని తొలగించడానికి ప్రోగ్రాం.

```
>>> str1="How about you"
>>> print (str1)
        How about you
>>> del str1
>>> print (str1)
Traceback (most recent call last):
        File "<pyshell#14>", line 1, in <module>
            print (str1)
        NameError: name 'str1' is not defined
```

8.5 స్ట్రింగ్ ఆపరేటర్స్ (String Operators):

పైథాన్ క్రింది స్ట్రింగ్ ఆపరేటర్‌లను అందిస్తుంది. ఈ ఆపరేటర్‌లు స్ట్రింగ్‌ను మార్చడానికి ఉపయోగపడతాయి.

(i) అనుక్రమము (Concatenation (+))

రెండు లేదా అంతకన్నా ఎక్కువ స్ట్రింగ్‌లు కలపడాన్ని అనుక్రమము (Concatenation) పైథాన్‌లో స్ట్రింగ్‌ను అనుక్రమము చేయడానికి (+) ఆపరేటర్ ఉపయోగపడుతుంది.

ఉదాహరణ

```
>>> "welcome" + "Python"
        'welcomePython'
```

(ii) అనుబంధం (Append (+ =))

ఇప్పటికే ఉన్న స్ట్రింగ్ చివరిలో ఎక్కువ స్ట్రింగ్‌లను జోడించడాన్ని అనుబంధం అంటారు. ఇప్పటికే ఉన్న స్ట్రింగ్‌తో క్రొత్త స్ట్రింగ్‌ను జోడించడానికి (+=) ఆపరేటర్ ఉపయోగపడుతుంది.

ఉదాహరణ

```
>>> str1="Welcome to "
```


12th_CS_Telugu_Medium_Chapter_8.indd 120 20-02-2023 16:21:52

```
>>> str1+="Learn Python"
>>> print (str1)
```

Welcome to Learn Python

(iii) పునరావృతం (Repeating (*))

గుణకారం ఆపరేటర్ (*) ఒక స్ట్రింగ్‌ను బహుళ సంఖ్యలో అనేకమార్లు ప్రదర్శించడానికి ఉపయోగిస్తారు.

ఉదాహరణ

```
>>> str1="Welcome "
>>> print (str1*4)
```

Welcome Welcome Welcome Welcome

(iv) స్ట్రింగ్ విభజన (String slicing)

ఒక ప్రధాన స్ట్రింగ్ యొక్క ఉప స్ట్రింగ్‌ను విభజన అంటారు. ఒక [ ] ఆపరేటర్ మరియు ఇండెక్స్ లేదా పాధిక విలువలను ఉపయోగించి దాని నిజమైన స్ట్రింగ్ నుండి ఉప స్ట్రింగ్‌ను తీసుకొనబడును. కావున [ ] ను విభజన ఆపరేటరు అంటారు. విభజన ఆపరేటరును ఉపయోగించి ప్రధాన స్ట్రింగ్‌ను ఒకటి లేదా అంతకన్నా ఎక్కువ ఉపస్ట్రింగ్‌లను విభజన చేయవచ్చును.

విభజన ఆపరేటర్ యొక్క సాధారణ రూపము (General format of slice operation):

str[start:end]

start అనేది ప్రారంభ సూచిక మరియు ***end*** అనేది స్ట్రింగ్‌లోని చివరి అక్షరం యొక్క సూచిక విలువ. పైథాన్ పేర్కొన్న వాస్తవ సూచిక నుండి తుది విలువను ఒకటి కంటే తక్కువ తీసుకుంటుంది. ఉదాహరణకు మీరు స్ట్రింగ్ నుండి మొదటి నాలుగు 4 అక్షరాలను విభజన చేయాలి అనుకుంటే మీరు దీనిని 0 నుండి 5 గా పేర్కొనాలి ఎందుకనగా పైథాన్ తుది విలువను n-1 గా మాత్రమే పరిగణిస్తుంది.

ఉదాహరణ I : స్ట్రింగ్ నుండి ఒక అక్షరమును విభజన చేయండి.

```
>>> str1="THIRUKKURAL"
>>> print (str1[0])
```

T

ఉదాహరణ II : ఇండెక్స్ 0 నుండి 4 లో ఉన్న ఉపస్ట్రింగ్‌లను విభజించడం :

```
>>> print (str1[0:5])
```

THIRU

ఉదాహరణ III : ముగింపు సూచికను పేర్కొనకుండా సూచిక 0 నుండి 4 వరకు ఉప స్ట్రింగ్‌లను విభజన చేయడం :

```
>>> print (str1[:5])
```

THIRU

12th_CS_Telugu_Medium_Chapter_8.indd 121 20-02-2023 16:21:52

ఉదాహరణ IV : ప్రారంభ సూచికను ఉపయోగించకుండా సూచిక 0 నుండి 6 వరకు ఉపస్ట్రింగ్‌లను విభజన చేయడం :

```
>>> print (str1[6:])
```

KURAL

ఉదాహరణ V : లూప్‌కు, ఉపయోగపడు ఉపస్ట్రింగ్‌లను విభజించడం

```
str1="COMPUTER"
index=0
for i in str1:
        print (str1[:index+1])
        index+=1
```

అవుట్‌పుట్

```
C
C O
C O M
C O M P
C O M P U
C O M P U T
C O M P U T E
C O M P U T E R
```

(v) స్ట్రింగ్ ను విభజన చేయునపుడు Stride చేయడం (Stride when slicing string)

విభజన ఆపరేషన్ చేసినపుడు మీరు మూడవ ఆర్గ్యుమెంట్‌ను stride గా పేర్కొనవచ్చు. ఇది మొదటి అక్షరమును స్ట్రింగ్ నుండి తిరిగి పొందిన తరువాత ముందుకు సాగడానికి ఉపయోగపడు అక్షరాల సంఖ్యను సూచిస్తుంది. stride యొక్క డీఫాల్ట్ విలువ 1 అగును.

ఉదాహరణ

```
>>> str1 = "Welcome to learn Python"
>>> print (str1[10:16])
        learn
>>> print (str1[10:16:4])
        r
>>> print (str1[10:16:2])
        er
>>> print (str1[::3])
Wceoenyo
```

గమనిక : పైథాన్ చివరి విలువను n-1 గా తీసుకుంటుందని గుర్తించుకొండి.

మీరు ప్రతికూల విలువను stride (మూడవ ఆర్గ్యుమెంట్) గా కూడా ఉపయోగించవచ్చు. మీరు ప్రతికూల విలువను పేర్కొంటే అది తలక్రిందులుగా ముద్రిస్తుంది.

12th_CS_Telugu_Medium_Chapter_8.indd 122 20-02-2023 16:21:52

ఉదాహరణ

```
>>> str1 = "Welcome to learn Python"
>>> print(str1[::-2])
nhy re teolW
```

8.6 స్ట్రింగ్ ఫార్మేటింగ్ ఆపరేటర్‌లు (String Formatting Operators):

స్ట్రింగ్ ఫార్మాటింగ్ ఆపరేటర్ పైథాన్ యొక్క అత్యంత ఉత్తేజకరమైన లక్షణాలలో ఒకటి. ఈ (%) ఫార్మేటింగ్ ఆపరేటర్‌లను స్ట్రింగ్‌లను నిర్మించడానికి ఉపయోగిస్తారు, స్ట్రింగ్ యొక్క భాగాలను మార్పుచేసి డేటాను వేరియబుల్స్‌లో నిక్షిప్తం చేయుటకు.

Syntax:

("String to be display with %val1 and %val2" %(val1, val2))

ఉదాహరణ

```
name = "Rajarajan"
mark = 98
print ("Name: %s and Marks: %d" %(name,mark))
```

అవుట్‌పుట్

```
Name: Rajarajan and Marks: 98
```

8.7 అక్షరాలను ఆకృతీకరించడం (Formatting characters)

అక్షరాల అమరిక	ఉపయోగము
%c	అక్షరం
%d (లేదా) %i	దశాంశ పూర్ణసంఖ్య
%s	స్ట్రింగ్
%u	గుర్తించని దశాంశ పూర్ణసంఖ్య
%o	ఆక్టల్ పూర్ణాంకము
%x లేదా %X	హెక్సాడెసిమల్ పూర్ణాంకము (లోయర్ కేస్ x, a-f ను సూచిస్తుంది అప్పర్ కేస్ X, A-F ను సూచిస్తుంది.)
%e లేదా %E	ఘాతాంక సంజ్ఞామానము
%f	ఫ్లోటింగ్ పాయింట్ సంఖ్యలు
%g లేదా %G	ఫ్లోటింగ్ పాయింట్లలలో చిన్న సంఖ్యలు లేదా ఘాతాంక సంజ్ఞామానము.

12th_CS_Telugu_Medium_Chapter_8.indd 123 20-02-2023 16:21:52

పైథాన్ లో తప్పించుకొను వరుసలు (Escape sequence in python)

పైథాన్‌లో తప్పించుకొను వరుసలు (Escape sequences) బ్యాక్ స్లాష్‌తో మొదలగును మరియు వీటిని, భిన్నంగా అర్థంచేసుకోవచ్చు. మీరు స్ట్రింగ్‌ను సూచించడానికి ఒకే కోట్‌ను ఉపయోగించినపుడు, స్ట్రింగ్‌లోని అన్ని సింగిల్ కోడ్స్ తప్పించుకోవాలి. డబుల్‌కోడ్స్‌కు ఈ నియమం వర్తిస్తుంది,

ఉదాహరణ

```
#సింగిల్ కోట్‌లో ట్రిపుల్ కోట్‌ను ప్రదర్శించడానికి ట్రిపుల్ కోట్స్‌తో స్ట్రింగ్
>>> print ('''They said, "What's there?"''')
        They said, "What's there?"
#తప్పించుకొను వరుసను ఉపయోగించి సింగిల్ కోట్స్‌తో స్ట్రింగ్‌ను ప్రదర్శించడానికి సింగిల్ కోట్ లోపల
స్ట్రింగ్
>>> print ('They said, "What\'s there?"')
        They said, "What's there?"
#తప్పించుకొను వరుసను ఉపయోగించి సింగిల్ కోట్‌తో స్ట్రింగ్‌ను ప్రదర్శించడానికి డబుల్ కోట్‌తో స్ట్రింగ్
>>> print ("They said, \"What's there?\"")
        He said, "What's there?"
```

పైథాన్ మద్దతు ఉన్న తప్పించుకొను వరుసలు (Escape sequences supported by python):

తప్పించుకొను వరుస	వివరణ
\newline	బ్యాక్‌స్లాష్ మరియు కొత్త పంక్తిని విస్మరించడం
\\	బ్యాక్ స్లాష్
\'	సింగిల్ కోట్
\"	డబుల్ కోట్
\a	ASCII గంట శబ్దం
\b	ASCII బ్యాక్‌స్పెస్
\f	ASCII ఫాంమ్ ఫీడ్
\n	ASCII లైన్ ఫీడ్
\r	ASCII క్యారేజ్ రిటర్న్
\t	ASCII క్షితి సమాంతర ట్యూబ్
\v	ASCII లంబ ట్యాగ్
\ooo	విలువ 000 తో అక్షం
\xHH	హెక్సాడెసిమల్ విలువ

8.8 ఫార్మేట్ format( ) (ఫంక్షన్)

స్ట్రింగ్‌తో ఉపయోగించిన ఫార్మేట్ ( ) function చాలా బహుముఖ మరయు శక్తిమంతమైన ప్రమేయం, కుండలీకరణములను (curly braces) { } ను placeholders గా లేదా పున:స్థాపన fields లుగా ఉపయోగిస్తారు. ఇది ఫార్మేట్ ( ) ఫంక్షన్‌లతో పాటు భర్తీ, చేయబడతాయి.

12th_CS_Telugu_Medium_Chapter_8.indd 124 20-02-2023 16:21:52

ఉదాహరణ

```
num1=int (input("Number 1: "))
num2=int (input("Number 2: "))
print ("The sum of { } and { } is { }".format(num1, num2,(num1+num2)))
```

అవుట్‌పుట్

```
Number 1: 34
Number 2: 54
The sum of 34 and 54 is 88
```

8.9 అంతర్నిర్మిత స్ట్రింగ్ విధులు (Built-in String functions):

స్ట్రింగ్‌ను మార్చడానికి పైథాన్ క్రింది అంతర్నిర్మిత ఫంక్షన్‌లకు మద్దతిస్తుంది.

Syntax	వివరణ	ఉదాహరణ

len(str)	స్ట్రింగ్ పొడవును (అక్షరాల సంఖ్యను) తిప్పి పంపును	>>> A="Corporation"
		>>> print(len(A))
		11

capitalize( )	స్ట్రింగ్‌లోని మొదటి అక్షరమును పెద్ద అక్షరముగా మార్చడానికి ఉపయోగపడును.	>>> city="chennai"
		>>> print(city.capitalize())
		Chennai

center(width, fillchar)	చెప్పబడిన స్ట్రింగ్ యొక్క మొత్తం పొడవును ఇవ్వబడిన స్ట్రింగ్ మధ్యలో వెలువడును మరియు అక్షరములు లేని వరుసలలో ఫిల్‌చర్ .... చే పూరింపబడును	>>> str1="Welcome"
		>>> print(str1.center(15,'*'))
		******Welcome*******

find(sub[, start[, end]])	ఇచ్చిన స్ట్రింగ్‌లోని ఉపస్ట్రింగ్ యొక్క మొదటి సంఘటన, శోధించుటకు ఈ ఫంక్షన్ ఉపయోగపడును. ఈ ఉపస్ట్రింగ్ (string) ప్రారంభమయ్యే సూచికపై తిరిగి వస్తుంది. స్ట్రింగ్‌లో జరగకపోతే ఇది −1 నుండి (substring) తిరిగివస్తుంది.	>>>str1='mammals' >>>str1.find('ma') ***0*** ***On omitting the start parameters, the function starts the search from the beginning.*** >>>str1.find('ma',2) 3 >>>str1.find('ma',2,4) -1 ***Displays -1 because the substring could not be found between the index 2 and 4-1.*** >>>str1.find('ma',2,5) 3

12th_CS_Telugu_Medium_Chapter_8.indd 125 20-02-2023 16:21:52

Syntax	వివరణ	ఉదాహరణ
isalnum( )	ఒక స్ట్రింగ్‌లో కేవలం అక్షరాల మరియు అంకెలు మాత్రమే ఉంటే ఒప్పు అని త్రిప్పి పంపుతుంది. ఒక వేళ స్ట్రింగ్‌లో ప్రత్యేక అక్షరాలు (_, @, #, *,) ఉన్నట్లయితే తప్పు అని త్రిప్పి పంపును	>>>str1='Save Earth' >>>str1.isalnum() ***False*** The function returns False as space is an alphanumeric character. >>>'Save1Earth'.isalnum() ***True***
isalpha( )	స్ట్రింగ్‌లో అక్షరాలు మాత్రం ఉన్నట్లు అయిన సరి అని త్రిప్పి పంపును. లేని యెడల తప్పు అని త్రిప్పి పంపును.	>>>'Click123'.isalpha() ***False*** >>>'python'.isalpha( ) ***True***
isdigit( )	స్ట్రింగ్‌లో అంకెలు మాత్రమే ఉన్నట్లు అయితే సరి అని తిరిగివస్తుంది. లేని యెడల తప్పు అని తిరిగివస్తుంది.	>>> str1='Save Earth' >>>print(str1.isdigit( )) ***False***
lower( )	దిగువ వరుసలో ఉన్న అక్షరాల్నింటితో వాస్తవ స్ట్రింగ్‌ను త్రిప్పి పంపును.	>>>str1='SAVE EARTH' >>>print(str1.lower()) ***save earth***
islower ( )	ఇవ్వబడిన స్ట్రింగ్ క్రింది వరుసలో ఉన్నచో సరి అని తిరిగి వస్తుంది.	>>> str1='welcome' >>>print (str1.islower( )) ***True***
isupper ( )	ఇవ్వబడిన స్ట్రింగ్ ఎగువ వరుసలో ఉన్నచో సరి అని తిరిగి పంపును.	>>> str1='welcome' >>>print (str1.isupper( )) ***False***
upper ( )	వాస్తవ స్ట్రింగ్‌ను పై వరుసలో గల అన్ని అక్షరాలతో తిరిగి పంపును.	>>> str1='welcome' >>>print (str.upper( )) ***WELCOME***

title ( )	టైటిల్ కేసులో ఉన్న స్ట్రింగ్‌ను త్రిప్పి పంపును.	>>> str1='education department'
		>>> print(str1.title())
		Education Department

swap case ( )		>>> str1="Tamil Nadu"
		>>> print (strl.swap case())
		Tamil Nadu

12th_CS_Telugu_Medium_Chapter_8.indd 126 20-02-2023 16:21:52

Syntax	వివరణ	ఉదాహరణ

count(str, beg, end)	ఇవ్వబడిన స్ట్రింగ్ పరిధిలో ఉన్న ఉపస్ట్రింగ్‌లును త్రిప్పి పంపుతుంది. ఉప ఒకే అక్షరము అయి కూడా ఉండవచ్చును. పరిధి వాదనలు ఒక ఐచ్ఛికము మాత్రమే. పైథాన్ మొత్తము స్ట్రింగ్‌ను వెతుకుతుంది. వెతకడం అనునది ఒక సున్నితమైన అంశంగా గుర్తించండి.	>>> str1="Raja Raja Chozhan"
		>>> print(str1.count('Raja'))
		2
		>>> print(str1.count('r'))
		0
		>>> print***(str1.count('R'))***
		2
		>>> print(str1.count('a'))
		5
		>>> print(str1.count('a',0,5))
		2
		>>> print(str1.count('a',11))
		1

ord(char)	ఇవ్వబడిన అక్షరం యొక్క ASCII విలువను త్రిప్పి పంపును.	>>> ch = 'A'
		>>> print(ord(ch))
		65
		>>> print(***ord('B'))***
		66

chr(ASII)	ఇవ్వబడిన ASCII చే ప్రాతిపదించబడిన అక్షరమును త్రిప్పి పంపును.	>>> ch=97
		>>> print(chr(ch))
		a
		>>> print(chr(87))
		W

8.10 సభ్యత్వ ఆపరేటర్లు (Membership Operators)

ఇచ్చిన స్ట్రింగ్ మరొక స్ట్రింగ్‌లో ఉందో లేదో తెలుసుకొనుటకు '**in**' మరియు '**not in**' ఆపరేటర్లలోని తీగలను ఉపయోగించవచ్చు. కాబట్టి ఈ ఆపరేటర్‌లను ఆపరేటర్ల సభ్యత్వం అంటారు.

ఉదాహరణ

```
str1=input ("Enter a string: ")
str2="chennai"
if str2 in str1:
        print ("Found")
else:
        print ("Not Found")
```

అవుట్‌పుట్ : 1

```
Enter a string: Chennai G HSS, Saidapet
Found
```

అవుట్‌పుట్ : 2

```
Enter a string: Govt G HSS, Ashok Nagar
Not Found
```

ఉదాహరణ 8.11 స్ట్రింగ్‌ను ఉపయోగించు ప్రోగ్రామ్‌లు :

ఉదాహరణ 8.11.1 : ఇచ్చిన స్ట్రింగ్ పాలండ్రోమ్, కాదా తనిఖీ చేసే ప్రోగ్రామ్.

```
str1 = input ("Enter a string: ")
str2 = ' '
index=-1
for i in str1:
        str2 += str1[index]
        index -= 1
print ("The given string = { } \n The Reversed string = { }".format(str1, str2))
if (str1==str2):
        print ("Hence, the given string is Palindrome")
else:
        print ("Hence, the given is not a palindrome")
```

అవుట్‌పుట్ : 1

```
Enter a string: malayalam
The given string = malayalam
The Reversed string = malayalam
Hence, the given string is Palindrome
```

అవుట్‌పుట్ : 2

```
Enter a string: welcome
The given string = welcome
The Reversed string = emoclew
Hence, the given string is not a palindrome
```

ఉదాహరణ 8.11.2: క్రింది నమూనాను ప్రదర్శించడానికి ప్రోగ్రామ్

```
*
* *
* * *
* * * *
* * * * *
str1=' * '
i=1
while i<=5:
        print (str1*i)
        i+=1
```

అవుట్‌పుట్

```
*
* *
* * *
* * * *
* * * * *
```


12th_CS_Telugu_Medium_Chapter_8.indd 128 20-02-2023 16:21:52

ఉదాహరణ 8.11.3 : ఇవ్వబడిన స్ట్రింగ్‌లో అచ్చులు మరియు హల్లుల సంఖ్యను ప్రదర్శించే ప్రోగ్రామ్

```
str1=input ("Enter a string: ")
str2="aAeEiIoOuU"
v,c=0,0
for i in str1:
        if i in str2:
                v+=1
        elif i.isalpha():
                c+=1
print ("The given string contains { } vowels and { } consonants".format(v,c))
```

అవుట్‌పుట్

```
Enter a string: Tamilnadu School Education
The given string contains 11 vowels and 13 consonants
```

ఉదాహరణ 8.11.4 : అబెసిడిరియన్ సిరీస్‌ను సృష్టించే ప్రోగ్రామ్ (అబెసిడిరియన్ అక్షరాల క్రమంలో కనిపించే మూలకాల జాబితాను సూచిస్తుంది.)

```
str1="ABCDEFGH"
str2="ate"
for i in str1:
        print ((i+str2),end='\t')
```

అవుట్‌పుట్

```
Aate    Bate    Cate    Date    Eate    Fate    Gate    Hate
```

ఉదాహరణ 8.11.5 : వినియోగదారు నుండి స్ట్రింగ్‌ను అంగీకరించే ప్రోగ్రామ్ దాని నుండి అచ్చులను తీసివేసిన తరువాత మునుపటి స్ట్రింగ్‌నే ప్రదర్శిస్తుంది.

```
def rem_vowels(s):
        temp_str=''
        for i in s:
                if i in "aAeEiIoOuU":
                        pass
                else:
                        temp_str+=i
        print ("The string without vowels: ", temp_str)
str1= input ("Enter a String: ")
rem_vowels (str1)
```

అవుట్‌పుట్

```
Enter a String: Mathematical fundations of Computer Science
The string without vowels:  Mthmtcl fndtns f Cmptr Scnc
```


12th_CS_Telugu_Medium_Chapter_8.indd 129 20-02-2023 16:21:52

ఉదాహరణ 8.11.6 : ఇవ్వబడిన స్ట్రింగ్‌లోని అక్షరం ప్రదర్శనమగు సంఖ్యను లెక్కించే ప్రోగ్రామ్

```
def count(s, c):
        c1=0
        for i in s:
                if i == c:
                        c1+=1
        return c1
str1=input ("Enter a String: ")
ch=input ("Enter a character to be searched: ")
cnt=count (str1, ch)
print ("The given character {} is occurs {} times in the given string".format(ch,cnt))
```

అవుట్‌పుట్

```
Enter a String: Software Engineering
Enter a character to be searched: e
The given character e is occurs 3 times in the given string
```

గుర్తించుకొనవలసిన అంశాలు

- పైథాన్‌లో స్ట్రింగ్ ఒకరకమైన డేటా టైపు అని గుర్తించుకొండి.
- స్ట్రింగ్ డేటా రకం స్థిరంగా ఉంటుంది. అనగా స్ట్రింగ్ నిర్వచించబడిన తరువాత మనము దానిని రన్ చేసే సమయంలో మార్చలేము.
- మూడు కోటేషన్ మార్కులలో స్ట్రింగ్‌లను నిర్వచించవచ్చు. ఇది బహుళ పంక్తి స్ట్రింగ్‌లను సృష్టించడానికి సహాయపడుతుంది.
- స్ట్రింగ్‌లో, పైథాన్ సబ్‌స్క్రిప్ట్ అని పిలవబడే ప్రతి అక్షరానికి సూచిక విలువను కేటాయిస్తుంది.
- సబ్‌స్క్రిప్ట్ అనుకూల లేదా ప్రతికూల పూర్ణాంక సంఖ్యలు కావచ్చు.
- స్లైస్ (Slice) ఒక ప్రధాన స్ట్రింగ్ యొక్క సబ్‌స్ట్రింగ్
- స్లైసింగ్ ఆపరేషన్‌లో స్లైడ్ మూడవ వాదన.
- తప్పించుకునే సన్నివేశాలు బ్యాక్‌స్లాష్‌తో మొదలవుతుంది మరియు దీనిని భిన్నంగా అర్థంచేసుకొనవచ్చును.
- స్ట్రింగ్‌లను ఉపయోగించి ఫార్మాట్ ( ) ఫంక్షన్ స్ట్రింగ్‌లను ఆకృతీకరించడానికి ఉపయోగించే బహుముఖ మరియు శక్తివంతమైన ఫంక్షన్.
- ఇవ్వబడిన స్ట్రింగ్ మరొక స్ట్రింగ్‌లో ఉందో లేదో తెలుసుకోవడానికి ఇన్ మరియు నాట్ ఇన్ ('not in') ఆపరేటర్‌లను స్ట్రింగ్‌లలో ఉపయోగించవచ్చు.



ప్రయోగపద్ధతి (Hands on Experience)

1. స్ట్రింగ్ యొక్క పొడవును కనుగొనడానికి పైథాన్ ప్రోగ్రామ్ రాయండి.

2. ఇచ్చిన స్ట్రింగ్‌లోని ప్రతి పదం ఎన్నిసార్లు కనిపించునో లెక్కించడానికి ఒక ప్రోగ్రామ్ రాయండి.

12th_CS_Telugu_Medium_Chapter_8.indd 130 20-02-2023 16:21:53

3. స్ట్రింగ్‌లోని అన్ని పంక్తులకు ఉపసర్గ వచనాన్ని జోడించడానికి ప్రోగ్రామ్‌ను రాయండి.

4. కుడివైపున '*' తో పూర్ణాంకాలను ముద్రించడానికి ఒక ప్రోగ్రామ్ రాయండి.

5. ఇచ్చిన స్ట్రింగ్‌కు mirror image సృష్టించడానికి ఒక ప్రోగ్రామ్ రాయండి.
ఉదాహరణకు, "Wel"= "lew"

6. స్ట్రింగ్‌లోని గివ్ క్యారెక్టర్ యొక్క అన్ని సంఘటనలను తొలగించడానికి ప్రోగ్రామ్ రాయండి.

7. += ఆపరేటర్లను ఉపయోగించకుండా మరొక స్ట్రింగ్‌కు స్ట్రింగ్‌ను జోడించడానికి ప్రోగ్రామ్‌ను వ్రాయండి.

8. రెండు స్ట్రింగ్‌లను మార్చుకోవడానికి ఒక ప్రోగ్రామ్ రాయండి.

9. పున:స్థాపన ( )ను ఉపయోగించకుండా స్ట్రింగ్‌ను మరొక స్ట్రింగ్‌తో భర్తీ చేయడానికి ప్రోగ్రామ్‌ను వ్రాయండి.

10. ఇచ్చిన స్ట్రింగ్‌లో అక్షరాలు, పదాలు మరియు పంక్తుల సంఖ్యను లెక్కించడానికి ఒక ప్రోగ్రామ్ రాయండి.

మూల్యాంకనం

F9P0Q

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. క్రింది వానిలో పైథాన్ అవుట్‌పుట్ ఏది?
```
str1="TamilNadu"
print(str1[::-1])
```
(అ) తమిళనాడు (ఆ) తళమి (ఇ) నాడు తమిళ్ ఈ) డనామిళత

2. క్రింది కోడ్ యొక్క అవుట్‌పుట్ ఏది?
```
str1 = "Chennai Schools"
str1[7] = "-"
```
(అ) Chennai-Schools (ఆ) Chenna-School (ఇ) Type error (ఈ) Chennai

3. క్రింది వాటిలో అనుక్రమము కొరకు ఉపయోగించు ఆపరేటర్ ఏది?
(అ) + (ఆ) & (ఇ) * (ఈ) =

4. ట్రిపుల్ కోడ్స్‌లో స్ట్రింగ్‌లను నిర్వచించడానికి, సృష్టించడానికి ఏది అనుమతిస్తుంది?
(అ) ఒక పంక్తి స్ట్రింగ్ (ఆ) బహుళ పంక్తి స్ట్రింగ్ (ఇ) ద్విపంక్తి స్ట్రింగ్ (ఈ) బహుళ స్ట్రింగ్‌లు

5. పైథాన్‌లోని స్ట్రింగ్‌లు:
(అ) మార్పునకు వీలగునది (ఆ) మార్పునకు అంగీకరించునది
(ఇ) మార్పునకు అంగీకరించనది (ఈ) సౌకర్యవంతమైనది

12th_CS_Telugu_Medium_Chapter_8.indd 131 20-02-2023 16:21:53

6. క్రింది వానిలో ఏది విభజన ఆపరేటర్?

(అ) { } (ఆ) [ ] (ఇ) < > (ఈ) ( )

7. Stride అనునది క్రింది వానిలో దేనిని గుర్తిస్తుంది.

(అ) Slide ఆపరేషన్ యొక్క సూచిక విలువ

(ఆ) Slice ఆపరేషన్ యొక్క మొదటి విలువ

(ఇ) Slice ఆపరేషన్ యొక్క రెండవ విలువ

(ఈ) Slice ఆపరేషన్ యొక్క మూడవ విలువ

8. క్రింది వాటిలో ఏది ఘాతాంక సంజ్ఞామానమును ముద్రించడానికి క్రింది ఆకృతీకంగా అక్షరం ఉపయోగపడుతుంది.

(అ) %e (ఆ) %E (ఇ) %g (ఈ) %n

9. క్రింది వాటిలో ఏది placeholders గా లేదా భర్తీ ఫిల్డ్ గా ఉపయోగించబడుతుంది.

(అ) { } (ఆ) < > (ఇ) ++ (ఈ) ^^

10. స్ట్రింగ్ యొక్క సబ్‌స్క్రిప్ట్ అనునది

(అ) అనుకూల సంఖ్యలు (ఆ) ప్రతికూల సంఖ్యలు (ఇ) (అ) మరియు (ఆ) (ఈ) (అ) లేదా (ఆ)

భాగము -II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. స్ట్రింగ్ అనగానేమి?

2. పైథాన్‌లో స్ట్రింగ్‌ను మార్చుటకు వీలు అగునా?

3. పైథాన్‌లో స్ట్రింగ్‌ను నీవు ఎలా తొలగించెదవు?

4. క్రింది పైథాన్ కోడ్ యొక్క అవుట్‌పుట్ ఏమి?

```
str1 = “School”
print(str1*3)
```

5. విభజన (slicing) అనగానేమి?

12th_CS_Telugu_Medium_Chapter_8.indd 132 20-02-2023 16:21:53

భాగము -III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. క్రింది నమూనాను ప్రదర్శించుటకు పైథాన్‌లో ప్రోగ్రామ్ రాయండి.

```
C O M P U T E R
C O M P U T E
C O M P U T
C O M P U
C O M P
C O M
C O
C
```

2. క్రింది వాటిని ఉదాహరణలతో వివరించండి.

   (అ) capitalize( ) (ఆ) swapcase( )

3. క్రింది పైథాన్ ప్రోగ్రామ్ యొక్క అవుట్‌పుట్ ఏమి?

```
str1 = "welcome"
str2 = "to school"
str3=str1[:2]+str2[len(str2)-2:]
print(str3)
```

4. format( ) యొక్క ఉపయోగము ఏమి? ఉదాహరణ ఇవ్వండి.

5. పైథాన్‌లో count( ) గురించి గమనిక రాయండి.

భాగము -IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. పైథాన్‌లో ఉపయోగించు స్ట్రింగ్ ఆపరేటర్స్ గురించి సరియైన ఉదాహరణలతో వివరించండి.

Reference Books

1. *https://docs.python.org/3/tutorial/index.html*
2. *https://www.techbeamers.com/python-tutorial-step-by-step/#tutorial-list*
3. *Python programming using problem solving approach – Reema Thareja – Oxford University press.*
4. *Python Crash Course – Eric Matthes – No starch press, San Francisco.*

యూనిట్ III

అధ్యాయం 9

LISTS, TUPLES, SETS మరియు నిఘంటువు

అభ్యసనా లక్ష్యములు

ఈ అధ్యాయము చదివిన తర్వాత విద్యార్థులు తెలుసుకోదగినవి.

- పైథాన్‌లో List, Tuples, sets మరియు Dictionary మొదలగు వివిధ రకాల సమూహ జాబితాల ప్రాథమిక విషయములను అర్థము చేసుకొనుట.
- వివిధ రకములైన ఫంక్షన్‌లను ఉపయోగించి List, Tuples, sets మరియు నిఘంటువులలో పనిచేయుట.
- List, Tuples, sets మరియు Dictionaries (నిఘంటువు) లకు పైథాన్ ప్రోగ్రామ్‌లను ఉపయోగించి వ్రాయుట.
- List, Tuples, sets మరియు నిఘంటువులకు మధ్యగల సంబంధములను అర్థము చేసుకొనుట.

9.1 List యొక్క పరిచయము

పైథాన్ ప్రోగ్రామ్‌లో నాలుగు రకములైన సమూహ జాబితాలు ఉన్నవి. అవి List, Tuples, Set మరియు Dictionary. పైథాన్ ప్రోగ్రామ్‌లోని List ని సీక్వన్సీ దేటా టైప్ అందురు. strings లాగా ఇది ఒక క్రమానుగత శ్రేణిలోని విలువలను చదురపు బ్రాకెట్టులో అమర్చిన విలువలగును. List లోని ప్రతి విలువను ఒక మూలకము (element) అని చెప్పబడుచున్నది. ఇది ఏ రకమైన సమూహముగా అనగా సంఖ్యలు, కేరెక్టర్స్, strings మరియు నెస్టడ్ జాబితాగా, బిన్నములైన List లాగా ఏ విధముగానైనను ఉండవచ్చును. మూలకములు మార్చదగినదిగాను, అనగా మూలకములను మార్చి అమర్చుటకును, చేర్చుటకును లేక తీసి వేయుటకును సాధ్యమగును. ఒక్కొక్క మూలకము List లో ఒక ప్రత్యేకమైనటు వంటి బాగములో అమరియుండును. ఒక మూలకము ఉండు ప్రాంతమును గుర్తించుటకును లేక మార్చడానికి ఉపయోగిస్తారు. కాబట్టి List అనునది పదకొండవ తరగతిలో మీరు నేర్చుకున్న arrays లకు సమానమని గుర్తించుకోండి.

9.1.1 పైథాన్‌లో List ను ఏర్పరచడం (Create a List in Python)

పైథాన్‌లో సాధారణంగా List ను చదురపు బ్రాకెట్టును ఉపయోగించి ఏర్పరుస్తారు. List లోని మూలకములను చదురపు బ్రాకెట్టులో గుర్తిస్తారు. ఈ క్రింద ఇవ్వబడిన సిన్‌టేక్స్ లిస్ట్‌ను ఏర్పరచడాన్ని వివరిస్తుంది.

Syntax:

Variable = [element-1, element-2, element-3 …… element-n]

12th_CS_Telugu_medium_Chapter_9.indd 134 27-03-2023 10:39:50

ఉదాహరణ

```
Marks = [10, 23, 41, 75]
Fruits = [“Apple”, “Orange”, “Mango”, “Banana”]
MyList = [ ]
```

పై ఉదాహరణలో ఇవ్వబడిన Marks అనే జాబితా నాలుగు పూర్ణసంఖ్యల మూలకములు; రెండవ లిస్ట్‌లోని పండ్లలో నాలుగు రకాలైన పండ్ల మూలకములు; మూడవ లిస్ట్‌లోని మూలకములు ఒక శూన్య లిస్ట్ అగును. list లోని మూలకములు ఒకే విధమైన మూలకములుగా ఉండనవసరము లేదు. క్రింద పేర్కొనబడిన list లో పలుకరములైన మూలకములను కలిగియున్నది.

Mylist = [“Welcome”, 3.14, 10, [2, 4, 6]]

పైన తెలుపబడిన ఉదాహరణలో Mylist లోని మూలకములు వేర్వేరు మూలకములుగా ఉన్నవి. అందువలన దీనిని **“Nested List”** గా చెప్పబడుచున్నది.

Nested list అనునది మరియొక list ను మూలకముగా యుండు list అగును.

9.1.2 List మూలకాలను ప్రాప్యత (Accessing List elements)

పైథాన్‌లోని List లో ప్రతి విలువలు 0 (Zero) తో మొదలవుతుంది. లిస్ట్‌లోని మూలకములను తెలిసికొనుటకు Index value ను ఉపయోగిస్తారు. పైథాన్‌లోని విలువలు ఒక Integer సంఖ్య అగును. ఈ సంఖ్య అనునది ధనాత్మకముగాను లేక ఋణాత్మకముగాను ఉండవచ్చును.

ఉదాహరణ

Marks = [10, 23, 41, 75]

Marks	10	23	41	75
Index (Positive)	0	1	2	3
IndexNegative)	-4	-3	-2	-1

లిస్ట్‌లో ధనాత్మక విలువలు ఆరంభములోను (ముందువైపుగా) మరియు ఋణాత్మక విలువలు వెనుకు వైపుగా లెక్కించబడును.

లిస్ట్‌లోని మూలకమును తెలిసికొనుటకు, లిస్ట్ యొక్క పేరును తెలుసుకొనుటకు, దీనిలోని మూలకములను చదురపు బ్రాకెట్‌లో వ్రాయుదురు.

Syntax:

List_Variable = [E1, E2, E3 …… En]

print (List_Variable[index of a element])

12th_CS_Telugu_medium_Chapter_9.indd 135 27-03-2023 10:39:50

ఉదాహరణ (Accessing single element):

```
>>> Marks = [10, 23, 41, 75]
>>> print (Marks[0])
10
```

పైన పేర్కొనబడిన ఉదాహరణలో ప్రింట్ కమాండ్ ప్రింట్‌లో 10 output గాను, 10 యొక్క సూచిక '0'గాను ఉండును.

ఉదాహరణ: Accessing elements in revevrse order

```
>>> Marks = [10, 23, 41, 75]
>>> print (Marks[-1])
  75
```

గమనిక

ఋణాత్మక విలువ మూలకాలను తెలుసుకొనడానికి ప్రతికూల పద్ధతిని ఉపయోగిస్తారు.

(i) జాబితా నుండి అన్ని మూలకాలను ప్రాప్యత చేయడం (Accessing all elements of a list)

జాబితా నుండి అన్ని మూలకాలను ప్రాప్యత చేయడానికి లూప్స్ ఉపయోగపడతాయి. లూప్ యొక్క ప్రారంభ విలువ సున్నా అయి ఉండాలి. జీరో అనేది జాబితా యొక్క ప్రారంభసూచిక విలువ.

ఉదాహరణ

```
Marks = [10, 23, 41, 75]
i = 0
while i < 4:
        print (Marks[i])
        i = i + 1
```

అవుట్‌పుట్

```
10
23
41
75
```

పై ఉదాహరణలో మార్కుల జాబితా నాలుగు పూర్ణాంక మూలకాలను కలిగి ఉంటుంది. అంటే 10,23,41,75 ప్రతిమూలకం 0 నుండి సూచిక విలువను కలిగి ఉంటుంది. మూలకాల సూచిక విలువ వరుసగా 0,1,2,3 ఇక్కడ అన్ని అంశాలను చదవడానికి అయితే లూప్ ఉపయోగపడుతుంది. లూప్ యొక్క ప్రారంభ విలువ సున్నా, మరియు పరీక్ష పరిస్థితి (i < 4) పరీక్ష పరిస్థితి నిజం అయినంతవరకు లూప్ సంబంధిత అవుట్‌పుట్‌ను అమలుపరచి ప్రింట్ చేస్తుంది.

12th_CS_Telugu_medium_Chapter_9.indd 136 27-03-2023 10:39:50

మొదట పునరావృత సమయంలో **i** యొక్క విలువ సున్నా, ఇక్కడ కండీషన్ నిజమైత. ఇప్పుడు క్రింది స్టేట్‌మెంట్ ప్రింట్ Marks[**i**]) అమలు అవుతుంది. Marks[**i**] మూలకం యొక్క విలువ `10' ను ప్రింట్ చేస్తుంది.

10 తదుపరి స్టేట్‌మెంట్ (**i = i + 1**) యొక్క విలువను 0 నుండి 1 వరకు పెంచుతుంది. ఇప్పుడు ప్రవాహం పరీక్ష స్థితిని తనిఖీ చేయడానికి కంట్రోల్ షిఫ్ట్ స్టేట్‌మెంట్‌కు మారుతుంది. లూప్ యొక్క పరీక్ష స్థితి తప్పు అయ్యేవరకు మార్కుల జాబితా యొక్క మిగిలిన అంశాలను ముద్రించడానికి ప్రక్రియ పునరావృత మవుతుంది.

ఈ క్రింది పట్టిక లూప్ యొక్క అమలు మరియు ముద్రించ వలసిన విలువను చూపుతుంది.

పునరావృతము	i	while i < 4	print (Marks[i])	i = i + 1
1	0	0 < 4 True	Marks [0] = 10	0 + 1 = 1
2	1	1 < 4 True	Marks [1] = 23	1 + 1 = 2
3	2	2 < 4 True	Marks [2] = 41	2 + 1 = 3
4	3	3 < 4 True	Marks [3] = 75	3 + 1 = 4
5	4	4 < 4 False	--	--

(ii) రివర్స్ ఇండెక్సింగ్ (Reverse Indexing):

పైథాన్ జాబితా మూలకాలకు రివర్స్ లేదా నెగటివ్ ఇండెక్స్‌ను అనుమతిస్తుంది. ఈ విధంగా పైథాన్ సూచికను వ్యతిరేక క్రమంలో జాబితా చేస్తుంది. పైథాన్ −1 జాబితాలోని చివరి మూలకం యొక్క సూచిక విలువగా మరియు మునుపటి మూలకానికి −2 అనే విలువను ఉంచుతుంది. దీనిని రివర్స్ ఇండెక్సింగ్ అంటారు.

ఉదాహరణ

```
Marks = [10, 23, 41, 75]
i = -1
while i >= -4:
        print (Marks[i])
        i = i + -1
```

అవుట్‌పుట్

```
75
41
23
10
```

పై పైథాన్ కోడింగ్ పనిచేయు ప్రక్రియను పట్టిక చూపిస్తుంది.

12th_CS_Telugu_medium_Chapter_9.indd 137 27-03-2023 10:39:50

పునరావృతము	i	while i >= -4	print (Marks[i])	i = i + -1
1	-1	-1 >= -4 True	Marks[-1] = 75	-1 + (-1) = -2
2	-2	-2 >= -4 True	Marks[-2] = 41	-2 + (-1) = -3
3	-3	-3 >= -4 True	Marks[-3] = 23	-3 + (-1) = -4
4	-4	-4 >= -4 True	Marks[-4] = 10	-4 + (-1) = -5
5	-5	-5 >= -4 False	--	--

9.1.3 List జాబితా పొడవు (List Length)

పైథాన్‌లోని len( ) ఫంక్షన్ జాబితా పొడవును కనుకొనడానికి ఉపయోగించబడుతుంది. (అనగా, జాబితాలోని మూలకాల సంఖ్య) సాధారణంగా, జాబితా యొక్క ఈ మూలకాల వద్ద చదవడానికి len( ) ఫంక్షన్ పై పరిమితిని లూప్‌లో సెట్ చేయడానికి ఉపయోగిస్తారు. ఒక జాబితాలో మరొక జాబితాను మూలకం వలె కలిగి ఉంటే, len( ) ఆ మూలకాన్ని ఒకే మూలకాన్ని తిరిగిఇస్తుంది.

ఉదాహరణ : ఒకే మూలకాన్ని యాక్సెస్ చేస్తుంది.

```
>>> MySubject = [“Tamil”, “English”, “Comp. Science”, “Maths”]
>>> len(MySubject)
    4
```

ఉదాహరణ : లూప్‌ను ఉపయోగించి ఒక list లోని మూలకాలను ప్రదర్శించే ప్రోగ్రామ్

```
MySubject = ["Tamil", "English", "Comp. Science", "Maths"]
i = 0
while i < len(MySubject):
        print (MySubject[i])
        i = i + 1
```

అవుట్‌పుట్

```
Tamil
English
Comp. Science
Maths
```

9.1.4 ఫర్ లూప్ ఉపయోగించి మూలకాలను ప్రాప్యత చేయడం (Accessing elements using for loop)

List లోని అన్ని అంశాలను, ఒక్కొక్కటిగా యాక్సెస్ చేయుటకు పైథాన్‌లోని, ఫర్ లూప్ ఉపయోగించబడుతుంది. List లోని అన్ని అంశాలను ఒక్కొక్కటిగా యాక్సెస్ చేయండి. ఇది C++ వంటి ఇతర ప్రోగ్రామింగ్ భాషలోని ***for*** కీవర్డ్ లాంటిది.

Syntax:

```
for index_var in list:
    print (index_var)
```


12th_CS_Telugu_medium_Chapter_9.indd 138 27-03-2023 10:39:50

ఇక్కడ, ***index_var*** అనునది list లోని ప్రతి మూలకము యొక్క సూచిక విలువను సూచిస్తుంది. పైథాన్ ఆంగ్ల భాషలో చదివేవిధంగా స్టేట్‌మెంట్‌ను రీడ్ చేస్తుంది. *"For (every) element in (the list of) list and print (the name of the) list items"*

ఉదాహరణ

```
Marks=[23, 45, 67, 78, 98]
for x in Marks:
        print( x )
```

అవుట్‌పుట్

```
23
45
67
78
98
```

పై ఉదాహరణలో Marks list లో 5 మూలకాలు ఉన్నాయి. ప్రతి మూలకం 0 నుండి 4 వరకు సూచించబడుతుంది. పైథాన్ ఆంగ్ల భాషలో చదివే విధంగా for loop మరియు english స్టేట్‌మెంట్ "For (every) element (represented as x) in (the list of) Marks and print (the values of the) elements" ను print చేయును.

9.1.5 List లోని మూలకాలను మార్చడం (Changing list elements)

పైథాన్‌లో, list లోని మూలకాలను మార్చగలవు. అంటే వాటిని మార్చవచ్చు. సాధారణ అసైన్‌మెంట్ ఆపరేటర్ (=) ను ఉపయోగించడం ద్వారా list లోని మూలకం లేదా మూలకాల పరిధిని మార్చవచ్చు.

Syntax:

List_Variable [index of an element] = Value to be changed
List_Variable [index from : index to] = Values to changed

సూచిక పరిధి యొక్క ప్రారంభ సూచిక : ఇండెక్స్ టు ఎగువ పరిమితి పరిధిలో మినహాయించబడింది. ఉదాహరణకు, మీరు పరిధిని (0:5) సెట్ చేస్తే, పైథాన్ పరిధి మూలకం సూచికగా 0 నుండి 4 మాత్రమే పడుతుంది. అందువల్ల, మీరు 1 నుండి 4 వరకు మూలకాల పరిధిని నవీకరించాలనుకుంటే దానిని (1:5) గా పేర్కొనాలి.

12th_CS_Telugu_medium_Chapter_9.indd 139 27-03-2023 10:39:50

ఉదాహరణ 9.1. MyList లో ఒక విలువను నవీకరించడం / మార్చడం

```
MyList = [2, 4, 5, 8, 10]
print ("MyList elements before update... ")
for x in MyList:
    print (x)
MyList[2] = 6
print ("MyList elements after updation... ")
for y in MyList:
    print (y)
```

అవుట్‌పుట్

```
MyList elements before update...
2
4
5
8
10
MyList elements after updation...
2
4
6
8
10
```

ఉదాహరణ 9.2. పైథాన్ ప్రోగ్రామ్ విలువల శ్రేణిని నవీకరించడానికి / మార్చడానికి

```
MyList = [1, 3, 5, 7, 9]
print ("List Odd numbers... ")
for x in MyList:
    print (x)
MyList[0:5] = 2,4,6,8,10
print ("List Even numbers... ")
for y in MyList:
    print (y)
```

అవుట్‌పుట్

```
List Odd numbers...
1
3
5
7
9
List Even numbers...
2
4
6
8
10
```


12th_CS_Telugu_medium_Chapter_9.indd 140 27-03-2023 10:39:50

9.1.6 List లో మూలకాలను చేర్చుట (Adding more elements in a list)

పైథాన్‌లో, ఒక మూలకాన్ని జోడించడానికి append ( ) ఫంక్షన్ ఉపయోగించబడుతుంది. మరియు ఇప్పటికే ఉన్న జాబితాలో ఒకటి కంటె ఎక్కువ మూలకాలను జోడించడానికి extend () ఫంక్షన్ ఉపయోగపడుతుంది.

Syntax:

List.append (element to be added)
List.extend ([elements to be added])

జోడించ వలసినటువంటి అంశాలను extend ఫంక్షన్‌లో, ఫంక్షన్ యొక్క వాదనలుగా చదురపు బ్రాకెట్టులో బహుళ మూలకాలను పేర్కొనవలయును.

ఉదాహరణ

```
>>> Mylist=[34, 45, 48]
>>> Mylist.append(90)
>>> print(Mylist)
      [34, 45, 48, 90]
```

పై ఉదాహరణలో, Mylist మూడు అంశాలలో సృష్టించబడుతుంది. >>>**Mylist.append** (90) స్టేట్‌మెంట్, అదనపు విలువ 90 ఇప్పటికీ ఉన్న జాబితాలో చివరి మూలకము వలె చేర్చబడి ఉంది. ఈ క్రింది ప్రింట్ స్టేట్‌మెంట్ MyList జాబితాలోని అన్ని అంశాలను చూపిస్తుంది.

ఉదాహరణ

```
>>> Mylist.extend([71, 32, 29])
>>> print(Mylist)
      [34, 45, 48, 90, 71, 32, 29]
```

పై కోడ్, extend() ఫంక్షన్ బహుళ అంశాలను చేర్చడానికి ఉపయోగపడుతుంది. అదనపు అంశాలను చేర్చిన తరువాత జాబితా యొక్క అన్ని అంశాలను print స్టేట్మెంటు చూపుతుంది.

9.1.7 List లో మూలకాలను చొప్పించడం (Inserting elements in a list)

మీరు ఇప్పటికే నేర్చుకొన్నట్లుగా పైథాన్‌లో append( ) ఫంక్షన్ list లో మరిన్ని అంశాలను జోడించడానికి ఉపయోగించబడుతుంది. కాని ఇది list చివరిలోని అంశాలను కలిగి ఉంటుంది. మీరు కోరుకున్న స్థానంలో ఒక మూలకాన్ని చేర్చాలనుకుంటే మీరు insert ( ) ఫంక్షన్‌ను ఉపయోగించవచ్చు. జాబితా యొక్క ఏదైనా స్థానంలో ఒక మూలకాన్ని చొప్పించడానికి insert ( ) ఫంక్షన్ ఉపయోగపడుతుంది.

Syntax:

List.insert (position index, element)

ఉదాహరణ

```
>>> MyList=[34,98,47,'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan' ]
>>> print(MyList)
      [34, 98, 47, 'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan']
>>> MyList.insert(3, 'Ramakrishnan')
>>> print(MyList)
[34, 98, 47, 'Ramakrishnan', 'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan']
```

పై ఉదాహరణలో insert( ) ఫంక్షన్ మూలక విలువ 3 వద్ద క్రొత్త మూలకము 'Ramakrishnan' ను చొప్పించును అంటే 4 స్థానంలో ఉంటుంది. ఇప్పటికీ ఉన్న మూలకాల మధ్య ఒక నిర్ధిష్ట ప్రదేశంలో క్రొత్త మూలకాన్ని చొప్పించేటప్పుడు ఉన్న మూలకాలు ఒక స్థానాన్ని కుడివైపున మారుస్తాయి.

12th_CS_Telugu_medium_Chapter_9.indd 141 27-03-2023 10:39:50

9.1.8 List నుంచి మూలకాలను తొలగించుట (Deleting elements from a list)

List నుండి ఒక మూలకాన్ని తొలగించడానికి రెండు మార్గాలు ఉన్నాయి. **del** మరియు **remove()** ఫంక్షన్ ఎలిమెంట్లను తొలగించడానికి ఉపయోగించబడుతుంది. తెలిసిన అంశాలను తొలగించడానికి డెల్‌స్టేట్‌మెంట్ ఉపయోగించబడుతుంది, అయితే List యొక్క సూచిక తెలియకపోతే దానిని తొలగించడానికి **remove()** ఫంక్షన్ ఉపయోగించ బడుతుంది. మొత్తం జాబితాను తొలగించడానికి **del** స్టేట్‌మెంట్ ఉపయోగించవచ్చును.

Syntax:

del List [index of an element]
to delete a particular element
del List [index from : index to]
to delete multiple elements
del List
to delete entire list

ఉదాహరణ

```
>>> MySubjects = ['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> print (MySubjects)
        ['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> del MySubjects[1]
>>> print (MySubjects)
        ['Tamil', 'Telugu', 'Maths']
```

పై ఉదాహరణలో, **MySubjects** అను list నాలుగు మూలకాలతో సృష్టించబడింది. print statement list యొక్క అన్ని మూలకాలను చూపుతుంది. >>> **del MySubjects[1** స్టేట్‌మెంట్లో, ఇండెక్స్ విలువ 1 అయిన ఒక మూలకాన్ని తొలగిస్తుంది. క్రింది ముద్రణ జాబితా యొక్క మిగిలిన అంశాలను చూపుతుంది.

ఉదాహరణ

```
>>> del MySubjects[1:3]
>>> print(MySubjects)
        ['Tamil']
```

పై దానిలో >>> **del MySubjects[1:3]** list నుండి రెండవ మరియు మూడవ అంశాలను తొలగిస్తుంది. ఇండెక్స్ యొక్క ఎగువ పరిమితి చదరపు బ్రాకెట్లలో పేర్కొనబడింది. పైథాన్ చేత –1గా తీసుకోబడుతుంది.

12th_CS_Telugu_medium_Chapter_9.indd 142 27-03-2023 10:39:50

ఉదాహరణ

```
>>> del MySubjects
>>> print(MySubjects)
Traceback (most recent call last):
File "<pyshell#9>", line 1, in <module>
print(MySubjects)
NameError: name 'MySubjects' is not defined
```

ఇక్కడ >>> del MySubjects list MySubjects అనే list ను పూర్తిగా తొలిగించును. ఇప్పుడు మూలకములను ప్రింట్ చేయడానికి ప్రయత్నించినపుడు list నిర్వహించబడనందున పైథాన్ లోపము చూపిస్తుంది. అంటే MySubjects జాబితాను పూర్తిగా తొలగించబడింది.

ఇప్పటికే చెప్పినట్లుగా సూచిక విలువ తెలియకపోతే remove( ) ఫంక్షన్ ఒకటి లేదా అంతకంటే ఎక్కువ మూలకాలను తొలగించడానికి కూడా ఉపయోగించవచ్చు. ఇచ్చిన సూచిక విలువను ఉపయోగించి ఒక మూలకాన్ని తొలగించడానికి remove( ) ఫంక్షన్ కాకుండా pop( ) ఫంక్షన్ కూడా ఉపయోగించవచ్చును. pop( ) ఫంక్షన్ సూచిక ఇవ్వనిచో జాబితా యొక్క చివరి మూలకాన్ని తొలగించి, తిరిగి ఇస్తుంది.

జాబితాలోని అన్ని అంశాలను తొలగించడానికి clear( ) ఫంక్షన్ ఉపయోగించబడుతుంది. ఇది మూలకాలను మాత్రమే తొలగిస్తుంది. మరియు జాబితాను కలిగి ఉంటుంది. Del స్టేట్‌మెంట్ మొత్తం జాబితాను తొలగిస్తుంది.

Syntax:

```
List.remove(element)          # to delete a particular element
List.pop(index of an element)
List.clear( )
```

ఉదాహరణ

```
>>> MyList=[12,89,34,'Kannan', 'Gowrisankar', 'Lenin']
>>> print(MyList)
        [12, 89, 34, 'Kannan', 'Gowrisankar', 'Lenin']
>>> MyList.remove(89)
>>> print(MyList)
        [12, 34, 'Kannan', 'Gowrisankar', 'Lenin']
```

పై ఉదాహరణలో, MyList మూడు ఇంటీజర్స్ మరియు మూడు స్ట్రింగ్ ఎలిమెంట్స్‌తో సృష్టించబడింది. ఈ క్రింది ప్రింటో స్టేటోమింట్స్ అందుబాటులో ఉన్న అన్ని అంశాలను చూపిస్తుంది. >>> **MyList.remove(89)** లో list నుండి 89 మూలకాన్ని తొలగిస్తుంది. మరియు print స్టేట్మెంట్ మిగిలిన అంశాలను చూపుతుంది.

ఉదాహరణ

```
>>> MyList.pop(1)
      34
>>> print(MyList)
      [12, 'Kannan', 'Gowrisankar', 'Lenin']
```

పై కోడ్‌లో pop( ) ఫంక్షన్ ఒక నిర్ధిష్ట మూలకాన్ని దాని సూచిక విలువను ఉపయోగించి తొలగించడానికి ఉపయోగించబడుతుంది. మూలకం తొలగించబడిన వెంటనే pop( ) ఫంక్షన్ మూలకము తొలగించబడిందని చూపిస్తుంది. జాబితా నుండి ఒక మూలకాన్ని తొలగించడానికి pop( ) ఫంక్షన్ ఉపయోగపడుతుంది. Del స్టేట్‌మెంట్ బహుళ అంశాలను తొలగిస్తోందని గుర్తించుకోండి.

ఉదాహరణ

```
>>> MyList.clear( )
>>> print(MyList)
          [ ]
```

పై కోడ్‌లో clear( ) ఫంక్షన్ మూలకాలను మాత్రమే తొలగించి, list ను నిలుపుకొంటుంది. మీరు ఇప్పటికే క్లియర్ చేసిన జాబితాను ముద్రించడానికి ప్రయత్నించినపుడు ఖాళీ మూలకం లేకుండా ఖాళీ చదరపు బ్రాకెట్టు ప్రదర్శించబడుతుంది, అంటే list ఖాళీగా ఉంది.

9.1.9 List మరియు range ( ) ఫంక్షన్: (List and range ( ) function)

range పైథాన్‌లో విలువల శ్రేణిని రూపొందించడానికి ఉపయోగించే ఫంక్షన్, range( ) ఫంక్షన్ ఉపయోగించి మీరు విలువల శ్రేణితో జాబితాను సృష్టించవచ్చు. range( ) ఫంక్షన్‌కు మూడు వాదనలు ఉంటాయి.

range యొక్క సింటాక్స్ **( ) function:**

range (start value, end value, step value)

ఇక్కడ,

- **start value** – శ్రేణి ప్రారంభ విలువ, సున్నా డిపాల్ట్ ప్రారంభ విలువ.
- **end value** – శ్రేణి యొక్క ఎగువ పరిమితి; పైథాన్ ముగింపు విలువను ఎగువ పరిమితి–1 తీసుకొంటుంది.
- **step value** – ఇది ఒక ఐచ్ఛిక వాదన ఇది ఒక విభిన్న విరామాన్ని ఉత్పత్తి చేయడానికి ఉపయోగపడుతుంది.

12th_CS_Telugu_medium_Chapter_9.indd 144 27-03-2023 10:39:50

ఉదాహరణ : x పరిధిలో x కోసం 10 వరకు గల పుర్ణాంక సంఖ్యలను ఉత్పత్తి చేస్తుంది.

```
for x in range (1, 11):
    print(x)
```

అవుట్‌పుట్

```
1
2
3
4
5
6
7
8
9
10
```

ఉదాహరణ : పరిధిలో X కోసం మొదటి 10 సరి సంఖ్యలను ఉత్పత్తి చేస్తుంది.

```
for x in range (2, 11, 2):
    print(x)
```

అవుట్‌పుట్

```
2
4
6
8
10
```

(i) విలువల శ్రేణితో list ను సృష్టించడం (Creating a list with series of values)

range( ) ఫంక్షన్‌ను ఉపయోగించి మీరు విలువల శ్రేణితో list ను సృష్టించవచ్చు. range( ) ఫంక్షన్ ఫలితాన్ని జాబితాగా మార్చడానికి మాకు list అనే మరో ఫంక్షన్ అవసరం. list ఫంక్షన్ range( ) యొక్క ఫలితాన్ని జాబితాగా చేస్తుంది.

Syntax:

List_Varibale = list (range ( ))

గమనిక

ఫైథాన్‌లో list ( ) ను సృష్టించడానికి list ( ) ఫంక్షన్ కూడా ఉపయోగించబడుతుంది.

12th_CS_Telugu_medium_Chapter_9.indd 145 27-03-2023 10:39:51

ఉదాహరణ

```
>>> Even_List = list(range(2,11,2))
>>> print(Even_List)
[2, 4, 6, 8, 10]
```

పై కోడ్‌లో, list( ) ఫంక్షన్ range( ) యొక్క ఫలితాన్ని Even_List ఎలిమెంట్స్‌గా తీసుకుంటుంది. అందువల్ల (list) జాబితాలో మొదటి ఐదు సరిసంఖ్యల అంశాలు పొంది ఉన్నాయి.

అదేవిధంగా, మేము range( ) ఫంక్షన్‌ను ఉపయోగించి విలువల శ్రేణిని సృష్టించవచ్చు. ఈ క్రింది ఉదాహరణ మొదటి 10 సహజ సంఖ్యల వర్గములతో జాబితాను ఎలా సృష్టించాలో వివరిస్తుంది.

ఉదాహరణ : మొదటి 10 సహజ సంఖ్యల వర్గాలను ఉత్పత్తి చేస్తుంది.

```
squares = [ ]
for x in range(1,11):
    s = x ** 2
    squares.append(s)
print (squares)
```

పై ప్రోగ్రామ్‌లో, వర్గములు (squares) పేరుతో ఖాళీ జాబితా సృష్టించబడుతుంది. అప్పుడు, ఫర్ లూప్ range( ) ఫంక్షన్‌ను ఉపయోగించి 1 నుండి 10 వరకు సహజ సంఖ్యలను ఉత్పత్తి చేస్తుంది. లూప్ లోపల (x) యొక్క ప్రస్తుత విలువ శక్తి 2 కు పెంచబడుతుంది మరియు వేరియబుల్స్ నిల్వచేయబడుతుంది. వర్గపు యొక్క ప్రతి క్రొత్త విలువ "వర్గాలు" జాబితాకు జోడించబడుతుంది. చివరగా ప్రోగ్రామ్ ఈ క్రింది విలువలను అవుట్‌పుట్‌గా చూపిస్తుంది.

అవుట్‌పుట్

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

9.1.10 జాబితా గ్రహణ

కాంప్రహెన్షన్ అనేది ఒక నిర్ధిష్ట పరిస్థితిని సంతృప్తిపరచే మూలకాల యొక్క సున్నితత్వాన్ని సృష్టించే సరళమైన మార్గం.

Syntax:

List = [expression for variable in range]

ఉదాహరణ : జాబితా గ్రహణ భావనను ఉపయోగించి మొదటి 10 సహజ సంఖ్యల వర్గాలను సృష్టించడం.

```
>>> squares = [ x ** 2 for x in range(1,11) ]
>>> print (squares)
```

అవుట్‌పుట్

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```


12th_CS_Telugu_medium_Chapter_9.indd 146 27-03-2023 10:39:51

పై ఉదాహరణలో వ్యక్తీకరణలోని x ** 2 ప్రతిసారీ మళ్ళించినప్పుడు అంచనా వేయబడుతుంది. విలువల శ్రేణిని ఉత్పత్తి చేసే shortcut పద్ధతి ఇది.

9.1.11 ముఖ్యమైన ఇతర జాబితా ఫంక్షన్ (Other important list funcion)

ఫంక్షన్	వివరణ	సింటేక్స్	ఉదాహరణ
copy ( )	జాబితాలోని మూలకాలను కాపీ చేసి తిరిగి ఇస్తుంది.	List.copy( )	MyList=[12, 12, 36] x = MyList.copy() print(x) **అవుట్‌పుట్** [12, 12, 36]
count ( )	జాబితాలోని ఒకే విధమైనటువంట మూలకముల సంఖ్యలను తెలుపుతుంది.	List.count(value)	MyList=[12, 12, 36] x = MyList.count(12) print(x) **అవుట్‌పుట్** 2
index ( )	మొదటి మూలకము యొక్క పునరావృత సంఖ్య విలువను ఇస్తుంది.	List.index(element)	MyList=[12, 12, 36] x = MyList.index(12) print(x) **అవుట్‌పుట్** 1
reverse ( )	జాబితాలోని మూలకాల శ్రేణిని వ్యుత్క్రమమును త్రిప్పిపంపును.	List.reverse( )	MyList=[12, 23, 36] MyList.reverse() print(MyList) **అవుట్‌పుట్** [36, 23, 12]
sort ( )	జాబితాలోని మూలకములను క్రమబద్ధీకరించును.	List.sort(reverse=True|False, key=myFunc)	
రెండు వాదనలు ఐచ్ఛికములు. • రివర్స్‌ను ట్రూగా సెట్ చేయబడితే జాబితా సార్టింగ్ (My list) అవరోహణ క్రమంలో ఉంది. • ఆరోహణ డిపాల్ట్‌గా ఉంటుంది. • కీ = మైఫంక్ (“myFunc” - sorting) ప్రమాణాలు నిర్దేశించే వినియోగదారు నిర్వహించిన ఫంక్షన్ యొక్క పేరు ముద్రణ (మై లిస్ట్) **గమనిక :** క్రమబద్ధీకరణ ( ) అసలు జాబితాను ప్రభావితం చేస్తుంది.		MyList=['Thilothamma', 'Tharani', 'Anitha', 'SaiSree', 'Lavanya'] MyList.sort( ) print(MyList) MyList.sort(reverse=True) print(MyList) **అవుట్‌పుట్** ['Anitha', 'Lavanya', 'SaiSree', 'Tharani', 'Thilothamma'] ['Thilothamma', 'Tharani', 'SaiSree', 'Lavanya', 'Anitha']	

12th_CS_Telugu_medium_Chapter_9.indd 147 27-03-2023 10:39:51

max( )	ఒక జాబితాలోని విలువలలో అత్యధిక (పెద్ద) విలువను ఇచ్చును.	max(list)	MyList=[21,76,98,23] print(max(MyList)) **అవుట్‌పుట్** 98
min( )	ఒక జాబితాలోని విలువలలో అతి చిన్న విలువనిచ్చును.	min(list)	MyList=[21,76,98,23] print(min(MyList)) **అవుట్‌పుట్** 21
sum( )	ఒక జాబితాలోని విలువల మొత్తమును తెలియజేసి ఇచ్చును.	sum(list)	MyList=[21,76,98,23] print(sum(MyList)) **అవుట్‌పుట్** 218

9.1.12 జాబితాలో ఉపయోగించిన ప్రోగ్రామ్‌లు (Programs using List)

1 నుండి 20 వరకు సంఖ్యల జాబితాను సృష్టించే ఒక ప్రోగ్రామ్‌ను రాయండి ఇది 4 చే భాగించబడాలి.

```
divBy4=[ ]
for i in range(21):
        if (i%4==0):
                divBy4.append(i)
print(divBy4)
```

అవుట్‌పుట్

[0, 4, 8, 12, 16, 20]

ప్రోగ్రామ్ 2 : బ్రిక్స్ (BRICS)లో సభ్యులైన దేశాల జాబితాను నిర్వచించడానికి ఒక ప్రోగ్రామ్ రాయండి. ఆ దేశం బ్రిక్స్‌లో సభ్యుడా కాదా అని తనిఖీ చేయండి.

```
country=["India", "Russia", "Srilanka", "China", "Brazil"]
is_member = input("Enter the name of the country: ")
if is_member in country:
        print(is_member, " is the member of BRICS")
else:
        print(is_member, " is not a member of BRICS")
```

అవుట్‌పుట్

Enter the name of the country: India
India is the member of BRICS

అవుట్‌పుట్

Enter the name of the country: Japan
Japan is not a member of BRICS

12th_CS_Telugu_medium_Chapter_9.indd 148 27-03-2023 10:39:51

ప్రోగ్రామ్ 3 : పైథాన్ ప్రోగ్రామ్ ఆరు సబ్జెక్టుల మార్కులను చదవడం మరియు ప్రతి సబ్జెక్టులో సాధించిన మార్కులను ప్రింట్ చేయడం మరియు మొత్తం మార్కులను చూపించడం.

```
marks=[]
subjects=['Tamil', 'English', 'Physics', 'Chemistry', 'Comp. Science', 'Maths']
for i in range(6):
        m=int(input("Enter Mark = "))
        marks.append(m)
for j in range(len(marks)):
        print("{ }. { } Mark = { } ".format(j+1,subjects[j],marks[j]))
print("Total Marks = ", sum(marks))
```

అవుట్‌పుట్

```
Enter Mark = 45
Enter Mark = 98
Enter Mark = 76
Enter Mark = 28
Enter Mark = 46
Enter Mark = 15
  1. Tamil Mark = 45
  2. English Mark = 98
  3. Physics Mark = 76
  4. Chemistry Mark = 28
  5. Comp. Science Mark = 46
  6. Maths Mark = 15
  Total Marks =  308
```

ప్రోగ్రామ్ 4 : పైథాన్ ప్రోగ్రామ్ ఒక జాబితాలోని 5 వస్తువుల ధరలను చదివి, ఆపై ధరల మొత్తాన్ని ప్రదర్శిస్తుంది, అన్ని ధరల ఉత్పత్తి మరియు సగటు వస్తువులను కనుగొనండి.

```
items=[]
prod=1
for i in range(5):
        print ("Enter price for item { } : ".format(i+1))
        p=int(input())
        items.append(p)
for j in range(len(items)):
        print("Price for item { } = Rs. { }".format(j+1,items[j]))
        prod = prod * items[j]
print("Sum of all prices = Rs.", sum(items))
print("Product of all prices = Rs.", prod)
print("Average of all prices = Rs.",sum(items)/len(items))
```


12th_CS_Telugu_medium_Chapter_9.indd 149 27-03-2023 10:39:51

అవుట్‌పుట్

```
Enter price for item  1 :
5
Enter price for item  2 :
10
Enter price for item  3 :
15
Enter price for item  4 :
20
Enter price for item  5 :
25
Price for item  1 = Rs. 5
Price for item  2 = Rs. 10
Price for item  3 = Rs. 15
Price for item  4 = Rs. 20
Price for item  5 = Rs. 25
Sum of all prices = Rs. 75
Product of all prices = Rs. 375000
Average of all prices = Rs. 15.0
```

ప్రోగ్రామ్ 5 : సంవత్సరానికి 1 లక్ష కంటే ఎక్కువ సంపాదించే ఉద్యోగుల సంఖ్యను లెక్కించడానికి పైథాన్ ప్రోగ్రామ్. ఉద్యోగుల సంఖ్యపై నెలవారి జీతాల ఇవ్వబడ్డాయి .

```
count=0
n=int(input("Enter no. of employees: "))
print("No. of Employees",n)
salary=[]
for i in range(n):
    print("Enter Monthly Salary of Employee { } Rs.: ".format(i+1))
    s=int(input())
    salary.append(s)
for j in range(len(salary)):
    annual_salary = salary[j] * 12
    print ("Annual Salary of Employee { } is:Rs. { }".format(j+1,annual_salary))
    if annual_salary >= 100000:
        count = count + 1
print("{ } Employees out of { } employees are earning more than Rs. 1 Lakh per annum"
format(count, n))
```


12th_CS_Telugu_medium_Chapter_9.indd 150 27-03-2023 10:39:51

అవుట్‌పుట్

```
Enter no. of employees: 5
No. of Employees 5
Enter Monthly Salary of Employee 1 Rs.:
3000
Enter Monthly Salary of Employee 2 Rs.:
9500
Enter Monthly Salary of Employee 3 Rs.:
12500
Enter Monthly Salary of Employee 4 Rs.:
5750
Enter Monthly Salary of Employee 5 Rs.:
8000
Annual Salary of Employee 1 is:Rs. 36000
Annual Salary of Employee 2 is:Rs. 114000
Annual Salary of Employee 3 is:Rs. 150000
Annual Salary of Employee 4 is:Rs. 69000
Annual Salary of Employee 5 is:Rs. 96000
2 Employees out of 5 employees are earning more than Rs. 1 Lakh per annum
```

ప్రోగ్రామ్ 6 : 1 నుండి 10 పరిధిలో ఉన్న సంఖ్యల జాబితాను పునరుద్ఘాటించడానికి ఒక ప్రోగ్రామ్‌ను వ్రాయండి. తర్వాత దానిలోని అన్ని సరిసంఖ్యలను తొలగించి తుది జాబితా ముద్రణను ముద్రించండి.

```
Num = []
for x in range(1,11):
    Num.append(x)
print("The list of numbers from 1 to 10 = ", Num)

for index, i in enumerate(Num):
    if(i%2==0):
        del Num[index]
print("The list after deleting even numbers = ", Num)
```

అవుట్‌పుట్

```
The list of numbers from 1 to 10 =  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
The list after deleting even numbers =  [1, 3, 5, 7, 9]
```

ప్రోగ్రామ్ 7 : పెబోనిశ్రేణిని ఏర్పరచి దానిని జాబితాలో సేకరించుటకు కావలసిన ప్రోగ్రామ్‌ను తయారు చేయుము తర్వాత ఆ శ్రేణిలోని మొత్తమును కనుగొనుము.

```
a=-1
b=1
n=int(input("Enter no. of terms: "))
i=0
sum=0
Fibo=[]
while i<n:
    s = a + b
    Fibo.append(s)
    sum+=s
    a = b
    b = s
    i+=1
print("Fibonacci series upto "+ str(n) +" terms is : " + str(Fibo))
print("The sum of Fibonacci series: ",sum)
```

అవుట్‌పుట్

```
Enter no. of terms: 10
Fibonacci series upto 10 terms is : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
The sum of Fibonacci series:  88
```

9.2 టుపుల్స్

Tuples పరిచయము (Introduction to Tuples) :

టుపుల్స్ కమాతో వేరుచేయబడిన మరియు కుండలీకరణాలలో జత చేయబడిన అనేక విలువలను కలిగిఉంటుంది. టుపుల్స్ జాబితాకు సమానంగా ఉంటుంది, జాబితాలోని విలువలను మార్చవచ్చు కాని టుపుల్స్‌లో కాదు.

మీకు తెలుసా?

టుపుల్స్ అనే పదం లాటిన్ పదం నుండి ఉద్భవించింది. ఇది సంఖ్యల శ్రేణి యొక్క సంగ్రహణను సూచిస్తుంది.

single(1), double(2), triple(3), quadruple(4), quinTuples(5), sexTuples(6), sepTuples(7), ocTuples(8), ..., n-Tuples, ...,

9.2.1 జాబితాలో టుపుల్స్ యొక్క ప్రయోజనాలు (Advantages of Tuples over list):

1. జాబితా జాబితాలోని మూలకములు మార్చదగినవి. అయితే ఒక టుపుల్స్ యొక్క అంశాలు మారవు. ఇది టుపుల్స్ మరియు జాబితా మధ్య ఉన్న ప్రధాన వ్యత్యాసం.
2. జాబితాలోని మూలకాలు చదరపు బ్రాకెట్టులో ఉంటాయి. కాని ఒక టుపుల్ యొక్క మూలకాలు పారాంథిసిస్ చేత జతచేయబడతాయి.
3. టుపుల్స్‌ను మళ్ళించడం జాబితా కంటే వేగంగా ఉంటుంది.

12th_CS_Telugu_medium_Chapter_9.indd 152 27-03-2023 10:39:51

9.2.2 టుపుల్స్ సృష్టించడం (Creating Tuples)

టుపుల్స్ సృష్టించడం అనునది జాబితాకు సమానం. జాబితాలో మూలకాలు చదరపు బ్రాకెట్లలో నిర్వచించబడతాయి. అయితే టుపుల్స్‌లో అవి కుండలీకరణం ద్వారా జతచేయబడతాయి. టుపుల్ యొక్క మూలకాలను కుండలీకరణం లేకుండా కూడా నిర్వచించవచ్చు. కుండలీకరణాల్లో లేదా కుండలీకరణాలు లేకుండా నిర్వచించిన అంశాలు, దాని పనితీరులో తేడా ఉండదు.

Syntax:

 # Empty tuple

Tuple_Name = ( )

 # Tuple with n number elements

Tuple_Name = (E1, E2, E2 ……. En)

 # Elements of a tuple without parenthesis

Tuple_Name = E1, E2, E3 ….. En

ఉదాహరణ

```
>>> MyTup1 = (23, 56, 89, 'A', 'E', 'I', "Tamil")
>>> print(MyTup1)
        (23, 56, 89, 'A', 'E', 'I', 'Tamil')

>>> MyTup2 = 23, 56, 89, 'A', 'E', 'I', "Tamil"
>>> print (MyTup2)
        (23, 56, 89, 'A', 'E', 'I', 'Tamil')
```

(i) టుపుల్ ( ) ఫంక్షన్ ఉపయోగించి టుపుల్స్‌ను సృష్టించడం (Creating tuples using tuple( ) function)

tuple( ) ఫంక్షన్ జాబితా నుండి టుపుల్స్ సృష్టించడానికి ఉపయోగిస్తారు. మీరు ఒక టుపుల్‌ను సృష్టించినప్పుడు, జాబితా నుండి మూలకాలను చదరపు బ్రాకెట్లలో ఉండాలి.

Syntax:

Tuple_Name = tuple([list elements])

ఉదాహరణ

```
>>> MyTup3 = tuple( [23, 45, 90] )
>>> print(MyTup3)
        (23, 45, 90)
>>> type (MyTup3)
        <class 'tuple'>
```


12th_CS_Telugu_medium_Chapter_9.indd 153 27-03-2023 10:39:51

గమనిక

రకం : ఫంక్షన్ పైథాన్ వస్తువు యొక్క డేటా రకాన్ని తెలుసుకోవడానికి ఉపయోగించబడుతుంది.

(ii) సింగిల్ ఎలిమెంట్ టుపుల్‌ను సృష్టించడం (Creating Single element tuple)

ఒకే మూలకంతో టుపుల్‌ను సృష్టించేటప్పుడు, మూలకము చివర కామాను జోడించండి కామా లేనప్పుడు పైథాన్ మూలకాన్ని సాధారణ డేటా రకంగా పరిగణిస్తుంది. ఒక మూలకంతో టుపుల్‌ను సృష్టించడం "సింగిల్టన్" టుపుల్ అంటారు.

ఉదాహరణ

```
>>> MyTup4 = (10)
>>> type(MyTup4)
        <class 'int'>
>>> MyTup5 = (10,)
>>> type(MyTup5)
        <class 'tuple'>
```

9.2.3 టుపుల్‌లో విలువలను యాక్సెస్ చేస్తోంది (Accessing values in a Tuple)

జాబితా విధంగా, టుపుల్ యొక్క ప్రతి మూలకం సున్నా నుండి ప్రారంభమయ్యే సూచిక సంఖ్యను కలిగి ఉంటుంది. సూచిక సంఖ్యను ఉపయోగించడం ద్వారా టుపుల్ యొక్క అంశాలను సులభంగా యాక్సెస్ చేయవచ్చు.

ఉదాహరణ

```
>>> Tup1 = (12, 78, 91, "Tamil", "Telugu", 3.14, 69.48)
# to access all the elements of a tuple
>>> print(Tup1)
        (12, 78, 91, 'Tamil', 'Telugu', 3.14, 69.48)
#accessing selected elements using indices
>>> print(Tup1[2:5])
        (91, 'Tamil', 'Telugu')
#accessing from the first element up to the specified index value
>>> print(Tup1[:5])
        (12, 78, 91, 'Tamil', 'Telugu')
# accessing from the specified element up to the last element.
>>> print(Tup1[4:])
        ('Telugu', 3.14, 69.48)
# accessing from the first element to the last element
>>> print(Tup1[:])
        (12, 78, 91, 'Tamil', 'Telugu', 3.14, 69.48)
```


12th_CS_Telugu_medium_Chapter_9.indd 154 27-03-2023 10:39:51

9.2.4 టుపుల్స్‌ను నవీకరించడం మరియు తొలగించడం (Update and Delete Tuple)

ఒక టుపుల్స్ మార్పు లేనిదని మీకు తెలిసినట్లుగా టుపుల్స్‌లోని మూలకాలను మార్చలేము. టుపుల్స్‌లో విలువలను మార్చడానికి బదులుగా, రెండు టుపుల్స్‌లో చేర్చడం లేదా మొత్తం టుపుల్స్ తొలగించడం సాధ్యమే.

ఉదాహరణ

```
# రెండు టుపుల్స్ చేర్చడానికి ప్రోగ్రామ్
Tup1 = (2,4,6,8,10)
Tup2 = (1,3,5,7,9)
Tup3 = Tup1 + Tup2
print(Tup3)
```

అవుట్‌పుట్

```
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

మొత్తం టుపుల్స్‌ను తొలగించడానికి డెల్ (del) కమాండ్‌ను ఉపయోగిస్తారు.

Syntax:

del tuple_name

ఉదాహరణ

```
Tup1 = (2,4,6,8,10)
print("The elements of Tup1 is ", Tup1)
del Tup1
print (Tup1)
```

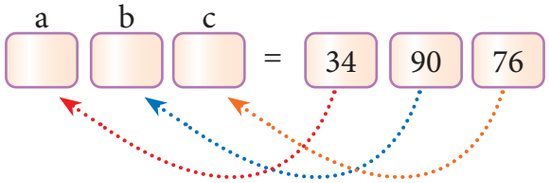
అవుట్‌పుట్

```
The elements of Tup1 is  (2, 4, 6, 8, 10)
Traceback (most recent call last):
File "D:/Python/Tuple Examp 1.py", line 4, in <module>
print (Tup1)
NameError: name 'Tup1' is not defined
```

గమనిక: ముద్రణ ప్రకటనపై కోడ్‌లో మూలకాలను ముద్రిస్తుంది. అప్పుడు డెల్ స్టేట్‌మెంట్ మొత్తము టుపుల్స్‌ను తొలగిస్తుంది. మీరు తొలగించిన టుపుల్‌ను ప్రింట్ చేయడానికి ప్రయత్నించినప్పుడు పైథాన్ లోపం చూపిస్తుంది.

9.2.5. టుపుల్ అసైన్‌మెంట్ చేయుట (Tuple Assignment)

టుపుల్ అసైన్‌మెంట్ ఒక శక్తివంతమైన లక్షణం. ఇది టుపుల్స్ వేరియబుల్ అనుమతిస్తుంది. అసైన్‌మెంట్ ఆపరేటర్ యొక్క ఎడమ వైపున అసైనమెంట్ యొక్క కుడివైపున ఉన్న విలువలకు కేటాయించబడుతుంది. ఒక్కొక్క విలువ వాటికి కేటాయించిన ప్రాంతములో నిలువ చేయబడుచున్నది.



ఉదాహరణ

```
>>> (a, b, c) = (34, 90, 76)
>>> print(a,b,c)
        34 90 76
# expression are evaluated before assignment
>>> (x, y, z, p) = (2**2, 5/3+4, 15%2, 34>65)
>>> print(x,y,z,p)
        4 5.666666666666667 1 False
```

గమనిక : మీరు ఒక టుపుల్స్‌కు విలువలను కేటాయించినపుడు రెండింటిలో ఉన్న విలువల సంఖ్య ఒకేలా ఉందని నిర్ధారించుకోండి లేకపోతే పైథాన్‌లో తప్పులు జరుగును.

9.2.6 టుపుల్స్‌లో బహుళ విలువలను తిరిగి ఇవ్వడం (Returning multiple values in Tuples)

ఒక సమయంలో ఒక ఫంక్షన్ వద్ద ఒక విలువను మాత్రమే తిరిగి ఇవ్వగలదు. కాని పైథాన్ ఒక ఫంక్షన్ నుండి ఒకటి కంటె ఎక్కువ విలువలను తిరిగి ఇస్తుంది. పైథాన్ బహుళ విలువలను సమూహపరుస్తుంది. మరియు వాటిని తిరిగి ఇస్తుంది.

ఉదాహరణ 9.7 లిస్ట్‌లో గరిష్ఠ మరియు కనీస విలువలను తిరిగి పొందుటకు ప్రోగ్రామ్.

```
def Min_Max(n):
    a = max(n)
    b = min(n)
    return(a, b)
Num = (12, 65, 84, 1, 18, 85, 99)
(Max_Num, Min_Num) = Min_Max(Num)
print("Maximum value = ", Max_Num)
print("Minimum value = ", Min_Num)
```

అవుట్‌పుట్

```
Maximum value =  99
Minimum value =  1
```


12th_CS_Telugu_medium_Chapter_9.indd 156 27-03-2023 10:39:51

9.2.7 నెస్టెడ్ టుపుల్స్ (Nested Tuples)

పైథాన్‌లో, మరొక టుపుల్ లోపల ఒక టుపుల్‌ను నిర్వచించవచ్చు దీనిని నెస్టెడ్ టుపుల్ అని పిలుస్తారు. ఒక నెస్టెడ్ టుపుల్‌లో, ప్రతి టుపుల్‌ను ఒక మూలకంగా పరిగణిస్తారు. ఒక సమూహ టుపుల్‌లోని అన్ని మూలకాలను ప్రాప్యత చేయడానికి for లూప్ ఉపయోగపడుతుంది.

ఉదాహరణ

```
Toppers = (("Vinodini", "XII-F", 98.7), ("Soundarya", "XII-H", 97.5),
                ("Tharani", "XII-F", 95.3), ("Saisri", "XII-G", 93.8))
for i in Toppers:
        print(i)
```

అవుట్‌పుట్

```
('Vinodini', 'XII-F', 98.7)
('Soundarya', 'XII-H', 97.5)
('Tharani', 'XII-F', 95.3)
('Saisri', 'XII-G', 93.8)
```

గమనిక

జాబితాలో ఉపయోగించిన అన్ని ఫంక్షన్‌లు టుపుల్స్‌కు కూడా వర్ధిస్తాయి.

9.2.8. టుపుల్స్‌ను ఉపయోగించి ప్రోగ్రామ్‌లు (Programs using Tuples)

ప్రోగ్రామ్1 : టుపుల్ అసైన్‌మెంట్ ఉపయోగించి రెండు విలువలను మార్చుకోవడానికి ఒక ప్రోగ్రామ్‌ను వ్రాయండి.

```
a = int(input("Enter value of A: "))
b = int(input("Enter value of B: "))
print("Value of A = ", a, "\n Value of B = ", b)
(a, b) = (b, a)
print("Value of A = ", a, "\n Value of B = ", b)
```

అవుట్‌పుట్

```
Enter value of A: 54
Enter value of B: 38
Value of A =  54
 Value of B =  38
Value of A =  38
 Value of B =  54
```

ప్రోగ్రామ్ 2 : ఒక వృత్తం యొక్క వైశాల్యం మరియు చుట్టుకొలతను తిరిగి ఇచ్చే ఫంక్షన్‌ను ఉపయోగించి ఒక ప్రోగ్రామ్‌ను వ్రాయండి. దీని వ్యాసార్ధం ఆర్గ్యుమెంట్‌గా ఆమోదించబడుతుంది. టుపుల్ అసైన్‌మెంట్ ఉపయోగించండి.

```
pi = 3.14
def Circle(r):
    return (pi*r*r, 2*pi*r)
radius = float(input("Enter the Radius: "))
(area, circum) = Circle(radius)
print ("Area of the circle = ", area)
print ("Circumference of the circle = ", circum)
```

అవుట్‌పుట్

```
Enter the Radius: 5
Area of the circle =  78.5
Circumference of the circle =  31.400000000000002
```

ప్రోగ్రామ్ 3 : ధనాత్మక మరియు ఋణాత్మక సంఖ్యల జాబితాను కలిగి ఉన్న ప్రోగ్రామ్‌ను వ్రాయండి.

```
Numbers = (5, -8, 6, 8, -4, 3, 1)
Positive = ( )
for i in Numbers:
    if i > 0:
        Positive += (i, )
print("Positive Numbers: ", Positive)
```

అవుట్‌పుట్

```
Positive Numbers:  (5, 6, 8, 3, 1)
```

9.3 సమితులు పరిచయం

పరిచయము (Introduction)

పైథాన్‌లో, సమితి మరొక రకమైన డేటా సేకరణ రకం. ఒక సెట్ అనేది నకిలీలు లేని మూలకాల యొక్క మార్చగల మరియు క్రమం లేని సేకరణ. అంటే సమితిలోని అంశాలు పునరావృతం కావు. సభ్యత్వ పరీక్ష మరియు నకిలీ అంశాలను తొలగించడానికి ఈ లక్షణం ఉపయోగించబడుతుంది.

12th_CS_Telugu_medium_Chapter_9.indd 158 27-03-2023 10:39:51

9.3.1 సమితిని సృష్టించడం (Creating a Set):

కామాతో వేరు చేయబడిన అన్ని మూలకాలను ఒక జత కుండలీకరణా గుర్తులో ఉంచడం ద్వారా సమితిని సృష్టించబడుతుంది. సెట్ ( ) ఫంక్షన్ పైథాన్‌లో సెట్లను సృష్టించడానికి కూడా ఉపయోగించవచ్చు.

Syntax:

Set_Variable = {E1, E2, E3 …….. En}

ఉదాహరణ

```
>>> S1={1,2,3,'A',3.14}
>>> print(S1)
        {1, 2, 3, 3.14, 'A'}

>>> S2={1,2,2,'A',3.14}
>>> print(S2)
        {1, 2, 'A', 3.14}
```

పై ఉదాహరణలో set S1 నకలు విలువలు లేకుండా వివిధ రకాల మూలకాలతో సృష్టించబడుతుంది. set S2 నకలు విలువలతో సృష్టించబడుతుంది. అయితే పైథాన్ నకిలీలలో ఒక మూలకాన్ని మాత్రమే అంగీకరిస్తుంది. అంటే పైథాన్ నకలు విలువను తీసివేసింది. ఎందుకంటే పైథాన్‌లోని సెట్‌లో నకిలీ అంశాలు ఉండవు.

గమనిక

మీరు సమితి నుండి మూలకాలను ముద్రించునపుడు పైథాన్ విలువలను వేర్వేరు క్రమంలో చూపిస్తుంది.

9.3.2 జాబితా లేదా టుపుల్ ఉపయోగించి సెట్‌ను సృష్టించడం (Creating Set using List or Tuple)

set( ) ఫంక్షన్‌ను ఉపయోగించి జాబితా లేదా టుపుల్‌ను సెట్‌గా మార్చవచ్చు. ఇది చాలా సులభమైన విధానం. మొదట మీరు List లేదా Tuple ను సృష్టించాలి. అప్పుడు దాని వేరియబుల్‌ను set( ) లో ప్రత్యామ్నాయం చేయండి. ఇది ఆర్గ్యుమెంట్‌గా పనిచేస్తుంది.

ఉదాహరణ

```
MyList=[2,4,6,8,10]
MySet=set(MyList)
print(MySet)
```

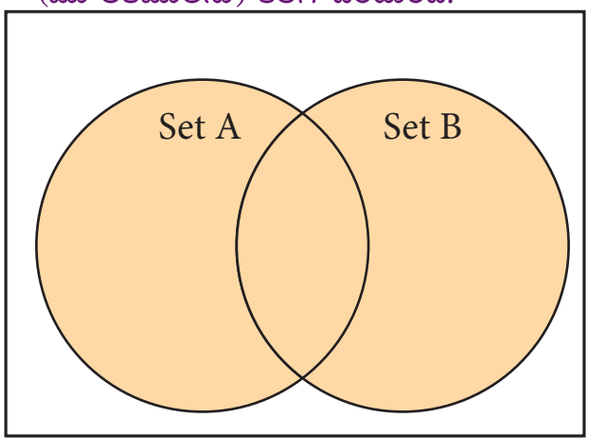
అవుట్‌పుట్

```
        {2, 4, 6, 8, 10}
```

9.3.3 సమితి ప్రక్రియలు (Set Operations):

గణితంలో మీరు నేర్చుకొన్నట్లుగా పైథాన్ కూడా సమ్మేకనం, ఖండన, వ్యత్యాసం మరియు సిమెట్రిక్ డిఫెరన్స్ వంటి సమితి ప్రక్రియలను మద్దతు ఇస్తుంది.

(i) యూనియన్ సమ్మేకనం : ఇది రెండు లేదా అంతకంటె ఎక్కువ సెట్ల నుండి అన్ని అంశాలను (మూలకములను) కలిగి ఉంటుంది.



పైథాన్‌లో, ఆపరేటర్ | రెండు సెట్‌ల యూనియన్ కు ఉపయోగిస్తారు. ఫంక్షన్ యూనియన్ union( ) పైథాన్‌లో రెండు సమితులను చేర్చడానికి కూడా ఉపయోగించబడుతుంది.

ఉదాహరణ : యూనియన్ చేర్పు చిహ్నమును ఉపయోగించి రెండు సమితులను చేర్చుటకు ప్రోగ్రామ్‌ను ఉపయోగించుట.

```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)
```

అవుట్‌పుట్

```
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

ఉదాహరణ : రెండు ప్రమేయములను ఉపయోగించి చేర్చుటకు ప్రోగ్రామ్‌ను చేయుట.

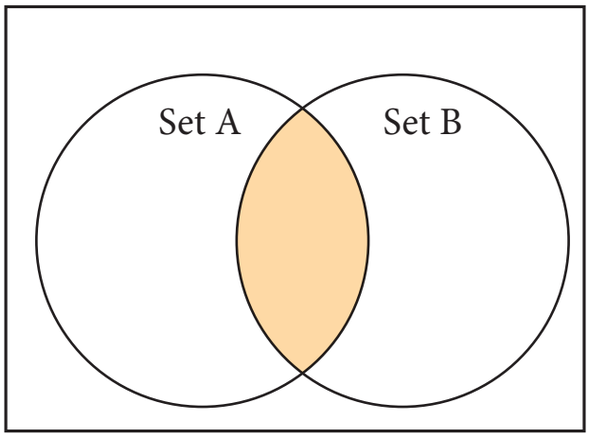
```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
set_U=set_A.union(set_B)
print(set_U)
```

అవుట్‌పుట్

```
{'D', 2, 4, 6, 8, 'B', 'C', 'A'}
```


12th_CS_Telugu_medium_Chapter_9.indd 160 27-03-2023 10:39:51

(ii) Intersection : ఇది రెండు సెట్లలోని సాధారణ అంశాలను కలిగి ఉంటుంది.



పైథాన్‌లో రెండు సమితులను ఖండన చేయుటకు ఆపరేటర్ & ఉపయోగించబడుతుంది పైథాన్‌లో **intersection( )** ఫంక్షన్ రెండు సమితులను ఖండన చేసే ప్రోగ్రామ్.

ఉదాహరణ : ఖండన ఆపరేటర్ ఉపయోగించి రెండు సెట్లను ఖండన చేసే ప్రోగ్రామ్.

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```

అవుట్‌పుట్

```
{'A', 'D'}
```

ఉదాహరణ : ఖండన ఫంక్షన్ ఉపయోగించి రెండు సెట్లను ఖండన చేసే ప్రోగ్రామ్.

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.intersection(set_B))
```

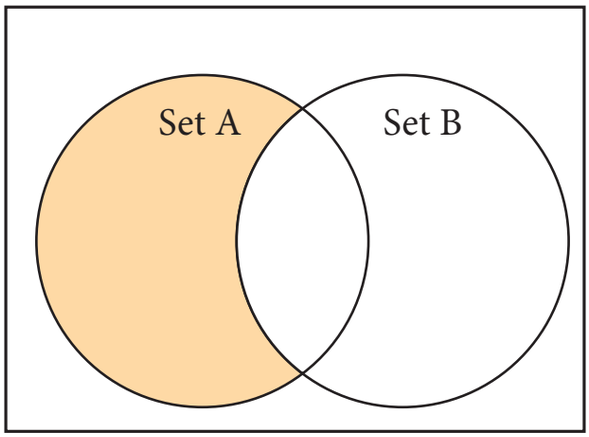
అవుట్‌పుట్

```
{'A', 'D'}
```


12th_CS_Telugu_medium_Chapter_9.indd 161 27-03-2023 10:39:51

(iii) వ్యత్యాసం (Difference):

దీని యందు మొదటి సెట్ నందలి అన్ని అంశములు (say set A) కాని రెండవ సెట్ నందు ఏమి ఉండవు (say set B) .



మైనస్ (-) ఆపరేటర్‌ను పైథాన్ నందలి సెట్ ఆపరేషన్ నందలి వ్యత్యాసములు కనుగొనుటకు ఉపయోగింతురు. (-) ఫంక్షన్ వ్యత్యాసము ( ) ఆపరేషన్ నందలి వ్యత్యాసము కనుగొనుటకు కూడా ఉపయోగిస్తారు.

ఉదాహరణ : మైనస్ ఆపరేటర్‌ను ఉపయోగించి రెండు సెట్ల మధ్య వ్యత్యాసమును ప్రోగ్రామ్ చేయుట.

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
```

అవుట్‌పుట్

{2, 4}

ఉదాహరణ : Difference డిపరెన్స్ ఫంక్షన్‌ను ఉపయోగించి రెండు సెట్ల మధ్య గల వ్యత్యాసముల ప్రోగ్రామ్ చేయుట.

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.difference(set_B))
```

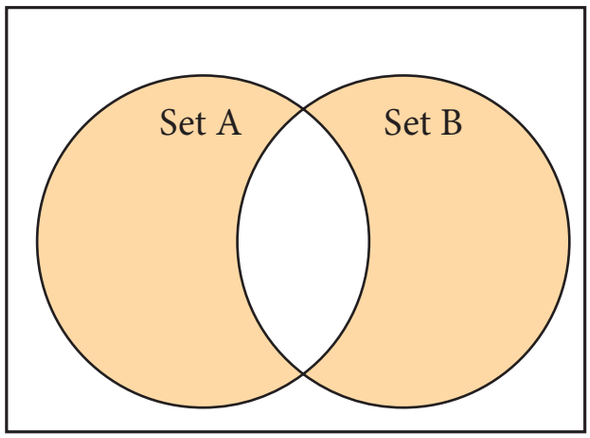
అవుట్‌పుట్

{2, 4}

12th_CS_Telugu_medium_Chapter_9.indd 162 27-03-2023 10:39:51

(iv) సిమెట్రిక్ వ్యత్యాసం (Symmetric difference):

ఇది రెండు సమితులలో ఉన్న అన్ని మూలకాలను కలిగి ఉంటుంది. (సమితి A మరియు సమితి B) కాని రెండు సమితులు సాధారణమైనది కాదు.



పైథాన్‌లో సిమిట్రిక్ డిఫెరెన్స్ సెట్ ఆపరేషన్‌కు కేరెట్ ( ) ఆపరేషన్ ఉపయోగపడుతుంది. ఫంక్షన్ సిమెట్రిక్ డిఫెరెన్స్ ( ) ఫంక్షన్ కూడా ఒకే ఆపరేషన్ చేయడానికి ఉపయోగపడుతుంది.

ఉదాహరణ : కేరెట్ ఆపరేటర్ సెట్ ఉపయోగించి రెండు సెట్ల యొక్క సిమెట్రిక్ వ్యత్యాసమునకు ప్రోగ్రామ్

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A ^ set_B)
```

అవుట్‌పుట్

```
{2, 4, 'B', 'C'}
```

ఉదాహరణ : సిమెట్రిక్ డిఫరెన్స్ ఫంక్షన్ సెట్ ఉపయోగించి రెండు సెట్ల తేడాకు ప్రోగ్రామ్.

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.symmetric_difference(set_B))
```

అవుట్‌పుట్

```
{2, 4, 'B', 'C'}
```


12th_CS_Telugu_medium_Chapter_9.indd 163 27-03-2023 10:39:51

9.3.4 సమితులను ఉపయోగించి ప్రోగ్రామ్‌లు (Programs using Sets):

ప్రోగ్రామ్ 1 : ప్రధాన సంఖ్యల సమితి మరియు సరిసంఖ్యల సమితిని ఉత్పత్తి చేసే ప్రోగ్రామ్‌ను వ్రాయండి, యూనియన్ (సమ్మేళనము), ఖండన, వ్యత్యాసం, సమరూప వ్యత్యాస కార్యకలాపాల ఫలితాన్ని ప్రదర్శించండి.

ఉదాహరణ

```
even=set([x*2 for x in range(1,11)])
primes=set()
for i in range(2,20):
        j=2
        f=0
        while j<=i/2:
                if i%j==0:
                        f=1
                j+=1
        if f==0:
                primes.add(i)
print("Even Numbers: ", even)
print("Prime Numbers: ", primes)
print("Union: ", even.union(primes))
print("Intersection: ", even.intersection(primes))
print("Difference: ", even.difference(primes))
print("Symmetric Difference: ", even.symmetric_difference(primes))
```

అవుట్‌పుట్

```
Even Numbers:  {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
Prime Numbers:  {2, 3, 5, 7, 11, 13, 17, 19}
Union:  {2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20}
Intersection:  {2, 4}
Difference:  {6, 8, 10, 12, 14, 16, 18, 20}
Symmetric Difference:  {3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20}
```

9.4 నిఘంటువులు

పరిచయం (Introduction):

పైథాన్‌లో నిఘంటువుల మూలకాల మిశ్రమ సేకరణ, ఇతర సేకరణల మాదిరిగా కాకుండా జాబితా లేదా టుపుల్ వంటి, డేటారకాలు, నిఘంటువు రకం దాని మూలకంతో పాటు, పైథాన్ డిక్షనరీలోని కీలను key () కోలన్ లో వేరుచేయబడును. (:) అయితే కామా (,) లు మూలకాలను వేరు చేయునదిగా పనిచేయును. కే విలువల జతలో కుండలీకరణాలలో జతచేయు జడతాయి.

12th_CS_Telugu_medium_Chapter_9.indd 164 27-03-2023 10:39:51

నిఘంటువును నిర్వహించే సింటాక్స్:

```
Dictionary_Name = {     Key_1: Value_1,
                        Key_2:Value_2,
                        ……..
                        Key_n:Value_n
                        }
```

నిఘంటువులోని కీ ప్రత్యేకమైన కేస్ సెన్సిటివ్‌గా ఉండాలి. మరియు ఏదైన చెల్లుబాటు అయ్యే పైథాన్ రకానికి చెందినది కావచ్చు.

9.4.1 నిఘంటువును సృష్టించడం (Creating a Dictionary)

శున్య నిఘంటువు

Dict1 = { }

కీతో కూడిన నిఘంటువు

Dict_Stud = { 'RollNo': '1234', 'Name':'Murali', 'Class':'XII', 'Marks':'451'}

9.4.2 Dictionary Comprehensions

పైథాన్‌లో డిక్షనరీని సృష్టించే మరోమార్గం కాంప్రెహెన్షన్. ఈ క్రిందివి dictionaryని సృష్టించే వాక్యనిర్మాణం.

Syntax

Dict = { expression for variable in sequence [if condition] }

ఒక వేళ షరతు ఐచ్ఛికం మరియు పేర్కొనబడితే, సీక్వెన్సీలోని విలువలు మాత్రమే పరిస్థితిని సంతృప్తి పరిచే వ్యక్తీకరణను ఉపయోగించి మదింపు చేయబడతాయి.

ఉదాహరణ

*Dict = { x : 2 * x for x in range(1,10)}*

Output of the above code is

{1: 2, 2: 4, 3: 6, 4: 8, 5: 10, 6: 12, 7: 14, 8: 16, 9: 18}

9.4.3 ఒక నిఘంటువు నుండి మూలకాలను ప్రాప్యత చేయడం, జోడించడం సవరించడం మరియు తొలగించడం (Accessing, Adding, Modifying and Deleting elements from a Dictionary)

ఒక నిఘంటువు నుండి అన్ని అంశాలను యాక్సెస్ చేయడం ప్రాప్యత చేయడం జాబితాలు మరియు టుపుల్స్ మాదిరిగానే ఉంటుంది.సాధారణ print ఫంక్షన్ అన్ని మూలకాలను యాక్సస్ చేయడానికి ఉపయోగించబడుతుంది. మీరు ఒక నిర్ధిష్ట మూలకాన్ని యాక్సెస్ చేయాలనుకుంటే key తో పాటు చదరపు బ్రాకెట్టును ఉపయోగించవచ్చు

12th_CS_Telugu_medium_Chapter_9.indd 165 27-03-2023 10:39:51

ఉదాహరణ: నిఘంటువులో నిల్వచేసిన అన్ని విలువలను యాక్సెస్ చేయడానికి తగిన ప్రోగ్రామ్.

```
MyDict = {      'Reg_No': '1221',
                'Name' : 'Tamilselvi',
                'School' : 'CGHSS',
                'Address' : 'Rotler St., Chennai 112' }
print(MyDict)
print("Register Number: ", MyDict['Reg_No'])
print("Name of the Student: ", MyDict['Name'])
print("School: ", MyDict['School'])
print("Address: ", MyDict['Address'])
```

అవుట్‌పుట్

```
{'Reg_No': '1221', 'Name': 'Tamilselvi', 'School': 'CGHSS', 'Address': 'Rotler St., Chennai 112'}
Register Number:  1221
Name of the Student:  Tamilselvi
School:  CGHSS
Address:  Rotler St., Chennai 112
```

పై విషయాన్ని గమనించినట్లయితే మొదటి ముద్రణ స్టేట్‌మెంట్ డిక్షనరీ యొక్క అన్ని విలువలను ముద్రిస్తుంది. ఇతర స్టేట్‌మెంట్‌లు చదరపు బ్రాకెట్టులో ఇవ్వబడిన పేర్కొన్న విలువలు మాత్రమే ముద్రించబడతాయి.

ఇప్పటికి ఉన్న నిఘంటువులో, keyతో పాటు విలువలను కేటాయించడం ద్వారా మీరు మరిన్ని విలువలను జోడించవచ్చు. డిక్షనరీలో విలువలను జోడించడానికి క్రింది వాక్యనిర్మాణమును ఉపయోగించెదము.

dictionary_name [key] = value/element

ఉదాహరణ : క్రొత్త విలువలను డిక్షనరీలో చేర్చుటకు గాను కావలసిన ప్రోగ్రామ్

```
MyDict = {      'Reg_No': '1221',
                'Name' : 'Tamilselvi',
                'School' : 'CGHSS', 'Address' : '
                Rotler St., Chennai 112'}
print(MyDict)
print("Register Number: ", MyDict['Reg_No'])
print("Name of the Student: ", MyDict['Name'])
MyDict['Class'] = 'XII - A'                 # Adding new value
print("Class: ", MyDict['Class'])           # Printing newly added value
print("School: ", MyDict['School'])
print("Address: ", MyDict['Address'])
```

డిక్షనరీలో విలువను సవరించడం మూలకాలను జోడించడం వలె వుంటుంది. మీరు ఒక key కి విలువ ఇచ్చినపుడు అది పాత విలువను over write చేస్తుంది.

పైథాన్ డిక్షనరీలో del keyword ఒక నిర్దిష్ట మూలకాన్ని తొలగించడానికి సహాయపడుతుంది. clear( ) ఫంక్షన్ నిఘంటువులోని అన్ని అంశాలను తొలగించడానికి ఉపయోగపడుతుంది. నిఘంటువును తొలగించడానికి మీరు నిఘంటువు పేరుతో del keyword ఉపయోగించవచ్చు.

12th_CS_Telugu_medium_Chapter_9.indd 166 27-03-2023 10:39:51

Syntax:

```
# To delete a particular element.
del dictionary_name[key]
# To delete all the elements
dictionary_name.clear( )
# To delete an entire dictionary
del dictionary_name
```

ఉదాహరణ : డిక్షనరీలోని మూలకాలను తొలగించుటకు మరియు చివరగా నిఘంటువును తొలగించడానికి ప్రోగ్రామ్.

```
Dict = {'Roll No' : 12001, 'SName' : 'Meena', 'Mark1' : 98, 'Marl2' : 86}
print("Dictionary elements before deletion: \n", Dict)
del Dict['Mark1']              # Deleting a particular element
print("Dictionary elements after deletion of a element: \n", Dict)
Dict.clear()                   # Deleting all elements
print("Dictionary after deletion of all elements: \n", Dict)
del Dict
print(Dict)                    # Deleting entire dictionary
```

అవుట్‌పుట్

```
Dictionary elements before deletion:
    {'Roll No': 12001, 'SName': 'Meena', 'Mark1': 98, 'Marl2': 86}
Dictionary elements after deletion of a element:
    {'Roll No': 12001, 'SName': 'Meena', 'Marl2': 86}
Dictionary after deletion of all elements:
    { }
Traceback (most recent call last):
    File "E:/Python/Dict_Test_02.py", line 8, in <module>
    print(Dict)
NameError: name 'Dict' is not defined
```

9.4.4 జాబితా మరియు నిఘంటువు మధ్యగల వ్యత్యాసము (Difference between List and Dictionary)

1. జాబితా అనేది క్రమయుగ్మాల మూలకాల సమితి. నిఘంటువు అనేది ఒక మూలకం ను మరొక విలువ తో సరిపోల్చడానికి ఉపయోగించే డేటా నిర్మాణం అవుతుంది.
2. సూచిక విలువలు ఒక నిర్దిష్ట మూలకాన్ని యాక్సెస్ చేయడానికి ఉపయోగించవచ్చు. కాని నిఘంటువు key సూచికను సూచిస్తుంది అని తెలుసుకోవాలి. key అనునది సంఖ్యాకములైన స్ట్రింగ్‌గా ఉంటుందని గుర్తుంచుకోవాలి.
3. విలువలు చూడడానికి జాబితాలు ఉపయోగించబడతాయి డిక్షనరీలో ఒక విలువను తీసుకోవడానికి మరియు మరొక విలువను చూడడానికి నిఘంటువు ఉపయోగించబడుతుంది.

12th_CS_Telugu_medium_Chapter_9.indd 167 27-03-2023 10:39:51

గుర్తించుకొనవలసిన అంశాలు

- పైథాన్ ప్రోగ్రామింగ్ భాషలో జాబితా వంటి నాలుగు రకాల డేటా సేకరణలు ఉన్నాయి అవి లిస్ట్, టుపుల్స్, సమితులు మరియు నిఘంటువు.
- లిస్టను "సీక్వెన్స్ డేటా టైప్" అని కూడా అంటారు. ప్రతి లిస్ట్‌లోని విలువను మూలకము అంటారు.
- లిస్ట్‌లోని మూలకములను చదరపు బ్రాకెట్టులో గుర్తించెదరు.
- ప్రతిమూలకం ఒకే రకమైన విలువను కలిగియుండే సూచిక విలువ '0' తో మొదలవుతుంది.
- పైథాన్ ధనాత్మక మరియు ఋణాత్మక విలువలను సూచిస్తుంది.
- లిస్ట్‌లోని అన్ని మూలకములను చొప్పించుటకు లూప్స్‌ను ఉపయోగిస్తారు.
- ఫర్ ("for") లూప్ అన్ని మూలకాలను ఒక్కొక్కటిగా చొప్పించడానికి లూప్ అనువైనది.
- జాబితాలో ఎక్కువ మూలకాలను చేర్చడానికి append( ),extend( ) మరియు insert( ) ఉపయోగిస్తారు.
- జాబితా నుండి మూలకములను తొలగించడానికి del, remove( ) మరియు pop( ) ఉపయోగించబడతాయి.
- శ్రేణి విలువను రూపొందించడానికి range ( ) ఫంక్షన్‌ను ఉపయోగిస్తారు.
- టుపుల్స్ సంఖ్యాత్మకమైన విలువలను కలిగియుండి అవి కామాలతో వేరుపరచి, వాటిని కుండలీకరణాలలో సూచిస్తారు.
- టుపుల్స్‌ను పునగావృతా చేయడం జాబితా కంటె వేగంగా ఉంటుంది.
- జాబితా నుండి టుపుల్స్‌ను సృష్టించడానికి టుపుల్ ( ) ఫంక్షన్ కూడా ఉపయోగిస్తారు.
- ఒక మూలకంతో టుపుల్స్‌ను సృష్టించడం 'సింగిల్టన్' టుపుల్ అంటారు.
- ఒక సమితి, ఒక మార్చగల మరియు నకిలీలు లేని మూలకాల క్రమం లేని సేకరణ.
- కామాతో వేరు చేయబడిన అన్ని మూలకాలను ఒక జత కుండలీకరణ బ్రాకెట్టులో ఉంచడం ద్వారా సమితి (set) ఏర్పడుతుంది.
- నిఘంటువు అనేది అన్ని మూలకాల మిశ్రమ సేకరణ అగును.



అనుభవ విషయాలు

1. జాబితా నుండి నకిలీలను తొలగించడానికి ఒక ప్రోగ్రామ్‌ను వ్రాయండి.
2. టుపుల్‌లో గరిష్ట విలువను ముద్రించే ప్రోగ్రామ్‌ను వ్రాయండి.
3. లూప్‌ను ఉపయోగించి టుపుల్స్‌లోని అన్ని సంఖ్యల మొత్తాన్ని కనుగొనే ప్రోగ్రామ్‌ను వ్రాయండి.
4. జాబితాలోని అన్ని సరి సంఖ్యల మొత్తాన్ని కనుగొనే ప్రోగ్రామ్‌ను వ్రాయండి.
5. లూప్‌ను ఉపయోగించి జాబితాను రివర్స్‌చేసే ప్రోగ్రామ్ వ్రాయండి.
6. పేర్కొన్న ప్రదేశంలో జాబితాలో విలువను చొప్పించడానికి ఒక ప్రోగ్రామ్ వ్రాయండి.

12th_CS_Telugu_medium_Chapter_9.indd 168 27-03-2023 10:39:52

7. 1 నుండి 50 మధ్య గల సంఖ్యలలో 3 చే లేదా 6 చే భాగించాబడు సంఖ్యల జాబితా కొరకు ప్రోగ్రామ్ రాయండి.
8. 1 నుండి 20 వరకు గల సంఖ్యలను ఒక లిస్ట్‌ను తయారు చేసే ప్రోగ్రామ్ వ్రాసి, తర్వాత 3చే భాగించబడగలిగిన సంఖ్యలను జాబితానుండి తొలగించండి.
9. జాబితాలో విలువలు ఎన్నిసార్లు కనబడుతుంది అని లెక్కించే ప్రోగ్రామ్‌ను వ్రాయండి. అదే చేయడానికి లూప్‌ను ఉపయోగించండి.
10. నిఘంటువులో గరిష్ట మరియు కనిష్ఠ విలువను ముద్రించే ప్రోగ్రామ్‌ను వ్రాయండి.

మూల్యాంకనం

TS1FQ

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. కలక్షన్ డేటా టైపులో సంబందం కలిగిన ఒక దానిని ఎంచుకొనుము.
   (a) List (b) Tuple (c) Dictionary (d) Loop
2. Let list1=[2,4,6,8,10], అయిన (List1[-2]) యొక్క ఫలితం.
   (a) 10 (b) 8 (c) 4 (d) 6
3. జాబితాలోని మూలకాల సంఖ్యను లెక్కించడానికి క్రింది వాటిలో ఏది ఉపయోగపడుతుంది.
   (a) count() (b) find() (c)len() (d) index()
4. జాబిత =[10, 20, 30, 40,50] అయితే List[2]=35 యొక్క ఫలితం.
   (a) [35,10,20,30,40,50] (b) [10,20,30,40,50,35]
   (c) [10,20,35,40,50] (d) [10,35,30,40,50]
5. List=[17,23,41,10] అయితే List.append(32) యొక్క ఫలితం.
   (a) [32,17,23,41,10] (b) [17,23,41,10,32]
   (c) [10,17,23,32,41] (d) [41,32,23,17,10]
6. క్రింది వాటిలో ఏ పైథాన్ ఫంక్షన్ ఇప్పటికే ఉన్న జాబితాలో ఒకటి కంటే ఎక్కువ మూలకాలను జోడించగలదు.
   (a) append() (b) append_more() (c)extend() (d) more()
7. క్రింది పైథాన్ కోడ్ ఫలితము ఏమిటి?

```
S=[x**2 for x in range(5)]
print(S)
```

   (a) [0,1,2,4,5] (b) [0,1,4,9,16] (c) [0,1,4,9,16,25] (d) [1,4,9,16,25]

12th_CS_Telugu_medium_Chapter_9.indd 169 27-03-2023 10:39:52

8. పైథాన్‌లో type() యొక్క ఉపయోగము ఏమిటి?

(a) Tuple ను సృష్టించడానికి

(b) Tuple లోని మూలకం యొక్క రకాన్ని తెలుసుకోవడానికి

(c) పైథాన్ ఆబ్జెక్ట్ యొక్క డేటా టైపును తెలుసుకొనడానికి

(d) జాబితాను సృష్టించడానికి

9. క్రిందివానిలో సరికాని స్టేట్‌మెంట్ ఏది?

(a) జాబిత మార్చదగినది

(b) Tuple మార్పులేనిది

(c) Append ( ) ను ఒక మూలకమును add చేయుటకు ఉపయోగపడును.

(d) tuple లో extend ( )ని జాబితాలోని మూలకాలను add చేయుటకు ఉపయోగపడును.

10. setA={3,6,9}, setB={1,3,9} అయిన క్రింది snippet ఫలితం ఏమిటి?

print(setA|setB)

(a) {3,6,9,1,3,9} (b) {3,9} (c) {1} (d) {1,3,6,9}

11. క్రింది set operation లో రెండు sets లలో ఉన్న అన్ని అంశాలు ఉంటాయి కాని రెండు సెట్‌లకు సాధారణమైనవి కావు?

(a) symmetric వ్యత్యాసం (b) వ్యత్యాసము (c) ఖండనము (d) సమ్మేళనము

12. పైథాన్ నిఘంటువులోని key దీని ద్వారా పేర్కొనబడినది.

(a) = (b) ; (c)+ (d) :

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. పైథాన్‌లో జాబితా అంటే ఏమిటి?
2. మీరు రివర్స్ ఆడర్‌లో జాబితా అంశాలను ఎలా access చేస్తారు?
3. పైథాన్ కోడ్‌ను అనుసరించి x యొక్క విలువ ఏమిటి?

```
List1=[2,4,6[1,3,5]]
x=len(List1)
```

4. జాబిత యొక్క remove( ) ఫంక్షన్‌తో del ను భేదపరచండి?
5. Tuple తో n సంఖ్యల elements ను సృష్టంచడానికి syntax ను వ్రాయండి.
6. పైథాన్‌లో set అనగానేమి?

12th_CS_Telugu_medium_Chapter_9.indd 170 27-03-2023 10:39:52

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. జాబితాలో Tuples యొక్క ఉపయోగాలు ఏమి?
2. sort( ) గురించి లఘు సంచికను రాయండి?
3. క్రింది కోడ్ యొక్క అవుట్‌పుట్ ఏమి?

```
list = [2**x for x in range(5)]
print(list)
```

4. డిక్షనరిలోdel మరియు clear( ) ఫంక్షన్ మధ్యగల భేదాలను ఉదాహరణలతో వివరించండి.
5. పైథాన్ మద్దతు తెలిపే set ఆపరేషన్‌ను జాబితా పరుచుము.
6. List మరియు Dictionary మధ్య గల భేదములు ఏవి?

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. జాబితాలో మూలకాన్ని చేర్చుటకు ఉన్న వివిధ మార్గాలు ఏవి? సరియైన ఉదాహరణతో వివరించండి?
2. range( ) ఫంక్షన్ యొక్క ప్రాముఖ్యత ఏమి? ఉదాహరణతో వివరించుము.
3. nested tuple అనగానేమి? ఉదాహరణతో వివరించుము.
4. పైథాన్ మద్దతు తెలిపే వివిద set ఆపరేషన్‌లను సరియైన ఉదాహరణలతో వివరించండి.

References

1. https://docs.python.org/3/tutorial/index.html
2. https://www.techbeamers.com/python-tutorial-step-by-step/#tutorial-list
3. Python programming using problem solving approach – Reema Thareja – Oxford University press.
4. Python Crash Course – Eric Matthes – No starch press, San Francisco.

12th_CS_Telugu_medium_Chapter_9.indd 171 27-03-2023 10:39:52

యూనిట్ III

అధ్యాయం 10

పైథాన్ క్లాసెస్ మరియు ఆబ్జెక్ట్స్

అభ్యసనా లక్ష్యములు

ఈ అధ్యాయము పూర్తి అయిన తరువాత విద్యార్థి నేర్చుకొను విషయములు.

- ఆబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రామ్ యొక్క ప్రాథమిక విషయాలను అవగాహన చేసుకొనుటకు అనగా క్లాసెస్, ఆబ్జెక్ట్స్, కన్స్ట్రక్టర్ మరియు డిస్ట్రక్టర్.
- పైథాన్ క్లాసెస్‌ను కూడిన మరియు ఆబ్జెక్ట్‌లను తయారు చేయుటకు సంబంధించిన జ్ఞానమును పొందును.
- కన్స్ట్రక్టరు కూడిన క్లాసెస్‌ను ఏర్పరచుట.
- క్లాసెస్‌ను ఉపయోగించుకొని పైథాన్‌లో కాంప్లెక్స్ ప్రోగ్రామ్‌ను వ్రాయుట.

10.1 పరిచయం

పైథాన్ అనునది ఒక ఆబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రామింగ్ లాంగ్వేజ్ (Obective oriented language) అగును. క్లాసెస్ మరియు ఆబ్జెక్ట్‌లు, ఆబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రామ్ యొక్క ముఖ్యలక్షణాలు అగును తరగతులు మరియు వస్తువు యొక్క సైద్ధాంతిక భావనలు C++ సమానంగా ఉంటాయి. C++ తో పోలిస్తే పైథాన్‌లో క్లాసెస్ మరియు ఆబ్జెక్ట్‌లు సృష్టించడం మరియు అమలు పరచడం సైద్ధాంతికంతో చాలా సులభం.

పైథాన్‌లో క్లాస్ ప్రధాన అంశంగా అగును. ఆబ్జెక్ట్ అనేది ఆ డేటాపై పనిచేసే ఫంక్షన్ మరియు డేటా యొక్క సేకరణ అగును. క్లాస్ అనేది ఆబ్జెక్ట్ యొక్క టెంప్లేట్ అగును. ఆబ్జెక్ట్ ఓరియంబెడ్ ప్రోగ్రామింగ్ కాన్సెప్ట్ ప్రకారం ఆబ్జెక్ట్‌లు ఒక క్లాస్ యొక్క సందర్భాలు. పైథాన్ లో ప్రతిదీ ఒక ఆబ్జెక్ట్. ఉదాహరణకు మనం ప్రోగ్రామ్‌లో ఉపయోగించే అన్ని పూర్ణాంకాం వేరియబుల్స్ క్లాస్ పూర్ణంకానికి చెందిన ఆబ్జెక్ట్. అదే విధంగా అన్ని స్ట్రింగ్ వేరియబుల్స్ కూడా క్లాస్ స్ట్రింగ్ యొక్క ఆబ్జెక్ట్.

10.2 తరగతుల నిర్వచనము

పైథాన్‌లో క్లాసెస్ నిర్ణయించడానికి "Class" అను ప్రత్యేకపదముగా ఉపయోగిస్తారు. ప్రతి క్లాస్‌లోను ప్రత్యేక గుర్తు colon (:) తో సూచిస్తారు.

Syntax:

```
class class_name:
    statement_1
    statement_2
    ……..
    ……..
    statement_n
```


12th_CS_Telugu_Medium_Chapter_10.indd 172 27-03-2023 10:40:27

ఇక్కడ క్లాస్ నిర్వచనంలో స్టేట్‌మెంట్ వేరియబుల్ డిక్లరేషన్, అనునది డెసిషన్ కంట్రోల్ కావచ్చు. లూప్ లేదా ఫంక్షన్ డెపెనిషన్ కూడా ఉండవచ్చు. తగరతి లోపల నిర్వచింపబడిన వేరియబుల్స్‌ను "క్లాస్ వేరియబుల్" అని పిలుస్తారు. మరియు ఫంక్షన్‌ను "మెథెడ్స్" అని కూడా పిలుస్తారు. క్లాస్ వేరియబుల్ మరియు పద్ధతులు కలిపి క్లాస్ సభ్యులు (Members) అంటారు. క్లాస్ సభ్యులను ఆబ్జక్ట్ ద్వారా లేదా క్లాస్ ఉదాహరణ ద్వారా యాక్సెస్ చేయాలి. పైథాన్ ప్రోగ్రామ్‌లో ఎక్కడైనా ఒక క్లాస్‌ని నిర్వచించవచ్చు.

ఉదాహరణ : క్లాస్‌ని నిర్వచించే ప్రోగ్రామ్.

క్లాస్ నమూనా:

```
x, y = 10, 20          # class variables
```

పై కోడ్ క్లాస్ నిర్వచనంలో *sample* అనునది క్లాస్ యొక్క పేరు దీనికి రెండు వేరియబుల్స్ x మరియు y వరుసగా ప్రారంభ విలువ 10 మరియు 20 కలిగి ఉంటాయి. క్లాస్ లోపల నిర్వచించిన విలువలను యాక్సెస్ చేయడానికి, మీకు క్లాస్ యొక్క ఒక ఆబ్జెక్ట్ లేదా ఉదాహరణ అవసరం.

10.3 ఆబ్జెక్ట్‌లను సృష్టించడం

ఒక క్లాస్‌ను సృష్టించబడిన తర్వాత మీరు ఆ క్లాస్ యొక్క ఒక ఆబ్జెక్ట్‌ను లేదా ఇన్‌స్టాంట్స్‌ను సృష్టించాలి. వస్తువును సృష్టించే ప్రక్రియను "క్లాస్ ఇన్‌స్టాంటిమేషన్" అంటారు.



Syntax:

Object_name = class_name( )

క్లాస్ పేరుతో, ఇన్‌స్టాంటిమేషన్ ఫంక్షన్‌లో సంజ్ఞామానాన్ని ఉపయోగిస్తుంది.

10.4 క్లాస్ సభ్యులను యాక్సెస్ చేయడం

క్లాస్ సభ్యులను యాక్సెస్ చేయడం ఏదైనా క్లాస్ సభ్యుడు డాట్ ఆపరేటర్‌తో ఆబ్జెక్ట్‌ను ఉపయోగించడం ద్వారా క్లాస్ వేరియబుల్ లేదా ఫంక్షన్‌ను యాక్సెస్ చేయవచ్చు.

Syntax:

Object_name . class_member

ఉదాహరణ : ఒక క్లాస్‌ని నిర్వచించడానికి మరియు దాని సభ్యుల వేరియబుల్స్‌ను యాక్సెస్ చేసే ప్రోగ్రామ్ క్లాస్ నమూనా

```
class Sample:
        x, y = 10, 20          #class variables
S=Sample( )            # class instantiation
print("Value of x = ", S.x)
print("Value of y = ", S.y)
print("Value of x and y = ", S.x+S.y)
```

Output :

```
Value of x =  10
Value of y =  20
Value of x and y =  30
```


12th_CS_Telugu_Medium_Chapter_10.indd 173 27-03-2023 10:40:27

పై కోడ్‌లో, క్లాస్ పేరు *sample*, క్లాస్ లోపల, ప్రారంభ విలువ 10 మరియు 20 తో **x** మరియు **y** వేరియబుల్స్‌ను కేటాయించబడుతుంది. ఈ రెండు వేరియబుల్స్ క్లాస్ వేరియబుల్స్ లేదా క్లాస్ మెంబర్ వేరియబుల్స్ అంటారు. క్లాస్ ఇన్స్‌స్టాంటియేషన్ ప్రాసెస్‌లో, క్లాస్ సభ్యులను యాక్సెస్ చేయడానికి ఒక **S** అని ఆబ్జెక్టును సృష్టించబడినది. మొదటి రెండు ప్రింట్ స్టేట్‌మెంట్‌లు క్లాస్ వేరియబుల్ **x** మరియు **y** యొక్క విలువను ప్రింట్ చేస్తాయి. మరియు చివరి ప్రింట్ స్టేట్‌మెంట్ రెండు విలువలను జోడించి ఫలితాన్ని ప్రింట్ చేస్తుంది.

10.5 క్లాస్ మెథడ్

పైథాన్‌లోక్లాస్ ఫంక్షన్ లేదా మెథడ్ ఒక చిన్న వ్యత్యాసంతో సాధారణ ఫంక్షన్‌కు చాలా దగ్గర పోలికలో ఉంటుంది. క్లాస్ పద్ధతికి సెల్ఫ్ అని పేరు పెట్టబడిన మొదటి ఆర్గ్యుమెంట్ ఉండాలి. పైథాన్ దాని విలువను అందించే పద్ధతిని పిలిచినప్పుడు ఈ వాదనకు ఒక విలువను అందించవలసిన అవసరం లేదు. స్వయంచాలకంగా, ఒక పద్ధతి ఎటువంటి వాదనలు తీసుకోకపోతే, దానిని 'self' పద్ధతి అని పిలిచే మొదటి వాదనతో నిర్వచించాలి. ఇది ఒకే ఒక ఆర్గ్యుమెంట్‌ను మాత్రమే అంగీకరించడానికి నిర్వచించబడుతుంది. అది రెండు ఆర్గ్యుమెంట్‌ను తీసుకుంటుంది.

అంటే 'self' నిర్వచించిన ఆర్యుమెంట్. మీరు క్లాస్ లోపల క్లాస్ వేరియబుల్‌ను ప్రకటించినప్పుడు, పద్ధతులు తప్పక క్లాస్ యొక్క పేరు మరియు Dot (.) అపరేటర్‌ను కలిగి ఉండాలి.

గమనిక

క్లాస్ ద్వారా ఉపసర్గ ఇవ్వండి క్లాస్ లోపల నిర్వచించిన స్టేట్‌మెంట్‌లు సరిగ్గా ఇండెంట్ చేయాలి.

ఉదాహరణ : 10:2 క్లాస్ ను ఉపయోగించి స్టూడెంట్ మార్క్ యొక్క మొత్తం మరియు సరాసరి గణించుటకు ప్రోగ్రామ్

```
class Student:
        mark1, mark2, mark3 = 45, 91, 71          #class variable
        def process(self):                        #class method
            sum = Student.mark1 + Student.mark2 + Student.mark3
            avg = sum/3
            print("Total Marks = ", sum)
            print("Average Marks = ", avg)
            return
S=Student()
S.process()
```

పై ప్రోగ్రామ్‌లో క్లాస్‌ను నిర్వచించిన తరువాత ఒక 'S' అనే ఆబ్జెక్ట్‌ను సృష్టించబడుతుంది. S process () అవసరమైన అవుట్‌పుట్ పొందడానికి ఫంక్షన్‌ను పిలుస్తుంది.

12th_CS_Telugu_Medium_Chapter_10.indd 174 27-03-2023 10:40:27

Mark1, Mark2 మరియు Mark3 అనే మూడు వేరియబుల్స్‌కు 45, 91, 71 గా ప్రకటించబడినది అని గమనించండి. Process () అనే పద్ధతి Self తో నిర్వచింపబడినది. ఆ పద్ధతికి ఎటువంటి విలువను ఇవ్వలేము. మొదట, Process method క్లాస్ వేరియబుల్స్ యొక్క విలువలను కూడబడినది. ఫలితాన్ని Sum అని వేరియబుల్ లో నిల్వచేస్తుంది. మరియు సగటును కనుగొంటుంది.

ఈ విధంగా పై కోడ్ క్రింది అవుట్‌పుట్‌ను చూపుతుంది.

అవుట్‌పుట్

Total Marks = 207

Average Marks = 69.0

ఉదాహరణ : క్లాస్‌ను ఉపయోగించి ఇచ్చిన సంఖ్య బేసి (లేదా) సరి సంఖ్య అని తనిఖీ చేసి ప్రింట్ చేసే ప్రోగ్రామ్.

```
class Odd_Even:
        def check(self, num):
                if num%2==0:
                print(num," is Even number")
        else:
                print(num," is Odd number")
n=Odd_Even()
x = int(input("Enter a value: "))
n.check(x)
```

When you execute this program, Python accepts the value entered by the user and passes it to the method check through object.

అవుట్‌పుట్ 1

```
Enter a value: 4
4  is Even number
```

అవుట్‌పుట్ 2

```
Enter a value: 5
5  is Odd number
```

10.6 పైథానలో Constructor మరియు Distructor

ఒక క్లాస్ యొక్క Objectను సృష్టించబడినప్పుడు Constructor స్వయం చాలకంగా అమలు చేయబడినచో దానంతట అదే అమలు అగును. పైథాన్‌లో **"inir"** అనే ప్రత్యేక ఫంక్షన్ కన్స్ట్రక్టర్‌గా పనిచేయును. ఇది డబుల్ అండర్ స్కోర్‌తో ప్రారంభమై, డబుల్ అండర్ స్కోర్‌తో ముగుస్తుంది. ఈ ఫంక్షన్ సాధారణ ఫంక్షన్ వలె పనిచేస్తుంది. కానీ తేడా ఏమిటంటే, Objective సృష్టించబడినప్పుడు అది స్వయంచాలకంగా అమలు అవుతుంది. ఈ కన్స్ట్రక్టర్ ఫంక్షన్‌ను వాదనలతో లేదా వాదన లేకుండనో నిర్వచించవచ్చు. క్లాస్ వేరియబుల్స్ ప్రారంభించడానికి ఈ పద్ధతి ఉపయోగించబడుతుంది.

12th_CS_Telugu_Medium_Chapter_10.indd 175 27-03-2023 10:40:27

General format of __init__ method (Constructor function)

def __init__(self, [args ……..]):

<statements>

ఉదాహరణ : కన్స్ట్రక్టర్ క్లాస్‌ను వివరించే ప్రోగ్రామ్ నమూనా

```
class Sample:
        def __init__(self, num):
                print("Constructor of class Sample...")
                self.num=num
                print("The value is :", num)
S=Sample(10)
```

పై పైథాన్ ప్రోగ్రామ్‌లో 'self, num' ఒకే ఒక objective తో కూడిన కన్స్ట్రక్టర్‌ను కలిగి ఉండును. కన్స్ట్రక్టర్‌ను num చేసేటప్పుడు మొదటి print statement, "Constructor of class Sample….", ను చేయును. ఒక ఆబ్జెక్ట్ సృష్టించబడినప్పుడు self num పొందుపరుచుము. ఈ విధంగా పైథాన్ క్రింది అవుట్‌పుట్‌ను ప్రదర్శిస్తుంది.

పై కన్స్ట్రక్టర్ self, num అమలు అవుతుంది. ఒక వస్తువు objectలో పారామితి 10ని సృష్టించినపుడు పైథాన్ కింది ఉత్పత్తి (output) ను ప్రదర్శిస్తుంది.

Constructor of class Sample...

The value is : 10

కన్స్ట్రక్టర్‌లో నిర్వచించిన క్లాస్ వేరియబుల్ క్లాస్‌తో సృష్టించబడిన వస్తువుల సంఖ్యను పొంది ఉండును.

గమనిక

Instance variables are the variables whose value varies from object to object. For every object, a separate copy of the instance variable will be created. Instance variables are declared inside a method using the self keyword. In the above example, we use constructor to define instance variable.

Example : Program to illustrate class variable to keep count of number of objects created.

```
class Sample:
        num=0
        def __init__(self, var):
                Sample.num+=1
                self.var=var
                print("The object value is = ", var)
                print("The count of object created = ", Sample.num)

S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```


12th_CS_Telugu_Medium_Chapter_10.indd 176 27-03-2023 10:40:27

పై ప్రోగ్రామ్ నందు, క్లాస్ వేరియబుల్ num మరియు class sample యొక్క మూడు వస్తువులచే భాగస్వామ్యం చేయబడుతుంది. ఇది సున్నాతో ప్రారంభించబడుతుంది మరియు ప్రతిసారీ ఒక ఆబ్జెక్ట్ సృష్టించబడినప్పుడు num 1 ద్వారా పెరుగుతుంది. కాబట్టి అన్ని objectiveలో పంచుకోబడిన వేరియబుల్, ఒక object ద్వారా num కు చేసిన మార్పు ఇతర వస్తువులలో కూడా ప్రతిబింబిస్తుంది. ఈ విధంగా పై ప్రోగ్రామ్ క్రింద ఇచ్చిన అవుట్‌పుట్‌ని ఉత్పత్తి చేస్తుంది.

అవుట్‌పుట్

```
The object value is             =    15
The count of object created     =    1
The object value is             =    35
The count of object created     =    2
The object value is             =    45
The count of object created     =    3
```

డిస్ట్రక్టర్ అనేది ఒక ప్రత్యేక పద్ధతి, ఒక వస్తువు పరిధి నుండి నిష్క్రమించినప్పుడు స్వయంగా అమలు అవుతుంది. ఇది కన్స్ట్రక్టర్‌కు వ్యతిరేకం. పైథాన్‌లో __del__( ) పద్ధతిని డిస్ట్రక్టర్‌గా ఉపయోగిస్తారు.

ఉదాహరణ : __del__( ) పద్ధతి గురించి వివరించడానికి ప్రోగ్రామ్

```
class Sample:
        num=0
        def __init__(self, var):
                Sample.num+=1
                self.var=var
                print("The object value is = ", var)
                print("The value of class variable is= ", Sample.num)
        def __del__(self):
                Sample.num-=1
                print("Object with value %d is exit from the scope"%self.var)
S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```

గమనిక

డెలిటెడ్ ఆబ్జెక్ట్ – __del__ ను ఉపయోగించినపుడు ఈ– పద్ధతి స్వతంత్రంగా పిలస్తుంది.

10.7 పబ్లిక్ (Public) మరియు ప్రైవేటు (Private) వేరియబుల్ డేటా

నిర్వచించిన వేరియబుల్స్ క్లాస్ లోపల డీపాల్ట్‌గా Public variable అగును. ఈ వేరియబుల్స్‌కు డాట్ ఆపరేటర్‌ను ఉపయోగించి ప్రోగ్రామ్‌లో ఎక్కడైనా యాక్సెస్ చేయవచ్చు.

డబుల్ అండర్ స్కోరుతో ప్రిఫిక్స్ చేయబడిన వేరియబుల్ Private అవుతుంది. ఈ వేరియబుల్స్ class లోపల మాత్రమే యాక్సెస్ చేయబడతాయి.

ఉదాహరణ : ప్రైవేట్ మరియు పబ్లిక్ వేరియబుల్స్‌ను వివరించు ప్రోగ్రామ్.

```
class Sample:
        def __init__(self, n1, n2):
                self.n1=n1
                self.__n2=n2
        def display(self):
                print("Class variable 1 = ", self.n1)
                print("Class variable 2 = ", self.__n2)
S=Sample(12, 14)
S.display()
print("Value 1 = ", S.n1)
print("Value 2 = ", S.__n2)
```

పై ప్రోగ్రామ్‌లో n1 మరియు n2 అనే రెండు క్లాస్ వేరియబుల్స్ ఉన్నాయి. n1 అనునది పబ్లిక్ వేరియబుల్ n2 ఒక ప్రైవేట్ వేరియబుల్. display ( ) సభ్యుల పద్ధతి ప్రోగ్రామ్ అనునది రెండు చరరాశులకు పంపబడిన విలువలను చూపించుటకు ఉపయోగపడును.

ఈ class variable n2 ప్రైవెట్ variable అయినప్పటికి క్లాస్ లోపల నిర్వహించనందువలన ప్రింట్ స్టేట్మెంట్ n1, n2 యొక్క విలువలను విజయవంతంగా ప్రదర్శిస్తుంది. ఎందుకంటే ఈ సందర్భంలో class లోపల నిర్వహించిన n2 ను పిలుస్తారు. కాని మనము n2 పై నుండి n2 విలువను యాక్సెస్ చేయడానికి ప్రయత్నించినపుడు లోపం చూపిస్తుంది. ఎందుకంటే, క్లాస్ వెలుపల నుండి ప్రైవేట్ వేరియబుల్ యాక్సస్ చేయలేము.

అవుట్‌పుట్

```
Class variable 1 =  12

Class variable 2 =  14

Value 1 =  12

Traceback (most recent call last):

    File "D:/Python/Class-Test-04.py", line 12, in <module>

        print("Value 2 = ", S.__n2)

AttributeError: 'Sample' object has no attribute '__n2'
```


12th_CS_Telugu_Medium_Chapter_10.indd 178 27-03-2023 10:40:27

10.8. క్లాస్ మరియు వస్తువులను వివరించు మాదిరి ప్రోగ్రామ్

ప్రోగ్రామ్ 1 : వృత్త వైశాల్యము మరియు చుట్టుకొలత కనుగొనుటకు ప్రోగ్రామ్‌ను వ్రాయుము.

```
class Circle:
        pi=3.14
def __init__(self,radius):
        self.radius=radius
def area(self):
        return Circle.pi*(self.radius**2)
def circumference(self):
        return 2*Circle.pi*self.radius
r=int(input("Enter Radius: "))
C=Circle(r)
print("The Area =",C.area())
print("The Circumference =", C.circumference())
```

అవుట్‌పుట్

```
Enter Radius: 5
The Area = 78.5
The Circumference = 31.400000000000002
```

ప్రోగ్రామ్ 2 : మీ స్కూల్ లైబ్రరీ క్లాస్ లైబ్రరీలో అందుబాటులో ఉన్న బుక్స్ రికార్డును ఉంచే మెనూడ్రివెన్ ప్రోగ్రామ్‌ను వ్రాయండి.

```
class String:
    def __init__(self):
        self.upper=0
        self.lower=0
        self.vowel=0
        self.consonant=0
        self.space=0
        self.string=""
    def getstr(self):
        self.string=str(input("Enter a String: "))
    def count (self):
```

```
            for ch in self.string:
                if (ch.isupper()):
                    self.upper+=1
                if (ch.islower()):
                    self.lower+=1
                if (ch in ('AEIOUaeiou'):
                    self.vowel+=1
             if (ch.isspace()):
                self.space+=1
                self.consonant = self.upper+self.lower - self.
vowel
    def display(self):
        print("The given string contains...")
        print("%d Uppercase letters"%self.upper)
        print("%d Lowercase letters"%self.lower)
        print("%d Vowels"%self.vowel)
        print("%d Consonants"%self.consonant)
        print("%d Spaces"%self.space)
S = String()
S.getstr()
S.count()
S.display()
```

అవుట్‌పుట్:

```
Enter a String:Welcome To Learn Computer Science
The given string contains...
4 Uppercase letters
25 Lowercase letters
12 Vowels
13 Consonants
4 Spaces
```


12th_CS_Telugu_Medium_Chapter_10.indd 180 27-03-2023 10:40:27

పైథాన్ అనునది గుర్తించుకోవలసిన అంశాలు

- ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామింగ్ భాష.
- తరగతులు మరియు ఆబ్జెక్ట్లు అనునది ఓరియెంటెడ్ ప్రోగ్రామింగ్ యొక్క ముఖ్య లక్షణాలు.
- పైథాన్లో, క్లాస్ అనే కీవర్డ్ ఉపయోగించి క్లాస్ నిర్వచించబడుతుంది.
- క్లాస్ లోపల నిర్వచించబడిన వేరియబుల్స్ను క్లాస్ "వేరియబుల్" అని పిలుస్తారు. మరియు ఫంక్షన్ను "మెథడ్స్" అని పిలుస్తారు.
- ఆబ్జెక్ట్ను సృష్టించే ప్రక్రియను "క్లాస్ ఇన్స్టాంటియేషన్" అంటారు.
- కన్స్ట్రక్టర్ అనేది ఒక క్లాస్ యొక్క ఆబ్జెక్ట్ సృష్టించబడినప్పుడు స్వయంచాలకంగా అమలు చేయబడే ప్రత్యేక ఫంక్షన్.
- పైథాన్లో "init" అని పిలువబడే ఒక ప్రత్యేక ఫంక్షన్ కన్స్ట్రక్టర్ను ఉపయోగించబడుతుంది.
- ఒక ఆబ్జెక్ట్ పరిధి నుండి నిష్క్రమించినప్పుడు డిస్ట్రక్టర్ కూడా ఒక ప్రత్యేకపద్ధతి.
- డబుల్ అండర్ స్కోర్ ప్రకృతిలో ప్రైవేట్ అవుతుంది.



హేండ్స్ ఆన్ ఎక్సీపిరియన్స్

1. క్లాస్లోని విద్యార్థుల పేర్లను, లిస్టులోని మార్కులను క్లాస్ని ఉపయోగించి నిల్వయించుటకు మరియు అందులోని మార్కుల మొత్తమును ప్రింట్ చేయుటకు ఒక ప్రోగ్రామ్ను వ్రాయండి.

2. మూడు భుజములను ఉపయోగించి త్రిభుజమును ఏర్పరచుటకు మరియు దాని వైశాల్యమును ప్రింట్ చేయుటకు క్లాస్ని ఉపయోగించి ఒక ప్రోగ్రామ్ను వ్రాయండి.

3. చదువుటకు, చూపించుటకు (display), కూడుటకు మరియు రెండు దూరముల వ్యత్యాసములకు ఒక (menu driven) ప్రోగ్రామ్ను వ్రాయండి.

12th_CS_Telugu_Medium_Chapter_10.indd 181 27-03-2023 10:40:27

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. ఒక ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామింగ్ లాగ్వేజ్ యొక్క ముఖ్య లక్షణాలు క్రింది వాటిలో ఏది.

   (a) కన్‌స్ట్రక్టరు మరియు క్లాసేస్

   (b) కన్‌స్ట్రక్టర్ మరయు ఆబ్జెక్ట్

   (c) క్లాసెస్ మరియు ఆబ్జెక్ట్

   (d) కన్‌స్ట్రక్టర్ మరియు డెస్‌ట్రక్టర్

2. క్లాస్‌లోపల వివరించబడు ఫంక్షన్‌లు.

   (a) ఫంక్షన్స్ (b) మాడులే (c) పద్ధతులు (d) సెక్షన్ (విభాగము)

3. ఏ ఆపరేటరు ద్వారా క్లాస్ సభ్యులను చొప్పించెదరు.

   (a) & (b) . (c) # (d) %

4. ఒక ఆబ్జెక్ట్ ఉత్పన్నమైనపుడు సాధారణంగా ఏ పద్ధతిని ఈ క్రింది వానిలో దేనిని అమలు చేస్తారు.

   (a) __object__( ) (b) __del__( ) (c) __func__( ) (d) __init__( )

5. ఒక ప్రైవేట్ వేరియబుల్ ఉంచడానికి ముందు దీనిని ఉంచుతారు.

   (a) __ (b) && (c) ## (d) **

6. ఒక డిస్ట్రెక్టర్‌గా ఉపయోగించడానికి ఈ క్రింది పద్ధతిని ఉపయోగిస్తారు.

   (a) __init__( ) (b) __dest__( ) (c) __rem__( ) (d) __del__( )

7. ఈ క్రింది వాటిలో క్లాస్ డిక్లరేషన్‌కు ఏది సరియైనది.

   (a) క్లాస్ క్లాస్ – పేరు (b) క్లాస్ క్లాస్ – పేరు

   (c) క్లాస్ క్లాస్ – పేరు (d) క్లాస్ క్లాస్ – పేరు ( )

8. క్రింద తెలుపబడిన ప్రోగ్రామ్‌లో (output) ఏది?

```
class Student:
    def __init__(self, name):
    self.name=name
    print (self.name)
    S=Student("Tamil")
```

   (a) Error (b) Tamil (c) name (d) self

12th_CS_Telugu_Medium_Chapter_10.indd 182 27-03-2023 10:40:27

9. ప్రైవేట్ క్లాస్ వేరియబుల్ క్రింది వాటిలో ఏది?

(a) __num (b) ##num (c) $$num (d) &&num

10. ఒక ఆబ్జెక్ట్‌ను సృష్టింప చేయుటకు ఉపయోగించే పద్ధతిని ఈ విధముగా పిలుస్తారు.

(a) కన్‌స్ట్రక్టరు (b) డిస్ట్రక్టర్ (c) ఇనిషలైజ్ (d) ఇన్‌స్టానియేషన్

భాగము -II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. క్లాస్ అనగానేమి?
2. ఇన్స్టాంటీయేషన్ అనగానేమి?
3. ఈ క్రింది ప్రోగ్రామ్‌లో (output) ఏది?

```
class Sample:
    __num=10
    def disp(self):
        print(self.__num)
S=Sample()
S.disp()
print(S.__num)
```

4. పైథాన్‌లో కన్‌స్ట్రక్టరు మీరు ఎలా సృష్టిస్తారు.
5. డిస్ట్రక్టర్ యొక్క ఉద్దేశమేమిటి?

భాగము -III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. క్లాస్ మెంబర్స్ అంటే ఏమిటి? మీరు దానిని ఎలా నిర్వచించాలి.
2. రెండు ప్రైవేట్ క్లాస్ వేరియబుల్స్‌తో క్లాస్ వ్రాయండి. మరియు ఒక పద్ధతిని ఉపయోగించి మొత్తాన్ని ప్రింట్ చేయండి.
3. ఇచ్చిన అవుట్‌పుట్ పొందటానికి ఈ క్రింది ప్రోగ్రామ్‌లో లోపమును కనుగొనుము.

```
class Fruits:
    def __init__(self, f1, f2):
        self.f1=f1
        self.f2=f2
    def display(self):
        print("Fruit 1 = %s, Fruit 2 = %s" %(self.f1, self.f2))
F = Fruits ('Apple', 'Mango')
del F.display
F.display()
```


12th_CS_Telugu_Medium_Chapter_10.indd 183 27-03-2023 10:40:27

అవుట్‌పుట్

Fruit 1 = Apple, Fruit 2 = Mango

4. క్రింది ప్రోగ్రామ్ యొక్క అవుట్‌పుట్ ఏంటి?

```
class Greeting:
    def __init__(self, name):
        self.__name = name
    def display(self):
        print("Good Morning ", self.__name)
obj=Greeting('Bindu Madhavan')
obj.display()
```

5. పైథాన్‌లో కన్‌స్ట్రక్టర్ మరియు డిస్ట్రక్టర్‌ను ఎలా నిర్వచించాలి?

భాగము -IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : (5 మార్కులు)

1. స్థిర వస్తువులను జోడించడానికి లేదా తొలగించడానికి మెనుడ్రివన్ ప్రోగ్రామ్‌ను వ్రాయండి. నిఘంటువును ఖచ్చితంగా ఉపయోగించుకొని ఐటమ్స్ మరియు బ్రాండ్ నిల్వచేయండి.

References

1. *https://docs.python.org/3/tutorial/index.html*
2. *https://www.techbeamers.com/python-tutorial-step-by-step/#tutorial-list*
3. *Python programming using problem solving approach – Reema Thareja – Oxford University press.*
4. *Python Crash Course – Eric Matthes – No starch press, San Francisco.*

12th_CS_Telugu_Medium_Chapter_10.indd 184 27-03-2023 10:40:27

యూనిట్ IV

అధ్యాయం 11

డేటాబేస్ భావనలు

అభ్యసనా లక్ష్యములు

ఈ అధ్యాయము పూర్తి అయిన తర్వాత విద్యార్థి నేర్చుకొనే అంశాలు.

- డేటాబేస్ భావనలు మరియు సంబంధాలు
- డేటాబేస్ యొక్క వేరువేరు అంశాలు
- డేటాబేస్ నమూనా యొక్క రకములు
- సంబంధాలు యొక్క రకాలు
- బీజగణిత సంబంధము యొక్క భావనలు

పరిచయం

డేటాబేస్ అనేది ఒక వ్యవస్తీకృత డేటా సేకరణ సాధారణంగా కంప్యూటర్ సిస్టమ్ నుండి ఎలక్ట్రానిక్‌గా నిల్వచేయబడుతుంది. డేటాబేస్ అనే పదాన్ని ఏదైనా (DBMS) డేటాబేస్ సిస్టమ్ లేదా డేటాబేస్‌తో అనుబంధించబడిన అనువర్తనాన్ని సూచించడానికి కూడా ఉపయోగిస్తారు. వాటి మధ్య సన్నిహిత సంబంధం ఉన్నందున, "డేటాబేస్" అనే పదాన్ని తరచుగా ఒక డేటాబేస్ మరియు దానిని మార్చటానికి ఉపయోగించే ("DBMS") రెండింటిని సూచించడానికి ఉపయోగిస్తారు. ఒక పాఠశాలలోని తరగతి హాజరు పట్టిక అనేది అక్షర క్రమంలో అమర్చబడిన ఒక డేటాబేస్. ప్రజలు విషయాలను రికార్డ్ చేయడం ప్రారంభించినప్పటి నుండి డేటా బేస్లు ఉన్నాయి. ఇక్కడ మనం ఎలక్ట్రానిక్ అంశములుపై దృష్టి పెడదాము.

11.1 డేటా

డేటా కంప్యూటర్‌లో నిల్వచేయబడిన ముడివాస్తవాలు. డేటా ఏదైనా అక్షరం, వచనం పదం లేదా సంఖ్యగా ఉండవచ్చు.

ఉదాహరణ : 600006, DPI క్యాంపస్, SCERT, చెన్నై, కాలేజ్ రోడ్

11.2 సమాచారము

సమాచారము అనేది సూత్రీకరింపబడిన డేటా. ఇది డేటాను కొన్ని నిర్దేశించిన మార్గాలలో ఉపయోగించుటకు అనుమతించబడును.

ఉదాహరణ

SCERT
కాలేజ్ రోడ్
DPI క్యాంపస్
చెన్నై – 600006

SYC51

పై ఉదాహరణల నుండి కొన్ని పదాలు మరియు సంఖ్యల యొక్క విభాగాలుగా డేటా కనబడుచున్నది. అయినప్పటికి, డేటా ప్రాసెస్ చేయబడినపుడు (ఆకృతీకరించబడినపుడు), నిర్వహించబడినపుడు మరియు ఫార్మట్ (formatted) చేయబడినపుడు, ఇది SCERT సంస్థ సంప్రదింపు చిరునామా గురించి అర్థవంతమైన సమాచారాన్ని ఇస్తుంది.

11.3 డేటాబేస్

డేటాబేస్ అనేది డేటాను సులభంగా యాక్సిస్ చేయడానికి, నిర్వహించడానికి మరియు నవీకరించడానికి వీలుగా నిర్వహించే సంబంధిత డేటా యొక్క రిపోజిటరీ సేకరణ. డేటాను నిల్వచేయాలనే ఏకైక ఉద్దేశ్యంతో డేటాబేస్ అనేది సాఫ్ట్‌వేర్ ఆధారమవుతుంది.

11.4 డేటాబేస్ నిర్వహణ వ్యవస్థ (DBMS)

DBMS అనేది సాఫ్ట్‌వేర్, ఇది డేటాబేస్‌ను సృష్టించడానికి, నిర్వహించడానికి మరియు మార్చడానికి అనుమతిస్తుంది. డేటాను సులభంగా నిల్వచేయడానికి, ప్రాసెస్ చేయడానికి మరియు

12th_CS_Telugu_Medium_Chapter_11.indd 185 28-03-2023 10:04:59

విశ్లేషించడానికి వినియోగదారులను అనుమతిస్తుంది. డేటాబేస్‌ను సృష్టించడానికి, డేటాను నవీకరించడానికి వివిధ కార్యకలాపాలను నిర్వహించడానికి DBMS మాకు ఇంటర్ఫేస్ లేదా సాధనాన్ని అందిస్తుంది. DBMS డేటాబేస్‌లకు రక్షణ మరియు భద్రతను కూడా అందిస్తుంది. ఇది బహుళ వినియోగదారుల విషయంలో డేటా స్థిరత్వాన్ని కూడా నిర్వహిస్తుంది.

11.4.1 రిలేషనల్ డేటాబేస్ మేనేజ్‌మెంట్ సిస్టంమ్ (RDBMS)

RDBMS యొక్క నూతన వర్షన్ ఇది డేటాను మరింత సమర్థవంతమైన మార్గంలో అనుమతిస్తుంది. ఇది పట్టిక రూపంలో ఉన్న డేటాను నిల్వచేయడానికి మరియు మాన్యుపులేట్ చేయడానికి ఉపయోగపడుతుంది.

RDBMS సాఫ్ట్‌వేర్ యొక్క ఉదాహరణలు : ఫాక్స్ ప్రో, డి బేస్

11.4.2 RDBMS యొక్క లక్షణాలు

1. డేటాను మార్చగల సామర్థ్యం	RDBMS డేటాబేస్‌లో డేటాను (భద్రపరచడం, సవరించడం మరియు తొలగించడం) మార్చగల సామర్థ్య సదుపాయాన్ని అందిస్తుంది.
2. అదనపు (డిమాండ్ కన్నా ఎంచుక ఉన్నవాటిని) తగ్గించడం	ఆధునిక ప్రపంచములో హార్డ్‌డిస్కుల ధర చాలా తక్కువ. మునపటి కాలంలో హార్డ్‌డిస్క్‌లు చాలా ఖరీదైనవి. అందువల్ల డేటాబేస్‌లో అవాంఛిత డేటా చాలా సంక్లిష్టముగా ఉన్నది. కాబట్టి RDBMS డేటా విభజించినను సాధారణ స్థితిని అనుసరించటం ద్వారా అవాంఛిత డేటాను తిరిగి వచ్చుటను తగ్గిస్తుంది.
3. డేటా స్థిరత్వం	ప్రత్యక్ష డేటాలో ఇది నిరంతరం నవీకరించబడుతుంది మరియు జోడించబడుతుంది. డేటా యొక్క స్థిరత్వాన్ని నిర్వహించడం సవాలుగా మారుతుంది. కాని RDBMS దానిని స్వయంగా నిర్వహిస్తుంది.
4. మద్దతు బహుళ వినియోగదారు మరియు ఏకకాలిక విధానాన్ని పెంచుతుంది.	RDBMS దానిపై పనిచేయడానికి బహుళ వినియోగదారులను అనుమతిస్తుంది. (నవీకరించండి, చొప్పించండి, తొలగిస్తుంది) అదే సమయంలో ఇప్పటికి నిర్వహించడానికి డేటా స్థిరత్వం నిర్వహిస్తుంది.
5. ప్రశ్నభాష	RDBMS వినియోగదారులకు సాధారణ ప్రశ్న భాషను అందిస్తుంది. డేటాబేస్‌లో ఏ డేటాబేస్‌లో ఏ డేటా అయినా సులభంగా పొందవచ్చు, చొప్పించవచ్చు, తొలిగించవచ్చు మరియు నవీకరించవచ్చు.
6. భద్రత	అనధికార ప్రాప్యత నుండి డేటా యొక్క భద్రతను RDBMS కూడ చూసుకుంటుంది. ఒక సాధారణ RDBMS లో మనం వివిధ యాక్సెస్ అనుమతులతో వినియోగదారు ఖాతాలను సృష్టించవచ్చు. ఇది వినియోగదార ప్రాప్యతను పరిమితం చేయడం ద్వారా మా డేటాను సులభంగా భద్రపరచగలను.
7. RDBMS మద్దతు లావాదేవీలు	డేటా సమగ్రతను వాస్తవముగా నిర్వహించడానికి మరియు మంచిగా నిర్వహించడానికి ఇది అనుమతిస్తుంది. ఇక్కడ బహుళ-థ్రెడింగ్ విస్తృతముగా ఉపయోగించబడే ప్రపంచ అనువర్తనాలు.

12th_CS_Telugu_Medium_Chapter_11.indd 186 28-03-2023 10:05:00

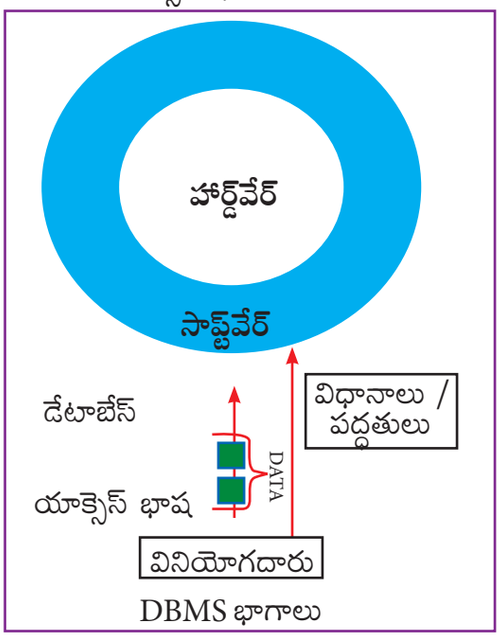
11.4.2 RDBMS యొక్క ప్రయోజనాలు

- ఆప్లికేషన్ ప్రోగ్రామ్ యొక్క విభజన
- కనిష్ట డేటా నకిలి లేదా డేటా తగ్గించడం
- ప్రశ్న భాషను ఉపయోగించి డేటాను సులభంగా తిరిగి పొందడం
- అభివృద్ధి సమయం మరియు నిర్వాహణ తగ్గుట

11.4.3 RDBMS యొక్క భాగాలు

డేటాబేస్ నిర్వాహణ వ్యవస్థను 5 ప్రధాన భాగాలుగా క్రింది విధములుగా విభజించవచ్చును.

1. హార్డ్‌వేర్ 2. సాఫ్ట్‌వేర్
3. డేటా 4. విధానాలు / పద్ధతులు
5. డేటాబేస్ యాక్సెస్ భాషలు



పటము 11.1

1. హార్డ్‌వేర్ (Hardware): కంప్యూటర్ హార్డ్‌డిస్క్ (I/O) డేటా కోసం ఛానెల్స్ మరియు ఏదైన ఇతర భౌతిక డేటా నిల్వలలో పాల్గొన్న భాగం

2. సాఫ్ట్‌వేర్ (Software): ప్రతిది నియంత్రించే ప్రోగ్రామ్‌లో ప్రధాన భాగము DBMS సాఫ్ట్‌వేర్ అర్థము చేసుకోగలదు. డేటాబేస్ యాక్సెస్ భాషలు మరియు అమలు కోసం డేటాబేస్ ఆదేశాలలోకి వివరిస్తుంది.

3. డేటా (Data): ఇది DBMS ను రూపొందించుటకు ఒక వనరు. DBMS సృష్టించి నిలువచేసి మరియు డేటాను ఉపయోగించుకోవడం.

4. విధానాలు / పద్ధతులు (Procedures/ Methods) : ఇది సాధారణమైన డేటాబేస్ నిర్వాహణను ఉపయోగించటానికి సూచనగా ఉంటుంది. DBMS యొక్క సంస్థాపన వంటి వ్యవస్థ, బ్యాకప్‌లు తీసుకోవటానికి డేటాబేస్‌లు నిర్వహించడం, తరం నివేదించడం మొదలైనవి.

5. డేటాబేస్ యాక్సెస్ భాషలు (DataBase Access Languages) : ఇవి డేటాబేస్‌లో ఏదైనా ఆదేశాలు వ్రాయడానికి ఉపయోగించే భాషలు, నిల్వచేసిన డేటాను ప్రాప్యత చేయడానికి, చొప్పించడానికి, నవీకరించడానికి మరియు తొలిగించడానికి దేనిని ఉపయోగిస్తారు.

ప్రసిద్ధి చెందిన RDBMS యొక్క ఉదాహరణలు:
Dbase, FoxPro

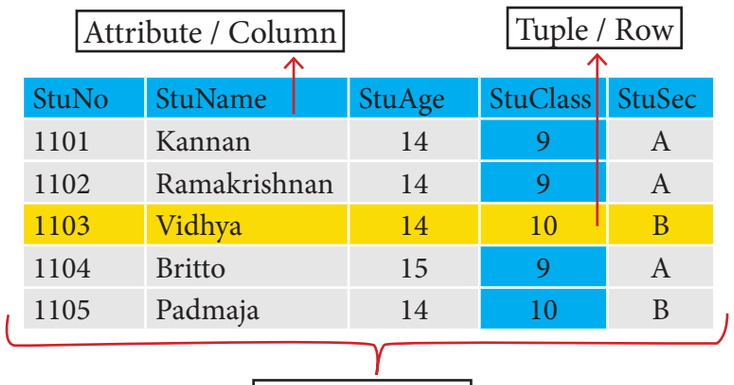
11.5 డేటాబేస్ నిర్మాణం

ఒక పట్టికలో సంబంధిత డేటా యొక్క మొత్తం సేకరణను ఒక ఫైల్‌గా సూచిస్తారు లేదా డేటాను అడ్డు మరియు నిలువ వరుసగా నిర్వహించే పట్టిక. పట్టికలోని ప్రతి అడ్డు వరుస ఒక రికార్డ్‌గాను, ఇది ప్రతి డేటాబేస్ ప్రవేశమునకు ఒక డేటా సమితిగా ఉంటుంది.

ప్రతి పట్టిక కాలము ఒక ఫీల్డ్‌ను (Field) సూచిస్తుంది. ఇది డేటా యొక్క ప్రతి భాగాన్ని లేదా అంశాన్ని సమూహపరుస్తుంది. నిర్దిష్ట వర్గాలలో, రికార్డులలో డేటా యొక్క రకములు : ఉదాహరణ : స్టూనో, స్టూనేమ్, స్టూఅజ్, స్టూక్లాస్, స్టూసెక్.

ఒక పట్టికను RELATION అంటారు.
ఒక అడ్డు వరుసను TUPLE అంటారు.
ఒక నిలువ వరుసను ATTRIBUTE అంటారు.

12th_CS_Telugu_Medium_Chapter_11.indd 187 28-03-2023 10:05:00



StuNo	StuName	StuAge	StuClass	StuSec
1101	Kannan	14	9	A
1102	Ramakrishnan	14	9	A
1103	Vidhya	14	10	B
1104	Britto	15	9	A
1105	Padmaja	14	10	B

DataBase Structure

పటము 11.2

11.6 డేటా నమూనా

- సాఫ్ట్‌వేర్ అమలు పూర్తయిన తర్వాత డేటాను ఎలా ప్రాతినిధ్యం వహించవచ్చు మరియు యాక్సెస్ చేయవచ్చు అనే విషయాన్ని డేటా నమూనా వివరిస్తుంది.
- వాతావరణం యొక్క సాధారణ సంగ్రహణ అనునది సంక్లిష్టమైన వాస్తవ ప్రపంచ డేటా సేకరణ అవుతుంది.
- డేటా నమూనా యొక్క ముఖ్య ఉద్దేశం, చివరి సాఫ్ట్‌వేర్ దాని అభివృద్ధి పూర్తి అయిన తర్వాత ఎలా ఉంటుందో ఒక ఆలోచన ఇస్తుంది.

11.6.1 డేటా నమూనా రకాలు

డేటా నమూనా యొక్క వివిధ రకాలు :

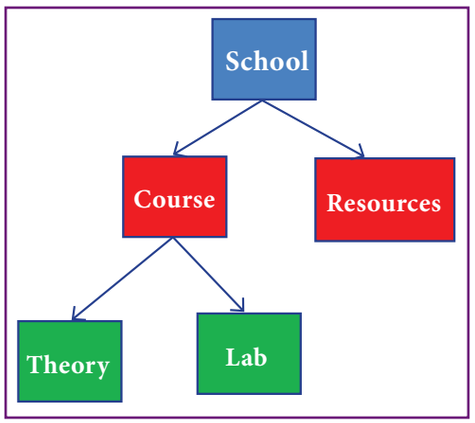
- క్రమానుగత నమూనా
- రిలేషనల్ (Relational Model) నమూనా
- నెట్‌వర్క్ డేటాబేస్ నమూనా
- (Entity Relationship) నమూనా (ఎంటిటి రిలేషన్ షిప్ మోడల్)
- (Object) నమూనా

1. క్రమానుగత నమూనా (Hierarchical Model):

క్రమానుగత నమూనాను (IBM) చేత (Information Management) గా విభజించబడింది.

క్రమానుగత నమూనాలో డేటా నిర్మాణ రూపమును చెట్టుగా సూచించబడుతుంది. ఇది ఒకటి నుండి అనేక సంబంధాలను సూచిస్తుంది. అంటే తల్లిదండ్రుల – పిల్లల సంబంధం. ఒక బిడ్డ ఒక పేరంట్ మాత్రమే కలిగి ఉంటారు. కాని ఒక పేరంట్ చాలా మంది పిల్లలను కలిగి ఉంటారు. ఈ మోడల్ ప్రధానంగా IBM ప్రధాన ప్రేమ్ కంప్యూటర్‌లో ఉపయోగించబడుతుంది.

12th_CS_Telugu_Medium_Chapter_11.indd 188 28-03-2023 10:05:00



Hierarchical Model పటము. 11.3

2. రిలేషన్ నమూనా (Relational Model):

రిలేషన్ డేటాబేస్ నమూనాను 1970లో E.F. కాడ్ మొదట ప్రతిపాదించారు. ఈ రోజుల్లో ఇది ప్రపంచ వ్యాప్తంగా డేటాబేస్ అనువర్తనాల కోసం ఉపయోగించే అత్యంత విస్తృతమైన డేటా నమూనా.

రిలేషన్ మోడల్‌లో డేటా యొక్క ప్రాథమిక నిర్మాణ పట్టికలు (సంబంధాలు). ఒక నిర్దిష్ట రకానికి సంబంధించిన ఆ పట్టిక యొక్క వరుసలలో అన్ని సమాచారముల నిల్వ చేయబడుతుంది. అందువల్ల రిలేషన్ నమూనాలోని రిలేషన్‌ను పట్టికలు అని కూడా అంటారు. రిలేషన్ కీ అనేది ప్రత్యేకంగా గుర్తించే లక్షణం ఒక ప్రత్యేకమైన టుపుల్ (సంబంధంలో అడ్డు వరుస పట్టిక)

Stu_id	Name	Age
1	Malar	17
2	Suncar	16
3	Velu	16

Subj_id	Name	Teacher
1	C++	Kannan
2	Php	Ramakrishnan
3	Python	Vidhya

Stu_id	Subj_id	Marks
1	1	92
1	2	89
3	2	96

Relational Model
పటము. 11.4

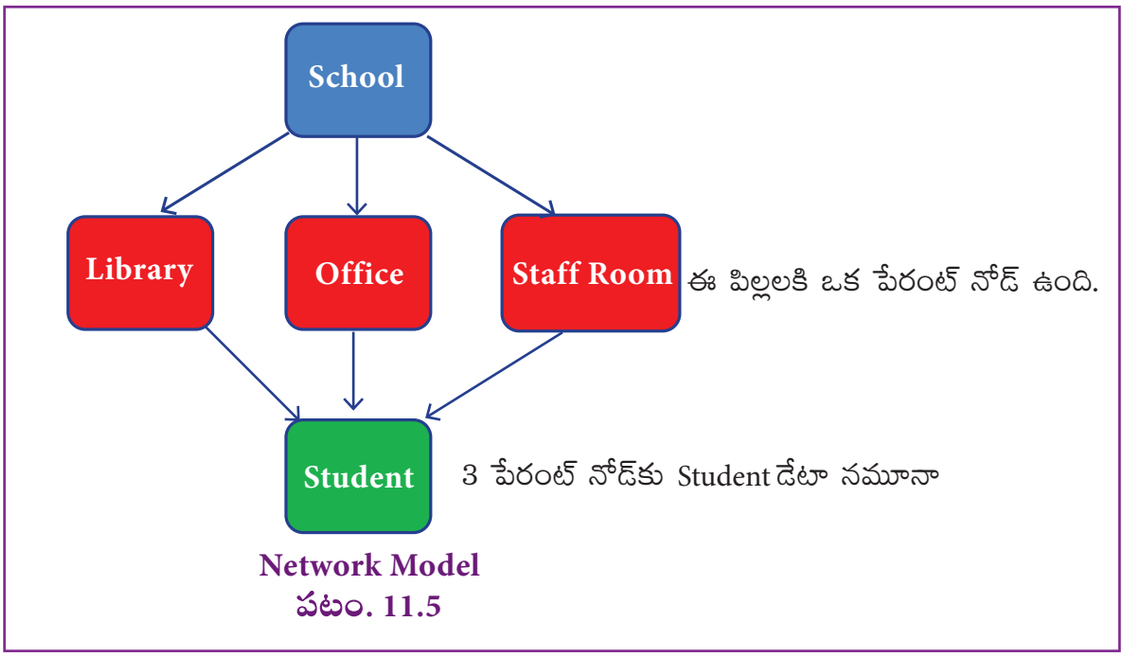
3. నెట్‌వర్క్ డేటాబేస్ నమూనా (Network Model):

నెట్‌వర్క్ డేటాబేస్ నమూనా క్రమానుగత డేటా నమూనా యొక్క విస్తరించిన రూపము క్రమానుగత మరియు నెట్‌వర్క్ డేటా నమూనాల మధ్య

- క్రమానుగత నమూనాలో పిల్లల రికార్డ్‌కు ఒకే ఒక పేరంట్ నోడ్ ఉంది.

12th_CS_Telugu_Medium_Chapter_11.indd 189 28-03-2023 10:05:00

- నెట్‌వర్క్ నమూనాలో పిల్లలకు చాలా పేరంట్ నోడ్స్ ఉండవచ్చు. ఈ డేటా Many -to- Many వరకు సంబంధమును సూచిస్తుంది.
- డేటాను యాక్సెస్ చేయడానికి ఈ నమూనా సులభం మరియు వేగంగా ఉంటుంది.



Network Model
పటం. 11.5

School పేరంట్ నోడ్‌ను తెలుపుతుంది.

Library, Office మరియు Staff room అనునది school యొక్క పిల్లలు (పేరంట్ నోడ్)

Student అనునది library మరియు staff room కు పిల్లలు

4. ER (ఎంటిటీ రిలేషన్ షిప్ నమూనా) model (Entity Relationship Model. (ER model)):

ఈ డేటాబేస్ నమూనాలో వస్తువును ఎంటిటీగా మరియు దాని Atttributes లక్షణాలుగా విభజించడం ద్వారా సంబంధం ఏర్పడుతుంది.

దీనిని 1976లో (Chen) (చెన్) అభివృద్ధి చేశారు. ఈ నమూనా డేటాబేస్‌కు Conceptual design అభివృద్ధికి ఉపయోగపడుతుంది. డేటా యొక్క తార్కిక వీక్షణను రూపకల్పన చేయడం చాలా సరళము మరియు సులభము. డెవలపర్ నిర్మించిన ER మోడల్‌ను చూడటం ద్వారా వ్యవస్థను సులభంగా అర్థం చేసుకోవచ్చు.

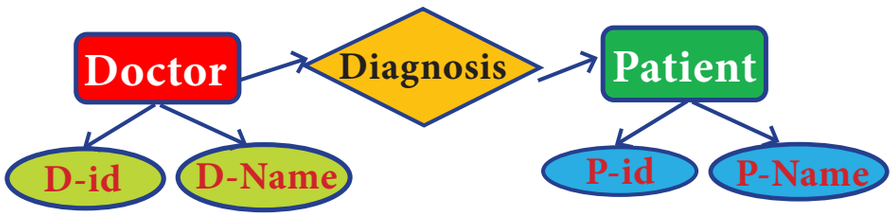
ధీర్ఘచతురస్రం ఎంటిటీలను సూచిస్తుంది. ఉదా : డాక్టర్ మరియు రోగి.

ధీర్ఘవృత్తం (Ellipse) లక్షణాలను సూచిస్తుంది. ఉదా: డి-ఐడి, డి-పేరు, పి-ఐడి, పి-పేరు లక్షణాలు గుణాలను వివరిస్తాయి. మరియు ప్రతి ఎంటిటీ డేటాబేస్లో నిల్వచేయబడిన డేటాలో ప్రధాన భాగం అవుతుంది.

డైమండ్ (ER) రేఖా చిత్రాలలో సంబంధాన్ని సూచిస్తుంది.

ఉదా: డాక్టర్ రోగిని నిర్ధారణ చేస్తారు.

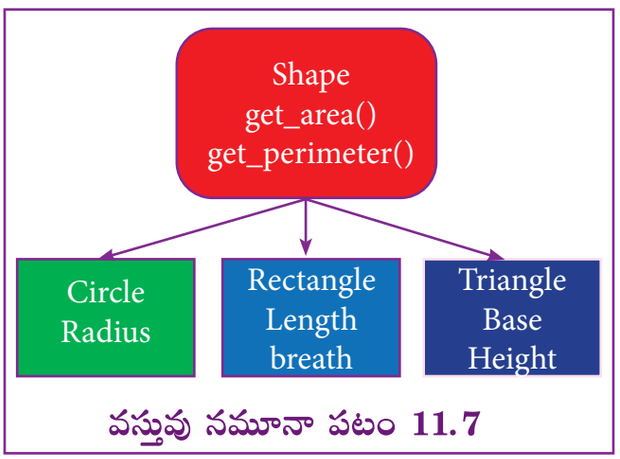
12th_CS_Telugu_Medium_Chapter_11.indd 190 28-03-2023 10:05:00



ER నమూనా పటము 11.6

5. వస్తువు నమూనా (Object Model):

వస్తువు నమూనా అనునది వస్తువులను, లక్షణాలను మరియు పద్ధతులు, తరగతులు మరియు వారసత్వ రూపంలో డేటాను నిల్వచేస్తుంది. ఈ నమూనా భౌగోళిక సమాచార వ్యవస్థ వంటి మరింత సంక్లిష్ట (complex) అనువర్తనాలను శాస్త్రీయ ప్రయోగాలు, ఇంజనీరింగ్ రూపకల్పన (design) మరియు తయారీలను నిర్వహిస్తుంది. ఇది ఫైల్ మేనేజ్‌మెంట్ సిస్టమ్‌లో ఉపయోగించబడుతుంది. ఇది నిజమైన ప్రపంచ వస్తువుల లక్షణాలను మరియు ప్రవర్తనలను సూచిస్తుంది. ఇది స్పష్టమైన నవీన (modular) నిర్మాణాన్ని అందిస్తుంది. ఇప్పటికే ఉన్న కోడ్‌ను నిర్వహించడం మరియు సవరించడం సులభం.



వస్తువు నమూనా పటం 11.7

వస్తువు నమూనాకు ఉదాహరణలు : **Shape, Circle, Rectangle** మరియు **Triangle** మొదలైనవి ఈ నమూనాలో ఉన్న వస్తువులు.

- **Circle** అనునది లక్షణ వ్యాసార్ధం కలిగి ఉంది.
- **Rectangle** పొడవు మరియు వెడల్పు లక్షణాలను కలిగి ఉంది.
- త్రిభుజంలో **base** మరియు **height** లక్షణాలు ఉన్నవి
- Circle, Rectangle మరియు Triangle అనే వస్తువులు Shape అను వస్తువుల నుండి వారసత్వంగా వస్తాయి.

11.6.2 DBMS వినియోగదారులు రకాలు :

డేటాబేస్ నిర్వాహకుడు (DBA)

డేటా బేస్ నిర్వాహకుడు లేదా DBA కూడా మొత్తం డేటాబేస్ నిర్వహణ వ్యవస్థను నిర్వహిస్తుంది. అతను DBMS భద్రతా లైసెన్స్‌లు, వినియోగదారు ఖాతాలు మరియు యాక్సెస్ మొదలైనవి జాగ్రత్తగా చూసుకుంటారు.

12th_CS_Telugu_Medium_Chapter_11.indd 191 28-03-2023 10:05:00

అప్లికేషన్ ప్రోగ్రామర్లు లేదా సాఫ్ట్‌వేర్ డెవలర్లు

ఈ వినియోగదారుల సమూహం DBMS భాగాలను అభివృద్ధి చేయడంలో మరియు రూపకల్పనలో పాల్గొంటారు.

తుది వినియోగదారుడు

అన్ని ఆధునిక అనువర్తనాలు, వెబ్ లేదా మొబైల్‌లో వినియోగదారుడు డేటాను నిల్వచేయండి. అనువర్తనాలు వినియోగదారు డేటాను సేకరించి వారి సర్వర్‌లో నడుస్తున్న DBMS సిస్టమ్‌లో డేటాను నిల్వచేసే విధంగా ప్రోగ్రామ్ చేయబడుతాయి. డేటాను నిల్వచేయడం, తిరిగి పొందడం, నవీకరించడం మరియు తొలగించడం తుది వినియోగదారులు.

డేటాబేస్ డిజైనర్లు :

డేటాను సూచించడానికి మరియు నిల్వచేయడానికి తగిన నిర్మాణాలను ఎన్నుకోవటానికి డేటాబేస్లో నిల్వచేయవలసిన డేటాను గుర్తించడానికి డేటాబేస్ డిజైనర్లు బాధ్యత వహిస్తారు.

11.7 DBMS మరియు RDBMS మధ్య గల వ్యత్యాసం

పోలిక యొక్క ఆధారం	DBMS	RDBMS
విస్తరణ	డేటాబేస్ నిర్వహణ వ్యవస్థ	RDBMS రిలేషనల్ డేటాబేస్ నిర్వహణ వ్యవస్థ
డేటానిల్వ	నావిగేషన్ నమూనా అనగా లింక్డ్ రికార్డుల ద్వారా డేటా	రిలేషనల్ నమూనా పట్టికలలో అనగా పట్టికలలోని నిలువు మరియు అడ్డు వరుసల డేటా
పునరావృత డేటా	ప్రదర్శించు	ప్రదర్శించబడలేదు
సాధారణీకరణ	బాగా పనిచేయలేదు	RDBMS పునరుక్తిని తగ్గించడానికి సాధారణీకరణను ఉపయోగిస్తుంది.
డేటా యాక్సెస్	ఎక్కువ సమయం తీసుకొంటుంది.	DBMS తో పోల్చిన వేగమైనది
ముఖ్యలక్ష్యాలు మరియు సూచికలు	ఉపయోగించబడదు	సంబంధాన్ని స్థాపించడానికి ఉపయోగిస్తారు. RDBMS లో ముఖ్య లక్ష్యాలను ఉపయోగిస్తారు.
లావాదేవీ నిర్వహణ	అసమర్థత లోపం సంభవించు మరయు సురక్షితమైనది కాదు	సమర్థవంతమైన మరియు సురక్షితమైనది
పంపిణీ చేసిన డేటాబేసెస్	మద్దతు లేదు	ద్వారా మద్దతు చేయబడుతుంది.
ఉదాహరణ	డిబేస్, ఫాక్స్ ప్రో	SQL సర్వర్, బరాకిల్, mysql మరియాడిబి (MariaDB) ఎస్‌క్యూలైట్ (SQLite)

12th_CS_Telugu_Medium_Chapter_11.indd 192 28-03-2023 10:05:00

డేటా పునరుక్తిని (redundancy) తగ్గించడానికి మరియు డేటా సమగ్రతను మెరుగుపరచడానికి RDBMS యొక్క అంతర్భాగంగా డేటాబేస్ సాధారణీకరణను మొదట డాక్టర్.ఎడ్గార్ ఎఫ్‌కోడ్ (***Dr. Edgar F Codd***) ప్రతిపాదించారు.

11.8 సంబంధాలు (Relationships) యొక్క రకాలు

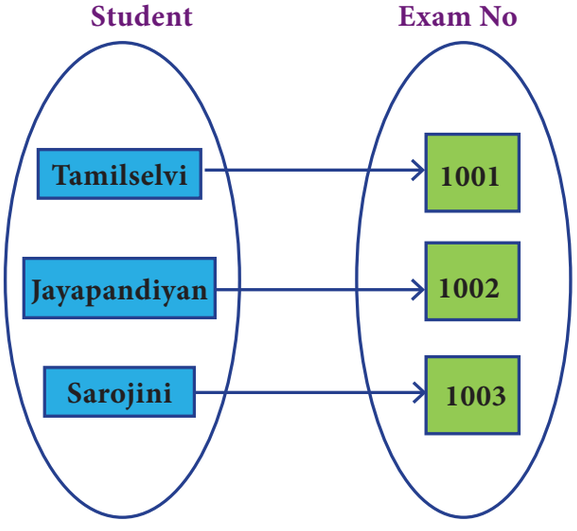
క్రింద ఇవ్వబడిన సంబంధాల రకాలను డేటాబేస్‌లో ఉపయోగిస్తున్నారు.

1. ఒకటి నుండి ఒకటి సంబంధం
2. ఒకటి నుండి అనేక సంబంధం
3. అనేక నుండి ఒక సంబంధం
4. అనేక నుండి అనేక సంబంధం

1. ఒకటి నుండి ఒక సంబంధం (One-to-One Relationship):

ఈ ఒకటి నుండి ఒకటి సంబంధంలో ఒక ఎంటిటీ మరొక ఎంటిటీతో మాత్రమే సంబంధం కలిగి ఉంటుంది. పట్టికలోని ఒక అడ్డు వరుస మరొక పట్టికలో ఒక అడ్డు వరుసతో మాత్రమే అనుసంధానించబడి ఉంటుంది. మరియు దీనికి విరుద్ధంగా.

ఉదాహరణ : ఒక విద్యార్థికి ఒక Exam no మాత్రమే ఉంటుంది.

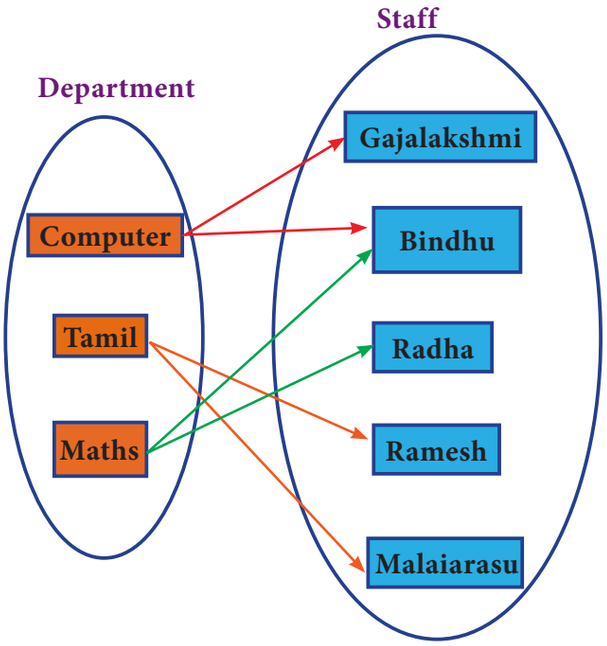


ఒకటి నుండి ఒకటి సంబంధం పటము 11.8

2. ఒకటి నుండి అనేక సంబంధం (One-to-Many Relationship):

ఈ ఒకటి నుండి అనేక సంబంధంలో ఒక ఎంటిటీ అనేక మరొక ఎంటిటీతో సంబంధం కలిగి ఉంటుంది.

పట్టికలో ఒక అడ్డు వరుస A పట్టికలోని అనేక వరుసలలో అనుసంధానించబడి ఉంది, కాని పట్టిక B లోని ఒక అడ్డు వరుస A లోని ఒక వరుసకు మాత్రమే అనుసంధానించబడి ఉంటుంది.



ఒకటి నుండి అనేక సంబంధం పటం 11.9

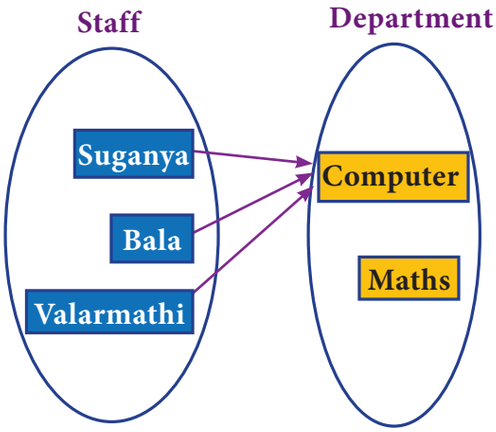
3. అనేక నుండి ఒక సంబంధం (Many-to-One Relationship):

అనేక నుండి ఒక సంబంధంలో అనేక ఎంటిటీలు ఇతర ఎంటిటీలో ఒక దానితో మాత్రమే సంబంధం కలిగి ఉంటాయి.

ఉదాహరణ : ఒక విభాగంలో పనిచేస్తున్న అనేక మంది సిబ్బంది.

సిబ్బంది సభ్యుల పట్టికలోని బహుళ వరుసలు సంస్థ పట్టికలోని ఒక వరుసతో మాత్రమే సంబంధం కలిగి ఉంటాయి.

12th_CS_Telugu_Medium_Chapter_11.indd 193 28-03-2023 10:05:00



అనేక నుండి ఒక సంబంధం పటం 11.10

4. అనేక నుండి అనేక సంబంధం (Many-to-Many Relationship)

పట్టికలోని బహుళ రికార్డులు మరొక పట్టికలోని బహుళ రికార్డులతో అనుబంధించబడినపుడు అనేక నుండి అనేక సంబంధం ఏర్పడుతుంది.

ఉదాహరణ 1 : వినియోగదారులు మరియు ఉత్పత్తి

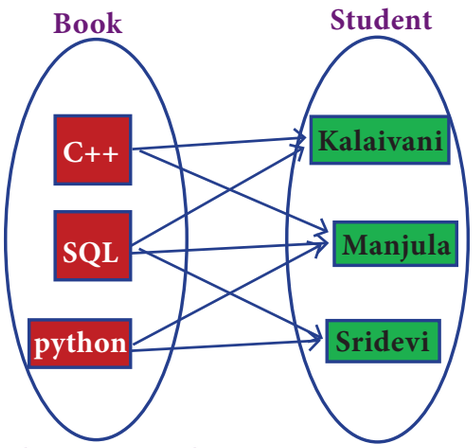
వినియోగదారులు వివిధ ఉత్పత్తులను కొనుగోలు చేయవచ్చు మరియు ఉత్పత్తులను చాలా మంది వినియోగదారులు కొనుగోలు చేయవచ్చు.

ఉదాహరణ 2 : ఒక విద్యార్థి అనేక కోర్సులకు

ఒక విద్యార్థి అనేక కోర్సులను నమోదు చేసుకోవచ్చు మరియు కోర్సులో చాలా మంది విద్యార్థులు ఉండవచ్చు.

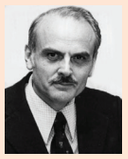
ఉదాహరణ 3: పుస్తకములు మరియు విద్యార్థి

గ్రంథాలయములోని చాలా పుస్తకాలు చాలా మంది విద్యార్థులకు జారీ చేయబడతాయి.



అనేక నుండి అనేక సంబంధం పటం 11.11

మీకు తెలుసా?



సంబంధం (relational model) నమూనాను ఎడ్గార్ ఫ్రాంక్ కాడ్ (సంబంధ డేటాబేస్ యొక్క తండ్రి) డేటా యొక్క సాధారణ నమూనాగా కనుగొన్నారు, తర్వాత క్రిస్ డేట్ మరియు హ్యూ డార్వెన్ ఇతరులు దీనిని ప్రోత్సహించారు.

11.9 DBMS లో సంబంధ Relational Algebra బీజగణితము

సంబంధ బీజగణితము అనగానేమి?

IBM (ఐ.బి.ఎమ్)లో ఉన్నప్పుడు ఎడ్గార్ ఎఫ్ కాడ్ చేత సంబంధ బీజగణితము మొదట సృష్టించబడింది. సంబంధ డేటాబేస్లో నిల్వచేసిన డేటాను మోడలింగ్ చేయడానికి మరియు దానిపై ప్రశ్నలను నిర్వచించడానికి ఇది ఉపయోగించబడింది.

సంబంధ బీజగణితము అనేది (SQL) ను ఉపయోగించి డేటాబేస్ పట్టికను ప్రశ్నించడానికి ఉపయోగించే ఒక విధానపరమైన ప్రశ్న భాష.

సంబంధ బీజగణిత కార్యకలాపాల అవుట్పుట్ను (output) ఇవ్వడానికి సంబంధించి పునరావృత్తమవుతాయి. ఈ కార్యకలాపాలు యొక్క అవుట్పుట్ కొత్త సంబంధం ఇది ఒకటి లేదా అంతకంటె ఎక్కువ ఇన్పుట్ (input) సంబంధాల ద్వారా ఏర్పడవచ్చు.

సంబంధ బీజగణితము అనేక గ్రూపులుగా విభజించబడినది.

అనామిక సంబంధిత కార్యకలాపాలు (Operations)

SELECT (symbol : σ)

PROJECT (symbol : $\prod$)

సెట్ (Set) సిద్ధాంతము నుండి సంబంధ బీజగణిత కార్యకలాపాలు (Operations)

- UNION ($\cup$) సమ్మేళనము
- INTERSECTION ($\cap$) ఖండనము

12th_CS_Telugu_Medium_Chapter_11.indd 194 28-03-2023 10:05:00

- DIFFERENCE $(-)$ వ్యత్యాసము
- CARTESIAN PRODUCT (X) కార్టీషియన్ లబ్ధము

SELECT (symbol : σ)

సాధారణ రూపము σ_c (R) అనునది సంబంధం మరియు దాని లక్షణాల షరతు

ఇచ్చిన షరతుల ప్రకారం టుపుల్స్‌తో ఉపసమితిని ఎంచుకోవడం కోసం (SELECT) కార్యకలాపాలను ఉపయోగిస్తారు.

C ని సంతృప్తిపరచని అన్ని టుపుల్స్ నుండి ఫిల్టర్‌ను ఎంచుకొండి.

STUDENT

Studno	Name	Course	Year
cs1	Kannan	Big Data	II
cs2	Gowri Shankar	R language	I
cs3	Lenin	Big Data	I
cs4	Padmaja	Python Programming	I

పట్టిక 11.1

σ_{course} = “Big Data” **(STUDENT)**

Studno	Name	Course	Year
cs1	Kannan	Big Data	II
cs3	Lenin	Big Data	I

PROJECT (symbol : Π)

ప్రొజెక్షన్ ఇన్‌పుట్ (input) రిలేషన్ యొక్క అన్ని లక్షణాలను తొలగిస్తుంది కాని ప్రొజెక్షన్ జాబితాలో పేర్కొన్నవి. ప్రొజెక్షన్ పద్ధతి రిలేషన్ యొక్క నిలువు ఉపసమితిని కలిగి ఉన్న సంబంధాన్ని నిర్వహిస్తుంది.

ఉదాహరణ 1 : (11.1 పట్టికను ఉపయోగించి)

Π_{course} (STUDENT)

Result

Course
Big Data
R language
Python Programming

గమనిక

ఫలితంలో నకిలీ అడ్డువరుస తొలగించబడుతుంది.

12th_CS_Telugu_Medium_Chapter_11.indd 195 28-03-2023 10:05:00

ఉదాహరణ 2 : (పట్టిక 11.1 ను ఉపయోగించి)

$\Pi_{\text{studno, course}}$ (STUDENT)

Result

Studno	Course
cs1	Big Data
cs2	R language
cs3	Big Data
cs4	Python Programming

సమ్మేళనం (Symbol :∪)

ఇది పట్టిక A లేదా B లో ఉన్న అన్ని టుపుల్స్‌ను కలిగి ఉంటుంది. ఇది నకిలీలను కూడా తొలగిస్తుంది. సమితి A సమ్మేళనము సమితి B ఒక $A \cup B$ గా వ్యక్తీకరించబడుతుంది.

ఉదాహరణ 3 :

క్రింది పట్టికను పరిశీలించండి.

Table A	
Studno	**Name**
cs1	Kannan
cs3	Lenin
cs4	Padmaja

Table B	
Studno	**Name**
cs1	Kannan
cs2	GowriShankarn
cs3	Lenin

పట్టిక 11.2.

ఫలితము (Result)

Table A ∪ B	
Studno	**Name**
cs1	Kannan
cs2	GowriShankar
cs3	Lenin
cs4	Padmaja

సమితి వ్యత్యాసము (Symbol : -)

A – B యొక్క ఫలితం A లో ఉన్న B లో లేని అన్ని టుపుల్స్‌ను కలిగి ఉన్న సంబంధం.

A యొక్క లక్షణం పేరు B లోని లక్షణపేరుతో సరిపోవాలి.

12th_CS_Telugu_Medium_Chapter_11.indd 196 28-03-2023 10:05:00

ఉదాహరణ 4 : (పట్టిక 11.2 ఉపయోగించి)
ఫలితము

Table A – B	
cs4	Padmaja

ఖండనము (symbol : ∩) A ∩ B

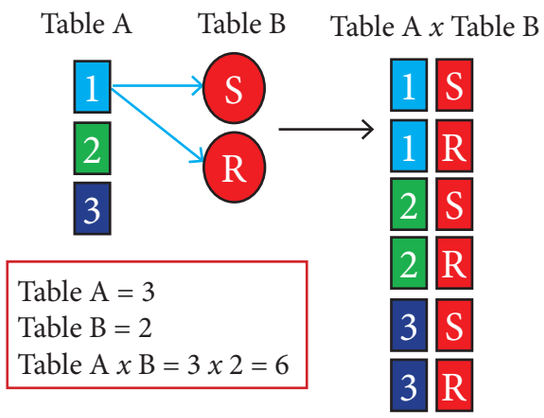
ఇది A మరియు B పట్టికలో సాధారణంగా ఉన్న అన్ని టుపుల్స్‌ను కలిగి ఉంటుంది.
ఉదాహరణ : (పట్టిక 11.2 ఉపయోగించి)

A ∩ B	
cs1	Kannan
cs3	Lenin

లబ్ధము లేదా కార్టీషియన్ లబ్ధము (Symbol : X) (PRODUCT OR CARTESIAN PRODUCT)

రెండు సంబంధాలను కలుపు మార్గం క్రాస్‌ప్రొడక్ట్ (Cross product) దీని ఫలితం రెండు సంబంధాల కలయికగా ఉంటుంది.

A x B అనునది A times B ఇక్కడ A మరియు B సంబంధం వేర్వేరు లక్షణాలను కలిగి ఉంటుంది. రెండు సంబంధాల నుండి నిలువు వరుసలను విలీనం చేయడానికి ఈ రకమైన ఆపరేషన్ (operation) సహాయపడుతుంది.



కార్టీషియన్ లబ్ధము
పటము 11.12

Table A		Table B	
studno	**name**	**course**	**subject**
cs1	Kannan	cs28	Big Data
cs2	Gowri Shankar	cs62	R language
cs4	Padmaja	cs25	python programming

పట్టిక 11.3

12th_CS_Telugu_Medium_Chapter_11.indd 197 28-03-2023 10:05:01

కార్టీషియన్ లబ్ధము : పట్టిక A x పట్టిక B

studno	name	course	subject
cs1	Kannan	cs28	Big Data
cs1	Kannan	cs62	R language
cs1	Kannan	cs25	python rogramming
cs2	Gowri Shankar	cs28	Big Data
cs2	Gowri Shankar	cs62	R language
cs2	Gowri Shankar	cs25	python programming
cs4	Padmaja	cs28	Big Data
cs4	Padmaja	cs62	R language
cs4	Padmaja	cs25	python programming

గుర్తించుకోవలసిన విషయములు:

- DBMS అనేది కంప్యూటర్ ఆధారత రికార్డ్ కపింగ్ వ్యవస్థ (system) (రికార్డులను ఉంచు వ్యవస్థ)
- డేటా ప్రాసెస్ (process) చేయని డేటా ఇందులో అక్షరం, వచనం, పదం లేదా సంఖ్యకు అర్థం లేదు.
- సమాచారం ప్రాసెస్ (process) చేయబడిన డేటా, వ్యవస్థీకృత మరియు ఆకృతీకరణ.
- mysql, oracle, sql server, ibm db2 అనునవి RDBMS యొక్క ఉదాహరణలు.
- పునరుక్తి అంటే డేటాబేస్లో డేటా యొక్క నకిలీ.
- డేటా స్థిరత్వం అంటే డేటాబేస్ యొక్క అన్ని సందర్భాల్లో డేటా విలువలు ఒకే విధంగా ఉంటాయి.
- డేటా సమగ్రత అనునది అనధికార వినియోగదారుల నుండి భద్రత.
- పట్టిక అనునది ఒక సంబంధం.
- ఒక వరుసను టుపుల్ అంటారు
- నిలువు వరుసను లక్షణంగా పిలుస్తారు.
- డేటా మోడల్ రకాలు క్రమానుగత, రిలేషనల్, నెట్వర్క్, ER మరియు ఆబ్జెక్ట్ మోడల్ (Object model)
- క్రమానుగత నమూనా అనేది పేరంట్–చైల్డ్ రిలేషన్ (Parent-Child Relation) అని పిలువబడే ఒకదానితో ఒకటి సంబంధంతో నిర్మాణ రూపం వంటి సాధారణ చెట్టు.
- రిలేషన్ మోడల్ డేటాను సంబంధాలు లేదా పట్టికలను (సంబంధ నమూనా) సూచిస్తుంది.
- నెట్వర్క్ మోడల్ (Network model) క్రమానుగత మోడల్తో సమానంగా ఉంటుంది. అయితే ఇది రికార్డుకు ఒకటి కంటె ఎక్కువ తల్లిదండ్రులను కలిగి ఉండటానికి అనుమతిస్తుంది.
- (ER) మోడల్ ఎంటిటీలు, లక్షణాలు మరియు సంబంధాలను కలిగి ఉంటుంది.
- ఆబ్జెక్ట్ మోడల్ డేటాను వస్తువులు, లక్షణాలు, పద్ధతులు, తరగతులు మరియు వారసత్వంగా నిల్వచేస్తుంది.

12th_CS_Telugu_Medium_Chapter_11.indd 198 28-03-2023 10:05:01

గుర్తించుకోవలసిన విషయములు:

- సాధారణీకరణ డేటా పునరుక్తిని (redundancy) తగ్గిస్తుంది మరియు డేటా సమగ్రతను మెరుగుపరుస్తుంది.
- వివిధ రకాలైన సంబంధాలు అనునవి ఒకటి నుండి ఒకటి, ఒకటి నుండి అనేక, అనేకం నుండి ఒకటి మరియు అనేకం నుండి అనేకం సంబంధాలు.
- డేటాబేస్ సాధారణీకరణను డాక్టర్.ఎడ్గార్.ఎఫ్ కోడ్ ప్రతిపాదించారు.
- సంబంధ బీజగణితము సంబంధ (Relational Algebra) డేటాబేస్లో నిల్వచేయబడిన డేటాను మోడలింగ్ చేయడానికి మరియు దానిపై ప్రశ్నలను నిర్వచించడానికి ఉపయోగిస్తారు.

మూల్యాంకనం

DGKEV

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. DBMS యొక్క సంక్షిప్త నామము ఏమి?

   a) డేటాబేస్ నిర్వహణ చిహ్నం b) డేటాబేస్‌ను నిర్వహించే వ్యవస్థ

   c) డేటాబేస్ నిర్వహణ వ్యవస్థ d) డేటాబేసిక్ (DataBasic) నిర్వహణ వ్యవస్థ

2. ఒక పట్టిక అనునది

   a) tuple టుపుల్ b) లక్షణాలు c) సంబంధం d) ఎంటిటీ

3. ఏ డేటాబేస్ నమూనా (database model) తల్లిదండ్రుల – పిల్లల సంబంధమును సూచిస్తుంది.

   a) రిలేషనల్ b) నెట్‌వర్క్ c) క్రమానుగత d) వస్తువు

4. రిలేషనల్ డేటాబేస్ నమూనాను (database model) మొదట ప్రతిపాదించినది.

   a) E F కోడ్ b) E E కోడ్ c) E F కాడ్ d) E F కోడర్

5. క్రమానుగత నమూనా ఏ రకమైన సంబంధాన్ని సూచిస్తుంది?

   a) ఒకటి నుండి ఒకటి b) ఒకటి నుండి అనేక

   c) అనేక నుండి ఒకటి d) అనేక నుండి అనేక

6. క్రింది వాటి నుండి రిలేషనల్ డేటాబేస్ యొక్క తండ్రి అని ఎవరిని పిలుస్తారు.

   a) క్రిస్ డేట్ b) హ్యూ డార్విన్ c) ఎడ్గార్ ఫ్రాంక్ కోడ్ d) ఎడ్గార్ ఫ్రాంక్ కాడ్

7. క్రిందివాటిలో RDBMS ఏది?

a) ది బేస్ b) ఫాక్స్‌ప్రో (Foxpro) c) Mongo DB d) SQLite (ఎస్‌క్యూ లైట్)

8. SELECT statement కు ఏ గుర్తును ఉపయోగిస్తారు.

a) σ b) Π c) X d) Ω

9. ఒక టుపుల్‌ను ఈ విధంగా కూడా అంటారు.

a) పట్టిక b) అడ్డు వరుస c) లక్షణం d) ఫీల్డ్ (field)

10. ER నమూనాను model ఎవరు అభివృద్ధి చేశారు?

a) చెన్ b) EF కోడ్ d) చెండ్ (Chend) c) చంద్ (Chend)

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. DBMS యొక్క కొన్ని ఉదాహరణలు పేర్కొనుము?
2. RDBMS యొక్క కొన్ని ఉదాహరణలను జాబితా చేయండి.
3. డేటా స్థిరత్వం అంటే ఏమిటి?
4. క్రమానుగత మరియు నెట్‌వర్క్ డేటా నమూనా మధ్య తేడా ఏమిటి?
5. సాధారణీకరణ అంటే ఏమిటి?

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. సెలెక్ట్ (Select) మరియు ప్రాజెక్ట్ (Project) మధ్య తేడా ఏమిటి?
2. DBA పాత్ర ఏమిటి?
3. తగిన ఉదాహరణతో కార్టీషియన్ లబ్ధమును వివరించుము?
4. ఆబ్జెక్ట్ నమూనాను (Object Model) ఉదాహరణతో వివరించుము?
5. వివిధ రకాల DBMS వినియోగదారులపై సంచికను (note) వ్రాయుము.

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. వివిధ రకాల డేటా నమూనను (model) వివరించండి?
2. వివిధ రకాలైన రిలేషన్ మ్యాపింగ్ (relationship mapping) గురించి వివరించండి?
3. DBMS మరియు RDBMS మధ్యగల భేదాలను వివరించండి?
4. సంబంధ బీజగణితములోని వేర్వేరు ఆపరేటర్లను తగిన ఉదాహరణలతో వివరించండి?
5. DBMS యొక్క లక్షణాలను వివరించండి?

12th_CS_Telugu_Medium_Chapter_11.indd 200 28-03-2023 10:05:01

యూనిట్ IV

అధ్యాయం 12

నిర్మాణాత్మక ప్రశ్నాభాష

OPE46

అభ్యసనా లక్ష్యములు

ఈ పాఠాన్ని అధ్యయనం చేసిన తర్వాత విద్యార్థులు అర్థం చేసుకొనునవి.

- SQL యొక్క ప్రాసెసింగ్ నైపుణ్యాలు
- SQL యొక్క భాగాలు
- ఫీల్డులు (క్షేత్రాలు) మరియు రికార్డులు యుండునట్లు ఒక పట్టికను సృష్టించడం
- ఒక పట్టికలో రికార్డులను చేర్చడం, నవీనీకరించడం మరియు తొలగించడం వంటి అవకతవకలను వర్తింపజేయడం.
- వివిధ నిబంధనలను తెలుసుకోవడం, వాటిని పట్టికలతో వర్తింపజేయడాన్ని అభ్యసించడం.
- వివిధ నిబంధనలను వర్తింపజేయడం ద్వారా పట్టికలో ప్రశ్నలను రూపొందించడం.
- ఇప్పటికే ఉన్న పట్టిక యొక్క నిర్మాణాన్ని సవరించడం.
- పట్టిక మరియు రికార్డులను తొలగించడానికి ఆదేశాలు మరియు వాటిని ఉపసంహరించుకోవడానికి ఆదేశాలు.

12.1 SQL పరిచయం

నిర్మాణాత్మక ప్రశ్నా భాష SQL అనునది డేటాబేసస్‌ను (దత్తాంశనిధిని) పొందడానికి (యాక్సస్ చేయడానికి) మరియు మార్చడానికి ఒక ప్రామాణిక ప్రోగ్రామింగ్ భాష. డేటాబేస్లలోని సమాచారాన్ని సృష్టించడానికి, తిరిగి పొందడానికి, మార్చడానికి మరియు బదిలీ చేయడానికి వినియోగదారునుకి SQL అనుమతిస్తుంది. ఇది రిలేషనల్ డేటాబేస్ మేనేజ్‌మెంట్ సిస్టమ్ (RDBMS) లో డేటాను నిర్వహించడం మరియు పొందడం కోసం రూపొందించిన భాష.

SQL లో అనేక సంస్కరణలు (versions) గలవు. అసలు సంస్కరణ IBM యొక్క పరిశోధనా కేంద్రంలో అభివృద్ధి చేయబడినది మరియు మొదట దీనిని 1970ల ప్రారంభంలో సీక్వెల్ అని పిలిచిరి. తర్వాత ఈ భాష 1986లో SQL గా మార్చబడినది, ANSI (అమెరికన్ నేషనల్ స్టాండర్డ్ ఇన్స్టిట్యూబ్) ఒక SQL ప్రమాణాన్ని 1992లో నవీనీకరించబడింది, 2008లో ప్రస్తుతమున్న SQL విడుదలై, SQL 2008 గా పేరు పెట్టబడింది.

గమనిక

ప్రస్తుతమున్న SQL ప్రమాణం 2008లో విడుదలైన SQL 2008

12th_CS_Telugu_Medium_Chapter_12.indd 201 28-03-2023 10:05:45

12.2 RDBMS లో SQL యొక్క పాత్ర

RDBMS అంటే రిలేషనల్ డేటాబేస్ మేనేజ్‌మెంట్ సిస్టమ్ Relational DataBase Management System. Oracle, MySQL, MS SQL Server, IBM DB2 మరియు Microsoft Access మొదలగునవి RDBMS ప్యాకేజీలగును. SQL అనునది అటువంటి డేటాబేస్లో డేటాను యాక్సెస్ చేయడానికి ఉపయోగించే భాష.

సాధారణంగా, డేటాబేస్ అనేది ఇతర అనువర్తనాలలో ఉపయోగం కోసం ప్రశ్నించగల డేటా సమితులను నిల్వ చేసే పట్టికల సమాహారం అగును. డేటాబేస్ నిర్వహణ వ్యవస్థ డేటాబేస్ ఉపరితలముల అభివృద్ధి, నిర్వహణ మరియు ఉపయోగానికి మద్దతునిస్తుంది. RDBMS అనేది వరుస ఆధారత పట్టిక నిర్మాణంతో కూడిన DBMS యొక్క రకం, ఇది సంబంధిత డేటా మూలకాలను అనుసంధానిస్తుంది. మరియు Create, Read, Update మరియు Delete కార్యకలాపాల సమిష్ఠని CRUD అని పిలుతురు.

RDBMS లోని డేటా, డేటాబేస్ వస్తువులు (objects) గా నిల్వచేయబడు వాటిని పట్టికలు (Tables) అందురు. పట్టిక అనేది సంబంధిత డేటా ఎంట్రీల సమాహారం మరియు ఇది అడ్డు వరుసలు మరియు నిలువు వరుసలను కలిగి యుండును.

ఫీల్డ్ (క్షేత్రము) (field) అనేది పట్టికలోని ప్రతి రికార్డు గురించి నిర్ధిష్ట సంబంధిత సమాచారాన్ని నిర్వహించడానికి రూపొందించబడిన పట్టికలోని గడి. ఇది ఒక నిలువు వాస్తవికత ఇది పట్టికలోని నిర్ధిష్ట క్షేత్రం (ఫీల్డ్)తో అనుబంధించబడిన మొత్తం సమాచారాన్ని కలిగి ఉండును. విద్యార్థి పట్టికలోని క్షేత్రం (ఫీల్డ్) లు AdmnNo, StudName, StudAge, StudClass, Place మొదలైన వాటికి చెందినవి కావచ్చు.

ఒక పట్టికలో ఉన్న సంబంధిత ఫీల్డ్‌లు లేదా నిలువు వరుసల సమాహారం వరుసను రికార్డు అందురు. రికార్డ్ అనేది పట్టికలోని సమాంతర వాస్తవికత, ఇది విద్యార్థి పట్టికలోని ఒక నిర్ధిష్ట విద్యార్థి వివరాలను సూచిస్తుంది.

12.3 SQL యొక్క ప్రాసెసింగ్ నైపుణ్యాలు

SQL యొక్క ప్రాసెసింగ్ నైపుణ్యాలు:

1. **డేటా డెఫినిషన్ లాంగ్వేజ్ (DDL)** (దత్తాంశ నిర్వచన భాష) సంబంధిత వ్యూహాలను నిర్వచించడం, సంబంధాలను తీసివేయడం, సూచికలను సృష్టించడం మరియు సంబంధిత వ్యూహాలను సవరించడం మొదలగు వాటికి SQL DDL ఆదేశాలను అందిస్తుంది.

2. **డేటా మానిపులేషన్ లాంగ్వేజ్ (దత్తాంశ తారుమారు భాష) (DML) :** డేటాబేస్లో టుపుల్స్ చొప్పించడానికి, తొలగించడానికి మరియు సవరించడానికి ఆదేశాలను కలిగి ఉంటుంది.

3. **పొందుపరచబడిన డేటా తారుమారు భాష EDML** (ఎంబెడెడ్ డేటా మానిప్యులేషన్ లాంగ్వేజ్) SQL యొక్క పొందుపరచబడిన (ఎంబెడెడ్) రూపం ఉన్నత స్థాయి ప్రోగ్రామింగ్ భాషలలో ఉపయోగపడుతుంది.

4. **నిర్వచనాన్ని వీక్షించండి (View Definition) :** పట్టికల అభిప్రాయాలను నిర్వచించడానికి SQL ఆదేశాలను కూడ కలిగి ఉంటుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 202 28-03-2023 10:05:45

5. **అంగీకారం (Authorization) :** పట్టికలు యొక్క సంబంధాలు మరియు వీక్షణలు (అభిప్రాయాలకు) ప్రాప్యత హక్కుల కోసం SQL ఆదేశాలను కలిగి ఉంటుంది.

6. **సమగ్రత (Integrity) :** SQL పరిస్థితిని అనుసరించి సమగ్రత తనిఖీ యొక్క రూపాలను అందిస్తుంది.

7. **లావాదేవీల నియంత్రణ (Transaction control) :** SQL ఫైలు file లావాదేవీలు మరియు లావాదేవీల ప్రక్రియపై నియంత్రణ కోసం ఆదేశాలను కలిగి ఉంటుంది.

మీకు తెలుసా?

SQL- నిర్మాణాత్మక ప్రశ్నాభాష అనేది డేటాబేసస్‌ను యాక్సెస్ చేయడానికి ఉపయోగించే భాష. అయితే **MySQL** అనునది Server, Oracle, Informix, Postgres వంటి డేటాబేస్ మేనేజ్‌మెంట్ సిస్టమ్. **MySQL** ఒక **RDBMS** అగును.

MYSQL ని సంస్థాపనం **installing** చేయడాన్ని అనుబంధం –1లో చూడండి.

12.4 డేటాబేస్‌ను సృష్టించడం(Creating Database)



డేటాబేస్‌లను సృష్టించడానికి

1. ప్రాంప్ట్‌లో క్రింది ఆదేశాలను టైపు చేయండి.

   CREATE DATABASE database_name;

   ఉదాహరణకు CREATE DATABASE stud;

2. డేటాబేస్ (దత్తాంశము)తో పనిచేయడానికి, క్రింది ఆదేశాన్ని టైప్ చేయండి.

   USE DATABASE;

   ఉదాహరణకు సృష్టించబడిన stud అను దత్తాంశమును ఉపయోగించడానికి,

   USE stud; అను ఆదేశాన్ని ఇవ్వండి.

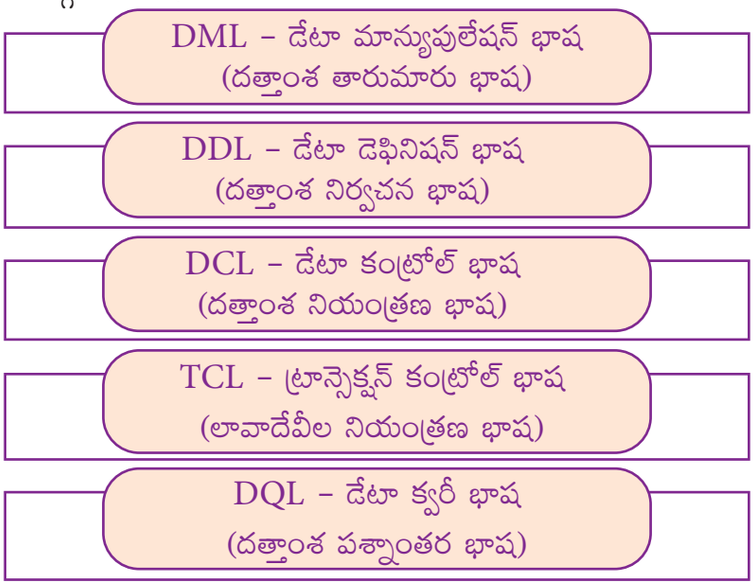
మీకు తెలుసా?

WAMP అంటే **"Windows, Apache, MySQL** మరియు **PHP". WAMP** అనేది విండోస్ సిస్టమ్స్ కోసం ఏర్పాటైన భిన్నత్వమే (వైవిధ్యమే) LAMP మరియు ఇది సాఫ్ట్‌వేర్ అంశాలుగా (Apache, MySQL and PHP) సంస్థాపన చేయబడింది. దీనిని తరచుగా వెబ్ అభివృద్ధి మరియు అంతర్గత శోధనల కొరకు ఉపయోగపడుతుంది. కానీ దీనిని ప్రత్యక్ష వెబ్‌సైట్‌లకు సేవచేయడానికి కూడా ఉపయోగించవచ్చు.

12th_CS_Telugu_Medium_Chapter_12.indd 203 28-03-2023 10:05:46

12.5 SQL యొక్క భాగాలు

SQL ఆదేశాలను ఐదు వర్గాలుగా విభజించారు.



12.5.1 దత్తాంశ నిర్వచన భాష (DATA DEFINITION LANGUAGE)

దత్తాంశ నిర్వచన భాష డేటాబేస్ నిర్మాణం లేదా సంక్షిప్తీకరణ (వ్యూహ). ఇది డేటాబేస్ వ్యూహాల యొక్క వివరణలతో వ్యవహరిస్తుంది. మరియు డేటాబేస్‌లలో, డేటాబేస్ వస్తువుల నిర్మాణాన్ని సృష్టించడానికి మరియు సవరించడానికి ఉపయోగపడుతుంది.

డేటాబేస్ సిస్టమ్ ఉపయోగించే నిల్వనిర్మాణం (స్టోరేజ్ స్ట్రక్చర్) మరియు Access పద్ధతులను పేర్కొనడానికి DDL నిర్వచనాల సమితిని అందిస్తుంది.

ఒక DDL క్రింది విధులను నిర్వహిస్తుంది:

1. డేటా అంశం, విభాగం, రికార్డ్ మరియు డేటాబేస్ ఫైలు వంటి డేటా డివిషన్ రకాన్ని గుర్తించాలి.
2. ఇది ప్రతి డేటా అంశం, రికార్డు, ఫైలు మరియు డేటాబేస్ రకాలకు ప్రత్యేకమైన పేరును ఇస్తుంది.
3. ఇది సరైన డేటా రకాన్ని పేర్కొనాలి.
4. ఇది డేటా అంశం యొక్క పరిమాణాన్ని నిర్వచించాలి.
5. ఇది డేటా అంశం ఉపయోగించగల విలువల పరిధిని నిర్వచించవచ్చు.
6. ఇది అనధికార డేటా ఎంట్రీని నిరోధించడానికి గోప్యతా తాళాలను పేర్కొనవచ్చు.

డేటా డెఫినిషన్ లాంగ్వేజ్ (దత్తాంశ నిర్వచన భాష) కింద వచ్చే SQL ఆదేశాలు :

Create	డేటాబేస్లో పట్టికలను సృష్టించడానికి
Alter	డేటాబేస్ యొక్క నిర్మాణాన్ని మార్చడానికి
Drop	డేటాబేస్ నుండి పట్టికలు తొలగించడానికి
Truncate	ఒక పట్టికలోని అన్ని రికార్డులను తొలగించును, ఆ రికార్డులు ఆక్రమించిన స్థలాన్ని కూడా విడుదల చేయును.

12th_CS_Telugu_Medium_Chapter_12.indd 204 28-03-2023 10:05:46

12.5.2 దత్తాంశ నిర్వహణ భాష (డేటా మానిప్యులేషన్ భాష) DML :

డేటా మానిప్యులేషన్ భాష అనునది డేటాబేస్లో డేటాను జోడించడం, తొలగించడం మరియు సవరించడానికి ఉపయోగించే కంప్యూటర్ ప్రోగ్రామింగ్ భాష.

SQL లో డేటా నిర్వహణ భాష డేటాబేస్లో భద్రపరచిన డేటాను సవరించడానికి ఉపయోగించిన SQL డేటా ప్రత్యామ్నాయ ప్రకటనలను కలిగి ఉంటుంది. కాని ఇది డేటాబేస్ పట్టిక యొక్క ప్రాజెక్ట్ డిజైన్ కాదు.

డేటాబేస్ను సృష్టించిన తర్వాత డేటాబేస్ ప్రాజెక్ట్ డిజైన్ ప్రకారం DML లో పేర్కొన్నట్లు ప్రక్రియల సమితిని ఉపయోగించి డేటాను మార్చవచ్చు.

దత్తాంశ నిర్వహణ అనునది

- డేటాబేస్లో క్రొత్త సమాచారములను చేర్చడం
- డేటాబేస్లో భద్రపరిచిన సమాచారమును తిరిగిపొందడం
- డేటాబేస్లో నుండి సమాచారాన్ని తొలగించడం
- డేటాబేస్లో నిల్వచేయబడిన సమాచారాన్ని మార్పు చేయడం

DML ప్రాథమికంగా రెండు రకాలు:

1. విధానపరమైన DML (ప్రొసీడ్యూరల్ DML) – ఏ డేటా అవసరం మరియు ఎలా పొందాలో పేర్కొనడానికి వినియోగదారుడు అవసరం.

2. విధానపరమైన కాని DML (నాన్–ప్రొసీడ్యూరల్ DML) – ఎలా పొందాలో పేర్కొనకుండా ఏ డేటా అవసరమో పేర్కొనడానికి వినియోగదారుడు అవసరం.

దత్తాంశ నిర్వహణ భాష (డేటా మానిప్యులేషన్ లాంగ్వేజ్) కింద వచ్చే SQL ఆదేశాలు :

Insert	డేటాను పట్టికలో చేర్చడం.
Update	పట్టికలో ఉన్న డేటాను నవీనీకరించడం
Delete	పట్టిక నుండి అన్ని రికార్డులను తొలగిస్తుంది. కాని ఆక్రమించిన స్థలం కాదు.

12.5.3 డేటా నియంత్రణ భాష (డేటా కంట్రోల్ లాంగ్వేజ్)

డేటా నియంత్రణ భాష (DCL) అనేది డేటాబేస్లో నిల్వచేయబడిన డేటా యొక్క ప్రాప్యతను నియంత్రించడానికి ఉపయోగించే ప్రోగ్రామింగ్ భాష. డేటాబేస్లోని అధికారులను నియంత్రించడానికి ఇది ఉపయోగపడుతుంది. సన్నివేశాలను సృష్టించడం, పట్టికలను వీక్షించడం మొదలగు అన్ని డేటాబేస్ కార్యకలాపాలను నిర్వహించడానికి అధికారాలు అవసరం.

డేటా కంట్రోల్ లాంగ్వేజ్ పరిధిలోకి వచ్చే SQL ఆదేశాలు :

Grant	నిర్దిష్ట పనులను చేయడానికి ఒకటి లేదా అంతకంటే ఎక్కువమంది వినియోగదారులకు అనుమతి మంజూరు చేయడం.
Revoke	స్టేట్మెంట్ (నివేదిక) GRANT ఆర్డర్ ఇచ్చిన యాక్సెస్ రద్దు చేయబడుతుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 205 28-03-2023 10:05:46

12.5.4 లావాదేవీల నియంత్రణ భాష (Transfer Control Language) (TCL)

డేటాబేస్‌లోని లావాదేవీలను నిర్వహించడానికి లావాదేవీ నియంత్రణ భాష (TCL) ఉపయోగపడతాయి. DML స్టేట్‌మెంట్స్ వాక్యాల ద్వారా పట్టికలోని డేటాకు చేసిన మార్పులను నిర్వహించడానికి ఇవి ఉపయోగపడతాయి.

బదిలీ నియంత్రణ భాష (Transfer Control Language) క్రిందకు వచ్చే SQL ఆదేశాలు:

COMMIT	ఏదైన లావాదేవీని డేటాబేస్లో శాశ్వతంగా భద్రపరచును.
ROLL BACK	డేటాబేసును చివరి నిబద్ధ స్థితికి పునరుద్ధరిస్తుంది.
SAVE POINT	లావాదేవీని తాత్కాలింగా సేవ్ చేయండి, తద్వారా మీరు రోల్‌బ్యాక్ చేయవచ్చు.

12.5.5 దత్తాంశ ప్రశ్నా భాష (DATA QUERY LANGUAGE) (DQL)

డేటా ప్రశ్నా భాష, ఒక డేటాబేస్ నుండి డేటాను ప్రశ్నించడానికి లేదా తిరిగి పొందడానికి ఉపయోగించే ఆదేశాలను కలిగి ఉంటుంది. డేటా ప్రశ్నాభాషలో అటువంటి SQL ఆదేశం SELECT ఒకటి.

SELECT	ఇది పట్టిక నుండి రికార్డులను ప్రదర్శిస్తుంది.

12.6 డేటా రకాలు

డేటాబేసులోని డేటా దానిలో సేవ్ చేయబడిన విలువ ఆధారంగా సేవ్ చేయబడుతుంది. డేటా యొక్క డేటా రకంగా లేదా ప్రతి ఫీల్డ్‌కు డేటా రకాన్ని గుర్తించడం ద్వారా గుర్తించబడుతుంది. ఇచ్చిన ఫీల్డ్‌లోని అన్ని విలువలు ఒకే రకంగా ఉండాలి.

ANSI, SQL ప్రమాణం Text మరియు Number డేటా రకాన్ని మాత్రమే గుర్తిస్తుంది. కొన్ని వాణిజ్య కార్యక్రమాలు తేది మరియు సమయం వంటి ఇతర డేటా రకాలను ఉపయోగిస్తాయి. ANSI డేటా రకాలు పట్టిక 12.1 లో ఇవ్వబడ్డాయి.

డేటా రకము	వివరణ
char (Character)	స్థిర వెడల్పు కలిగిన స్ట్రింగ్ విలువ. ఈ రకమైన విలువ ఏకవచనములో గుర్తించబడును. (ఉదా) Anu's అనునది 'Anu' 's'. గుర్తించబడును.
varchar	అసిర్థమైన (వేరియబుల్) వెడల్పు కలిగిన అక్షర స్ట్రింగ్ ఇది char డేటా రకం లాంటిది అయితే డేటా పరిమాణం ప్రకారం రికార్డు పరిమాణం మారుతుంది.
dec (Decimal)	ఇది 15.12, 0.123 వంటి దశాంశ సంఖ్యలను సూచిస్తుంది. ఇక్కడ పరిమాణం రెండు భాగాలను కలిగి ఉంటుంది. ఖచ్చితత్వం మరియు ప్రమాణాలు (కొలతలు) ఖచ్చితత్వం సంఖ్య ఎన్ని అంకెలు కలిగి ఉండవచ్చు అని సూచిస్తుంది. మరియు స్కేల్ దశాంశ బిందువు కుడివైపున ఉన్న అంకెల సంఖ్యను సూచిస్తుంది. పరిమాణం (5,2) లో ఖచ్చితత్వం 5గా, స్కేలు 2గా సూచిస్తుంది. స్కేల్, ఖచ్చితత్వాన్ని మించకూడదు.
numeric	ఇది decimal డేటా రకాన్ని పోలి ఉంటుంది. కానీ గరిష్ట సంఖ్యల సంఖ్య ఖచ్చితత్వానికి మించి ఉండకపోవచ్చు.

12th_CS_Telugu_Medium_Chapter_12.indd 206 28-03-2023 10:05:46

int (Integer)	ఇది దశంశ బిందువు లేని పూర్ణాంశాన్ని సూచించును. ఇక్కడ value argument ఉపయోగంలో లేదు.
smallint	ఇది పూర్ణాంకము వలె ఉండును కాని సాధారణ పరిమాణం పూర్ణాంకం కంటే తక్కువగా ఉండవచ్చు.
float	ఇది బేస్ 10 ఘాతాంక సంజ్ఞామానములో ఫ్లోటింగ్ పాయింట్ సంఖ్యను సూచిస్తుంది. మరియు గరిష్టంగా ఖచ్చితత్వం 64 అంకెలు వరకు ఉంటుందని నిర్వచించవచ్చు.
real	ఇది Float వలె ఉంటుంది. ఖచ్చితత్వం గరిష్టంగా 64 వరకు ఉంటుంది. కాని size argument ఉపయోగించబడును.
double	ఇది real వలె ఉంటుంది. కాని ఖచ్చితత్వం 64కు దాటవచ్చు.

పట్టిక *12.1*

12.7 SQL ఆదేశాలు మరియు దాని విధులు

డేటాను నిల్వచేయుటకు పట్టికలు మాత్రమే మార్గం అందువల్ల మొత్తం సమాచారం పట్టికల రూపంలో అమర్చాలి. డేటాబేస్ పైన పనిచేయడానికి SQL ముందుగా నిర్ణయించిన ఆదేశాల సమితిని అందిస్తుంది.

Keywords	వీటికి SQL లో ప్రత్యేకమైన అర్థం కలదు. వాటిని సూచనలుగా అర్థం చేసుకొందురు.
Commands	వినియోగదారునిచే డేటాబేసుకు ఇవ్వబడిన సూచనలను statements అందురు.
Clauses	ఇవి keyword తో ప్రారంభమగును. మరియు ఇందు keyword మరియు argument ఉండును.
Arguments	ఇవి నిబంధనను పూర్తిచేయడానికి ఇచ్చిన విలువలు.

12.7.1 DDL ఆదేశాలు

CREATE TABLE ఆదేశము

CREATE TABLE ఆదేశమును ఉపయోగించి పట్టిక (table) ను సృష్టంచవచ్చు. ఒక పట్టికను సృష్టించినపుడు, దాని నిలువు వరుసలకు పేరు పెట్టబడింది. డేటా రకాలు మరియు పరిమాణాలు గుర్తించబడాలి. ప్రతి పట్టికలో కనీసం ఒక column ఉండాలి. CREATE TABLE ఆదేశం యొక్క వాక్య నిర్మాణం.

CREATE TABLE *<table-name>*
(<column name><data type>[<size>]
(<column name><data type>[<size>].....
);

12th_CS_Telugu_Medium_Chapter_12.indd 207 28-03-2023 10:05:46

ఇప్పుడు ఒక తరగతి విద్యార్థుల గురించి కొంత సమాచారాన్ని డేటాబేస్‌లో నిల్వచేయడానికి పై వాక్య నిర్మాణాన్ని ఉపయోగిద్దాం. దీని కోసం మీరు మొదట పట్టికను సృష్టించాలి. విద్యార్థి పట్టికను సృష్టించడానికి, ప్రవేశసంఖ్య వంటి విద్యార్థులకు సంబంధించిన కొంత సమాచారాన్ని తీసుకుందాం. దానిని మనం సూక్ష్మ రూపంలో (admno) విద్యార్థిపేరు (name) మొదలైనవిగా ఉపయోగించవచ్చు. Admno, Name, Gender, Age మరియు Place ఫీల్డులు కలిగి ఉన్న పట్టికను సృష్టిద్దాం.

SQL ఆదేశం క్రింది విధంగా ఉంటుంది.

```
CREATE TABLE Student
    (Admno   integer,
     Name  char(20),
     Gender  char(1),
     Age     integer,
     Place char(10),
     );
```

పైన పేర్కొన్నది ఎటువంటి పరిమితులు లేని సాధారణ పట్టిక నిర్మాణం ఒక పట్టికలో నిల్వయున్న ఫీల్డ్‌లోని వెళ్ళే డేటారకాన్ని పరిమితం చేయడానికి కొన్ని అడ్డంకులను సెట్ చేయవచ్చు. పట్టికలోకి వెళ్ళే డేటా రకాన్ని పరిమితం చేయడానికి constraints (పరిమితులు) ఉపయోగించబడతాయి. ఇది డేటా బేస్లోని డేటా యొక్క ఖచ్చితత్వం మరియు విశ్వసనీయతను నిర్ధారిస్తుంది. పరిమితులు column స్థాయిలో లేదా table స్థాయిలో ఉండవచ్చును.

గమనిక

పరిమితి అనేది ఫీల్డ్ లేదా ఫీల్డ్‌ల సమితిపై వర్తించే షరతు.

నిలువ వరుస పరిమితి	నిలువ వరుస పరిమితి ప్రత్యేకమైన నిలువ వరుసకు మాత్రమే వర్తిస్తుంది.
పట్టిక పరిమితి	ఒకటి లేదా అంతకంటే ఎక్కువ నిలువవరుసల సమూహానికి పట్టిక పరిమితి వర్తిస్తుంది.

పరిమితి (constraint) తో సృష్టించబడిన పట్టిక యొక్క వాక్య నిర్మాణం (syntax) క్రింద ఇవ్వబడింది.

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]<column constraint>,
(<column name><data type>[<size>]<column constraint>……
<table constraint>(<column name>,[<column name>….])…..
);
```

క్రింది ఉదాహరణ “NOT NULL” నిలువ వరుస పరిమితితో విద్యార్థి పట్టిక. ఈ పరిమితి ఎల్లప్పుడూ విలువను కలిగి ఉండటానికి ఫీల్డ్‌ను అమలుపరుస్తుంది.

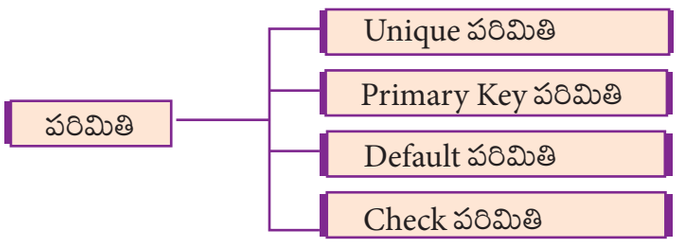
12th_CS_Telugu_Medium_Chapter_12.indd 208 28-03-2023 10:05:46

```
CREATE TABLE Student
(
Admno integer    NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20)NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

పై ఆదేశాలను ఉపయోగించి సృష్టించబడిన “విద్యార్థి” అను పట్టిక యందు integer రకమునకు చెందిన Admno అను ఫీల్డు, char రకమునకు చెందిన Name అను ఫీల్డు, "NOT NULL" అను నిర్వచింపబడింది. అంటే ఈ రెండు ఫీల్డులకు (క్షేత్రాలకు) విలువలు ఉండాలి. Gender, Age మరియు Place అనే ఫీల్డులకు (క్షేత్రాలకు) ఎటువంటి అవరోధాలు లేవు.

12.7.2 పరిమితుల (అవరోధముల) రకాలు

డేటాబేస్ సమగ్రతను పరిమితులు నిర్ధారిస్తాయి, కాబట్టి దేనిని డేటాబేస్ సమగ్రత పరిమితులు అంటారు. వివిధ రకములైన పరిమితులు.



12.7.2.1 Unique పరిమితి (ప్రత్యేక పరిమితి)

ఈ పరిమితి పేర్కొన్న నిలువు వరుసలలో ఏ రెండు వరుసలకు ఒకే విలువ లేదని నిర్ధారిస్తుంది ఉదాహరణకు student పట్టికలో Admno అను ఫీల్డ్ నందు Unique పరిమితిని ఉపయోగించి, ఇద్దరు విద్యార్థులకు ఒకే ప్రవేశ సంఖ్య (Admn no) లేదని నిర్ధారిస్తుంది. ఈ పరిమితిని క్రింది విధంగా ఉపయోగించుకొనవచ్చును.

```
CREATE TABLE Student
(
Admno integer NOT NULL UNIQUE, → Unique constraint
Name char (20) NOT NULL,
Gender char (1),
Age integer,
Place char (10),
);
```

UNIQUE పరిమితి NOT NULL అని ప్రకటించబడిన క్షేత్రాలకు (ఫీల్డ్‌లకు) మాత్రమే వర్తిస్తుంది.

ఒక ప్రత్యేకమైన క్షేత్రము (ఫీల్డ్)నకు రెండు పరిమితులు వర్తించినపుడు, దీనిని బహుళ పరిమితులు అంటారు. పైన పేర్కొన్న బహుళ పరిమితులలో NOT NULL మరియు UNIQUE అను బహుళ పరిమితులు ఒకే ఫీల్డ్ Admno లో ఉపయోగించబడ్డాయి. పరిమితుల మధ్య space తో వేరుచేయబడతాయి. మరియు

12th_CS_Telugu_Medium_Chapter_12.indd 209 28-03-2023 10:05:46

ఫీల్డ్ (క్షేత్ర) నిర్వచనం చివరిలో కామా (,) జోడించబడుతుంది. Admno అను ఫీల్డు (క్షేత్రము)కు ఈ రెండు పరిమితులు జోడించడం ద్వారా, ఈ ఫీల్డ్ Admno ఏదైన కొంత విలువను కలిగియుండవలయును. అదే సమయంలో NULL కాని నకిలీ కానివిగా యుండరాదు.

12.7.2.2 Primary Key పరిమితి

ఈ పరిమితి ఒక ఫీల్డును ప్రాథమిక కీ (Primary key) గా ప్రకటిస్తుంది. ఇది రికార్డును ప్రత్యేకంగా గుర్తించడానికి సహాయపడుతుంది. ఇది ఒక పట్టికలోని ఒక ఫీల్డ్‌ను మాత్రమే ప్రాథమిక కీగా అమర్చగల ప్రత్యేక పరిమితిని పోలి ఉండును ప్రాథమిక కీ NULL విలువలను అనుమతించదు.

విద్యార్థి పట్టికలో Primary Key పరిమితిని చూపించే ఉదాహరణ

```
CREATE TABLE Student
(
Admno integer    NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20)NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

పై ఉదాహరణలో Admno ఫీల్డు ప్రాథమిక కీగా అమర్చబడినది. అందువలన రికార్డును ప్రత్యేకంగా గుర్తించడంలో మనకు సహాయపడుతుంది. ఇది NOT NULL గా అమర్చబడినది కావున ఫీల్డు విలువ ఖాళీగా యుండరాదు.

12.7.2.3 DEFAULT పరిమితి

ఒక ఫీల్డు నందు పూర్వస్థితి (Default) విలువలను కేటాయించడానికి DEFAULT పరిమితి ఉపయోగించబడుచున్నది. DEFAULT పరిమితి కేటాయించిన ఫీల్డుకు విలువను ఇవ్వనపుడు, స్వయంచాలకంగా పూర్వస్థితి (DEFAULT) విలువ ఫీల్డు కు కేటాయించబడుతుంది.

విద్యార్థి పట్టికలో DEFAULT పరిమితిని చూపించే ఉదాహరణ :

```
CREATE TABLE Student
(
Admno integer    PRIMARY KEY,
Name char(20)NOT NULL,
Gender char(1),
Age integer  DEFAULT 17, → Default Constraint
Place char(10),
);
```


12th_CS_Telugu_Medium_Chapter_12.indd 210 28-03-2023 10:05:46

పై ఉదాహరణలో "Age" అను ఫీల్డు నకు 17 అని default value కేటాయించినందున, వినియోగదారుడు ఆ ఫీల్డులో విలువను నమోదు చేయనపుడు, Age అను ఫీల్డు అది స్వయంచాలకంగా 17 అను విలువను కేటాయిస్తుంది.

12.7.2.4 CHECK (నిర్బంధ) పరిమితి

CHECK పరిమితి ఒక క్షేత్రము (field) లోని పరిమితి విలువను నిర్ణయించడానికి సహాయపడుతుంది. CHECK పరిమితిని ఒకే నిలువ వరుస (column) కు నిర్వచించినపుడు, ఆ క్షేత్రంలో పరిమితం చేయబడిన విలువలను మాత్రమే అనుమతిస్తుంది.

విద్యార్థి పట్టికలో CHECK పరిమితిని చూపించే ఉదాహరణ :

```
CREATE TABLE Student
(
Admno integer   PRIMARY KEY
Name char(20)NOT NULL,
Gender char(1),
Age integer CHECK (Age <=19), → Check Constraint
Place char(10),
);
```

పై ఉదాహరణలో, 'Age' అను క్షేత్రంలో CHECK పరిమితిని ఉపయోగించి, Age అను క్షేత్రం field లో పరిమితం చేయబడిన విలువ 19 కంటే తక్కువ లేదా సమానంగా ఉండాలి అని నిర్వచించబడియున్నది.

గమనిక

షరతు కొఅకు సంబంధిత మరియు తార్కిక ప్రక్రియలకు CHECK పరిమితి ఉపయోగపడుతుంది.

12.7.2.5 పట్టిక పరిమితి (TABLE CONSTRAINT)

ఒక పట్టికలోని క్షేత్రాల (field) సమూహానికి పరిమితిని వర్తించినపుడు, దానిని పట్టిక పరిమితి (table constraint) అందురు. సాధారణంగా పట్టిక పరిమితి పట్టిక నిర్వచనం చివరిలో ఇవ్వబడుతుంది. Admno, Firstname, Lastname, Gender, Age, Place అను క్షేత్రములతో field Student1 అను ఒక క్రొత్త పట్టికను తీసుకుందాం.

```
CREATE TABLE Student 1
(
Admno integer        NOT NULL,
Firstname char(20),
Lastname char(20),
Gender char(1),
Age integer,
Place char(10),
PRIMARY KEY (Firstname, Lastname) → Table constraint
);
```

పై ఉదాహరణలో, Firstname మరియు Lastname అనే క్షేత్రాలు Primary key పట్టిక పరిమితిగా నిర్వచించబడ్డాయి.

12.7.3 DML ఆదేశాలు

ఒక పట్టిక యొక్క స్కీమా లేదా నిర్మాణం సృష్టించబడిన తర్వాత, విలువలను పట్టికకు చేర్చవచ్చు. DML ఆదేశాలు, పట్టికలోని అడ్డు వరుసలను చొప్పించడం, తొలగించడం మరియు నవీనీకరించడం మొదలైన ఆదేశాలు కలిగి ఉంటాయి.

12.7.3.1 INSERT ఆదేశం

డేటాబేసుకు కొత్త డేటాను చేర్చడానికి లేదా పట్టికలో కొత్త రికార్డులను చేర్చడానికి INSERT ఆదేశం సహాయపడుతుంది. ఈ ఆదేశంను ఏ విధంగా ఉపయోగించబడుతుంది క్రింద ఇవ్వబడినది.

INSERT INTO <table-name> [column-list] VALUES (values);

INSERT INTO Student (Admno, Name, Gender, Age, Place)
VALUES (100,' Ashish',' M', 17,' Chennai');
INSERT INTO Student (Admno, Name, Gender, Age, Place)
VALUES (101, 'Adarsh', 'M', 18, 'Delhi');

క్రింద చూపిన విధంగా రెండు కొత్త రికార్డులు పట్టికకు చేర్చబడ్డాయి.

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi

విలువల క్రమం తప్పనిసరిగా CREATE TABLE ఆదేశంలోని నిలువు వరుస క్రమంతో సరిపోవునట్లు యుండవలయును. అన్ని నిలువు వరుసలకు డేటా చేర్చాలి అంటే నిలువు వరస column పేర్లను పేర్కొనడం ఐచ్ఛికం. విద్యార్థి పట్టికలో విలువలను insert చేయు ఆదేశాన్ని ఈ క్రింది విధంగా ఉపయోగించవచ్చు.

INSERT INTO Student VALUES (102, 'Akshith', 'M', '17,' 'Bangalore');

102	Akshith	M	17	Bangalore

పై ఆదేశం రికార్డును విద్యార్థి పట్టికలో చేర్చును.

ఒక రికార్డులోని కొన్ని నిలువు వరుసలకు (columns) మాత్రమే డేటాను చేర్చడానికి columns పేరు మరియు వాటి డేటాను మాత్రమే పేర్కొనడం ద్వారా వీటిని చేయవచ్చు.

INSERT INTO Student(Admno, Name, Place) VALUES (103, 'Ayush', 'Delhi');

103	Ayush	M	18	Delhi

పై ఆదేశం 'M' మరియు Age డిఫాల్ట్ విలువలుగా M మరియు 18గా జతచేస్తుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 212 28-03-2023 10:05:46

INSERT INTO Student (Admno, Name, Place) VALUES (104, 'Abinandh', 'Chennai');

104	Abinandh	M	18	Chennai

విద్యార్థి పట్టిక క్రింది డేటాను కలిగి ఉంటుంది:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai

INSERT ఆదేశంలో ఇవ్వని క్షేత్రాలు (ఫీల్డులు) నిర్వచించి డిఫాల్ట్ విలువలను తీసుకుంటాయి. లేనిచో NULL విలువను తీసుకొనును.

మీకు తెలుసా?

INSERT ఆదేశంలో విస్మరించబడిన ఫీల్డులు నిర్వచించిన డిఫాల్ట్ విలువను లేదా NULL విలువను కలిగి యుండును.

12.7.3.2 DELETE ఆదేశం

DELETE ఆదేశం పట్టిక నుండి ఒకటి లేదా అంతకంటె ఎక్కువ రికార్డులను శాశ్వతంగా తొలగిస్తుంది. ఇది మొత్తం అడ్డు వరుసను తొలగిస్తుంది, అడ్డు వరుసలోని ప్రత్యేక ఫీల్డును కాదు, కావున ఫీల్డు argument అవసరం లేదు. DELETE ఆదేశం క్రింది విధంగా ఉపయోగించబడుతుంది.

DELETE FROM table-name WHERE condition;

ఉదాహరణకు : admon 104 అయిన రికార్డును తొలగించడానికి ఆదేశం ఈ క్రింది విధంగా ఇవ్వబడింది.

DELETE FROM Student WHERE Admno=104;

104	Abinandh	M	18	Chennai

విద్యార్థి పట్టిక నుండి ఈ రికార్డు తొలగించబడినది.

పట్టిక యొక్క అన్ని అడ్డు వరుసలను తొలగించడానికి,

ఉపయోగించు ఆదేశం :

DELETE * FROM Student;

ఇప్పుడు పట్టిక ఖాళీగా కనబడును. మరియు DROP ఆదేశాన్ని ఉపయోగించి సంపూర్ణముగా నాశనము చేయవచ్చును. (విభాగం 12.7.4.5 లో చర్చించబడింది.)

12th_CS_Telugu_Medium_Chapter_12.indd 213 28-03-2023 10:05:46

12.7.3.3 UPDATE ఆదేశం

UPDATE ఆదేశం, డేటాబేస్‌లోని కొన్ని లేదా అన్ని డేటా విలువలను నవీనీకరిస్తుంది. ఇది పట్టికలోని ఒకటి లేదా అంతకంటే ఎక్కువ రికార్డులను నవీనీకరించగలదు. UPDATE ఆదేశం, WHERE clause (నిబంధన) ఉపయోగించి మార్చవలసిన అడ్డువరుసలను, SET ప్రత్యేక పదమును ఉపయోగించి క్రొత్త డేటాను నిర్దేశిస్తుంది UPDATE ఆదేశంను క్రింది విధంగా ఉపయోగించబడుతుంది.

UPDATE <table-name> SET column-name = value, column-name = value,... WHERE condition;

ఉదాహరణకు క్రింది ఫీల్డులను (క్షేత్రాలను) నవీనీకరించడానికి

UPDATE Student SET Age = 20 WHERE Place = "Bangalore";

పై ఆదేశం "Bangalore" అను స్థలములోని విద్యార్థుల వయస్సు 20 అని మార్చి అమర్చబడును. నవీనీకరించబడిన పట్టిక క్రింది విధంగా :

The table will be as updated as below:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	20	Bangalore
103	Ayush	M	18	Delhi

బహుళ క్షేత్రాలను (ఫీల్డులను) నవీనీకరించడానికి, SET నిబంధన ఉపయోగించి వాటి విలువలను కామాతో (,) వేరు చేసి పేర్కొనవచ్చును. ఉదాహరణకు, ఒక విద్యార్థి పట్టికలో బహుళక్షేత్రాలను నవీనీకరించడానికి, ఆదేశం ఇలా ఇవ్వబడింది.

UPDATE Student SET Age=18, Place = 'Chennai' WHERE Admno = 102;

102	Akshith	M	18	Chennai

పై ఆదేశం ఈ క్రింది విధంగా రికార్డును సవరించును.

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	18	Chennai
103	Ayush	M	18	Delhi

12.7.4 కొన్ని అదనపు DDL ఆదేశాలు

12.7.4.1 ALTER ఆదేశం

ALTER ఆదేశంను ఉపయోగించి పట్టిక నిర్మాణములో నిలువ వరుసను జోడించడం, ఉన్న column పేరు మార్చడం, ఏ column లో నైన డేటా రకాన్ని లేదా పరిమాణంను మార్చడం, పట్టిక నుండి తొలగించుట వంటి మార్పులను చేయవచ్చును. అది క్రింది విధంగా ఉపయోగించబడుతుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 214 28-03-2023 10:05:46

ALTER TABLE <table-name> ADD <column-name><data type><size>;

విద్యార్థి పట్టికలో 'char' కు సంబంధించిన ఫీల్డు "Address" అను column చేర్చుటకు ఆదేశం ఇలా ఉపయోగపడుతుంది.

ALTER TABLE Student ADD Address char;

పట్టిక యందు ఇప్పటికే ఉన్న నిలువు వరుసను సవరించడానికి, ALTER TABLE ఆదేశాన్ని MODIFY నిబంధనతో ఉపయోగించవచ్చు:

ALTER TABLE <table-name> MODIFY<column-name><data type><size>;

ALTER TABLE Student MODIFY Address char (25);

పై ఆదేశం విద్యార్థి పట్టిక యందు address అడ్డువరుస ఇప్పుడు 25 అక్షరాలను కలిగి ఉండటానికి సవరించును.

ఇప్పటికే ఉన్న column పేరును ఈ విధంగా మార్చడానికి ALTER ఆదేశాన్ని ఉపయోగించవచ్చును.

ALTER TABLE <table-name> CHANGE old-column-name new-column-name new column definition;

ఉదాహరణకు, Address అను పేరు గల column కు City అని పేరు మార్చుటకు తోడ్పడు ఆదేశం:

ALTER TABLE Student CHANGE Address City char(20);

ఒక నిలువు వరుస (column) లేదా అన్ని నిలువవరుసలను తొలగించడానికి ALTER ఆదేశం ఉపయోగించవచ్చు. ఉదాహరణకు, ALTER TABLE తో DROP COLUMN ఉపయోగించి, నిర్ధిష్ట కాలమ్, నిర్ధిష్ట ఫీల్డ్‌ను తొలగించవచ్చును. ఆదేశాన్ని క్రింది విధంగా ఉపయోగించవచ్చు.

ALTER <table-name> DROP COLUMN <column-name>;

విద్యార్థి పట్టిక నుండి City అను కాలమ్‌ను తొలగించడానికి ఆదేశం ఇలా ఉపయోగించబడుతుంది.

ALTER TABLE Student DROP COLUMN City;

12.7.4.2 TRUNCATE command

TRUNCATE ఆదేశం, పట్టికలోని అన్ని అడ్డువరుసలను తొలగించడానికి ఉపయోగించబడుతుంది. పట్టిక యొక్క నిర్మాణం మారకుండ, స్థలం ఖాళీ అవుతుంది TRUNCATE ఆదేశం కోసం వాక్యనిర్మాణం :

TRUNCATE TABLE table-name;

ఉదాహరణకు, విద్యార్థి పట్టిక యొక్క అన్ని రికార్డులను తొలగించి, పట్టికను తొలగించడానికి SQL వాక్యము ఈ క్రింది విధంగా ఇవ్వబడింది.

TRUNCATE TABLE Student;

Student పట్టిక తొలగించబడి, స్థలము ఖాళీ చేయబడుతుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 215 28-03-2023 10:05:46

12.7.4.3 DROP TABLE ఆదేశం

డేటాబేస్ నుండి పట్టికను తొలగించడానికి DROP TABLE ఆదేశం ఉపయోగించబడుతుంది. మీరు పట్టికను DROP చేసిన పట్టికలోని అన్ని అడ్డువరసులు తొలగించబడి పట్టిక యొక్క నిర్మాణం వేటాబేస్ నుండి తొలగించబడుతుంది. ఒకసారి పట్టికను DROP చేసిన తర్వాత మనం దానిని తిరిగి పొందలేము, కావున, DROP TABLE ఆదేశాన్ని ఉపయోగించునపుడు హెచ్చరికతో యుండవలయును.

అన్ని అడ్డువరుసలను తొలగించిన తర్వాత, క్రింది విధంగా DROP TABLE ఆదేశాన్ని ఉపయోగించి పట్టికను తొలగించవచ్చును.

DROP TABLE table-name;

ఉదాహరణకు, విద్యార్థి పట్టికను తొలగించడానికి

DROP TABLE Student;

DELETE, TRUNCATE మరియు DROP వాక్య నిర్మాణం:

DELETE	DELETE ఆదేశం where clause లో ఇవ్వబడిన నిబంధన ప్రకారం, పట్టికలోని నిర్ధిష్ట అడ్డు వరుసలను మాత్రమే తొలగించును లేదా ఎటువంటి షరత్తు లేనిచో పట్టికలోని అన్ని అడ్డు వరుసలను తొలగించును. కానీ పట్టికలో స్థలాన్ని ఖాళీ చేయదు.
TRUNCATE	TRUNCATE ఆదేశం పట్టికలోని అడ్డు వరుసలను తొలగించి, పట్టిక నిర్మాణం మారకుండ, పట్టిక ఆక్రమించిన స్థలాన్ని ఖాళీ చేయును.
DROP	DROP ఆదేశం డేటాబేస్ నుండి ఒక object ను తొలగించడానికి ఉపయోగించబడుతుంది. ఒక పట్టికను DROP చేసిన, పట్టికలోని అన్ని అడ్డు వరుసలు తొలగించబడి, పట్టిక నిర్మాణం డేటాబేస్ నుండి తొలగించబడుతుంది. ఒక పట్టికను DROP చేసిన తర్వాత దానిని తిరిగి పొందలేము.

12.7.5 DQL ఆదేశం SELECT ఆదేశం

SQL తో పనిచేసేటప్పుడు, ప్రశ్నలను రూపొందించడం మరియు డేటాను తిరిగి పొందడం చాలా ముఖ్యమైన పనులలో ఒకటగును. ప్రశ్న అనేది డేటాబేస్ నుండి ఆశించిన ఫలితాన్ని పొందడానికి ఇవ్వబడే ఆదేశం. డేటాబేస్‌లోని పట్టిక నుండి డేటాను ప్రశ్నించడానికి లేదా తిరిగి పొందడానికి SELECT ఆదేశం ఉపయోగపడుతుంది. ఒకటి లేదా అంతకంటే ఎక్కువ పట్టికల నుండి రికార్డుల ఉపసమితిని తిరిగి పొందడానికి ఇది ఉపయోగపడుతుంది. SELECT ఆదేశాన్ని వివిధ రూపాల్లో ఉపయోగించవచ్చు :

12th_CS_Telugu_Medium_Chapter_12.indd 216 28-03-2023 10:05:46

SELECT ఆదేశం యొక్క వాక్య నిర్మాణం

SELECT <column-list>FROM<table-name>;

- Table-name అనేది సమాచారాన్ని తిరిగి పొందు పట్టిక పేరు.
- Column-list అనేది ఒకటి లేదా అంతకంటే ఎక్కువ నిలువు వరుసల నుండి డేటా తిరిగి పొందడానికి.

ఉదాహరణకు, విద్యార్థి పట్టిక నుండి ప్రవేశ సంఖ్య మరియు విద్యార్థుల పేరు మాత్రమే చూడటానికి ఆదేశం ఈ క్రింది విధంగా ఇవ్వబడుతుంది.

విద్యార్థి పట్టికలో ఈ క్రింది డేటా ఉంటే :

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
107	Hema	F	17	Chennai

SELECT Admno, Name FROM Student;

పై SELECT ఆదేశం క్రింది డేటాను ప్రదర్శిస్తుంది.

Admno	Name
100	Ashish
101	Adarsh
102	Akshith
103	Ayush
104	Abinandh
105	Revathi
106	Devika
107	Hema

పట్టికలోని అన్ని ఫీల్డులు (fields) మరియు అడ్డువరుసలను చూడటానికి SELECT ఆదేశం ఇవ్వవచ్చును.

SELECT * FROM STUDENT;

12.7.5.1 DISTINCT కీఒర్డ్

DISTINCT కీఒర్డ్‌తో SELECT ఆదేశమును ఉపయోగించి పట్టికలోని నకిలీ వరుసలను తొలగించవచ్చు. అవసరమైన డేటాను తొలగించడానికి ఇది సహాయపడుతుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 217 28-03-2023 10:05:46

ఉదాహరణకి :

SELECT DISTINCT Place FROM Student; అను ఆదేశం క్రింది డేటాను ప్రదర్శిస్తుంది.

Place
Chennai
Bangalore
Delhi

పై outputలో ఫీల్డు నందు ఎటువంటి నకిలీ వరుసలు ఉండవు. DISTINCT అను ప్రత్యేక పదము ఉపయోగించునపుడు, ఎక్కువ NULL విలువలు సంభవించినప్పటికీ, ఒక NULL మాత్రమే తిరిగి ఇవ్వబడుతుంది.

12.7.5.2 ALL కీఒర్డ్

ALL కీఒర్డ్ అను పదము నకిలీ ఆడ్డువరుసలను కూడా ఉంచుతుంది. ఇది నకిలీ ఎంట్రీలను పరిగణించకుండా పట్టికలోని ప్రతి అడ్డువరుసను ప్రదర్శిస్తుంది.

SELECT ALL Place FROM Student;

పై ఆదేశం నకిలీ ఎంట్రీలను పరిగణలోకి తీసుకోకుండా పట్టికలోని ప్రతి అడ్డువరుస నుండి place ఫీల్డ్‌లోని విలువలను ప్రదర్శిస్తుంది.

Place
Chennai
Delhi
Bangalore
Delhi
Chennai
Chennai
Bangalore
Chennai

SELECT ఆదేశంలోని WHERE నిబంధన ఆశించిన ఫలితాన్ని పొందడానికి ప్రమాణాలను నిర్దేశిస్తుంది. WHERE నిబంధనతో SELECT ఆదేశం యొక్క సాధారణ రూపం.

SELECT <column-name>[,<column-name>,….] FROM <table-name>WHERE condition>;

ఉదాహరణకు, చెన్నైకి చెందిన విద్యార్థుల ప్రవేశ సంఖ్య మరియు పేరును మాత్రమే ప్రదర్శించడానికి, SELECT ఆదేశం క్రింది విధంగా ఉపయోగించబడుతుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 218 28-03-2023 10:05:46

SELECT Admno, Name, Place FROM Student WHERE Place =”Chennai”;

Admno	Name	Place
100	Ashish	Chennai
104	Abinandh	Chennai
105	Revathi	Chennai
107	Hema	Chennai

SELECT Admno, Name, Age FROM Student WHERE Age >= 18;

Admno	Name	Age
101	Adarsh	18
103	Ayush	18
104	Abinandh	18
105	Revathi	19
106	Devika	19

రిలేషనల్ ఆపరేటర్లు =, <, <=, >, >=, <> మొదలైనవి రెండు విలువలను పోల్చడానికి SELECT ఆదేశంతో WHERE నిబంధన ఉపయోగించవచ్చు. WHERE లాజికల్ ఆపరేటర్లయిన OR, AND మరియు NOT నిబంధనలోని శోధన షరతును కలపడానికి కూడా ఉపయోగించవచ్చు.
ఉదాహరణకి

SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 AND Place = "Delhi");

Admno	Name	Age	Place
101	Adarsh	18	Delhi
103	Ayush	18	Delhi

SELECT ఆదేశాన్ని క్రింది విధంగా ఉపయోగించవచ్చు:

SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 OR Place ="Delhi");

SELECT Admno, Name, Place FROM Student WHERE (NOT Place ="Delhi");

12.7.5.3 BETWEEN మరియు NOT కీలక పదాలు

BETWEEN అను కీఓర్డ్, నిబంధనను నిజం చేయడానికి రికార్డు తప్పక విలువల శ్రేణిని నిర్వచిస్తుంది. విలువల విస్తృతి (శ్రేణి) గరిష్ట విలువ మరియు కనిష్ట విలువ మధ్య ప్రమాణాలు తప్పక వస్తాయి.

SELECT Admno, Name, Age, Gender FROM Student WHERE Age BETWEEN 18 AND 19;

Admno	Name	Age	Gender
101	Adarsh	18	M
103	Ayush	18	M
104	Abinandh	18	M
105	Revathi	19	F
106	Devika	19	F

12th_CS_Telugu_Medium_Chapter_12.indd 219 28-03-2023 10:05:46

BETWEEN కీఓర్డ్ యొక్క వ్యతిరేక పదము NOT BETWEEN పనిచేయును, నిబంధనను సంతృప్తిపరచని రికార్డులను ప్రదర్శించబడతాయి.

SELECT Admno, Name, Age FROM Student WHERE Age NOT BETWEEN 18 AND 19;

Admno	Name	Age
100	Ashish	17
102	Akshith	17
107	Hema	17

12.7.5.4 IN కీఓర్డ్

IN కీఓర్డ్, నిర్ధిష్ట జాబితాలోని విలువలను, రికార్డు విలువలతో పోల్చడానికి ఉపయోగపడుతుంది. మరోమాటలో చెప్పాలంటే, ఇది ఒకటి లేదా అంతకంటే ఎక్కువ విలువలు గల ఒక నిలువ వరుసను పోల్చడానికి ఉపయోగపడుతుంది. ఇది OR నిబంధనకు సమానమైనది.

ఉదాహరణకు :

SELECT Admno, Name, Place FROM Student WHERE Place IN (“Chennai”, “Delhi”);

Admno	Name	Place
100	Ashish	Chennai
101	Adarsh	Delhi
103	Ayush	Delhi
104	Abinandh	Chennai
105	Revathi	Chennai
107	Hema	Chennai

NOT IN కీఓర్డ్, పట్టికలో సరిపోని రికార్డులను మాత్రమే ప్రదర్శిస్తుంది.

ఉదాహరణకి :

SELECT Admno, Name, Place FROM Student WHERE Place NOT IN (“Chennai”, “Delhi”);

చెన్నై మరియు ఢిల్లీ కాకుండా ఇతర ప్రదేశాల విద్యార్థులను మాత్రమే ప్రదర్శిస్తుంది.

Admno	Name	Place
102	Akshith	Bangalore
106	Devika	Bangalore

NULL విలువ :

ఫీల్డ్‌లోని NULL విలువను WHERE నిబంధనలోని IS NULL ఉపయోగించి శోధించవచ్చును. ఉదాహరణకు, Age అను ఫీల్డ్ నందు విలువలు ఇవ్వని అందరి విద్యార్థులను జాబితా చేయుటకు ఉపయోగపడు ఆదేశము :

SELECT * FROM Student WHERE Age IS NULL;

12th_CS_Telugu_Medium_Chapter_12.indd 220 28-03-2023 10:05:47

గమనిక

NULL ఉపయోగించి పట్టికలోని IS NOT NULL లేని విలువలను జాబితా చేయవచ్చు.

12.7.5.5 ORDER BY clause

ఒకటి లేదా అంతకంటే ఎక్కువ నిలువు వరుసల ఆధారంగా డేటాను ఆరోహణ లేదా అవరోహణలో క్రమబద్ధీకరించడానికి SQL లోని ORDER BY నిబంధన ఉపయోగపడుతుంది.

1. అప్రమేయంగా డేటాను ఆరోహణ క్రమంలో క్రమబద్ధీకరిస్తుంది.
2. డేటాను, అవరోహణ క్రమంలో క్రమబద్ధీకరించడానికి DESC అను కీలక పదము, ఆరోహణ క్రమంలో క్రమబద్ధీకరించడానికి ASC అను కీలక పదము ఉపయోగించవచ్చు.

ORDER BY ఈ విధంగా ఉపయోగించవచ్చు.

SELECT <column-name>[,<column-name>,....] FROM <table-name>ORDER BY <column1>,<column2>,...ASC\| DESC ;

ఉదాహరణకి :

అక్షర క్రమంలో విద్యార్థుల పేర్లు ప్రదర్శించడానికి ఉపయోగపడు ఆదేశం.

SELECT * FROM Student ORDER BY Name;

పై విద్యార్థి పట్టిక ఈ క్రింది విధంగా అమర్చబడినది.

Admno	Name	Gender	Age	Place
104	Abinandh	M	18	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
100	Ashish	M	17	Chennai
103	Ayush	M	18	Delhi
106	Devika	F	19	Bangalore
107	Hema	F	17	Chennai
105	Revathi	F	19	Chennai

గమనిక

ORDER BY నిబంధన అసలు పట్టికను ప్రభావితం చేయదు.

12.7.5.6 WHERE clause

రికార్డులను వడకట్టుటకు (ఫీల్టర్) WHERE నిబంధన ఉపయోగించబడుతుంది. ఇచ్చిన నిబంధనను సంతృప్తి పరిచే రికార్డులను మాత్రమే సేకరించుటకు ఇది సహాయపడుతుంది ఉదాహరణకు, విద్యార్థి పట్టికలో, 18 ఏళ్ళు మరియు అంతకంటే ఎక్కువ వయస్సు గల విద్యార్థుల జాబితాను వారి పేర్ల క్రమంలో ప్రదర్శించడానికి, క్రింది ఆదేశం ఇవ్వబడింది.

SELECT * FROM Student WHERE Age>=18 ORDER BY Name;

Admno	Name	Gender	Age	Place
104	Abinandh	M	18	Chennai
101	Adarsh	M	18	Delhi
103	Ayush	M	18	Delhi
106	Devika	F	19	Bangalore
105	Revathi	F	19	Chennai

18 మరియు అంతకంటే ఎక్కువ వయస్సు గల పేర్లను అవరోహణ క్రమంలో జాబితా చేయుటకు ఆదేశం ఈ క్రింది విధంగా ఇవ్వబడింది :

SELECT * FROM Student WHERE Age>=18 ORDER BY Name DESC;

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai

గమనిక

క్రమబద్ధీకరణ బహుళ క్షేత్రాలలో చేయవచ్చు.

12.7.5.7 GROUP BY clause

GROUP BY నిబంధన SELECT వాక్యనిర్మాణంతో విద్యార్థులను అడ్డువరుసలు లేదా నిలువు వరుసలలో ఒకేలాంటి విలువలను కలిగి ఉండటానికి లేక పట్టికను సమూహాలుగా విభజించడానికి ఉపయోగపడుతుంది. ఒక తరగతిలోని అబ్బాయిలు లేదా అమ్మాయిల సంఖ్యను తెలుసుకోవడానికి, GROUP BY నిబంధన ఉపయోగించవచ్చు. డేటాబేస్ నుండి సారాంశ నివేదికలను రూపొందించడానికి ఇది ఎక్కువ మొత్తం functions లతో కలిపి ఉపయోగపడుతుంది.

GROUP BY నిబంధన యొక్క వాక్య నిర్మాణం

12th_CS_Telugu_Medium_Chapter_12.indd 222 28-03-2023 10:05:47

SELECT <column-names> FROM <table-name> GROUP BY <column-name>HAVING condition];

GROUP BY పై ఆదేశాన్ని విద్యార్థి పట్టికలో వర్తింపజేయడానికి :

SELECT Gender FROM Student GROUP BY Gender;

పై ఆదేశం క్రింది ఫలితాలను ఇస్తుంది.

Gender
M
F

గమనించదగ్గ విషయం ఏమిటంటే రెండు ఫలితాలు మాత్రమే తిరిగి ఇవ్వబడ్డాయి. దీనికి కారణం, మనకు రెండు లింగ రకాలు 'మగ' మరియు 'ఆడ' మాత్రమే. GROUP BY నిబంధన 'M' విద్యార్థులందరినీ ఒక చోట చేర్చి, వాటిని ఒకే వరుసలో తిరిగి ఇవ్వబడినది 'F' విద్యార్థులకు అదే పనిని చేసింది.

ఉదాహరణకు, విద్యార్థి పట్టికలో మగ, ఆడ విద్యార్థుల సంఖ్యను లెక్కించడానికి ఈ క్రింది ఆదేశం ఇవ్వబడింది :

SELECT Gender, count(*) FROM Student GROUP BY Gender;

Gender	count(*)
M	5
F	3

గమనిక

NULL విలువలను చేర్చడానికి COUNT తో చేర్చబడుతుంది.

GROUP BY మొత్తం విధులను సమూహాల శ్రేణికి స్వతంత్రంగా వర్తిస్తుంది. ఇవి క్షేత్రవిలువను ఉమ్మడిగా కలిగి ఉంటాయి. పై SELECT వాక్యనిర్మాణం యొక్క output మగ మరియు ఆడ విద్యార్థుల సంఖ్యను ఇస్తుంది.

12.7.5.8 HAVING clause

SELECT ఆదేశంతో HAVING మరియు GROUP BY లను ఉపయోగించి, సమూహాలపై షరుతు (నిబంధనను) ఉంచడానికి మరియు వాటిపై మొత్తం విధులను చేర్చవచ్చు. ఉదాహరణకు చెన్నైకి చెందిన మగ మరియు ఆడ విద్యార్థుల సంఖ్యను లెక్కించడానికి

SELECT Gender , FROM Student GROUP BY Gender HAVING count(*)> = 18;

Place	count
Chennai	2

12th_CS_Telugu_Medium_Chapter_12.indd 223 28-03-2023 10:05:47

పై output విద్యార్థి పట్టిక నుండి చెన్నైలోని మగ మరియు ఆడ విద్యార్థుల సంఖ్యను సూచిస్తుంది.

12.7.6 TCL ఆదేశాలు

12.7.6.1 COMMIT ఆదేశం

COMMIT ఆదేశం, డేటాబేసులోని ఏ ఒక్క లావాదేవీలనైనా శాశ్వతంగా save (భద్రపరచడం) చేయుటకు ఉపయోగపడుచున్నది. INSERT, UPDATE, DELETE వంటి DML ఆదేశాలను ఉపయోగించి చేసిన మార్పులు శాశ్వతంగా ఉండవు. SQL నుండి COMMIT ఆదేశం ఇచ్చిన తర్వాత ఇది శాశ్వతంగా గుర్తించబడుతుంది. COMMIT ఆదేశం ఇచ్చిన తర్వాత, చేసిన మార్పులను తిరిగి పొందలేము. COMMIT ఆదేశం క్రింది విధంగా ఉపయోగించబడుతుంది.

```
COMMIT;
```

12.7.6.2 ROLLBACK ఆదేశం

ROLLBACK ఆదేశం, డేటాబేసుని చివరి నిబద్ధస్థితికి పునరుద్ధరిస్తుంది. నిర్దిష్ట Save Point స్థానానికి వెళ్ళడానికి, ఈ ఆదేశంతో SAVEPOINT ఆదేశం ఉపయోగించబడుతుంది.

ROLLBACK ఆదేశం యొక్క వాక్యనిర్మాణం :

```
ROLL BACK TO  save point name;
```

12.7.6.3 SAVEPOINT ఆదేశం

SAVEPOINT ఆదేశం, లావాదేవీ తాత్కాలికంగా save చేయడానికి ఉపయోగించబడుతుంది. తద్వారా మీరు అవసరమైనప్పుడు పాయింట్‌కి rollback చేయవచ్చు. పట్టికలోని వేర్వేరు స్థితులు వేర్వేరు పేర్లను ఉపయోగించి save చేయవచ్చు. మరియు ROLLBACK ఆదేశంను ఉపయోగించి తిరిగి ఆ స్థితిని పొందగలము.

```
SAVEPOINT savepoint_name;
```

క్రింది డేటాను కలిగియున్న విద్యార్థి పట్టికలో COMMIT, SAVEPOINT మరియు ROLLBACK ఆదేశాలు చూపించే ఉదాహరణ :

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai

```
INSERT INTO Student  VALUES (107, 'Beena', 'F', 20 , 'Cochin');
COMMIT;
```


12th_CS_Telugu_Medium_Chapter_12.indd 224 28-03-2023 10:05:47

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

```
UPDATE Student SET Name = 'Mini' WHERE Admno=105;
SAVEPOINT A;
```

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

```
INSERT INTO Student VALUES(108, 'Jisha', 'F', 19, 'Delhi');
SAVEPOINT B;
```

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	vAbinandh	M	18	Chennai
107	Beena	F	20	Cochin
108	Jisha	F	19	Delhi

```
ROLLBACK TO A;
```

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore

103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

గుర్తించుకోవలసిన అంశాలు

- *SQL* అనేది Relational Databases ను సృష్టించడానికి, అమలు చేయుటకు సహాయపడు భాష.
- *MySQL* అనునది ఒక డేటాబేస్ నిర్వహణ వ్యవస్థ
- *SQL* యొక్క వివిధ భాగాలు డేటా డెఫినిషన్ లాంగ్వేజ్ (*DDL*), డేటా మానిప్యులేషన్ లాంగ్వేజ్ (*DML*), డేటా క్వరీ లాంగ్వేజ్ (*DQL*), ట్రాన్సాక్షనల్ కంట్రోల్ లాంగ్వేజ్ (*TCL*) మరియు డేటా కంట్రోల్ లాంగ్వేజ్ (*DCL*).
- పట్టికలను సృష్టించడానికి మరియు తొలగించడానికి *DDL* వాక్యనిర్మాణం statements ను అందిస్తుంది.
- పట్టిక యొక్క డేటాను చొప్పించడానికి, నవీనీకరించడానికి మరియు తొలగించడానికి *DML* వాక్యనిర్మాణం (statements) అందిస్తుంది.
- *DCL*, డేటా access చేయడానికి అధికార ఆదేశాలు అందిస్తుంది.
- డేటాబేస్‌లో లావాదేవీలను నిర్వహించడానికి *TCL* ఆదేశాలు ఉపయోగించబడతాయి.
- *DQL* ఆదేశాలు, డేటాబేస్‌లో ప్రశ్నలను రూపొందించడానికి సహాయపడతాయి.
- *CREATE TABLE* ఆదేశం, క్రొత్త పట్టికను సృష్టిస్తుంది.

12th_CS_Telugu_Medium_Chapter_12.indd 226 28-03-2023 10:05:47

Hands on Experience (అనుభవం)

1. విద్యార్థి పట్టికకు name, age, place మరియు admno అను క్రమంలో ప్రశ్నలను query సృష్టించుము.
2. ప్రత్యేకమైన నగరంతో 18 కంటే ఎక్కువ వయసు గల విద్యార్థుల వివరాలను ప్రదర్శించడానికి ఒక ప్రశ్నను సృష్టించండి

Hands on Experience

1. ఉద్యోగి సంఖ్య, ఉద్యోగిపేరు, హోదా, పనిలో చేరిన తేది, మరియు ప్రాథమిక వేతనం మొదలగు క్షేత్రములు కలిగిన ఒక ఉద్యోగి పట్టికను సృష్టించండి.
2. పై పట్టికలో ఉద్యోగి సంఖ్యను ప్రాథమిక కీ గా సెట్ చేయండి. మరియు ఏదైనా ఒక ఫీల్డ్ (క్షేత్రము)లో NULL విలువలను తనిఖీ చేయండి.
3. నిర్వాహకులు (Manager)గా ఉన్న ఉద్యోగులందరి జాబితాను సిద్ధం చేయండి.

2UG94

మూల్యాంకనం

భాగము - అ

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. ఏ ఆదేశాలు, పట్టిక నిర్మాణాన్ని సృష్టించడం, సంబంధాలను (relations) తొలగించడం మరియు స్కీమాలను సవరించడానికి నిర్వచనాలు అందిస్తాయి.

   అ) DDL **ఆ)** DML

   ఇ) DCL ఈ) DQL

2. పట్టిక యొక్క నిర్మాణాన్ని మార్చడానికి ఏ ఆదేశం అనుమతిస్తుంది.

   అ. SELECT **ఆ)** ORDER BY

   ఇ) MODIFY ఈ) ALTER

3. Structure తో పాటు పట్టికను తొలగించే అదేశం

అ) DROP ఆ) DELETE

ఇ) DELETE ALL ఈ) ALTER TABLE

4. ప్రశ్నలను సృష్టించడానికి సహాయపడునది

అ. SELECT **ఆ)** ORDER BY

ఇ) MODIFY ఈ) ALTER

5. డేటాబేస్‌లో డేటాను క్రమబద్ధీకరించడానికి ఉపయోగించు నిబంధన

అ. SORT BY **ఆ)** ORDER BY

ఇ) GROUP BY ఈ) SELECT

భాగం - ఆ

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. 18 ఏళ్ళలోపు విద్యార్థులందరినీ క్రమబద్ధంగా ఎంపిక చేసే ఒక ప్రశ్న (query) వ్రాయండి.
2. Unique మరియు Primary Key నిబంధనలను వేరుచేయుము.
3. పట్టిక పరిమితి మరియు నిలువు వరుస column పరిమితికి మధ్యగల భేదములను (వ్యత్యాసములను) వ్రాయుము.
4. పట్టికను సృష్టించడానికి, దానిలో విలువలను చొప్పించడానికి SQL యొక్క ఏ భాగం అనుమతిస్తుంది.
5. SQL మరియు MySQL మధ్యగల తేడాలు ఏమిటి?

భాగం - ఇ

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. పరిమితి అంటే ఏమిటి? Primary key పరిమితిపై ఒక జ్ఞాపికను వ్రాయుము.
2. క్రొత్త ఫీల్డ్‌ను చేర్చడం ద్వారా విద్యార్థుల పట్టిక నిర్మాణాన్ని సవరించడానికి SQL statement వ్రాయండి.
3. ఏదైన మూడు DDL ఆదేశాలను వ్రాయండి.
4. SAVEPOINT ఆదేశం వాడకాని ఉదాహరణతో వ్రాయండి.
5. DISTINCT కీలక పదము ఉపయోగించి ఒక SQL Statement వ్రాయుము.

12th_CS_Telugu_Medium_Chapter_12.indd 228 28-03-2023 10:05:47

భాగం - ఈ

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. వివిధ రకాలైన పరిమితులు మరియు వాటి విధులను వ్రాయండి.

2. క్రింది ఉద్యోగుల పట్టికను పరిశీలించి (i) నుండి (v) వరకు గల ప్రశ్నలకు SQL ఆదేశాలను వ్రాయుము.

EMP CODE	NAME	DESIG	PAY	ALLO WANCE
S1001	Hariharan	Supervisor	29000	12000
P1002	Shaji	Operator	10000	5500
P1003	Prasad	Operator	12000	6500
C1004	Manjima	Clerk	8000	4500
M1005	Ratheesh	Mechanic	20000	7000

(i) ఉద్యోగుల వారి pay ఆధారంగా అవరోహణ క్రమంలో ప్రదర్శించడాని

(ii) 5000 నుండి 7000 వరకు allowance తీసుకొను ఉద్యోగులందరిని ప్రదర్శించడానికి

(iii) mechanic రకమునకు చెందిన ఉద్యోగులను తొలగించడానికి

(iv) ఒక క్రొత్త అడ్డు వరుసకు (insert) చేర్చడానికి

(v) ఆపరేటర్లు అయిన ఉద్యోగులందరిని ప్రదర్శించడానికి

3. SQL యొక్క భాగాలు ఏమిటి? ప్రతి ఒక్కటికి ఆదేశాలు వ్రాయండి.

4. విద్యార్థి పట్టికలో క్రింది SQL statement రూపొందించండి.

   (i) SELECT statement లో GROUP BY నిబంధన ఉపయోగించి.

   (ii) SELECT statement లో ORDER BY నిబంధన ఉపయోగించి.

5. ఏదైన ఐదు ఫీల్డులు కలిగిన ఒక ఉద్యోగి పట్టికను సృష్టించడానికి ఒక SQL statement వ్రాసి ఆ ఉద్యోగి పట్టికకు ఒక పట్టిక పరిమితిని సృష్టించుము.

12th_CS_Telugu_Medium_Chapter_12.indd 229 28-03-2023 10:05:47

యూనిట్ IV

అధ్యాయం 13

పైథాన్ మరియు CSV ఫైళ్ళు

అభ్యసన లక్ష్యాలు

ఈ అధ్యాయం అభ్యసించిన తర్వాత విద్యార్థి తెలుసుకొనునవి

- CSV అంటే ఏమిటి?
- పైథాన్ ప్రోగ్రామ్‌లలో CSV ఫైల్‌ను దిగుమతి చేయగల సామర్థ్యం
- పైథాన్ ప్రోగ్రామ్‌లను అమలు (Execute) మరియు డీబగ్ చేయడం

KH9FZ

13.1 పరిచయం

పైథాన్ దాని పంపిణీతో కూడిన మాడ్యూళ్ళ యొక్క విస్తారమైన లైబ్రరీని కలిగి ఉంది. వాటిలో ఒకటి CSV మాడ్యూల్. ఇది పైథాన్ ప్రోగ్రామర్‌కు CSV ను అన్వయించే సామార్థ్యాన్ని ఇస్తుంది. CSV (కామాతో వేరు చేయబడిన విలువలు) files. ఒక CSV file అనేది ప్రతి పంక్తిలో మానవుడు చదవగలిగే file కామాలతో లేదా కొన్ని ఇతర డీలిమిటర్‌తో వేరు చేయబడిన అనేక ఫీల్డ్‌లను కలిగి ఉండును. మీరు ప్రతి వరుసను ఒక అడ్డువరుసగా మరియు ప్రతి ఫీల్డ్‌ను నిలువు వరుసగా ఊహించవచ్చును. CSV మాడ్యూల్ CSV లోకి అధిక భాగాన్ని విస్తారంగా చదవగలము మరియు వ్రాయగలదు.

13.2 CSV మరియు XLS ఫైల్ పార్మట్ మధ్యగల బేధములు.

కామా-సెపరేటెడ్ వాల్యూస్ (CSV) మరియు **eXceL** షీట్స్ (XLS) పైలే ఫార్మాట్ల మధ్య తేడాలు

ఎక్సెల్	CSV
ఎక్సెల్ అనేది బైనరీ ఫైల్. ఇది కంటెంట్ మరియు పార్మాంటింగ్ రెండింటితో సహా పైల్‌లోని అన్ని వర్క్‌షీట్‌ల గురించి సమాచారాన్ని కలిగి ఉంటుంది.	CSV పార్మాట్ అనేది కామాలతో వేరు చేయబడిన విలువల శ్రేణితో గల plain text (సాదా వచన) ఆకృతి.
XLS పైల్స్ వాటి ఫార్మాట్ చదవడానికి ప్రత్యేకంగా వ్రాయబడిన అనువర్తనాల ద్వారా మాత్రమే చదవబడతాయి మరియు అదే విధంగా వ్రాయబడతాయి.	విండోస్‌లోని నోట్‌ప్యాడ్, ఎంఎస్ ఎక్సెల్, ఓపెన్ ఆఫీస్ మొదలైన టెక్స్ట్ ఎడిటర్‌తో CSV ని తెరవవచ్చు.
ఎక్సెల్ అనేది స్ప్రెడ్‌షీట్, ఇది ఫైల్‌లను దాని స్వంత యాజమాన్య ఆకృతిలో సేవ్ చేస్తుంది xls లేదా xlsx	CSV అనేది .csv పొడిగింపుతో వేరు చేయబడిన టెక్స్ట్ ఫైల్‌లో పట్టిక సమాచారాన్ని సేవ్ చేయడానికి ఒక ఫార్మాట్
ఎక్సెల్ డేటాను దిగుమతి చేయడానికి అధిక మెమరీని వినియోగిస్తుంది.	CSV ఫైళ్ళను దిగుమతి చేయడం చాలా వేగంగా ఉంటుంది మరియు ఇది తక్కువ మెమరీని కూడా వినియోగిస్తుంది.

12th_CS_Telugu_Medium_Chapter_13.indd 230 28-03-2023 10:06:46

మీకు తెలుసా?

ఎక్సెల్‌లో సేవ్ చేసిన ఫైల్‌లను టెక్స్ట్ ఎడిటర్స్ తెరవలేరు లేదా సవరించలేరు

13.3 CSV ఫైల్ యొక్క ఉద్దేశ్యం

CSV అనేది స్ప్రెడ్‌షీట్ లేదా డేటాబేస్ వంటి పట్టిక టేబుల్ నిల్వచేయడానికి ఉపయోగించే ఒక సాధారణ ఫైల్ ఫార్మాట్. అవి సాధారణ వచనం కాబట్టి, మీరు ఉపయోగిస్తున్న నిర్దిష్ట సాఫ్ట్‌వేర్‌తో సంబంధం లేకుండా అవి స్ప్రెడ్‌షీట్ లేదా మరొక నిల్వ డేటాబేస్‌లోకి దిగుమతి చేసుకోవడం సులభం.

మీరు మైక్రోసాఫ్ట్ ఎక్సెల్ వంటి స్ప్రెడ్‌షీట్ ప్రోగ్రామ్‌లో లేదా టెక్స్ట్ ఎడిటర్‌లో లేదా డేటాబేస్ ద్వారా **CSV** .ఫైల్‌లను తెరవవచ్చు.

గమనిక

CSV ఫైల్ పటాలు లేదా గ్రాఫ్‌లను నిల్వచేయదు. ఇది డేటాను నిల్వచేస్తుంది కాని ఫార్మాటింగ్, సూత్రాలు, మాక్రోలు మొదలైనవి కలిగి ఉండదు.

మీకు తెలుసా?

CSV ఫైల్‌ను ఫ్లాట్ ఫైల్ (Flat File) అని కూడా అంటారు. మైక్రోసాఫ్ట్ ఎక్సెల్ లేదా ఓపెన్ ఆఫీస్ కాల్క్ వంటి పట్టికలలో డేటాను నిల్వ చేసే ప్రోగ్రామ్‌ల నుండి CSV ఫార్మాట్‌లోని ఫైల్‌లను దిగుమతి మరియు ఎగుమతి చేయవచ్చు.

13.4 నోట్‌ప్యాడ్ (లేదా ఏదైనా టెక్స్ట్ ఎడిటర్) ఉపయోగించి CSV ఫైల్‌ను సృష్టించడం

ఒక CSV ఫైల్ ఒక టెక్స్ట్ ఫైల్, కాబట్టి దీన్ని ఏదైనా టెక్స్ట్ ఎడిటర్ ఉపయోగించి సృష్టించవచ్చు. మరియు సవరించవచ్చు, కానీ చాలా తరచుగా CSV ఫైల్ దానిని సృష్టించిన ప్రోగ్రామ్‌లో స్ప్రెడ్‌షీట్ లేదా డేటాబేస్‌ని ఎగుమతి చేయడం ద్వారా సృష్టించబడుతుంది.

13.4.1. CSV సాధారణ ఫైల్‌ను సృష్టించడం

నోట్‌ప్యాడ్‌లో CSV ఫైల్‌ను సృష్టించడానికి, మొదట కొత్త ఫైల్‌ను తెరవడానికి

File →New or ctrl +N.

ఫైల్ → క్రొత్త లేదా ctrl +N

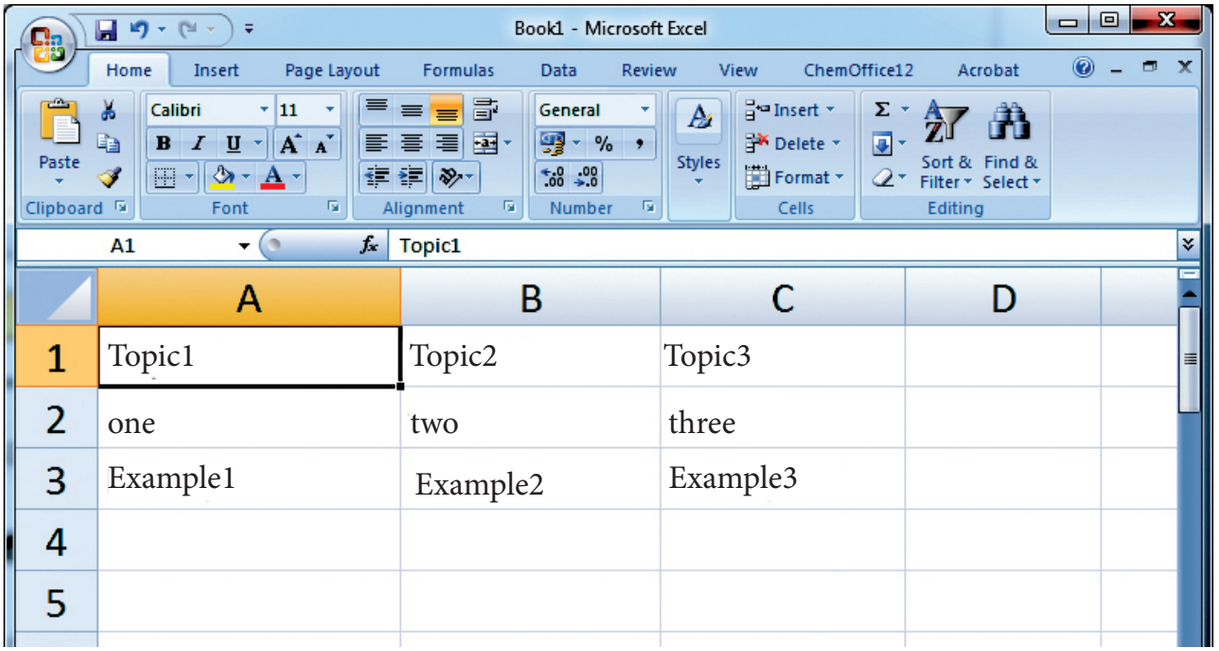
ఇప్పుడు మీరు ఫైల్ కలిగి ఉండాలనుకుంటున్న డేటాను నమోదు చేయండి. ప్రతి విలువను కామాతో మరియు ప్రతి అడ్డు వరుసను కొత్త పంక్తితో వేరు చేస్తుంది.

ఉదాహరణకు క్రింది వివరాలను పరిశీలించండి

```
Topic1,Topic2,Topic3
one,two,three
Example1,Example2,Example3
```


12th_CS_Telugu_Medium_Chapter_13.indd 231 28-03-2023 10:06:46

ఫైల్‌లోని కంటెంటెను ఈ పొడిగింపుతో సేవ్ చేయండి. అప్పుడు మీరు మైక్రోసాఫ్ట్ ఎక్స్‌ల్ లేదా మరేదైనా స్ప్రెడ్‌షీట్ ప్రోగ్రామ్‌ను ఉపయోగించి తెరవవచ్చు. ఇక్కడ మనము మైక్రోసాఫ్ట్ ఎక్స్‌ల్ ఉపయోగించి తెరిచాము. ఇది క్రింది వాటికి సమానమైన డేటా టేబుల్ సృష్టిస్తుంది.



పటము 13.4.1 *MS-Excel* లో తెరిచినప్పుడు *CSV* ఫైల్

పై CSV ఫైల్‌లో, డేటాఫీల్డ్‌లు కామాలతో వేరు చేయ చేయబడిందని మీరు గమనించవచ్చు. కానీ డేటాలో కామాలతో ఉంటే ఏమి జరుగుతుంది?

మీ CSV ఫైల్ డేటాలోని కామాలతో ఉంటే, మీరు ఆ డేటా ఫీల్డ్‌లను డబుల్ కోట్స్ (“) లో జతచేయడం ద్వారా వాటిని రక్షించవచ్చు. ఫీల్డ్‌లను వేరు చేయడానికి ఉపయోగించే డేటా కామాతో వేరుచేయబడిన భాగం, దీన్ని కామానుండి వేరు చేస్తుంది.

13.4.2 డేటాలో Comma కలిగినట్లు CSV ఫైల్‌ను సృష్టించడం

ఉదాహరణకు, చిరునామా ఫీల్డ్‌లో కామా ఉందని అనుకుందాం క్రింద ఇచ్చిన ఉదాహరణ డేటా ఉదాహరణకూ, మాఫీల్డ్‌లలో ఒకటి వివరణలో కామాలతో ఉందని చెప్పండి. మా డేటా క్రింది ఉదాహరణలా కనిపిస్తే

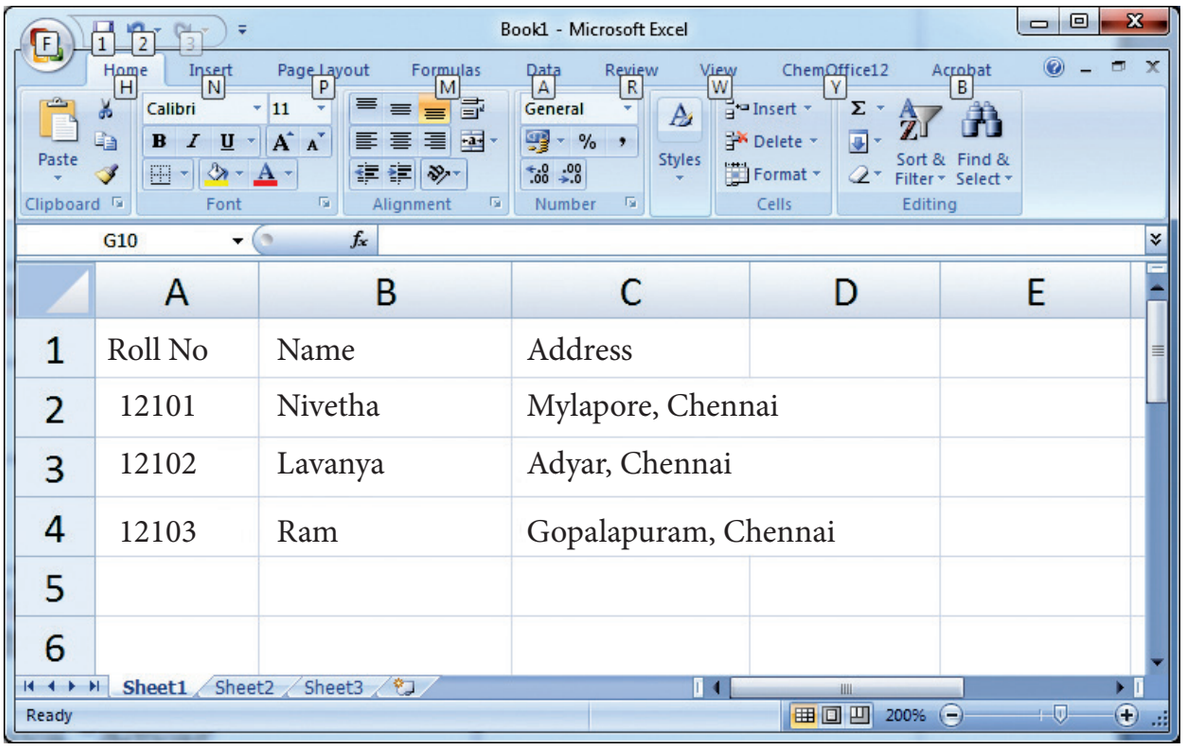
రోల్.నం	పేరు	చిరునామా
12101	నివేదా	మైలాపూర్, చెన్నై
12102	లావణ్య	అడయార్, చెన్నై
12103	రామ్	గోపాలపురం, చెన్నై

“చిరునామా” కాలమ్‌లో కామాలను నిలుపుకోవటానికి, మీరు ఫీల్డ్‌లను కొటేషన్ మార్కుల్లో జతచేయవచ్చు. ఉదాహరణకి:

```
RollNo, Name, Address
12101, Nivetha, “Mylapore, Chennai”
12102, Lavanya, “Adyar, Chennai”
12103, Ram, “Gopalapuram, Chennai”
```


12th_CS_Telugu_Medium_Chapter_13.indd 232 28-03-2023 10:06:46

మీరు గమనిస్తే, కామాలతో కూడిన ఫీల్డ్‌లు మాత్రమే కోట్స్‌లో ఉంటాయి. మీరు దీన్ని MS Excel లో తెరిస్తే ఇది క్రింది విధంగా కనిపిస్తుంది.



పటము 13.4.2 (ఎ) *Excel* లో కామాతో పీల్డ్ డేటా

మీ పీల్డ్ డేటాలో భాగమైన క్రొత్త లైన్‌లను (ఒకటి కంటే ఎక్కువ పంక్తి ఉదాహరణ చిరునామా కాలమ్‌లో డేటాను ప్రదర్శించండి) ప్రదర్శించడానికి అదే ఫంక్షన్‌ని ఉపయోగించవచ్చును. ఏదైనా ఫీల్డులోని డేటాని కొత్త పంక్తిలో ప్రదర్శించడానికి డబుల్ కోట్స్‌తో జత చేయాలి.

ఉదాహరణకి

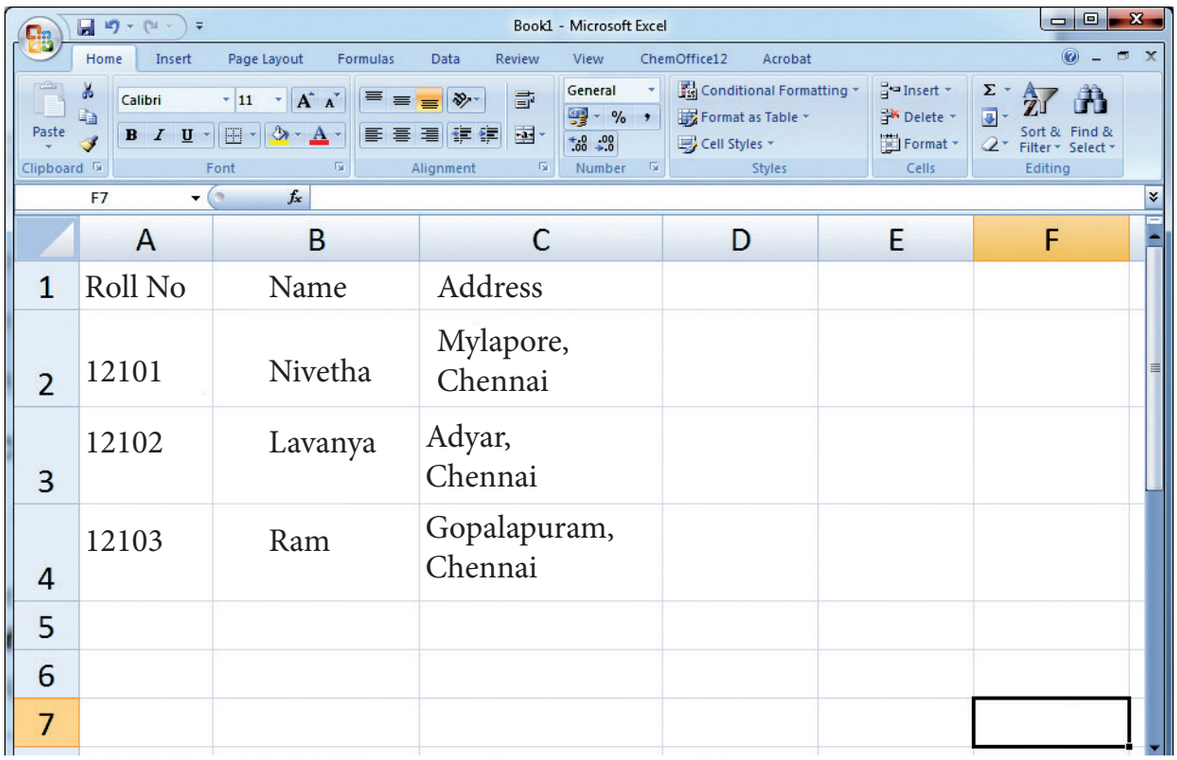
RollNo	Name	Address
12101	Nivetha	Mylapore, Chennai
12102	Lavanya	Adyar, Chennai
12103	Ram	Gopalapuram, Chennai

12th_CS_Telugu_Medium_Chapter_13.indd 233 28-03-2023 10:06:47

దీనిని CSV ఫైల్‌లో క్రింది విధంగా వ్రాయవలయును.

```
RollNo, Name, Address
12101, Nivetha, “Mylapore, Chennai”
12102, Lavanya, “Adyar, Chennai”
12103, Ram, “Gopalapuram, Chennai”
```

అవుట్పుట్ ఈ విధంగా



పటము 13.4.2 (బి) ఎక్సెల్‌లో క్రొత్త పంక్తి *CSV* ఫీల్డ్ డేటా

13.4.3 డబుల్ కోట్స్ ఉన్న డేటాతో CSV ఫైల్‌ను సృష్టించడం

మీ ఫీల్డ్‌లు వారి డేటాలో బాగంగా డబుల్ కోట్స్ కలిగి ఉంటే, అంతర్గత కొటేషన్ మార్కులు రెట్టింపు కావాలి, తద్వారా వాటిని సరిగ్గా అర్థం చేసుకోవచ్చు. ఉదాహరణ కోసం, క్రింది డేటా ఇవ్వబడింది.

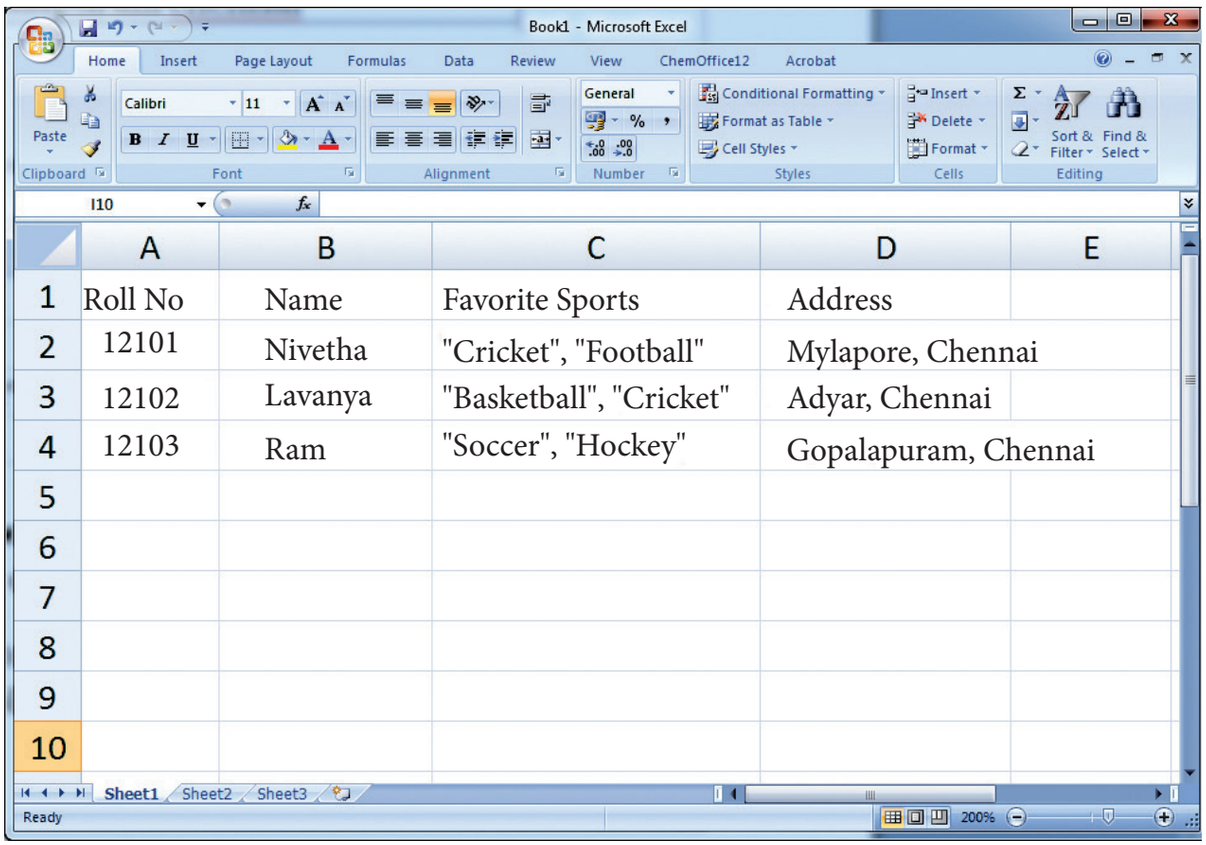
Roll No	Name	Favorite Sports	Address
12101	Nivetha	“Cricket”, “Football”	Mylapore Chennai
12102	Lavanya	“Basketball”, “Cricket”	Adyar Chennai
12103	Ram	“Soccer”, “Hockey”	Gopala puram Chennai

12th_CS_Telugu_Medium_Chapter_13.indd 234 28-03-2023 10:06:47

దీనిని csv ఫైల్‌లో క్రింది విధంగా వ్రాయాలి

```
RollNo, Name, Address
12101, Nivetha, “Mylapore Chennai”
12102, Lavanya, “Adyar Chennai”
12103, Ram, “Gopalapuram Chennai”
```

అవుట్‌పుట్ ఈ విధంగా



పటము 13.4.3 ఎక్సెల్‌లో డబుల్ కోట్స్‌తో *CSV* ఫీల్డ్ డేటా

13.4.4. CSV ఫైల్‌లో డేటాను ఫార్మాట్ చేయడానికి అనుసరించాల్సిన నియమాలు

1. ప్రతి రికార్డ్ (డేటా వరుస) ప్రత్యేక పంక్తిలో ఉండాలి. ఎంటర్ కీని నొక్కడం ద్వారా లైన్ బ్రేక్ ద్వారా వేరుచేయబడుతుంది. ఉదాహరణకి:↵

   xxx,yyy↵

   ↵ denotes enter Key to be pressed

2. ఫైల్‌లోని చివరి రికార్డ్‌కు ముగింపు పంక్తి విరామం ఉండవచ్చు లేదా ఉండకపోవచ్చు. ఉదాహరణకి:

   ppp, qqq ↵

   yyy, xxx

12th_CS_Telugu_Medium_Chapter_13.indd 235 28-03-2023 10:06:47

3. సాధారణ రికార్డ్ పంక్తుల మాదిరిగానే ఫార్మాట్ ఉన్న ఫైల్ యొక్క పంక్తిగా కనిపించే ఐచ్ఛిక శీర్షిక పంక్తి ఉండవచ్చు. హెడ్డర్ ఫైల్‌లోని ఫీల్డ్‌లకు అనుగుణమైన పేర్లను కలిగి ఉంటుంది. మరియు మిగిలిన ఫైల్‌లోని రికార్డుల వలె అదే సంఖ్యలో ఫీల్డ్‌లను కలిగి ఉండును. ఉదాహరణకు :

```
field_name1,field_name2,field_name3 ↵
aaa,bbb,ccc ↵
zzz,yyy,xxx CRLF( Carriage Return and Line Feed)
```

4. హెడ్డర్ మరియు ప్రతి రికార్డ్ లోపల, కామాలతో వేరు చేయబడిన ఒకటి లేదా అంతకంటే ఎక్కువ ఫీల్డ్‌లు ఉండవచ్చు. ఖాళీలు ఫీల్డ్‌లో భాగంగా పరిగణించబడతాయి మరియు వాటిని విస్మరించకూడదు. రికార్డ్‌లోని చివరి ఫీల్డ్‌ను కామాతో అనుసరించకూడదు. ఉదాహరణకు : ఎరుపు, నీలం

5. ప్రతి ఫీల్డ్ డబుల్ కోట్స్‌తో జతచేయబడవచ్చు లేదా ఉండకపోవచ్చు. ఫీల్డ్‌లు డబుల్ కోట్స్‌తో జతచేయబడకపోతే, ఫీల్డ్‌లలో డబుల్ కోట్స్ కనిపించకపోవచ్చు. ఉదాహరణకి :

```
"Red","Blue","Green"↵      #Field data with doule quotes
Black,White,Yellow         #Field data without doule quotes
```

6. లైన్ బ్రేక్‌లు (CRLF) డబుల్ కోట్స్ మరియు కామాలతో కూడిన ఫీల్డ్‌లను డబుల్ కోట్స్‌లో జతచేయాలి. ఉదాహరణకి:

```
ఎరుపు, ",", బ్లూ CRLF  # comma itself is a field value.so it is enclosed with double  quotes
Red, Blue , Green
```

7. ఫీల్డ్‌లను జతచేయడానికి డబుల్ కోట్స్ ఉపయోగించినట్లైయితే, ఫీల్డ్ లోపల కనిపించే డబుల్ కోట్స్ మరొక డబుల్ కోట్స్‌తో ముందు ఉండాలి. ఉదాహరణకి:

```
"Red, " "Blue", "Green",    # since double quotes is a field value it is enclosed with another double  quotes
, , White
```

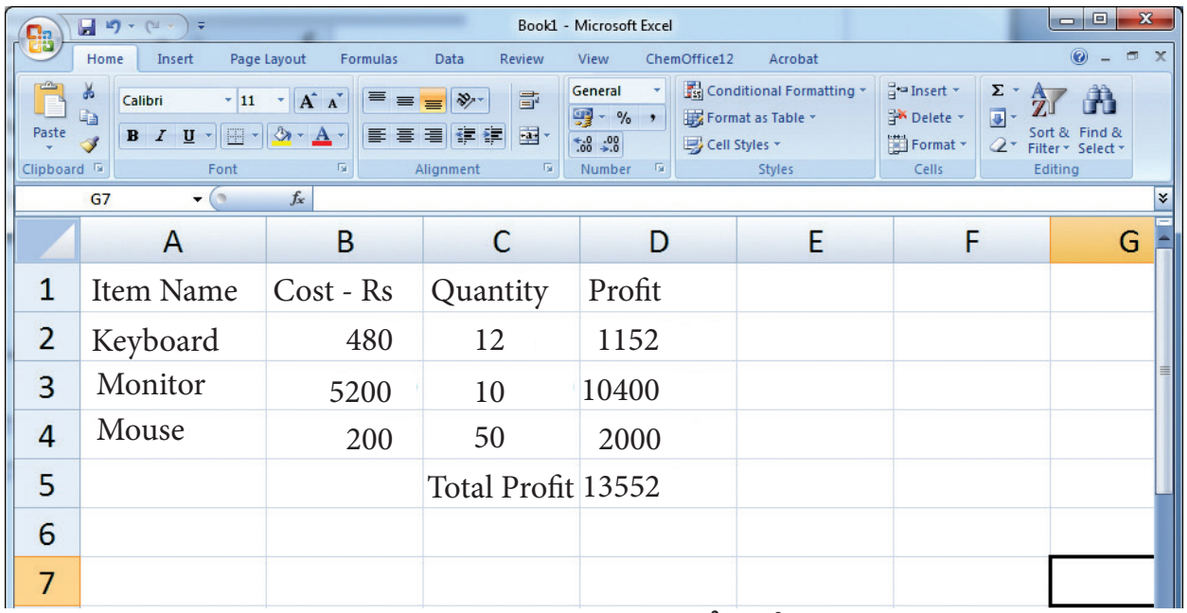
గమనిక

పై ఉదాహరణలోని చివరి అడ్డు వరుస (తెలుపు) రెండు కామాలతో మొదలవుతుంది ఎందుకంటే ఆ వరుసలోని మొదటి రెండు ఫీల్డ్‌లు మా స్ప్రెడ్‌షీట్‌లో ఖాళీగా ఉన్నాయి. వాటిని తొలగించవద్దు – రెండు కామాలతో అవసరం కాబట్టి ఫీల్డ్‌లు వరుస నుండి వరుసకు అనుగుణంగా ఉంటాయి. వాటిని వదిలివేయలేము.

12th_CS_Telugu_Medium_Chapter_13.indd 236 28-03-2023 10:06:47

13.5 మైక్రోసాఫ్ట్ ఎక్సెల్ ఉపయోగించి CSV ఫైలును సృష్టించండి
(Create A CSV File Using Microsoft Excel)

మైక్రోసాఫ్ట్ ఎక్సెల్ ఉపయోగించి CSV ఫైలును సృష్టించడానికి Excel ప్రారంభించి ఆపై మీరు CSV ఆకృతిలో సేవ్ చేయదలిచిన ఫైల్ను తెరవండి. ఉదాహరణకు, మా నమూనా Excel వర్క్‌షీట్‌లో ఉన్న డేటా క్రింద ఉంది:

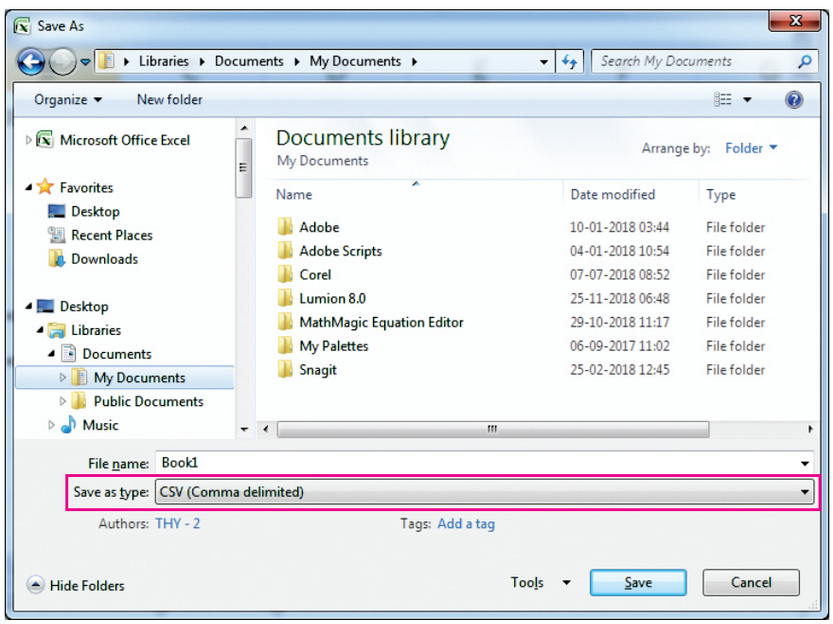


	A	B	C	D	E	F	G
1	Item Name	Cost - Rs	Quantity	Profit			
2	Keyboard	480	12	1152			
3	Monitor	5200	10	10400			
4	Mouse	200	50	2000			
5			Total Profit	13552			
6							
7							

పటము 13.5. నమూనా వర్క్‌షీట్ డేటా

వర్క్‌షీట్‌లో డేటా ఎంటర్ చేసిన తర్వాత, **File → Save As** ఎంపికను ఎంచుకోండి మరియు "టైప్ ఆప్షన్‌గా సేవ్ చేయడం" కోసం CSV (కామాతో వేరుచేయబడినది) ఎంచుకోండి లేదా .csv పొడిగింపుతో పాటు ఫైల్ పేరును టైప్ చేయండి.

ఎక్సెల్ ఫైల్‌ను CSV గా సేవ్ చేస్తోంది.



పటము 13.6 డైలాగ్ బాక్స్‌లో సేవ్ చేయండి.

12th_CS_Telugu_Medium_Chapter_13.indd 237 28-03-2023 10:06:47

మీరు ఫైల్‌ను సేవ్ చేసిన తర్వాత, దాన్ని చూడటానికి లేదా మానవీయంగా సవరించడానికి టెక్స్ట్ ఎడిటర్‌లో దాన్ని తెరవడానికి స్వేచ్ఛ ఉంది. దీని విషయాలు ఈ క్రింది వాటిని పోలి ఉంటాయి.

```
Item Name, Cost-Rs, Quantity, Profit
Keyboard, 480, 12, 1152
Monitor, 5200, 10, 10400
Mouse, 200, 50, 2000
,,Total Profit =,13552
```

13.5.1 CSV ఫైల్‌ను తెరవడానికి మైక్రోసాఫ్ట్ ఎక్సెల్ (Microsoft Excel to open a CSV file)

మైక్రోసాఫ్ట్ ఎక్సెల్ కంప్యూటర్లో ప్రతిష్టాపించబడితే, ఫైల్ డబుల్ క్లిక్ చేసినప్పుడు డిఫాల్టా CSV ఫైల్స్ ఎక్సెల్ లో స్వయంచాలకంగా తెరవబడతాయి. CSV ఫైల్‌ను తెరిచేటప్పుడు మీరు ఓపెన్ విత్ ప్రాంప్ట్‌ను పొందుతుంటే, ఫైల్‌ను తెరవడానికి అందుబాటులో ఉన్న ప్రోగ్రామ్‌ల నుండి మైక్రోసాఫ్ట్ ఎక్సెల్ ఎంచుకోండి.

ప్రత్యామ్నాయంగా, మీరు మైక్రోసాఫ్ట్ ఎక్సెలుకు తెరవవచ్చు మరియు మెనుబార్లో, **File → Open** ఎంచుకోండి మరియు CSV ఫైలును ఎంచుకోండి. ఫైల్ జాబితా చేయకపోతే, టెక్స్ట్ ఫైళ్ళకు తెరవవలసిన ఫైల్ రకాన్ని మార్చాలని నిర్ధారించుకోండి (*.prn, *.txt, *.csv).

మీకు తెలుసా?

MS Excel మరియు Open Office calc రెండూ కంప్యూటర్‌లో ఇన్‌స్టాల్ చేయబడితే, ఆ ప్రమేయంగా CSV ఫైల్ MS Excel లో తెరవబడుతుంది.

13.6 ఫైథాన్ ఉపయోగించి CSV ఫైల్‌ను చదవండి మరియు వ్రాయండి (Read a CSV file Using Python)

పైథాన్ **CSV** అనే మాడ్యూల్‌ను అందిస్తుంది, దీనిని ఉపయోగించి మీరు **CSV** ఫైల్‌లలో అనేక ఆపరేషన్లు చేయవచ్చు. **CSV** లైబ్రరీలో మరియు **CSV** ఫైళ్ళ నుండి డేటాను చదవడానికి, వ్రాయడానికి మరియు ప్రాసెస్ చేయడానికి వస్తువులు మరియు ఇతర కోడ్‌లు ఉన్నాయి.

మీకు తెలుసా?

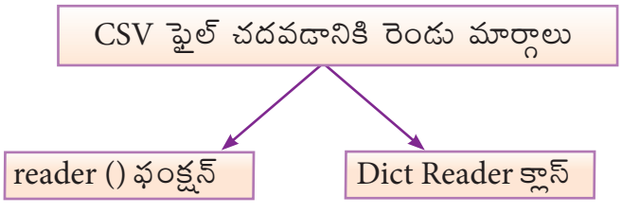
CSV ఫైల్స్ ఇ-కామార్స్ అనువర్తనాలలో విస్తృతంగా ఉపయోగించబడ్డాయి ఎందుకంటే అవి ప్రాసెస్ చేయడం చాలా సులభం.

13.6.1. ఫైథాన్ ఉపయోగించి CSV ఫైల్ చదవడం (Read a CSV File Using Python)

CSV ఫైల్ చదవడానికి రెండు మార్గాలు ఉన్నాయి

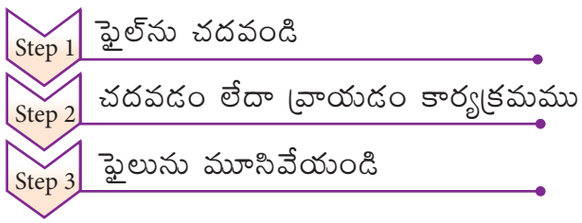
1. csv మాడ్యూల్ యొక్క రీడర్ ఫంక్షన్ ఉపయోగించడం
2. DictReader తరగతిని ఉపయోగించడం.

12th_CS_Telugu_Medium_Chapter_13.indd 238 28-03-2023 10:06:47



పటము 13.7 *CSV* ఫైల్ చదవడానికి మార్గాలు

మీరు ఫైల్ నుండి చదవానుకున్నప్పుడు లేదా వ్రాయాలనుకున్నప్పుడు మీరు దానిని తెరవాలి. పఠనం ముగిసిన తర్వాత దాన్ని మూసివేయాలి. దీనివలన ఫైల్‌లో మెమెరీ స్థలాన్ని ఖాళీచేస్తుంది. అందువల్ల పైథాన్‌లో క్రింది క్రమంలో ఫైల్ ఆపరేషన్ జరుగుతుంది.



గమనిక

ఫైల్ పేరు లేదా పూర్తి మార్గం పేరును "లేదా ఓపెన్ కమాండ్‌లోని" లో నూచించవచ్చు.

పైథాన్ ఫైల్‌ను తెరవడానికి అంతర్నిత ఫంక్షన్ **open()** ను కలిగి ఉంది. ఈ ఫంక్షన్ **file object** (ఆబ్జెక్ట్)ను తిరిగి ఇచ్చును. దీనిని **handle** అని కూడా పిలుస్తారు. ఎందుకంటే ఫైల్‌ను తదనుగుణంగా చదవడానికి లేదా సవరించడానికి ఇది ఉపయోగించబడుతుంది.

ఉదాహరణకి

```
>>> f = open("sample.txt")  # open file in current directory and f is file object
>>> f = open('c:\pyprg\ch13sample5.csv')# specifying full path
```

ఫైల్‌ను తెరిచేటపుప్పుడు మీరు మోడ్‌ను పేర్కొనవచ్చు. మోడ్‌లో మీరు **'r'** చదవాలనుకుంటున్నారా, **'w'** అని వ్రాయాలా లేదా ఫైల్‌కు **'a'** ను జోడించాలనుకుంటున్నారా అని పేర్కొనవచ్చు. తెరవలసిన ఫైల్ టెక్స్ట్ లేదా బైనరీ ను కూడా మీరు పేర్కొనవచ్చు.

డిపాల్ట్‌గా టెక్స్ట్ మోడ్‌లో చదువుతోంది. ఈ మోడ్‌లో, ఫైల్ నుండి చదివేటప్పుడు డేటా స్ట్రింగ్ ఆకృతిలో ఉంటుంది.

మరో వైపు బైనరీ మోడ్ **bytes** తిరిగి ఇస్తుంది మరియు ఇమేజ్ లేదా **exe** ఫైల్స్ వంటి టెక్స్ట్ కాని ఫైల్‌లతో వ్యవహరించేటప్పుడు ఉపయోగించాల్సిన మోడ్ ఇది.

12th_CS_Telugu_Medium_Chapter_13.indd 239 28-03-2023 10:06:47

పైథాన్ ఫైల్ మోడ్‌లు

మోడ్	వివరణ
'r'	చదవడానికి ఫైల్‌ను తెరవండి (డిపాల్ట్)
'w'	రాయడానికి ఒక ఫైల్ తెరవండి. క్రొత్త ఫైల్ ఉనికిలో లేకుంటే దాన్ని సృష్టిస్తుంది లేదా ఫైల్ ఉన్నట్లైయితే దానిని కత్తిరించుకుంటుంది.
'x'	ప్రత్యేకమైన సృష్టి కోసం ఫైల్‌ను తెరవండి. ఫైల్ ఇప్పటికీ ఉంటే, ఆపరేషన్ విఫలమవుతుంది.
'a'	ఫైల్‌ను కత్తిరించకుండా చివరిలో చేర్చడానికి తెరవండి క్రొత్త ఫైల్ ఉనికిలో లేకపోతే దాన్ని సృష్టిస్తుంది.
't'	టెక్స్ట్ మోడ్‌లో తెరవండి.
'b'	బైనరీ మోడ్‌లో తెరవండి.
'+'	సవరించడానికి ఫైల్‌ను తెరవండి (చదవడం మరియు వ్రాయడం)

f=**open("sample.txt")**

#equivalent to 'r' or 'rt'

f = open("sample.txt",'w') # టెక్స్ట్ మోడ్‌లో వ్రాయండి

f = open("image1.bmp",'r+b') # బైనరీ మోడ్‌లో చదవండి మరియు వ్రాయండి

మెమొరిలో ప్రస్తావించబడని **objects** ను శుభ్రపరచుటకు **garbage collector** పైథాన్‌లో కలదు. అయితే ఫైల్‌ను మూసివేయడానికి దానిపై ఆధారపడకూడదు.

```
f = open("test.txt")  # మోడ్ పేర్కొనబడనందున డిఫాల్ట్ మోడ్ rt ఉపయోగించబడుతుంది
# పైల్ ఆపరేషన్లను నిర్వహించండి.
f.close()
```

పై ఫంక్షన్ పూర్తిగా సురక్షితం కాదు. మీరు ఫైల్‌తో కొంత ఆపరేషన్ చేస్తున్నప్పుడు మినహాయింపు సంభవిస్తే ఫైల్‌ను మూసివేయకుండా కోడ్ నిష్క్రమిస్తుంది. దీనికి ఉత్తమ మార్గం **with** స్టేట్‌మెంట్‌ని ఉపయోగించడం. లోపల ఉన్న బ్లాక్ నుండి నిష్క్రమించినప్పుడు ఫైల్ మూసివేయబడిందని ఇది నిర్ధారిస్తుంది. మీరు క్లోజ్ () ఫంక్షన్‌ని స్పష్టంగా ఇవ్వవలసిన అవసరం లేదు. ఇది అంతర్నిర్మితంగా మూసివేయబడుతుంది.

```
with open("test.txt",'r') as f:
ఫైల్‌కు సంబంధించిన ఆపరేషన్లు     # అనునది ఫైల్ వస్తువు అగును.
```

3

12th_CS_Telugu_Medium_Chapter_13.indd 240 28-03-2023 10:06:47

ఫైలును మూసి వేయునపుడు దానితో సంబంధమున్న వనరులు, ఫైథాన్‌లో **close()** ఫంక్షన్ మూలంగా విడువడును.

```
f = open("sample.txt")
# ఫైల్ ఆపరేషన్లను నిర్వహించండి
f.close()
```

13.6.1.1 CSV మాడ్యూల్ యొక్క రీడర్ ఫంక్షన్ (CSV Module's Reader Function)

మీరు **csv.reader()** ఫంక్షన్ సహాయంతో CSV ఫైల్‌లోని విషయాలను చదవవచ్చు. రీడర్ ఫంక్షన్ ఫైల్ యొక్క ప్రతి పంక్తిని తీసుకొని అన్ని నిలువు వరుసల జాబితాను రూపొందించడానికి రూపొందించబడింది. అప్పుడు, మీరు వేరియబుల్ డేటాను కోరుకునే కాలమ్‌ను ఎంచుకోవచ్చు. ఈ ఫంక్షన్‌ని ఉపయోగించి కోట్స్ (" ") పైప్ (|) మరియు కామా (,) వంటి వివిధ పార్మాట్ యొక్క CSV ఫైల్ నుండి డేటాను చదవవచ్చు.

csv.reader() కొరకు వాక్యనిర్మాణం

```
csv.reader(fileobject,డిలిమిటర్,fmtparams)
```

ఎక్కడ

file object :- ఫైల్ యొక్క మార్గం మరియు ఫంక్షన్‌ని తిరిగి పంపును.

delimiter :- వంటి ప్రామాణిక డైలెక్ట్‌లను కలిగి ఉన్న ఐచ్ఛిక పారామితి మొదలైనవి విస్మరించవచ్చు.
fmtparams :- మాండలికాల యొక్క డిపాల్ట్ విలువలను భర్తీ చేయడానికి సహాయపడే ఐచ్ఛిక పారామితి.

Skipinitialspace :- కోటింగ్ మొదలైనవి వదిలివేయును.

1. CSV ఫైల్ డిఫాల్ట్ డీలిమిటర్ కామాతో (,) డేటా.
2. CSV ఫైల్–ప్రారంభంలో స్పేసేతో డేటా
3. CSV ఫైల్–కోట్స్‌తో డేటా
4. CSV ఫైల్–కస్టమ్ డీలిమిటర్లతో డేటా

13.6.1.1.1. డిఫాల్ట్ డీలిమిటర్ కామాతో CSV ఫైల్ (,)

క్రింది ప్రోగ్రామ్ డిపాల్ట్ డీలిమిటర్ కామా (,) తో "sample1.csv" అనే ఫైల్‌ను చదివి ఒక్కొక్క ప్రింట్ చేయండి.

12th_CS_Telugu_Medium_Chapter_13.indd 241 28-03-2023 10:06:47

```
#importing csv
import csv
#opening the csv file which is in different location with read mode
with open('c:\pyprg\sample1.csv', 'r') as F:
#other way to open the file is f= ('c:\pyprg\sample1.csv', 'r')
      reader = csv.reader(F)
# printing each line of the Data row by row
for row in reader:
     print(row)
F.close()
```

అవుట్పుట్

```
['SNO', 'NAME', 'CITY']
['12101', 'RAM', 'CHENNAI']
['12102', 'LAVANYA', 'TIRUCHY']
['12103', 'LAKSHMAN', 'MADURAI']
```

13.6.1.1.2 CSV ఫైల్స్ ప్రారంభంలో ఖాళీలతో డేటా (CSV files- data with Spaces at the beginning)

నోట్‌ప్యాడ్ ద్వారా తెరిచినప్పుడు క్రింది డేటాను కలిగి ఉన్న "sample2.csv" ఫైల్‌ను పరిగణించండి.

```
Topic1, Topic2, Topic3,
one, two, three
Example1, Example2, Example3
```

క్రింది ప్రోగ్రామ్‌ను పైథాన్‌లోని "csv.reader()" ను ఉపయోగించి ఫైల్‌ను చదవండి.

```
import csv
csv.register_dialect('myDialect',delimiter = ',',skipinitialspace=True)
F=open('c:\pyprg\sample2.csv','r')
reader = csv.reader(F, dialect='myDialect')
for row in reader:
        print(row)
F.close()
```

అవుట్పుట్

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```


12th_CS_Telugu_Medium_Chapter_13.indd 242 28-03-2023 10:06:47

"sample2.csv" లో మీరు చూడగలిగినట్లుగా డీలిమిటర్ తరువాత ఖాళీలు ఉన్నాయి. దీని వలన అవుట్పుట్ ఖాళీలతో ప్రదర్శించబడుతుంది.

csv మాడ్యూల్ యొక్క **csv.register_dialect()** తరగతిని ఉపయోగించి కొత్త మాండలికాలను నమోదు చేయడం ద్వారా ఈ వైట్స్పేస్లను తొలగించవచ్చు. ఒక మాండలికం చదవలసిన csv ఫైల్ యొక్క ఆకృతిని వివరిస్తుంది. మాండలికాలలో 'skipinitial space' పారామితి డీలిమిటర్ తర్వాత వైట్స్పేస్లను తొలగించడానికి ఉపయోగిస్తారు.

గమనిక

అప్రమేయంగా "skipinitialspace" విలువ తప్పు

క్రింది ప్రోగ్రామ్ **"sample2.csv"** ఫైల్ను చదువుతుంది. ఇది డీలిమిటర్ తర్వాత ఖాళీలను కలిగి ఉంటుంది.

```
import csv
csv.register_dialect('myDialect',delimiter = ',',skipinitialspace=True)
F=open('c:\pyprg\sample2.csv','r')
reader = csv.reader(F, dialect='myDialect')
for row in reader:
        print(row)
F.close()
```

అవుట్పుట్

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```

గమనిక

మాండలికం అనేది CSV మాడ్యూల్ యొక్క తరగతి, ఇది CSV చదవడానికి మరియు వ్రాయడానికి పారామితులను నిర్వచించడంలో సహాయపడుతుంది. ఇది మీ డేటా కోసం వివిధ ఆకృతీకరణ పారామితులను సృష్టించడానికి, నిల్వ చేయడానికి మరియు తిరిగి ఉపయోగించడానికి మిమ్మల్ని అనుమతిస్తుంది.

13.6.1.1.3 కోట్స్తో CSV ఫైల్-డేటా (CSV File-Data With Quotes)

csv మాడ్యూల్ యొక్క csv.register_dialect() తరగతిని ఉపయోగించి క్రొత్త మాండలికాలను నమోదు చేయడం ద్వారా మీరు csv ఫైల్ను కోట్స్తో చదవవచ్చును.

ఇక్కడ, ఈ క్రింది డేటాతో మనకు quotes.csv ఫైల్ ఉంది.

```
SNO,Quotes
1, "The secret to getting ahead is getting started."
2, "Excellence is a continuous process and not an accident."
3, "Work hard dream big never give up and believe yourself."
4, "Failure is the opportunity to begin again more intelligently."
5, "The successful warrior is the average man, with laser-like focus."
```

క్రింది ప్రోగ్రామ్ "quotes.csv" ఫైల్‌ను చదవండి. ఇక్కడ డీలిమిటర్ కామా (,) అయితే కోట్స్, కోట్స్‌లో ఉంటాయి (" ").

```
import csv
csv.register_dialect('myDialect',delimiter = ',',skipinitialspace=True)
f=open('c:\pyprg\quotes.csv','r')
reader = csv.reader(f, dialect='myDialect')
for row in reader:
    print(row)
```

అవుట్‌పుట్

```
['SNO', 'Quotes']
['1', 'The secret to getting ahead is getting started.']
['2', 'Excellence is a continuous process and not an accident.']
['3', 'Work hard dream big never give up and believe yourself.']
['4', 'Failure is the opportunity to begin again more intelligently.']
['5', 'The successful warrior is the average man, with laser-like focus. ']
```

పై ప్రోగ్రామ్‌లో myDialect పేరుతో మాండలికాన్ని నమోదు చేయండి. అప్పుడు **csv.QUOTE_ALL** లను ఉపయోగించిన అన్ని గుర్తులు డబుల్ కోట్స్ తరువాత ప్రదర్శించబడును.

13.6.1.1.4 CSV ఫైల్స్ కస్టమ్ డీలిమిటర్లతో (CSV files with Custom Delimiters)

csv.register_dialect() సహాయంతో క్రొత్త మాండలికాన్ని నమోదు చేయడం ద్వారా మీరు కస్టమ్ డీలిమిటర్ కలిగిన **CSV** ఫైల్‌ను చదవవచ్చును.

"sample4.csv" అని పిలువబడే క్రింది ఫైల్‌లో, ప్రతి కాలమ్ పైప్ గుర్తు (|) తో వేరు చేయబడుతుంది.

```
Roll No | Name | City
12101 | Arun | Chennai
12102 | Meena | Kovai
12103 | Ram | Nellai
```

క్రింది ప్రోగ్రామ్ వినియోగదారు నిర్వచించిన డీలిమిటర్ "|" తో "sample4.csv" ఫైల్‌ను చదవండి.

12th_CS_Telugu_Medium_Chapter_13.indd 244 28-03-2023 10:06:47

```
import csv
csv.register_dialect('myDialect', delimiter = '|')
with open('c:\pyprg\sample4.csv', 'r') as f:
    reader = csv.reader(f, dialect='myDialect')
    for row in reader:
        print(row)
f.close()
```

అవుట్పుట్

```
['RollNo', 'Name', 'City']
['12101', 'Arun', 'Chennai']
['12102', 'Meena', 'Kovai']
['12103', 'Ram', 'Nellai']
```

పై ప్రోగ్రామ్‌లో myDialect అనే కొత్త మాండలికాలు నమోదు చేయబడ్డాయి. డీలిమిటర్ = | ఉపయోగించండి ఇక్కడ పైపు (|) ని కాలమ్ సెపెరేటర్‌గా పరిగణిస్తారు.

13.6.2. ఫైల్‌లో నిర్ధిష్ట కాలమ్ చదవండి (Read a specific column In a File)

"sample5.csv" ఫైల్ కోసం అంశం పేరు మరియు లాభం వంటి నిర్ధిష్ట నిలువు వరుసలను పొందడానికి. అప్పుడు మీరు ఈ క్రింది వాటిని చేయాలి.

```
import csv
#opening the csv file which is in different location with read mode
f=open("c:\pyprg\ch13sample5.csv",'r')
#reading the File with the help of csv.reader()
readFile=csv.reader(f)
#printing the selected column
for col in readFile :
    print (col[0],col[3])
f.close()
```

sample5.csv File in Excel

	A	B	C	D
1	item Nam	Cost-Rs	Quantity	Profit
2	Keyboard	480	12	1152
3	Monitor	5200	10	10400
4	Mouse	200	50	2000

sample5.csv File with selected col

అవుట్పుట్

```
Item Name Profit
Keyboard 1152
Monitor 10400
Mouse 2000
```


12th_CS_Telugu_Medium_Chapter_13.indd 245 28-03-2023 10:06:47

13.6.3. ఒక CSV ఫైల్‌ను చదవండి మరియు దానిని జాబితాలో నిల్వచేయండి (Read A CSV File And Store It In A List)

ఈ అంశంలో మీరు CSV ఫైల్‌ను చదవబోతున్నారు మరియు ఫైల్‌లోని విషయాలు జాబితాగా నిల్వచేయబడతాయి. జాబితాలో నిల్వ చేయడానికి వాక్యనిర్మాణం

```
list = []                          # Start as the empty list
list.append(element)               # Use append() to add elements
```

ఉదాహరణకు "sample.csv" ఫైల్ యొక్క అన్ని వరుస విలువలు క్రింది ప్రోగ్రామ్‌ను ఉపయోగించి జాబితాలో నిల్వ చేయబడతాయి.

```
import csv
# other way of declaring the filename
inFile= 'c:\pyprg\sample.csv'
F=open(inFile,'r')
reader = csv.reader(F)
# declaring array
arrayValue = []
# displaying the content of the list
for row in reader:
    arrayValue.append(row)
    print(row)
F.close()
```

sample.csv opened in MS-Excel

A1	^	*fx* Topic 1
	A	B	C
1	Topic 1	Topic 2	Topic 3
2	One	two	three
3	Example 1	Example 2	Example 3
4			

అవుట్‌పుట్

```
['Topic1', 'Topic2', 'Topic3']
[' one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```


12th_CS_Telugu_Medium_Chapter_13.indd 246 28-03-2023 10:06:48

గమనిక

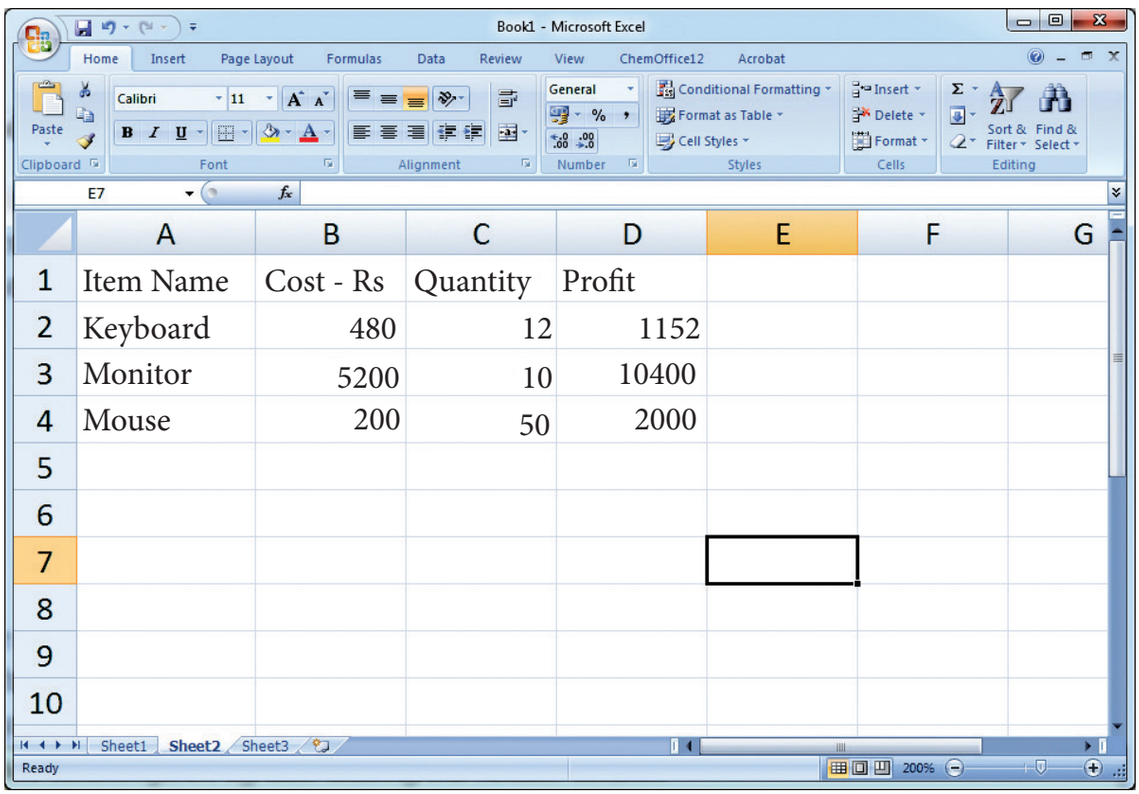
జాబితా ఫైథాన్ లోని డేటా నిర్మాణం, ఇది మార్చగల లేదా మార్చగల ఆదేశించిన మూలకాల క్రమం

మీకు తెలుసా?

జాబితా అక్షరాస్యులు చదరపు బ్రాకెట్లలో వ్రాయబడింది [ ] జాబితాలు స్ట్రింగ్ సమానంగా పనిచేస్తాయి.

13.6.4. ఒక CSV ఫైల్‌ను చదవండి మరియు క్రమబద్ధీకరించడానికి జాబితాలో కాలమ్ విలువను నిల్వ చేయండి (Read A CSV File And Store A Column Value In A List For Sorting)

ఈ ప్రోగ్రామ్‌లో మీరు వినియోగదారు నుండి కాలమ్ నెంబర్‌ను పొందడం ద్వారా "sample6.csv" ఫైల్ నుండి ఎంచుకున్న కాలమ్‌ను చదవబోతున్నారు మరియు కంటెంటెను జాబితాలో నిల్వ చేయబోతున్నారు.



	A	B	C	D	E	F	G
1	Item Name	Cost - Rs	Quantity	Profit			
2	Keyboard	480	12	1152			
3	Monitor	5200	10	10400			
4	Mouse	200	50	2000			
5							
6							
7							
8							
9							
10							

పటము 13.6.4 వరుస క్రమం కొఱకు *CSV* ఫైల్ డేటా జాబితా

మొదటి వరసను దాటవేయకుండా ఉండటానికి, అడ్డు వరుస శీర్షిక కూడా క్రమబద్ధీకరించబడుతుంది. "next()" కమాండ్ ఉపయోగించి ఇది చేయవచ్చును జాబితా క్రమబద్ధీకరించబడి, ప్రదర్శించబడుతుంది.

వినియోగదారు శీర్షిక కాలమ్‌ను వదిలిపెట్టి ఇచ్చిన ఎంచుకున్న కాలమ్‌ను క్రమబద్ధీకరించుట

12th_CS_Telugu_Medium_Chapter_13.indd 247 28-03-2023 10:06:48

```
# sort a selected column given by user leaving the header column in
# descending order of value
import csv
# other way of declaring the filename
inFile= 'c:\pyprg\sample6.csv'
# opening the csv file which is in the same location of where the current #
python file
```

```
F=open(inFile,'r')
# reading the File with the help of csv.reader()
reader = csv.reader(F)
# skipping the first row(heading)
next(reader)
# declaring a list
arrayValue = []
a = int(input ("Enter the column number between 0 to 3:-"))
# sorting a particular column-cost
for row in reader:
    arrayValue.append(row[a])
arrayValue.sort(reverse=True)
for row in arrayValue:
    print (row)
F.close()
```

అవుట్పుట్

```
Enter the column number between 0 to 3:- 2
50
12
10
```



ఒక csv ఫైల్‌లో ఒక నిర్ధిష్ట కాలమ్‌నుచదివి, దాని ఫలితాన్ని ఆరోహణ క్రమంలో ప్రదర్శిస్తుంది.

12th_CS_Telugu_Medium_Chapter_13.indd 248 28-03-2023 10:06:48

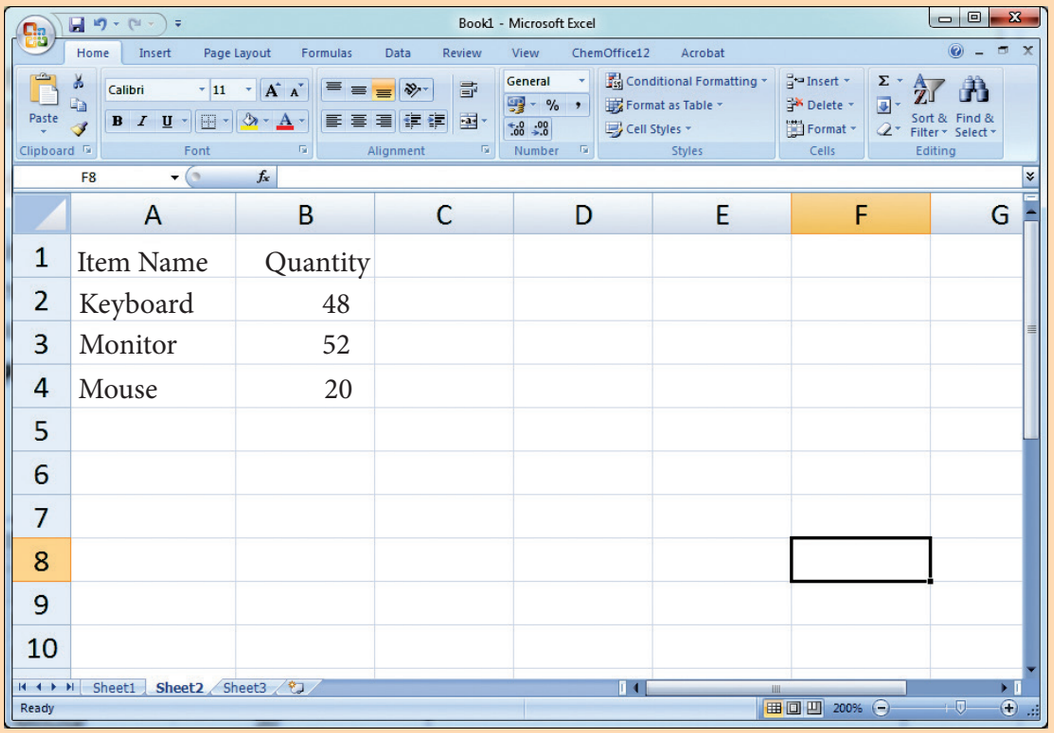
మీకు తెలుసా?

list_name.sort() ఆదేశం జాబితా విలువను ఆరోహణ క్రమంలో ఏర్పాటు చేస్తుంది sort(reverse=True) జాబితాను అవరోహణ క్రమంలో ఏర్పాటు చేయడానికి ఉపయోగిస్తారు.

13.6.5. ఒక నిర్ధిష్ట కాలమ్‌తో CSV ఫైల్‌ను క్రమబద్ధీకరించడం
Sorting A CSV File With A Specified Column

ఈ ప్రోగ్రామ్‌లో మీరు "sample8.csv" ఫైల్ యొక్క మొత్తం కంటెంట్ జాబితాకు బదిలీ చేయడాన్ని చూడబోతున్నారు అప్పుడు వరుసల జాబితా క్రమబద్ధీకరించబడుతుంది మరియు పరిమాణం యొక్క ఆరోహణ క్రమంలో ప్రదర్శించబడుతుంది. ఒకటి కంటే ఎక్కువ నిలువు వరుసల ద్వారా క్రమబద్ధీకరించడానికి మీరు బహుళ సూచికలతో itemgetter ను ఉపయోగించవచ్చు. ఆపరేటర్ itemgetter (1,2), "sample8.csv" యొక్క కంటెంట్

sample8.csv in Excel screen



sample8.csv in Notepad

ItemName ,Quantity
Keyboard, 48
Monitor,52
Mouse ,20

పటము 13.6.5 CSV ఫైల్ డేటాను జాబితాలోకి క్రమబద్ధీకరించడానికి

క్రింది ప్రోగ్రామ్ **operator.itemgetter(col_no)** ఉపయోగించి పైన పేర్కొన్న పనిని చేస్తుంది.

12th_CS_Telugu_Medium_Chapter_13.indd 249 28-03-2023 10:06:48

```
#Program to sort the entire row by using a specified column.
# declaring multiple header files
import csv ,operator
#One more way to read the file
data = csv.reader(open('c:\pyprg\sample8.csv'))
next(data)                    #(to omit the header)
#using operator module for sorting multiple columns
sortedlist = sorted (data, key=operator.itemgetter(1))    # 1 specifies we want to sort
                                                          # according to  second column
for row in sortedlist:
    print(row)
```

అవుట్పుట్

```
['Mouse ', '20']
['Keyboard ', '48']
['Monitor', '52']
```

గమనిక

క్రమబద్ధీకరించబడిన () పద్ధతి ఇచ్చిన వస్తువు యొక్క మూలకాలను ఒక నిర్దిష్ట క్రమంలో క్రమబద్ధీకరిస్తుంది. ఆరోహణ లేదా అవరోహణ. క్రమబద్ధీకరించిన () ఫంక్షన్ క్రమబద్ధీకరించిన () మాదిరిగానే పనిచేస్తుంది. తేడా క్రమబద్ధీకరణ () పద్ధతి ఎటువంటి విలువను తిరిగి ఇవ్వదు మరియు అసలు జాబితాను కూడా మారుస్తుంది.

"sample8.csv" లో మరో కాలమ్‌ని జోడించండి మరియు వాక్యనిర్మాణాన్ని ఉపయోగించి ఖర్చు యొక్క అవరోహణ క్రమంలో క్రమబద్ధీకరించండి

```
sortedlist = sorted(data, key=operator.itemgetter(Col_number) ,reverse=True)
```

13.6.6 CSV ఫైల్‌ను ఒక నిఘంటువులోకి చదవడం (Reading CSV File Into A Dictionary)

CSV ఫైల్‌ను డిక్షనరిలో చదవడానికి CSV మాడ్యూల్ యొక్క **DictReader** ఫంక్షన్‌ని ఉపయోగించడం ద్వారా చదవవచ్చు, ఇది రీడర్ ( ) క్లాస్‌కు సమానంగా పనిచేయును కాని డిక్షనరీకి డేటాను మ్యాప్ చేసే వస్తువును సృష్టిస్తుంది. ఫీల్డ్‌నేమ్‌ల ద్వారా ఇవ్వబడిన కీలు పారామితిగా ఇవ్వబడ్డాయి. CSV యొక్క మొదటి పంక్తిని చదవడం ద్వారా మరియు ఈ పంక్తిలోని విలువను కామాతో వేరు చేయుటకు డిక్షనరీ కీ ఉపయోగించడం ద్వారా **DictReader** పనిచేస్తుంది. ప్రతి వరుసలోని తరువాత నిలువు వరుసలు నిఘంటువు విలువల వలె ప్రవర్తిస్తాయి మరియు తగిన కీతో (అంటే ఫీల్డ్ పేరు) యాక్సెస్ చేయవచ్చు.

మీ CSV యొక్క మొదటి వరుసలో మీ కాలమ్ పేర్లు లేకపోతే, డిక్షనరీ కీలను మాన్యువల్‌గా కేటాయించడానికి మీరు ఫీల్డ్ నేమ్స్ పారామితిని DictReader's యొక్క కన్స్ట్రక్టర్‌లోకి పంపవచ్చు.

csv.reader() మరియు DictReader() ల మధ్య ప్రధాన వ్యత్యాసం csv.writer రీడర్ మరియు **list/ tuple,** పనిచేస్తుండగా, csv.DictReader మరియు DictWriter నిఘంటువు **keys** గా ఉపయోగించబడే అదనపు ఆర్గ్యుమెంట్ ఫీల్డ్ పేర్లను తీసుకుంటాయి.

ఉదాహరణ కోసం "sample8.csv" ఫైల్‌ను నిఘంటువులోకి చదవడం

12th_CS_Telugu_Medium_Chapter_13.indd 250 28-03-2023 10:06:48

```
import csv
filename = 'c:\pyprg\sample8.csv'
input_file =csv.DictReader(open(filename,'r'))
for row in input_file:
    print(dict(row))                    #dict() to print data
```

అవుట్పుట్

```
{'ItemName ': 'Keyboard ', 'Quantity': '48'}
{'ItemName ': 'Monitor', 'Quantity': '52'}
{'ItemName ': 'Mouse ', 'Quantity': '20'}
```

పై ప్రోగ్రామ్‌లో "sample8.csv" ఫైల్‌ను చదవడానికి మరియు డిక్షనరీలోకి మ్యాప్ చేయడానికి DictReader() ఉపయోగించబడుతుంది. అప్పుడు ఫంక్షన్ dict() ఆర్డర్ లేకుండా డేటాను డిక్షనరీ ఫార్మాట్‌లో ప్రింట్ చేయడానికి ఉపయోగించబడుతుంది.



పై ప్రోగ్రామ్ నుండి డిక్ట్ dict() ఫంక్షన్‌ను తీసివేసి, ప్రింట& (అడ్డువరుస) ను వాడండి. మీరు ఈ క్రింది అవుట్‌పుట్ పొందుతున్నారో లేదో తనిఖీ చేయండి.

```
OrderedDict([('ItemName ', 'Keyboard '), ('Quantity', '48')])
OrderedDict([('ItemName ', 'Monitor'), ('Quantity', '52')])
OrderedDict([('ItemName ', 'Mouse '), ('Quantity', '20')])
```

13.6.7. యూజర్ డిపైన్డ్ డీలిమిటర్‌తో CSV ఫైల్‌ను డిక్షనరీలోకి చదవడం (Reading CSV File With User Defined Delimiter Into A Dictionary)

కొత్త dialects ను నమోదు చేసుకొని DictReader() పద్ధతుల మనము ఉపయోగించవచ్చు.

```
ItemName|Quantity
Keyboard|48
Monitor|52
Mouse|20
```

అప్పుడు కొత్త dialect నమోదు చేయడం ద్వారా "sample8.csv" నిఘంటువులో చదవవచ్చు.

```
import csv
csv.register_dialect('myDialect',delimiter = '|',skipinitialspace=True)
filename = 'c:\pyprg\ch13\sample8.csv'
with open(filename, 'r') as csvfile:
    reader = csv.DictReader(csvfile, dialect='myDialect')
for row in reader:
    print(dict(row))
csvfile.close()
```

అవుట్పుట్

```
{'ItemName':'Keyboard','Quantity': 48}
{'ItemName' :'Monitor':'Quantity':52}
{'ItemName': 'Mouse':'Quantity': 20}
```


12th_CS_Telugu_Medium_Chapter_13.indd 251 28-03-2023 10:06:48

గమనిక

DictReader() దాని అవుట్‌పుట్‌లో అప్రమేయంగా OrderedDict ఇస్తుంది. OrderedDict అనేది డిక్షనరీ సబ్‌క్లాస్. ఇది దాని విషయాలు జోడించబడిన క్రమాన్ని ఆదా చేస్తుంది. OrderedDict ను dict() తొలగించడానికి OrderedDict ను ఉపయోగించవలయును.

13.7 Csv ఫైళ్ళలో వివిధ రకాలుగా డేటాను రాయడం

మీకు తెలిసినట్లుగా పైథాన్ CSV ఫైల్‌తో పనిచేయడానికి సులభమైన మార్గాన్ని అందిస్తుంది. మరియు CSV ఫైల్‌లో డేటాను చదవడానికి మరియు వ్రాయడానికి CSV మాడ్యూల్ ఉంది. మునుపటి అంశాలలో, పైథాన్‌లో CSV ఫైల్‌లను ఎలా చదవాలో మీరు నేర్చుకున్నారు. అదే విధంగా మీరు పైథాన్‌లో క్రొత్తదాన్ని వ్రాయవచ్చు లేదా ఇప్పటికే ఉన్న CSV ఫైల్‌లను సవరించవచ్చు.

1. క్రొత్త సాధారణ CSV ఫైల్‌ను సృష్టించడం
2. ఇప్పటికే ఉన్న ఫైల్‌ను సవరించడం
3. కోట్స్‌తో CSV ఫైల్‌లో రాయడం
4. కస్టమ్ డీలిమిటర్లతో CSV ఫైల్‌లో రాయడం
5. Lineterminator తో CSV ఫైల్‌లో రాయడం
6. కోట్చార్లతో ఒక CSV ఫైల్‌లో రాయడం
7. CSV ఫైల్‌ను ఒక నిఘంటువులోకి రాయడం
8. రన్‌టైమ్‌లో డేటాను పొందడం మరియు ఫైల్‌లో రాయడం

13.7.1 క్రొత్త సాధారణ CSV ఫైల్‌ను సృష్టించడం (Creating A New Normal CSV File)

మీరు CSV ఫైల్‌లో నిల్వ చేయదలచిన డేటా సమితిని కలిగి ఉన్నప్పుడు, దీనికి విరుద్ధంగా మరియు **write()** ఉపయోగించాలి.

csv.writer() ఫంక్షన్ రచయిత ఆబ్జెక్ట్‌ను తిరిగి ఇస్తుంది. ఇది వినియోగదారుని యొక్క డేటాను ఇచ్చిన ఫైల్ లాంటి వస్తువుపై వేరు చేయబడిన స్ట్రింగ్‌లుగా మారుస్తుంది **writerow()** ఫంక్షన్ పేర్కొన్న ఫైల్‌లోకి వరుస క్రమంలో డేటాను వ్రాస్తుంది.

csv.writer() కొరకు వాక్యనిర్మాణం

```
csv.writer(fileobject,delimiter,fmtparams)
```

ఇక్కడ

fileobject :-	ఫైల్ యొక్క మార్గం మరియు పద్ధతిని పంపును
delimiter :-	| Pipe వంటి ప్రామాణిక dilects ను కలిగి ఉన్న ఐచ్చిక పారామితి మొదలైనవి తొలగించవచ్చు.

12th_CS_Telugu_Medium_Chapter_13.indd 252 28-03-2023 10:06:48

fmtparams :	skipinitialspace, quoting మొదలైన dialects డిఫాల్ట్ విలువలను భర్తీ చేయడానికి సహాయపడే ఐచ్ఛిక పారామితిని వదిలివేయవచ్చు.

డిఫాల్ట్ డీలిమిటర్ కామా (,) కలిగి ఉన్న CSV మాడ్యూల్ యొక్క writer() ఫంక్షన్‌ని ఉపయోగించి మీరు సాధారణ CSV ఫైల్‌ను సృష్టించవచ్చు.

(ఉదాహరణకు)

క్రింది పైథాన్ ప్రోగ్రామ్ డేటా జాబితాను "Pupil.csv" అని పిలిచే CSV ఫైల్‌గా మారుస్తుంది. (కామా) విలువను వేరుచేసేదిగా ఉపయోగిస్తుంది.

```
Import csv
csvData = [['Student', 'Age'], ['Dhanush', '17'], ['Kalyani', '18'], ['Ram', '15']]
with open('c:\pyprg\ch13\Pupil.csv', 'w') as CF:
    writer = csv.writer(CF)                  # CF is the file object
    writer.writerows(csvData)                # csvData is the List name
CF.close()
```

మీరు టెక్స్ట్ ఎడిటర్‌తో "Pupil.csv" ఫైల్‌ను తెరిచినప్పుడు, అది ఈ క్రింది విధంగా contentను చూపుతుంది.

Student, Age
Dhanush, 17
Kalyani, 18
Ram, 15

పై ప్రోగ్రామ్‌లో csv.writer() ఫంక్షన్ "csvData" జాబితాలోని మొత్తం డేటాను తీగలుగా మారుస్తుంది మరియు కంటెంట్‌ను ఆబ్జెక్ట్ వంటి ఫైల్‌గా సృష్టిస్తుంది writerows () పద్ధతి అన్ని డేటాను క్రొత్త CSV ఫైల్ "Pupil.csv". కు వ్రాస్తుంది.

గమనిక

writerow() ఫంక్షన్ ఒక సమయంలో ఒక వరుసను వ్రాస్తుంది. మీరు మొత్తం డేటాను ఒకేసారి వ్రాయవలసి వస్తే మీరు writerow() ఫంక్షన్‌ని ఉపయోగించవచ్చు.

13.7.2. ఇప్పటికే ఉన్న ఫైల్‌ను సవరించడం (Modifying An Existing File)

ఇప్పటికీ ఉన్న ఫైల్ యొక్క డేటాలో కొన్ని మార్పులు చేయడం లేదా ఎక్కువ డేటాను జోడించడం సవరణ అంటారు. ఉదాహరణకు "student.csv" ఫైల్ క్రింది డేటాను కలిగి ఉంటుంది.

Roll No, Name, City
1, Harshini, Chennai
2, Adhith, Mumbai
3, Dhuruv, Bangalore
4, Krishna, Tiruchy
5, Venkat, Madurai

12th_CS_Telugu_Medium_Chapter_13.indd 253 28-03-2023 10:06:48

క్రింది ప్రోగ్రామ్ "student.csv" లో ఇప్పటికి ఉన్న అడ్డు వరుస విలువను సవరించడం ద్వారా student.csv ఫైల్‌ను సవరించడం.

```
import csv
row = ['3', 'Meena',"Bangalore']
with open('student.csv', 'r') as readFile:
    reader = csv.reader(readFile)
    lines = list(reader)        # list()- to store each row of data as a list
    lines[3] = row
with open('student.csv', 'w') as writeFile:
# returns the writer object which converts the user data with delimiter
    writer = csv.writer(writeFile)
#writerows()method writes multiple rows to a csv file
    writer.writerows(lines)
readFile.close()
writeFile.close()
```

మనము టెక్స్ట్ ఎడిటర్‌తో student.csv ఫైల్‌ను తెరిచినప్పుడు, అది క్రింది విధంగా చూపబడుతుంది.

Roll No, Name, City

1, Harshini, Chennai

2, Adhith, Mumbai

3, Meena, Bangalore

4, Krishna, Tiruchy

5, Venkat, Madurai

పై ప్రోగ్రామ్‌లో "student.csv" యొక్క మూడవ వరుస సవరించి సేవ్ చేయబడింది. మొదట "student.csv" ఫైల్ csv.reader() ఫంక్షన్ ఉపయోగించి చదవబడుతుంది. అప్పుడు జాబితా () ఫైల్ యొక్క ప్రతి అడ్డు వరుసను నిల్వ చేస్తుంది. పంక్తులు "lines[3] = row" అడ్డు వరుస అనే ప్రకటన, ఫైల్ యొక్క మూడవ వరుసను అడ్డు వరుసలోని క్రొత్త కంటెంట్‌తో మార్చింది. రైట్‌రోస్ (పంక్తులు) ఉపయోగించి ఫైల్ రైటర్ జాబితా యొక్క విలువలను "student.csv" ఫైల్‌కు వ్రాస్తాడు

13.7.2.1. క్రొత్త వరుసను జోడించడం (ADDING NEW ROW)

కొన్నిసార్లు మీరు ఇప్పటికే ఉన్న CSV ఫైల్‌లో కొత్త అడ్డు వరుసను జోడించాల్సి ఉంటుంది. ఫైల్ చివరిలో కొత్త అడ్డు వరుసను జోడించడంను వరుసను జోడించడం (appending a row) అంటారు.

క్రింది ప్రోగ్రామ్ ఇప్పటికే ఉన్న "student.csv" ఫైల్‌కు కొత్త అడ్డు వరుసను జోడిస్తుంది.

12th_CS_Telugu_Medium_Chapter_13.indd 254 28-03-2023 10:06:48

```
import csv
row = ['6', 'Sajini ', 'Madurai']
with open('student.csv', 'a') as CF:            # append mode to add data at the end
    writer = csv.writer(CF)
    writer.writerow(row)    # writerow() method write a single row of data in file
CF.close()
```

టెక్స్ట్ ఎడిటర్‌తో "student.csv" ఫైల్ తెరిచినప్పుడు అది ఈ క్రింది విధంగా ప్రదర్శిస్తుంది.

Roll No, Name, City

1, Harshini, Chennai

2, Adhith, Mumbai

3, Meena, Bangalore

4, Krishna, Tiruchy

5, Venkat, Madurai

6, Sajini, Madurai

పై ప్రోగ్రామ్‌లో, క్రొత్త వరుస "student.csv". లో చేర్చబడుతుంది. దీని కోసం CSV ఫైల్ మాత్రమే 'a' అనుబంధ మోడ్‌లో తెరవబడుతుంది. అపెండ్ మోడ్ "student.csv" ఫైల్ యొక్క చివరి పంక్తి తర్వాత అడ్డు వరుస విలువను వ్రాయండి

మీకు తెలుసా?

'w' మోడ్ క్రొత్త ఫైల్‌ను సృష్టిస్తుంది. ఫైల్ ఇప్పటికీ ఉన్నట్లయితే 'w' మోడ్ దానిపై వ్రాస్తుంది. ఫైల్ ఇప్పటికీ ఉన్నట్లయితే 'a' అనుబంధ మోడ్ ఫైల్ చివరిలో డేటాను జోడిస్తే లేకపోతే క్రొత్తదాన్ని సృష్టిస్తుంది.

గమనిక

writerow() 1 డైమన్షనల్ డేటా (ఒక అడ్డు వరుస) మరియు రైట్‌రోస్ 2 డైమెన్షనల్ డేటా (బహుళ వరుసలు) వ్రాయడానికి ఒక ఫైల్‌లో పడుతుంది.

13.7.3. కోట్స్‌తో CSV ఫైళ్ళు (CSV Files With Quotes)

csv మాడ్యూల్ యొక్క **csv.register_dialect()** క్లాజును ఉపయోగించి కొత్త **dialect** ను నమోదు చేయుట ద్వారా కోట్స్‌తో csv ఫైల్‌ను వ్రాయగలము. ఈ క్రింది ప్రోగ్రామ్ దీనిని విశదీకరించును.

```
import csv
info = [['SNO', 'Person', 'DOB'],
['1', 'Madhu', '18/12/2001'],
['2', 'Sowmya',19/2/1998'],
['3', 'Sangeetha',20/3/1999'],
['4', 'Eshwar', '21/4/2000'],
['5', 'Anand', '22/5/2001']]
csv.register_dialect('myDialect',quoting=csv.QUOTE_ALL)
with open('c:\pyprg\ch13\person.csv', 'w') as f:
    writer = csv.writer(f, dialect='myDialect')
    for row in info:
        writer.writerow(row)
f.close()
```

మీరు "person.csv" ఫైల్‌ను తెరిచినప్పుడు మనము ఈ క్రింది అవుట్‌పుట్‌ను పొందుతాము

```
"SNO","Person","DOB"
"1","Madhu","18/12/2001"
"2","Sowmya","19/2/1998"
"3","Sangeetha","20/3/1999"
"4","Eshwar","21/4/2000"
"5","Anand","22/5/2001"
```

పై ప్రోగ్రామ్‌లో myDialect అనే కొత్త Dialect నమోదు చేయబడింది (ప్రకటించబడింది) కోటింగ్ = csv. QUOTE_ALL అన్ని విలువలపై డబుల్ కోట్స్ రాయడానికి అనుమతిస్తుంది.

13.7.4. వినియోగదారుని డీలిమిటర్లతో CSV ఫైళ్ళు (CSV Files With Custom Delimiters)

డీలిమిటర్ అనేది ఫీల్డ్‌లను వేరు చేయడానికి ఉపయోగించే స్ట్రింగ్, డిఫాల్ట్ విలువ కామా **csv.register_dialect()** సహాయంతో క్రొత్త **dialect** నమోదు చేయడం ద్వారా క్రొత్త CSV ఫైళ్ళలో వినియోగదారుని డీలిమిటర్ కలిగి ఉండవచ్చు. ఈ ఉదాహరణ ప్రోగ్రామ్ కస్టమ్ డీలిమిటర్ ఫైప్ (|) ఉపయోగించి వ్రాయబడింది.

```
import csv
info = [['SNO', 'Person', 'DOB'],
['1', 'Madhu', '18/12/2001'],
['2', 'Sowmya',19/2/1998'],
['3', 'Sangeetha',20/3/1999'],
['4', 'Eshwar', '21/4/2000'],
['5', 'Anand', '22/5/2001']]
csv.register_dialect('myDialect',delimiter = '|')
with open('c:\pyprg\ch13\dob.csv', 'w') as f:
    writer = csv.writer(f, dialect='myDialect')
     for row in info:
        writer.writerow(row)
f.close()
```


12th_CS_Telugu_Medium_Chapter_13.indd 256 28-03-2023 10:06:48

మనము “dob.csv” ఫైల్‌ను తెరిచినపుడు మనము ఈ క్రింది అవుట్‌పుట్‌ను పొందుతాము.

```
SNO|Person|DOB
1|Madhu|18/12/2001
2|Sowmya|19/2/1998
3|Sangeetha|20/3/1999
4|Eshwar|21/4/2000
5|Anand|22/5/2001
```

పై ప్రోగ్రామ్‌లో డీలిమిటర్‌తో పైప్ (|) ఉన్న కొత్త dialect నమోదు చేయబడింది. అప్పుడు “info” list ను CSV ఫైల్ “dob.csv” లో వ్రాయబడుతుంది.

గమనిక

dialect పారామితి skipinitialspace యొక్క విలువ True అయిన డీలిమిటర్‌ను అనుసరించే వైట్‌స్పేస్ విస్మరించబడును దీని డీపాల్ట్ default విలువ False అగును.

13.7.5. లైన్ టెర్మినేటర్‌తో CSV ఫైల్ (CSV File With A Line Terminator)

లైన్ టెర్మినేటర్ అనేది () పంక్తులను ముగించడానికి ఉపయోగించే స్ట్రింగ్ డిఫాల్ట్ విలువ (లేదా) \r మాడ్యూల్ యొక్క \n తరగతిని ఉపయోగించి కొత్త csv.register_dialect() ను నమోదు చేయడం ద్వారా ఫైథాన్ లైన్ టెర్మినేటర్‌తో మనము csv ఫైల్‌ను వ్రాయవచ్చు. ఉదాహరణకి

```
import csv
Data = [['Fruit', 'Quantity'], ['Apple', '5'], ['Banana', '7'], ['Mango', '8']]
csv.register_dialect('myDialect', delimiter = '|', lineterminator = '\n')
with open('c:\pyprg\ch13\line.csv', 'w') as f:
    writer = csv.writer(f, dialect='myDialect')
    writer.writerows(Data)
f.close()
```

మనము line.csv ఫైల్‌ను తెరిచినపుడు పంక్తుల మధ్య అంతరం ఉన్న క్రింది అవుట్‌పుట్‌ను పొందుతాము.

```
Fruit|Quantity
Apple|5
Banana|7
Mango|8
```

పై కోడ్‌లో, క్రొత్త dialect “myDialect డీలిమిటర్‌ను ఉపయోగిస్తుంది ='|' ఇక్కడ ఒక | (pipe) (పైపు) కాలమ్ సెపరేటర్‌గా పరిగణించబడుతుంది. లైన్ టెర్మినేటర్ ='\r\n\r\n' ప్రతి అడ్డువరుసను వేరు చేస్తుంది మరియు నోట్‌ప్యాడ్‌లో ఒక ఖాళీ పంక్తి తర్వాత డేటాను ప్రదర్శిస్తుంది.

12th_CS_Telugu_Medium_Chapter_13.indd 257 28-03-2023 10:06:48

గమనిక

పైథాన్ యొక్క CSV మాడ్యూల్ line \r\n, \n లేదా line r ను లైను టర్మినేటర్‌గా మాత్రమే అంగీకరిస్తుంది.

13.7.6 కోట్ అక్షరాలతో CSV ఫైల్ (CSV File with quote characters)

CSV మాడ్యూల్ యొక్క csv.register_dialect() తరగతిని ఉపయోగించి క్రొత్త dialect ను నమోదు చేయడం ద్వారా మీరు CSV ఫైల్‌ను కస్టమ్ కోట్ అక్షరాల (custom quote characters) తో వ్రాయవచ్చు.

```
import csv
csvData = [['SNO','Items'], ['1','Pen'], ['2','Book'], ['3','Pencil']]
csv.register_dialect('myDialect',delimiter = '|',quotechar = '"',
    quoting=csv.QUOTE_ALL)
with open('c:\pyprg\ch13\quote.csv', 'w') as csvFile:
    writer = csv.writer(csvFile, dialect='myDialect')
    writer.writerows(csvData)
print("writing completed")
csvFile.close()
```

మీరు నోట్‌ప్యాడ్లో "quote.csv" ఫైల్ను తెరిచినప్పుడు మేము ఈ క్రింది అవుట్‌పుట్‌ని పొందుతాము.

```
"SNO|"Items
1|"Pen"
2|"Book"
3|"Pencil"
```

పై ప్రోగ్రామ్‌లో ఫైల్ లోపల వ్రాయడానికి myDialect పైప్ (|) ను డీలిమిటర్‌గా మరియు కోట్‌చార్‌ను డబుల్‌కోట్ "" గా ఉపయోగిస్తుంది.

13.7.7 CSV ఫైల్‌ను ఒక నిఘంటువులోకి రాయడం (Writing CSV File Into A Dictionary)

csv మాడ్యూల్ యొక్క DictWriter() తరగతిని ఉపయోగించి మనము ఒక csv ఫైల్‌ను నిఘంటువులోకి వ్రాయవచ్చును. ఇది నిఘంటువులోకి డేటాను మ్యాప్ చేసే ఒక వస్తువును సృష్టిస్తుంది. keys ఫీల్డ్ నేమ్స్ పారామితి ద్వారా ఇవ్వబడతాయి. క్రింది ప్రోగ్రామ్ ఫైల్‌లో నిఘంటువును వ్రాయడానికి సహాయపడుతుంది.

```
import csv
data = [{'MOUNTAIN' : 'Everest', 'HEIGHT': '8848'},
    {'MOUNTAIN' : 'Anamudi ', 'HEIGHT': '2695'},
    {'MOUNTAIN' : 'Kanchenjunga', 'HEIGHT': '8586'}]
with open('c:\pyprg\ch13\peak.csv', 'w') as CF:
fields = ['MOUNTAIN', 'HEIGHT']
    w = csv.DictWriter(CF, fieldnames=fields)
    w.writeheader()
    w.writerows(data)
    print("writing completed")
CF.close()
```


12th_CS_Telugu_Medium_Chapter_13.indd 258 28-03-2023 10:06:48

మీరు నోట్‌ప్యాడ్‌లో “peak.csv” ఫైల్‌ను తెరిచినప్పుడు మీరు ఈ క్రింది అవుట్‌పుట్‌ను పొందుతారు

```
MOUNTAIN,HEIGHT
Everest,8848
Anamudi,2695
Kanchenjunga,8586
```

పై ప్రోగ్రామ్‌లో ఫీల్డ్ పేర్లను csv ఫైల్‌లోని ప్రతి కాలమ్ యొక్క శీర్షికలుగా ఉండును. అప్పుడు డిక్షనరీ లోని డేటాను “peak.csv” ఫైల్‌లో వ్రాయడానికి డిక్ట్‌రైటర్ () ను ఉపయోగించండి.

13.7.7.1 కస్టమ్ మాండలికాలతో నిఘంటువును CSV ఫైల్‌లోకి రాయడం (Writing Dictionary Into CSV File With Custom Dialects)

```
import csv
csv.register_dialect('myDialect', delimiter = '|', quoting=csv.QUOTE_ALL)
with open('c:\pyprg\ch13\grade.csv', 'w') as csvfile:
    fieldnames = ['Name', 'Grade']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames, dialect="myDialect")
    writer.writeheader()
    writer.writerows([{'Grade': 'B', 'Name': 'Anu'},
                      {'Grade': 'A', 'Name': 'Beena'},
                      {'Grade': 'C', 'Name': 'Tarun'}])
print("writing completed")
```

మనము grade.csv ఫైల్‌ను తెరిచినప్పుడు అది క్రింది అవుట్‌పుట్‌ను కలిగి ఉంటుంది.

```
"Name"|"Grade"
"Anu"|"B"
"Beena"|"A"
"Tarun"|"C"
```

పై ప్రోగ్రామ్‌లో డీలిమిటర్‌గా myDialect పైప్ (|) అని పిలువబడే custom dialect ఒక csv ఫైల్‌లో వ్రాయడానికి ఫీల్డ్ పేర్లను ప్రతి కాలమ్ యొక్క శీర్షికలుగా ఉపయోగిస్తుంది. చివరగా, డిక్షనరీ డేటాను “grade.csv” ఫైల్‌లోకి వ్రాయడానికి మనము DictWriter() ను ఉపయోగిస్తాము.

13.7.8. రన్‌టైమ్‌లో డేటాను పొందడం మరియు CSV ఫైల్‌లో రాయడం (Getting Data At Runtime And Writing It In a CSV File)

మీరు కీబోర్డ్ ద్వారా డేటాను అంగీకరించవచ్చు మరియు CSV ఫైల్‌కు వ్రాయవచ్చు. ఉదాహరణకు, క్రింది ప్రోగ్రామ్ user నుండి డేటాను కీ బోర్డు ద్వారా అంగీకరిస్తుంది మరియు దానిని “dynamicfile.csv” అనే ఫైల్‌లో నిల్వ చేస్తుంది. ఇది ఫైల్ యొక్క విషయాన్ని కూడా ప్రదర్శిస్తుంది.

12th_CS_Telugu_Medium_Chapter_13.indd 259 28-03-2023 10:06:48

```
import csv
with open('c:\pyprg\ch13\dynamicfile.csv', 'w') as f:
    w = csv.writer(f)
    ans='y'
    while (ans=='y'):
        name = input("Name?: ")
        date = input("Date of birth: ")
        place = input("Place: ")
        w.writerow([name, date, place])
        ans=input("Do you want to enter more y/n?: ")
    F=open('c:\pyprg\ch13\dynamicfile.csv",'r')
    reader = csv.reader(F)
    for row in reader:
        print(row)
    F.close()
```

అవుట్పుట్

```
Name?: Nivethitha
Date of birth: 12/12/2001
Place: Chennai
Do you want to enter more y/n?: y
Name?: Leena
Date of birth: 15/10/2001
Place: Nagercoil
Do you want to enter more y/n?: y
Name?: Padma
Date of birth: 18/08/2001
Place: Kumbakonam
Do you want to enter more y/n?: n
['Nivethitha', '12/12/2001', 'Chennai']
[]
['Leena', '15/10/2001', 'Nagercoil']
[]
['Padma', '18/08/2001', 'Kumbakonam']
```

MS Excel output

H8 ^ *fx*

	A	B	C
1	Nivethitha	12/12/2001	Chennai
2			
3	Leena	15/10/2001	Nagercoil
4			
5	Padma	18/08/2001	Kumbakonam
6			

12th_CS_Telugu_Medium_Chapter_13.indd 260 28-03-2023 10:06:48

గుర్తించుకోవలసిన పాయింట్లు:

- CSV ఫైల్ అనేది మనిషి చదవగలిగే టెక్స్ట్ ఫైల్, ఇక్కడ ప్రతి వాక్యంలో అనేక ఫీల్డ్‌లు ఉంటాయి. వీటిని కామాలతో లేదా ఇతర డీలిమిటర్‌తో వేరు చేస్తారు.
- ఎక్సెల్ ఒక బైనరీ ఫైల్ అయితే CSV ఫార్మాట్ సాదా టెక్స్ట్ ఫార్మాట్
- CSV ఫైల్‌ను చదవడానికి రెండు మార్గాలు csv.reader() ఫంక్షన్‌ను ఉపయోగించడం మరియు DictReader క్లాస్‌ని ఉపయోగించడం.
- చదవడం మరియు వ్రాయడంలో CSV ఫైల్ యొక్క డిఫాల్ట్ మోడ్ టెక్స్ట్ మోడ్
- Image లేదా exe ఫైల్ వంటి టెక్స్ట్ కాని ఫైల్‌లతో వ్యవహరించేటప్పుడు బైనరీ మోడ్‌ను ఉపయోగించవచ్చు.
- close() ఫంక్షన్ ఫైల్‌తో ముడిపడి ఉన్న వనరులను విముక్తి చేస్తుంది.
- డిఫాల్ట్‌గా CSV ఫైల్‌లు ఎక్సెల్‌లో స్వయంచాలకంగా తెరవబడతాయి.
- CSV లైబ్రరీలో CSV ఫైళ్ళ డేటాను చదవడానికి వ్రాయడానికి మరియు ప్రాసెస్ చేయడానికి objects మరియు ఇతర కోడ్‌లు ఉన్నాయి.
- డీలిమిటర్ తర్వాత వైట్‌స్పేస్‌లను తొలగించడానికి “skipinitialspace” ఉపయోగించబడుతుంది.
- ఒకటికంటే ఎక్కువ కాలమ్‌లను క్రమబద్ధీకరించడానికి operator.itemgetter() ఉపయోగించవచ్చును.
- CSV మాడ్యూల్ యొక్క DictReader() తరగతి ఒక వస్తువు objects ను సృష్టిస్తుంది. ఇది డేటాను నిఘంటువుకు దారి చూపును.
- Custom డీలిమిటర్ కలిగిన CSV ఫైల్‌ను csv.register_dialect() సహాయంతో చదవబడుతుంది.
- బహుళ సూచికలతో ఒకటి కంటే ఎక్కువ column itemgetter() ద్వారా క్రమబద్ధీకరించడానికి ఉపయోగించబడుతుంది.
- ప్రస్తావించబడని objects ను శుభ్రపరచుటకు పైథాన్‌లో garbage collector కలదు.
- csv.reader మరియు csv.writer list/tuple తో పని చేస్తాయి. అయిన csv.DictReader మరియు csv.DictWriter నిఘంటువుతో పనిచేస్తాయి.
- csv.DictReader మరియు csv.DictWriter డిక్షనరీ keys గా ఉపయోగించబడే అదనపు ఆర్గ్యుమెంట్ ఫీల్డ్ పేర్లను తీసుకుంటాయి.
- Dict ( ) ఫంక్షన్, డిక్షనరీ ఫార్మాట్‌లో క్రమంగా డేటాను ప్రింట్ చేయడానికి ఉపయోగించబడుతుంది.
- csv.writer() ఫంక్షన్ వినియోగదారుని యొక్క డేటాను వేరుచేసి స్ట్రింగ్‌లుగా మార్చే writer object no గా తిరిగి ఇస్తుంది.
- write రైట్రో () ఫంక్షన్ ఒక సమయంలో ఒక వరుసను వ్రాస్తుంది. అన్ని డేటాను ఒకేసారి వ్రాయడానికి రైట్‌రోస్ () ఫంక్షన్ ఉపయోగించబడుతుంది.
- ఫైల్ చివరిలో క్రొత్త అడ్డు వరుసను జోడించడం, వరుసను జోడించడం appending అడ్డువరుస అంటారు.

12th_CS_Telugu_Medium_Chapter_13.indd 261 28-03-2023 10:06:49



అనుభవంపైక చేతులు

1. క్రింది Namelist.csv ఫైల్‌ను చదవడానికి ఫైథాన్ ప్రోగ్రామ్‌ను వ్రాసి, జాబితాలోని పేర్ల అక్షర క్రమంలో డేటాను క్రమబద్ధీకరించండి మరియు అవుట్‌పుట్‌ను ప్రదర్శిస్తుంది.

	A	B	C
1	SNO	NAME	OCCUPATION
2	1	NIVETHITHA	ENGINEER
3	2	ADHITH	DOCTOR
4	3	LAVANYA	SINGER
5	4	VIDHYA	TEACHER
6	5	BINDHU	LECTURER

2. 5 మంది విద్యార్థుల పేర్లు మరియు ఐదు సబ్జెక్టుల మార్కులను అంగీకరించడానికి ఫైథాన్ ప్రోగ్రామ్ రాయండి. మరియు మొత్తాన్ని కనుగొని విద్యార్థుల వివరాలన్నింటినీ ఒక CSV ఫైల్‌లో నిల్వ చేయండి.



మూల్యాంకనం

MQDPW

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. CSV ఫైల్ అనునది

   అ) Flat File ఆ) 3D File

   ఇ) String File ఈ) Random File

2. CRLF విస్తరణ

   అ) Control Return and Line Feed

   ఆ) Carriage Return and Form Feed

   ఇ) Control Router and Line Feed

   ఈ) Carriage Return and Line Feed

3. CSV ఫైళ్ళపై అనేక ఆపరేషన్లు ప్రకియలు చేయడానికి ఫైథాన్ క్రింది వాటిలో ఏ మాడ్యూల్ కలదు?

   అ) py ఆ) xls

   ఇ) csv ఈ) os

4. exe లేదా ఎక్స్‌ఫైల్స్ వంటి టెక్ట్స్ కాని ఫైళ్ళతో వ్యవహరించేటప్పుడు క్రింది మోడ్ ఏది ఉపయోగించబడుతుంది?

   అ) టెక్ట్స్ మోడ్ ఆ) బైనరీ మోడ్

   ఇ) xls మోడ్ ఈ) csv మోడ్

12th_CS_Telugu_Medium_Chapter_13.indd 262 28-03-2023 10:06:49

5. CSV ఫైల్‌లో వరుసను డాటవేయడానికి ఉపయోగించే ఆదేశం

అ) next()　　ఆ) skip()

ఇ) omit()　　ఈ) bounce()

6. సిఎస్వి మాడ్యూల్ యొక్క writer() ఫంక్షన్ ద్వారా ఉత్పత్తి చేయబడిన పంక్తులను ముగించడానికి ఈ క్రింది వాటిలో స్ట్రింగ్ ఏది?

అ) Line Terminator　　ఆ) Enter key

ఇ) Form feed　　ఈ) Data Terminator

7. క్రింది ప్రోగ్రామ్ యొక్క అవుట్‌పుట్ ఏమిటి? csv అను

```
d=csv.reader(open('c:\PYPRG\ch13\city.csv'))
next(d)
for row in d:
   print(row)
```

if the file called “city.csv” contain the following details

```
chennai,mylapore
mumbai,andheri
```

అ) chennai,mylapore　　ఆ) mumbai,andheri

ఇ) chennai
mumbai　　ఈ) chennai,mylapore
mumbai,andheri

8. క్రింది వాటిలో ఏది dictionary కు డేటాను మ్యాప్ చేసే వస్తువును సృష్టిస్తుంది?

అ) listreader()　　ఆ) reader()

ఇ) tuplereader()　　ఈ) DictReader ()

9. ఇప్పటికే ఉన్న ఫైల్ యొక్క డేటాలో కొన్ని మార్పులు చేయడం లేదా అధిక డేటాను జోడించడంను ఈ విధంగా అంటారు.

అ) ఎడిటింగ్　　ఆ) జోడించడం

ఇ) సవరణ　　ఈ) మార్పు

10. test.csv ఫైల్‌లో క్రింది ప్రోగ్రామ్ ఏ వివరన్ని ప్రదర్శించును.

```
import csv
D = [['Exam'],['Quarterly'],['Halfyearly']]
csv.register_dialect('M',lineterminator = '\n')
with open('c:\pyprg\ch13\line2.csv', 'w') as f:
```


12th_CS_Telugu_Medium_Chapter_13.indd 263　　28-03-2023 10:06:49

```
    wr = csv.writer(f,dialect='M')
    wr.writerows(D)
f.close()
```

అ) Exam Quarterly Halfyearly

ఆ) Exam Quarterly Halfyearly

ఇ) E
Q
H

ఈ) Exam,
Quarterly,
Halfyearly

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. CSV ఫైల్ అంటే ఏమిటి?
2. ఫైథాన్ ఉపయోగించి CSV ఫైల్ చదవడానికి రెండు మార్గాలను పేర్కొనండి.
3. ఫైల్ యొక్క డిఫాల్ట్ మోడ్‌లను పేర్కొనండి
4. next() ఫంక్షన్ యొక్క ఉపయోగం ఏమిటి?
5. మీరు csv ఫైల్ నుండి ఒకటి కంటే ఎక్కువ కాలమ్ (column) ఎలా క్రమబద్ధీకరిస్తారు? ఉదాహరణనిమ్ము.

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. ఫైథాన్ యొక్క open() ఫంక్షన్ పై గమనిక రాయండి. రెండు పద్ధతుల మధ్య తేడా ఏమిటి?
2. ఇప్పటికే ఉన్న ఫైల్‌ను సవరించడానికి ఫైథాన్ ప్రోగ్రామ్‌ను వ్రాయండి.
3. డిపాల్ట్ డీలిమిటర్ కామా (,) తో CSV ఫైల్‌ను చదవడానికి ఫైథాన్ ప్రోగ్రామ్‌ను వ్రాయండి.
4. write మోడ్ మరియు append మోడ్ మధ్య వ్యత్యాసము ఏమిటి?
5. reader() ఫంక్షన్ మరియు DictReader() తరగతికి మధ్యగల భేదములను వ్రాయుము?

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. ఫైల్ మరియు CSV ఫైల్‌ను వేరు చేయండి.
2. వివిధ మోడ్‌లను దాని అర్థంతో పట్టిక చేయండి
3. ఫైథాన్‌లో ఫైల్‌ను చదవడానికి వేర్వేరు పద్ధతులను రాయండి.
4. కస్టమ్‌కోట్స్ CSV ఫైల్ రాయడానికి ఫైథాన్ ప్రోగ్రామ్ రాయండి.
5. CSV ఫైల్‌లో డేటాను ఫార్మాట్ చేయడానికి అనుసరించాల్సిన నియమాలను వ్రాయండి.

REFERENCES

1. ***Python for Data Analysis, Data Wrangling with Pandas, NumPy, and IPython By William McKinney***
2. ***CSV File Reading and Writing - Python 3.7.0 documentation***
3. ***https://docs.python.org***

12th_CS_Telugu_Medium_Chapter_13.indd 264 28-03-2023 10:06:49

యూనిట్ V

అధ్యాయం 14

పైథాన్‌లో C++ ప్రోగ్రామ్‌ను ఉపయోగించుట

అభ్యసనా లక్ష్యములు

పైథాన్ C++ ప్రోగ్రామ్ ప్రాముఖ్యత అనే అధ్యాయం పూర్తి అయిన తర్వాత విద్యార్థులు ఈ క్రింది విషయాలను అర్థం చేసుకొనగలరు.

- చుట్టడం అంటే ఏమిటో అర్థం చేసుకొనుట.
- పైథాన్ ప్రోగ్రామ్‌లకు C++ ఫంక్షన్ క్లాస్‌లను దిగుమతి చేసుకోగలుగుతారు.
- రెండు భాషలతో పనిచేయటానికి కావలసిన వాతావరణం సృష్టించటం.
- పైథాన్ ప్రోగ్రామ్ అమలుచేయుట మరియు డిబగ్ చేయుట.

14.1 పరిచయము

పైథాన్ మరియు C++ సాధారణమైన ప్రయోజిత ప్రోగ్రామింగ్ అయినచో C++ నుండి చాలా భిన్నంగా ఉంటాయి.

S.NO	పైథాన్	C++
1	సాధారణంగా అన్వయించబడిన భాష (లాగ్వేంజ్)	C++ సాధారణ కంపైల్డ్ భాష
2	పైథాన్ డైనమిక్ టైప్ భాష	C++ సంఖ్యా పరంగా టైప్ చేయబడిన భాష
3	వేరియబుల్ అని ప్రకటించేటపుడు భాషారకం అవసరం లేదు.	వేరియబుల్ అని ప్రకటించేటపుడు డేటా రకము అవసరము.
4	పైథాన్ స్క్రిప్టింగ్ మరియు జనరల్ పర్పస్ భాషగా పనిచేస్తుంది.	ఇది సాధారణ ప్రయోజన భాష.

పైథాన్ మరియు C++ సంపూర్ణంగా ఉంటాయి. పైథాన్ ఎక్కువగా (scripting) లేదా ("glue") భాషగా ఉపయోగపడుతుంది. అంటే ఉన్నత స్థాయి ప్రోగ్రామ్ ఎక్కువగా C లేదా C++ లో వ్రాయబడి ఉండును. ఇప్పటికే ఉన్న Code పరంగా తర్కాన్ని వ్రాయగలిగినపుడు ఇది ఉపయోగ పడుతుంది. (ఉదాహరణకు ప్రోగ్రామ్ లో C++ వ్రాయబడింది). కాని పైథాన్ ప్రోగ్రామ్ ద్వారా దీన్ని పిలుస్తారు మరియు మార్చవచ్చును.

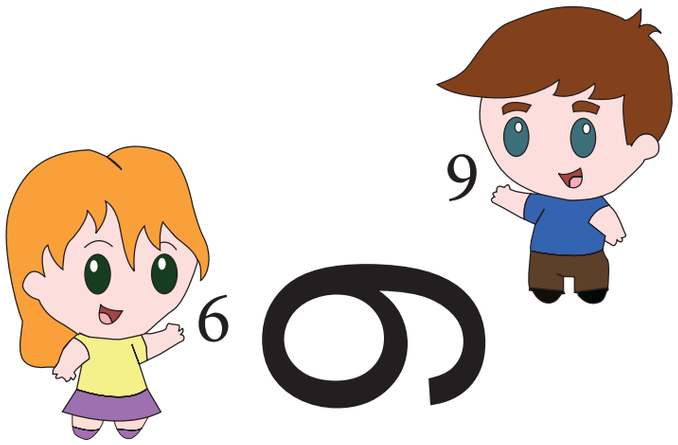
12th_CS_Telugu_Medium_Chapter_14.indd 265 28-03-2023 10:11:12

14.2 స్క్రిప్టింగ్ భాష

స్క్రిప్టింగ్ భాష ఇతర ప్రోగ్రామింగ్ భాషలతో సమగ్రపరచడానికి మరియు కమ్యూనికేట్ చేయడానికి రూపొందించిన ప్రోగ్రామింగ్ భాష. జావా స్క్రిప్ట్, VB స్క్రిప్ట్, PHP పెరల్, పైథాన్, ASP రూబీ మరియు Tcl చాలా విస్తృతంగా ఉపయోగించే స్క్రిప్టింగ్ భాషలు. స్క్రిప్టింగ్ భాష సాధారణంగా మరొక ప్రోగ్రామింగ్ భాషతో కలిసి ఉపయోగపడుతుంది. కాబట్టి, అవి తరచుగా HTML జావా లేదా C++ తో కలిసి కనిపిస్తాయి.

14.2.1 స్క్రిప్టింగ్ మరియు ప్రోగ్రామింగ్ భాషల మధ్య వ్యత్యాసం (Difference between Scripting and Programming Languages):

స్క్రిప్టింగ్ భాష మరియు ప్రోగ్రామింగ్ భాష ఈ క్రింది చిత్రంగా కనిపిస్తుంది.



ప్రాథమికంగా అన్నీ స్క్రిప్టింగ్ లాంగ్వేజస్‌ను ప్రోత్సహిస్తున్నాయి. రెండింటి మధ్య సిద్ధాంతపరమైన వ్యత్యాసము స్క్రిప్టింగ్ లాగ్వేంజ్‌కు సంకలన దశ అవసరము లేదు మరియు బదులుగా అన్వయం చేసుకొనవచ్చును. ఉదాహరణకు సాధారణంగా C++ ప్రోగ్రామ్‌ను రన్ చేయడానికి ముందు కంపైల్ చేయాలి. అయితే జావా స్క్రిప్ట్ లేదా పైథాన్ వంటి స్క్రిప్టింగ్ భాష కంపైల్ చేయబడదు. స్క్రిప్టింగ్ భాషకు వ్యాఖ్యాత అవసరం అయితే ప్రోగ్రామింగ్ భాషకు కంపైల్ సంగ్రహించుట అవసరం. ఇచ్చిన భాషను వారు ఉపయోగించే విధానాన్ని బట్టి స్క్రిప్టింగ్ లేదా ప్రోగ్రామింగ్ లాంగ్వేజ్ అని పిలుస్తారు.

14.3 స్క్రిప్టింగ్ లాంగ్వేజస్ యొక్క అనువర్తనాలు

1. ఒక ప్రోగ్రామ్‌లో కొన్ని పనులను తనకు తానే నియంత్రించడానికి.
2. డేటా సెట్ నుండి సమాచారాన్ని సంగ్రహించుట.
3. సాంప్రదాయ ప్రోగ్రామింగ్ భాషతో పోలిస్తే ఇంటెన్సివ్ కోడ్ తక్కువ.
4. అనువర్తనాలకు మరియు గ్రూప్ కాంప్లెక్స్ సిస్టమ్‌లకు కొత్త ఫంక్షన్‌లను తీసుకురాగలదు.

పైథాన్ వాస్తవానికి ఏదైనా ఆధునిక కంప్యూటర్ ఆపరేటింగ్ సిస్టమ్‌లో ఉపయోగించబడే ఒక వివరణాత్మక ఉన్నత స్థాయి సాధారణ ప్రయోజన ప్రోగ్రామింగ్ భాష. ఇది ప్రోసెసింగ్ సంఖ్యలు చిత్రాలు, సైంటిఫిక్ డేటా మరియు మీరు కంప్యూటర్‌లో ఏదైనా సేవ్ చేయగలుగడాని ఉపయోగించవచ్చును. ఇప్పటి రోజుల్లో పెద్ద అనువర్తనాలు పైథాన్‌లో ప్రత్యేకంగా వ్రాయబడ్డాయి.

12th_CS_Telugu_Medium_Chapter_14.indd 266 28-03-2023 10:11:13

14.4 C++ మీద పైథాన్ యొక్క లక్షణాలు

- పైథాన్ ఆటోమేటిక్ గార్బేజ్ కలెక్షన్‌ను ఉపయోగిస్తుంది. అయితే C++ ఉపయోగించదు.
- C++ అనునది స్థిరంగా టైప్ చేసిన భాష, పైథాన్ డైనమిక్ టైప్ చేసిన భాష.
- పైథాన్ ఒక వ్యాఖ్యత ద్వారా నడుస్తుంది. అయితే C++ ముందే కంపైల్ చేయబడింది.
- పైథాన్ కోడ్ C++ లో వ్రాసిన దాని కంటే 5 నుండి 10 రెట్లు తక్కువగా ఉంటుంది.
- పైథాన్‌లోని రకాలను స్పష్టంగా డిక్లర్ చేయవలసిన అవసరం లేదు. కాని C++ లోని రకాలను స్పష్టంగా డిక్లర్ చేయవలెను.
- ఒక ఫంక్షన్ ఏ రకమైన వాదననైనా అంగీకరించవచ్చును మరియు ముందే ఎలాంటి డిక్లరేషన్ లేకుండా బహుళంగా తిరిగి ఇస్తుంది. C++ రిటర్న్ స్టేట్ మెంట్‌లో తిరిగి ఒకే ఒక విలువ మాత్రమే తిరిగి వస్తుంది.

గమనిక

పైథాన్ అవాంఛిత వస్తువులను తొలగిస్తుంది. (అంతర్నిర్మిత రకాలు లేదా తరగతి ఉదంతాలు) ఆటోమేటిక్‌గా మెమొరీని ఫ్రీ చేస్తుంది. దీని ద్వారా పైథాన్ క్రమానుగతంగా పైథాన్‌లో చేయడం ద్వారా మెమరీలో ఫ్రీ ప్లేస్ తిరిగి ఏర్పడుతుంది. ఈ విధంగా ఉపయోగంలో లేని మెమరీ బ్లాక్‌లను గార్బేజ్ కలెక్షన్ అంటారు.

14.5 పైథాన్‌లో C++ ఫైళ్ళను దిగుమతి చేయడం

పైథాన్ C++ ఫైళ్ళను దిగుమతి చేయడం పైథాన్ ప్రోగ్రామ్‌లోని C++ ప్రోగ్రామ్‌ను దిగుమతి చేసుకోవడం చుట్టడం లేదా C++ ప్రోగ్రామ్‌ల కోసం పైథాన్ ఇంటర్ ఫేస్‌లను సృష్టించడం అని పిలుస్తారు. సాధారణంగా ఉపయోగించే అనేక ఇంటర్ ఫేస్‌లలో



- పైథాన్ C-API (API-**Application Programming Interface** యొక్క ఇంటర్ ఫేసింగ్ C ప్రోగ్రామ్)
- C రకాలు (C ప్రోగ్రామ్‌లతో ఇంటర్ పేసింగ్)
- SWIG (Simplified Wrapper Interface Generator- Both C మరియు C++ రెండూ)
- Cython సైథాన్ అనునది పైథాన్ లాంటి భాష వ్రాయడానికి మరియు C పొడిగింపు లాంటి రెండినీ చేయడానికి
- బూస్ట్ పైథాన్ (C++ మరియు పైథాన్ ఇంటర్ ఫేసింగ్ ఫ్రేమ్‌వర్క్)
- MinGW (*Minimalist GNU for Windows*)

14.5.1 MinGW ఇంటర్‌ఫేస్

MinGW అనేది రన్‌టైమ్ హెడర్ ఫైళ్ళ సమితిని సూచిస్తుంది. ఇది, కోడ్‌ను కంపైల్ చేయడానికి మరియు లింక్ చేయడానికి ఉపయోగిస్తారు. విండోస్ ఆపరేటింగ్ సిస్టమ్‌లో C, C++ మరియు FORTRAN ప్రోగ్రామ్‌లో అమలు చేయబడతాయి.

12th_CS_Telugu_Medium_Chapter_14.indd 267 28-03-2023 10:11:13

Windows లో C++ కోసం MinGw-W64 (MinGw యొక్క వెర్షన్) ఉత్తమ కంపైలర్. కంపైల్ చేయడానికి మరియు ఎక్సిక్యూట్ చేయడానికి C++ ప్రోగ్రామ్ అమలు చేయడము. మీరు Windows-OS కోసం 'g++' చేసారు. MinGw కంపైల్ చేయడానికి మరియు ఎక్సిక్యూట్ చేయడానికి 'g++' ఉపయోగించి పైథాన్ ప్రోగ్రామ్ ద్వారా C++ ప్రోగ్రామ్‌ను డైనమిక్‌గా అమలుచేయబడును.

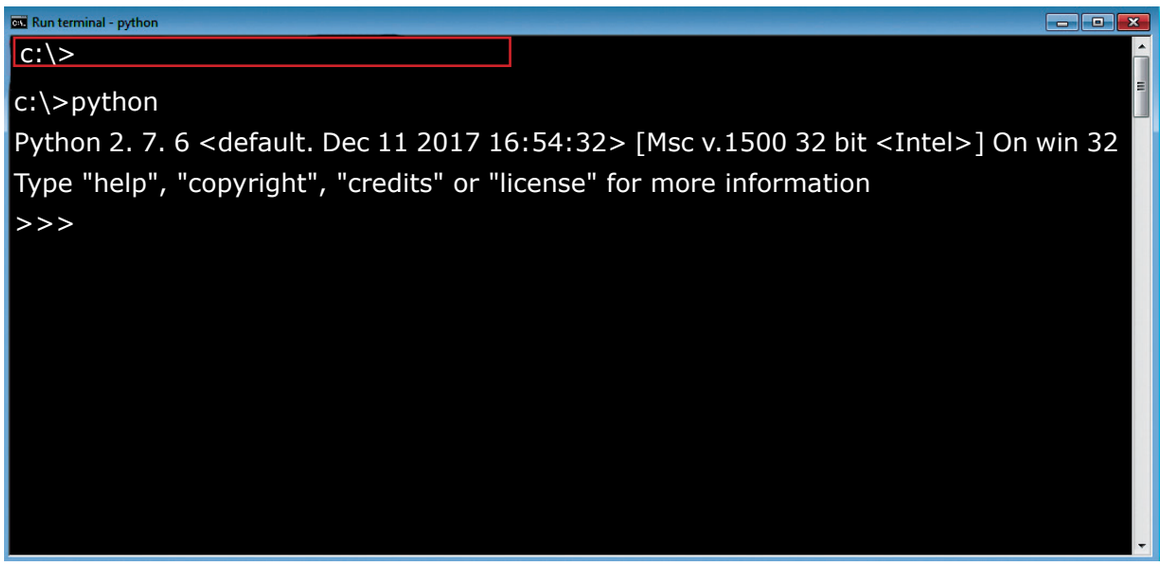
C++ కోడింగ్ కలిగివున్న పైథాన్ ప్రోగ్రామ్‌ను కమాండ్ ప్రామ్ట్ ద్వారా లేదా రన్‌టర్మినల్ ఉపయోగించి అమలు చేయవచ్చును.

g++ అనేది GCC (GNU C Compiler) అని పిలిచే ఒక ప్రోగ్రామ్. ఆటోమేటిక్‌గా లింక్ చేస్తుంది. ఆబ్జెక్ట్ కోడ్‌కు C++ లైబ్రరీ ఫైళ్ళు అవసరం

Refer installation of MinGW in Annexure -2

14.5.2 పైథాన్ ద్వారా C++ ప్రోగ్రామ్ ఎక్సిక్యూట్ చేయుట

1. కమాండ్ ప్రామ్ట్ లేదా రన్ టర్మినల్‌పై డబుల్ క్లిక్ చేయండి.



పటము *14. 1*

2. ఈ ఉదాహరణలో ప్రాంప్ట్ పైన పేర్కొన "C:\>" హైలెట్ చేసిన ప్రాంతాలను విండోస్‌తో చూడండి. డైరెక్టరీ మార్చడానికి 'cd' కమాండ్ ఉపయోగించండి. ఉదాహరణకు pyprg కమాండ్‌ను టైప్ చేసి ఉదాహరణకు 'cd pyprg' డైరక్టరిలో (pyprg) కమాండ్ టైప్ చేయండి 'cd pyprg' కమాండ్ ప్రామ్‌ంట్‌లో.

ఉదాహరణకు pycpp.py పైథాన్ ప్రోగ్రామ్‌లో C++ ప్రోగ్రామ్‌లో చదవచ్చును. "Pali.cpp". అని ప్రోగ్రామ్ నంబర్‌లో "Palindrome or Not". ఉదాహరణకు ఇన్‌పుట్‌లో "input". 232 అని టైప్ చేస్తే అవుట్ పుట్ (output) లో "Palindrome" అని display అవుతుంది. C++ ప్రోగ్రామ్ పాలీ నోట్‌ప్యాడ్‌లో టైప్ చేయబడి **pali.cpp.** అని (save) చేయబడుతుంది. అదే విధంగా పైథాన్ ప్రోగ్రామ్ **pycpp.py** code కూడా నోట్‌పాడ్‌లో (**pycpp.py**) గా save చేయబడుతుంది.

3. ప్రోగ్రామ్‌ను రన్ చేయటానికి పైథాన్ ప్రోగ్రాంమ్‌ను (folder) రెండు సార్లు (double) click చేయండి. పైథాన్ program ను అమలుచేయడానికి క్రింది syntax తెలుపును.

Python <filename.py> -i <C++ filename without cpp extension>

12th_CS_Telugu_Medium_Chapter_14.indd 268 28-03-2023 10:11:13

దీనిలో,

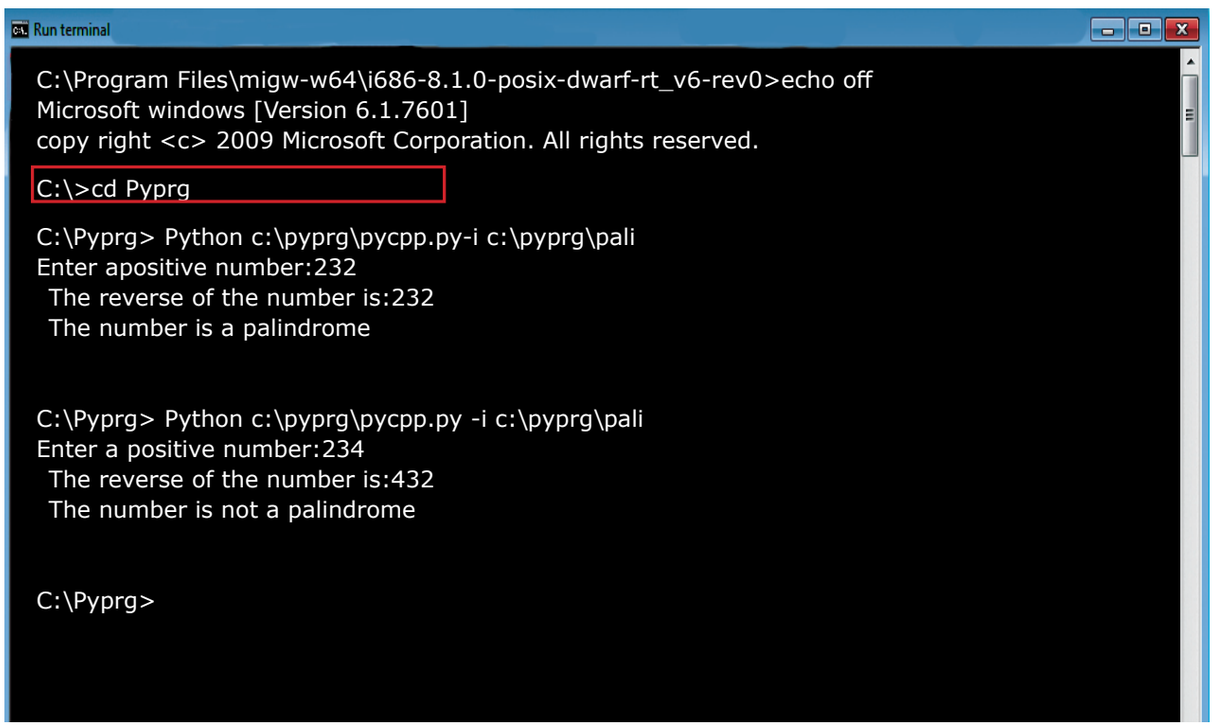
Python	పైథాన్ ప్రోగ్రామ్‌ను కమాండ్ నుండి రన్ చేయడానికి కమాండ్ లైన్.
filename.py	రన్ చేయవలసిన పైథాన్ ప్రోగ్రామ్ పేరు
- i	(input) ఇన్‌పుట్ మోడ్
C++ filename without cpp extension	కంపైల్ చేసి ఎక్జిక్యూట్ చేయవలసిన C++ఫైల్ పేరు

ఉదాహరణకు కమాండ్ ప్రాంప్ట్‌లో పైథాన్ pycpp.py –i pali ను టైప్ చేసి enter key ని ప్రెస్ చేయండి ఇప్పుడు విజయవంతమైతే మీకు కావలసిన అవుట్ పుట్ లభిస్తుంది. లేకపోతే లోపం (error) ప్రదర్శించబడుతుంది.

గమనిక

రన్ ఆదేశంలో ఇన్పుట్ ఫైల్‌కు దాని పొడిగింపు అవసరం లేదు. ఉదాహరణకు “pali.cpp”. కు బదులుగా “pali” పేరును పేర్కొనడం సరిపోతుంది.

ఇప్పుడు మనం pycpp.py మరియు pali.cpp. ఉదాహరణల ద్వారా ప్రోగ్రామ్‌లను రన్ చేయడం ఎలా అని చూస్తాం. ఈ రెండు ప్రోగ్రామ్‌లు c:\pyprg. అనే ఫోల్డర్‌లో ఉండవలెను. ఈ ప్రోగ్రామ్‌లు పైథాన్ ఫోల్డర్‌లలో లేనిచో రన్ చేసినపుడు ఫైల్ యొక్క (complete) part ని ఇవ్వవలెను. క్రింది విధంగా అవుట్‌పుట్ డిస్‌ప్లే అగును.



పటము *14.2*

12th_CS_Telugu_Medium_Chapter_14.indd 269 28-03-2023 10:11:13

గమనిక

కమాండ్ విండోను క్లియర్ చేయడానికి **cls** కమాండ్‌ను వాడండి.

ఇప్పుడు C++ కోడ్‌ను కంపైల్ చేయడానికి పైథాన్ ప్రోగ్రామ్‌ను ఎలా రాయాలో చూద్దాం.

14.6 C++ ను ఇంపోర్ట్ చేయడానికి పైథాన్ ప్రోగ్రామ్

పైథాన్ చాలా మాడ్యూళ్ళను కలిగి ఉంది. ఒక సమస్య కోసం పైథాన్ ప్రోగ్రామ్‌లను కలిగి ఉండడానికి అనుమతిస్తుంది. వారి సౌలభ్యం ప్రకారం వేర్వేరు మాడ్యూల్ ఉపయోగించడంలో ఆవశ్యత. పైథాన్ ప్రోగ్రామ్ మేము వ్రాసిన అన్ని కొత్త ఆదేశాలను కలిగి ఉంది. ఇవి ప్రాథమిక పైథాన్‌లో మనకు రాలేని ప్రోగ్రామ్. మా ప్రోగ్రామ్ రెండు వేర్వేరు భాషల అనుసంధానం కాబట్టి మేము దిగుమతి చూసుకోవలసిన మాడ్యుల్స్ os, sys మరియు getopt.

14.6.1 మాడ్యూల్ (MODULE)

మాడ్యూలర్ ప్రోగ్రామింగ్ అనేది మీకోడ్‌ను చిన్న చిన్న భాగాలుగా విభజించడానికి ఉపయోగపడే సాఫ్ట్‌వేర్ డిజైన్. ఈ భాగాలను మాడ్యూల్స్ అంటారు. ఈ విభజన కోసం ఫోకస్ ఇతర మాడ్యుల్స్‌పై ఎటువంటి కొన్ని డిపెండెన్సీలు లేసి మాడ్యుల్‌లను కలిగి ఉండాలి మరో విధంగా చెప్పాలంటే డిపెండెన్సల కుదించిన లక్ష్యం అగును.

పైథాన్‌లో మాడ్యూళ్ళను ఎలా సృష్టించగలం? గుణకాలు పైథాన్ ఉన్న ఫైల్‌ను సూచన ప్రకటనలు మరియు నిర్వచనలు పైథాన్ కోడ్డ్ కలిగివున్న ఫైల్‌కు e.g factorial.py, అనునది మాడ్యూల్ మరియు దాని యొక్క మాడ్యుల్ నేమ్ అనునది ఫాక్టోరియల్ పెద్ద ప్రోగ్రామ్‌లను చిన్నవిగా నిర్వహించదగినవిగా మరియు వ్యవస్థీకృత ఫైల్‌లుగా విచ్ఛిన్నం చేయడానికి మేము మాడ్యూళ్ళను ఉపయోగిస్తాము. ఇంకా గుణకాలు కోడ్ యొక్క పునర్వినియోగతను అందిస్తాయి.

ఉదాహరణ:

```
def fact(n):
        f=1
        if n == 0:
        return 0
elif n == 1:
        return 1
        else:
                for i in range(1, n+1):
                        f= f*i
                print (f)
```

Output:

```
        >>>fact (5)
        120
```

పై ఉదాహరణను ఫాక్టోరియర్ (factorial. py) అంటారు.

12th_CS_Telugu_Medium_Chapter_14.indd 270 28-03-2023 10:11:13

14.6.2 పైథాన్‌లో మాడ్యూళ్ళను ఎలా దిగుమతి చేయాలి?

మనము మాడ్యూల్ లోపల నిర్వచనాలను మరొక మాడ్యుల్‌కు దిగుమతి చేసుకోవచ్చును. మనము **import** అనే పదాన్ని ఉపయోగించి దీనిని చేయవచ్చును. ముందు నిర్వచించిన మాడ్యూల్ **factorial** అనే పైథాన్ ప్రాంప్ట్‌లో టైప్ చేస్తాము.

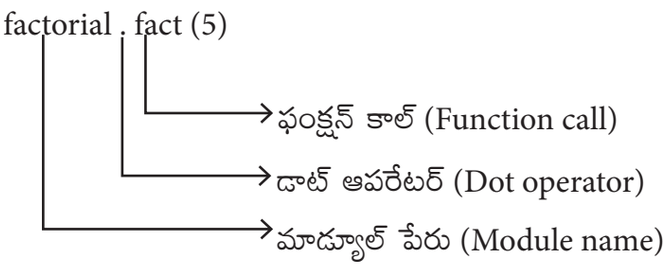
```
>>> import factorial
```

మాడ్యూల్ పేరును ఉపయోగించి మాడ్యూల్ లోపల నిర్వచించిన ఫంక్షన్లను యాక్సెస్ చేయవచ్చును. **dot (.)** అనునది ఫంక్షన్లను యాక్సెస్ చేయడానికి ఉపయోగించబడుతుంది. మాడ్యూల్ నుండి (syntax functions) యాక్సెస్ చేస్తుంది.

<module name> . <function name>

For example:

```
>>> factorial.fact(5)
120
```



పైథాన్ ప్రామాణిక (అంతర్నిర్మిత) మాడ్యూళ్ళ సంఖ్యను కలిగి ఉంది. నిర్వహించిన మాడ్యూళ్ళను దిగుమతి చేసే విధంగానే ప్రామాణిక మాడ్యుల్‌లను దిగుమతి చేసుకొనవచ్చును. C++ కోడ్‌ని అమలు చేయడానికి ప్రోగ్రామ్‌కు అవసరమైన స్టాండర్డ్ మాడ్యుల్‌లను చూడగలము.

14.6.2.1 పైథాన్ యొక్క (sys) మాడ్యూల్ (Python's sys module)

ఈ మాడ్యూల్ వ్యాఖ్యాత (interpreter) ఉపయోగించే అంతర్నిర్మిత వేరియబుల్స్‌కు యాక్సెస్‌ను అందిస్తుంది. **argv** అనునది **sys** వేరియబుల్‌లో ఒకటి.

sys.argv

sys.argv అనేది పైథాన్ ప్రోగ్రామ్‌కు పంపిన command-line ఆర్గ్యుమెంట్‌ల జాబితా, command-line ఇన్పుట్ ద్వారా వచ్చే అన్ని అంశాలను కలిగి ఉంది, ఇది ప్రాథమికంగా ప్రోగ్రామ్ యొక్క command-line ఆర్గ్యుమెంట్లను కలిగి ఉన్న జాబితా.

sys.argv అనునది పైథాన్ ప్రోగ్రామ్‌కు పంపే కమాండ్‌లైన్ వాదనలు కలిగి ఉన్న జాబితా. sys. argv ను ఉపయోగించడానికి **import sys** ను దిగుమతి చేసుకొనవలెను. మొదటి వాదన, పైథాన్ ప్రోగ్రామ్ (ఉదాహరణ pali. py) మరియు sys.argv[1] యొక్క పేరు కలిగి ఉంది. తదుపరి వాదన ప్రోగ్రామ్ (ఇక్కడ ఇది C++ ఫైల్). ఇది ప్రధాన () ద్వారా పంపబడిన వాదన అవుతుంది.

12th_CS_Telugu_Medium_Chapter_14.indd 271 28-03-2023 10:11:13

main(sys.argv[1])	ఇన్పుట్ఫైల్ (C++ఫైల్) దాని మార్గంలో పాటు **argv[1]** ను ఉపయోగించి జాబితా (array) గా పంపబడుతుంది. **argv[0]** పైథాన్ ప్రోగ్రామ్ను కలిగిఉంది. ఎందుకంటే ఇది పాస్ చేయవలసిన అవసరం లేదు. ఎందుకంటే డిఫాల్ట్గా మెయిన్ సోర్స్ కోడ్ రిఫరెన్స్ను కలిగి ఉంటుంది.

14.6.2.2 పైథాన్ యొక్క OS మాడ్యూల్

పైథాన్లోని OS మాడ్యూల్ ఆపరేటింగ్ సిస్టమ్ డిపెండెంట్ ఫంక్షనాలిటీని ఉపయోగించే మార్గాన్ని అందిస్తుంది.

OS మాడ్యూల్ పైథాన్ నడుస్తున్న విండోస్ ఆపరేటింగ్ సిస్టమ్లో ఇంటర్ ఫేస్ చేయడానికి మనల్ని అనుమతిస్తుంది.

os.system(): ఫైల్లో C++ కంపైలింగ్ కమాండ్ను (స్ట్రింగ్లో యునిక్స్, C కమాండ్ని కలిగి ఉంటుంది. ఇది C++ కమాండ్కు మద్దతు ఇస్తుంది) రన్ చేయండి (ఇక్కడ ఇది కమాండ్ విండో) C++ ప్రోగ్రామ్ను కంపైల్ చేయడానికి ఉదాహరణ కోసం **g++** ను ప్రారంభించాలి. అలా చేయడానికి క్రింది ఆదేశం ఉపయోగపడుతుంది.

os.system ('g++ ' + <variable_name1> +' -<mode> ' + <variable_name2>)

ఇక్కడ

os.system :-	ఆపరేటింగ్ సిస్టమ్లో ఇంటరాక్ట్ అవ్వడానికి OS మాడ్యూల్లో షంక్షన్ సిస్టమ్ () నిర్వచించబడింది.
g++ :-	విండోస్ ఆపరేటింగ్ సిస్టమ్ క్రింద C++ ప్రోగ్రామ్ను కంపైల్ చేయడానికి జనరల్ కంపైలర్.
variable_name1:-	పొడగింపు లేకుండా C++ ఫైల్ యొక్క పేరు స్ట్రింగ్ ఆకృతిలో ముందు చెప్పిన విధంగా సూచించబడి ఉంటుంది.
mode :-	ఇన్పుట్ లేదా అవుట్పుట్ మోడ్ను పేర్కొనడానికి. ఇక్కడ ఇది hyphen 'O' ప్రిఫిక్స్ (prefix) చేయబడి ఉంటుంది.
variable_name2 :-	స్రింగ్ పార్మాట్లో పొడిగింపుతో ఎక్సిక్యూటబుల్ ఫైల్ యొక్క పేరు సూచించబడి ఉంటుంది.

ఉదాహరణకు C++ ప్రోగ్రామ్ను కంపైల్ చేసి అమలు చేసే ఆదేశం క్రింద ఇవ్వబడింది.

os.system('g++ ' + cpp_file + ' -o ' + exe_file)	g++ కంపైలర్ cpp_ ఫైల్ను కంపైల్ చేస్తుంది. మరియు –o (అవుట్పుట్) exe_file కు పంపుతుంది.

12th_CS_Telugu_Medium_Chapter_14.indd 272 28-03-2023 10:11:13

గమనిక

os.system() లోని '+' అన్ని తీగలును ఒకే స్ట్రింగ్ వలె సంయోగం చేసినట్లు సూచిస్తుంది. కాబట్టి పై వాదనకు ప్రతి పదం తర్వాత ఖాళీని ఇవ్వండి. ఉదా: **'g++ ' + cpp_file + ' -o ' + exe_file**

14.6.2.3.3 పైథాన్ getopt మాడ్యూల్ (Python getopt module)

పైథాన్ యొక్క (getopt) మాడ్యూల్ కమాండ్‌లైన్ ఎంపికలు మరియు వాదనలను అన్వయించడానికి (split) మీకు సహాయపడుతుంది. ఈ మాడ్యూల్ కమాండ్‌లైన్ ఆర్గుమెంట్ పార్సింగ్‌ను ప్రారంభించడానికి getopt () పద్ధతిని అందిస్తుంది.

getopt.getopt ఫంక్షన్

ఈ పద్ధతి కమాండ్‌లైన్ ఎంపికలు మరియు పారామితి జాబితాను అన్వయించింది. ఈ పద్ధతికి వాక్యనిర్మాణం :

<opts>,<args>=getopt.getopt(argv, options, [long_options])

పారామితుల వివరాలు ఇక్కడ ఉన్నాయి.

argv – అన్వయించవలసిన విలువల యొక్క వాదన జాబితా ఇది (విభజించబడింది). మన ప్రోగ్రామ్‌లో పూర్తి ఆదేశం జాబితాగా ఇవ్వబడుతుంది. ఉదాహరణకు ***c:\pyprg\pali.py -i c:\pyprg\pali_cpp***

options – ఇది పైథాన్ ప్రోగ్రామ్ అవుట్‌పుట్ కోసం లేదా ఇన్‌పుట్ కోసం ఎంపిక చేసిన అక్షర వరుస క్రమము. ఇక్కడ ('i' లేదా 'o') మాదిరి ఎంపికలు దానిని అనుసరించు కోలన్ (:) ఉంది. ఇక్కడ మోడ్‌ను చూచించుటకు కోలన్‌ను ఉపయోగిస్తారు.

long_options –వరుస క్రమము జాబితాను కలిగి ఉంటుంది. సమానత్వపు గుర్తు ద్వారా లాంగ్ ఆప్షన్ యొక్క వాదనను అనుసరించాలి. మన ప్రోగ్రామ్‌లో C++ ఫైల్ పేరు దాని మార్గంతో పాటు స్ట్రింగ్‌గా పంపబడుతుంది మరియు ఇన్‌పుట్ ఫైల్‌గా సూచించడానికి 'i' i కూడా పంపబడుతుంది.

getopt() method returns value consisting of two elements. ఈ విలువలలో ప్రతి ఒక్కటి రెండు వేర్వేరు జాబితాలలో **opts** మరియు **args** నిల్వచేయబడతాయి. మోడ్ మరియు మార్గం వంటి విభజించిన వరుస క్రమము జాబితాను **opts** కలిగి ఉంటాయి. అన్ని వ్యాఖ్యలు తప్పు మార్గం లేదా మోడ్‌తో ఇవ్వబడితే **args** వరుసక్రమ లోపాలును కలిగి ఉంటాయి. లోపం లేకపోతే **args** ఖాళీ జాబితా అవుతుంది.

ఉదాహరణకు **C++ file p4** ఫైల్ ను కమాండ్‌లైన్‌లో అమలు చేయబోయే పైథాన్ కోడ్ క్రింది మాదిరిగానే **getopt** () పద్ధతిని కలిగి ఉంటుంది. **opts, args = getopt.getopt (argv, "i:",['ifile='])**

where **opts** contains	[('-i', 'c:\\pyprg\\p4')]
-i :-	ఎంపిక **(colon) :** ని మోడ్ తప్పకుండా అనుసరిస్తుంది.
'c:\\pyprg\\p4'	విలువ **C++** ఫైల్ యొక్క సంపూర్ణ మార్గము.

12th_CS_Telugu_Medium_Chapter_14.indd 273 28-03-2023 10:11:13

మన ఉదాహరణలలో మొత్తం కమాండ్‌లైన్ ఆదేశాలు అన్వయించబడినందువలన మరియు రెండవ ఆర్గ్యుమెంట్ [] ఖాళీగా ఉంటాయి. ప్రింట్ [ ] ఉపయోగించి (**args**) ప్రదర్శన పెడితే కమాండ్ అవుట్‌పుట్‌గా ప్రదర్శిస్తుంది.

```
>>>print(args)
[]
```

గమనిక

పైథాన్ ప్రామాణిక గుణకాల యొక్క పూర్తి జాబితాను మరియు అవి ఎందుకోసం అని మీరు తెలుసుకొనవచ్చును. ఈ ఫైళ్ళను పైథాన్ వ్యవస్థాపించబడిన ప్రదేశంలో లిబ్ డైరెక్టరీలో ఉన్నాయి.

C++ కోడ్‌ను చుట్టడానికి (wrap) మరొక ఆదేశం.

```
if __name__=='__main__':
    main(sys.argv[1:])
```

పైథాన్‌లో __name__ (ప్రత్యేక వేరియబుల్)

పైథాన్ main() ఫంక్షన్ లేనందున, పైథాన్ ప్రోగ్రామ్‌ని అమలు చేయమని ఇంటర్‌ప్రిటర్‌కు కమాండ్ ఇచ్చినప్పుడు, స్థాయి 0 ఇండెంటేషన్‌లో ఉన్న కోడ్ (ప్రోగ్రామ్ యొక్క ఫైలైన్) అమలు చేయబడుతుంది. అయితే, అలా చేయడానికి ముందు, ఇంటర్‌ప్రెటర్ కొన్ని ప్రత్యేక వేరియబుల్స్ నిర్వచిస్తారు. __name__ అనేది అటువంటి ప్రత్యేక వేరియబుల్, ఇది డీఫాల్ట్‌గా పైథాన్ ఫైల్ పేరున నిల్వచేస్తుంది. ప్రధాన ప్రోగ్రామ్‌గా Source ఫైల్ అమలు చేయబడితే, ఇంటర్‌ప్రెటర్ __name__ వేరియబుల్‌ను __main__ విలువను కలిగి ఉండేలా సెట్ చేస్తాడు.

__name__ అనేది ప్రస్తుత మాడ్యూల్ పేరును అంచనా వేసే అంతర్నిర్మిత వేరియబుల్. అందువల్ల ప్రస్తుత స్క్రిప్ట్ దాని స్వంతంగా నడుస్తుందో లేదో తనిఖీ చేయడానికి దీనిని ఉపయోగించవచ్చును.
ఉదాహరణకు ఈ క్రింది వాటిని పరిశీలించండి.

```
if __name__ == '__main__':
    main (sys.argv[1:])
```

ఆదేశ వరుస కమాండ్ లైన్‌లో పైథాన్ ప్రోగ్రామ్ మొదట అమలు చేయబోతున్నట్లయితే అప్పుడు __main__ స్ట్రింగ్‌ను కలిగి ఉంటుంది. షరతు __name__ నిజమైతే ప్రధాన ఫంక్షన్ అంటారు.

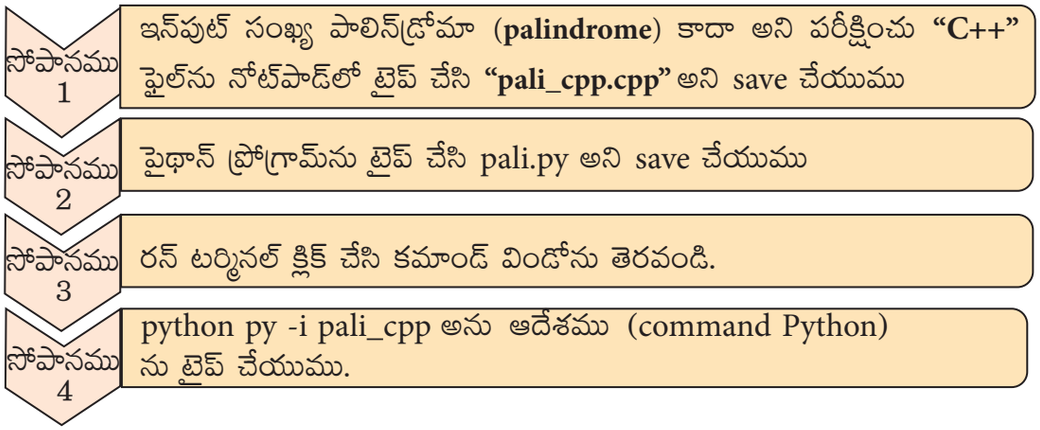
గమనిక

sys.argv[1:] - స్క్రిప్ట్ పేరు (ఫైల్ పేరు) తర్వాత ప్రతిదీ పొందండి.
sys.argv[0] అనేది స్క్రిప్ట్ పేరు (పైథాన్ ప్రోగ్రామ్) మీరు 8 వ అధ్యాయంలో అధ్యయనం చేసిన “స్ట్రింగ్ స్లైసింగ్” “string slicing” గుర్తుంచుకోండి.

12th_CS_Telugu_Medium_Chapter_14.indd 274 28-03-2023 10:11:13

14.7 కంట్రోల్ స్టేట్‌మెంట్‌ను ఉపయోగించి C++ ప్రోగ్రామ్‌ను అమలు చేస్తున్న పైథాన్ ప్రోగ్రామ్.

ఇప్పుడు C++ కోడింగ్ చదివి దాని ఫలితాన్ని అమలు చేయడానికి పైథాన్ ప్రోగ్రామ్‌ను రాద్దాము. ఇవ్వబడిన సంఖ్య పాలిండ్రోమ్ మా కాదా అని పరీక్షించు C++ ఫైల్ అమలు చేసే సోపానాలు క్రింద ఇవ్వబడినవి.



ఉదాహరణ 14.7.1 ఏదైనా ఒక సంఖ్యను ఇన్‌పుట్ చేసి (while loop) ను ఉపయోగించి ఆ సంఖ్య పాలిన్‌డ్రోమా కాదా అని తెలిసికొనుటకు ఒక C++ఫైల్‌ను వ్రాయండి.

```
/*. To check whether the number is palindrome or not using while loop.*/
//Now select File->New in Notepad and type the C++ program
#include <iostream>
using namespace std;
int main()
{
int n, num, digit, rev = 0;
cout<< "Enter a positive number: ";
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;
}
// Save this file as pali_cpp.cpp
```


12th_CS_Telugu_Medium_Chapter_14.indd 275 28-03-2023 10:11:13

```
#Now select File→New in Notepad and type the Python program
# Save the File as pali.py . Program that compiles and executes a .cpp file
# Python c:\pyprg\pali.py -i c:\pyprg\pali_cpp
import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)

def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

Output of the above program

Output 1

```
C:\Users\Dell>python c:\pyprg\pali.py -i c:\pyprg\pali_cpp
Enter a positive number: 56765
The reverse of the number is: 56765
The number is a palindrome
```

Output 2

```
C:\Users\Dell>python c:\pyprg\pali.py -i c:\pyprg\pali_cpp
Enter a positive number: 56756
The reverse of the number is: 65765
The number is not a palindrome
```


12th_CS_Telugu_Medium_Chapter_14.indd 276 28-03-2023 10:11:13

పైథాన్ కోడ్	పనిచేయు విధము
import sys, os, getopt	అవసరమైన ఫంక్షన్‌లకు sys , os మరియు getopt అను చేర్చి ఉపయోగించుకొనవలెను.
def main(argv):	main() ఫంక్షని 'argv' ని కలిగి ఉండే ఇన్‌పుట్ మోడ్ మరియు c++ ప్రోగ్రామ్‌గా ఫైల్‌గా నిర్వచించబడి ఉన్నది. అది ఏమిటంటే i.e ['-i', 'c:\ pyprg \pali_cpp']
opts, args = getopt.getopt(argv, "i:")	getopt() ఉన్న కమాండ్‌ను ఆప్షన్ మరియు ఆర్గ్యుమెంట్‌గా విభజించబడుతుంది. [('-i', 'c:\pyprg\pali_cpp')]. ను కలిగి ఉంటుంది. అందువలన తప్పులేదు అని 'args' చూపుతుంది.
for o, a in opts:	'o' మోడ్‌ను కలిగి ఉంటుంది, మరియు path of c++ ప్రోగ్రామ్ a'ను కలిగి ఉంటుంది. print("o = ",o) shows o = -i print("a = ",a) shows a = c:\pyprg\pali_cpp
if o in ("-i"):	Checks o == 'i' if true
run(a)	c++ ప్రోగ్రామ్‌ను పంపి, run() గా పిలువబడుతుంది.
def run():	run() ఫంక్షన్ యొక్క నిర్వచనము ఇక్కడ నుండి ఆరంభమవుతుంది.
inp_file=a+'.cpp'	'inp_file' అనే వేరియబల్ c++ ప్రోగ్రామ్‌ను మరియు .cpp అనే దానితోను కలిసి ఉంటుంది. print (inp-file) shows 'c'\pyprg\ pali- cpp.cpp.
exe_file=a+'.exe'	'exe_file' అనే వేరియబుల్ఎ c++ ప్రోగ్రామ్ పేరు మరియు cpp అను దానితో కలిగి ఉంటుంది. print (inp-file) shows 'c'\pyprg\ pali- cpp.cpp.
os.system('g++ ' + inp_file + ' -o ' + exe_file)	inp_file లో c++ ప్రోగ్రామ్‌ను g++ కంపైలర్‌లో కంపైల్ చేస్తుంది మరియు exe-file ను ఎక్సిక్యూటబుల్ ఫైల్‌లో నిల్వచేస్తుంది.
os.system(exe_file)	exe file ఎక్సిక్యూట్ చేస్తుంది.
if __name__=='__main__':	__name__ పైథాన్ ప్రోగ్రామ్‌ను నిల్వ చేస్తుంది. __ main__ కూడా పైథాన్ ప్రోగ్రామ్ యొక్క పేరును నిల్వచేస్తుంది.
main(sys.argv[1:])	name మరియు main సమానంగా ఉంటే పైథాన్ ప్రోగ్రామ్ పేరు మినహాయించి కమాండ్ లైన్ ఆర్గ్యుమెంట్‌లో మెయిన్ పిలవబడుతుంది మరియు పంపబడుతుంది. argv[1:] అనునది –i c:\pyprg\pali_cpp అని కలిగి ఉంటుంది.

12th_CS_Telugu_Medium_Chapter_14.indd 277 28-03-2023 10:11:13

పైథాన్ స్క్రిప్ట్ (ప్రోగ్రామ్) ప్రధానంగా C++ ఫైల్‌తో పాటు 'i'/'o'. 'getopt()' కమాండ్ లైన్ యొక్క ప్రతి విలువను అన్వచించడం (విభజిస్తుంది) మరియు ఎంపికలను (విలువలు) పాస్ చేస్తుంది. 'opt' జాబితాగ మరియు లోపం లేనందున (no error) ఖాళీ జాబితాను ఉత్పత్తి చేస్తుంది. [] 'for loop' ను ఉపయోగించడం జాబితాలోని డబుల్ పాక్ చేయబడదు. మోడ్‌ను నిల్వ చేస్తుంది మరియు c++ ఫైల్ యొక్క మార్గంతో పాటు పేరును నిల్వచేస్తుంది.

'inp_file' అనే వేరియబుల్ c++ ఫైల్ ను దాని పొడగింపుతో పాటు నిల్వచేస్తుంది మరియు 'exe_file' ఎక్సిక్యూటబుల్ ఫైల్‌ను exe ఎక్స్‌టెన్షన్‌తో నిల్వ చేస్తుంది. 'os.system()' తో పాటు 'g++' inp_file ని కంపైల్ చేస్తుంది. మోడ్ 'o' ఎక్జిక్యూటబుల్ ఫైల్‌ను 'exe_file'. కు పంపుతుంది.

'__name__' పైథాన్ స్క్రిప్ట్ (సున్నా పంక్తి) మొదటి నుండి ప్రారంభించడానికి వేరియబల్ ప్రోగ్రామ్‌ను నిర్దేశిస్తుంది. main() నిర్వచనం పార్సింగ్ మరియు run() అని పిలుస్తుంది. "run() g++" కంపైలర్‌ను ప్రారంభించి exe ఫైల్‌ను సృష్టిస్తుంది. os మాడ్యూల్ యొక్క సిస్టమ్ ()exe ఫైల్‌ను రన్ చేస్తుంది మరియు కావలసిన అవుట్‌పుట్ Output స్క్రీన్‌లో ప్రదర్శించబడుతుంది. ఫైల్ పొడిగింపులు పైథాన్ స్క్రిప్ట్ చేత జోడింపబడతాయి కాబట్టి C ప్రోగ్రామ్‌లను అమలు చేయడం కూడా సాధ్యమే.

14.8 C++ లోని లోపాలను పైథాన్ ఎలా నిర్వహిస్తుంది

పైథాన్ విజయవంతమైన C++ ప్రోగ్రామ్‌ను అమలు చేయడమే కాకుండా సంకలనం సమయంలో C++ స్టేట్‌మెంట్‌లో ఏదైనా ఉంటే లోపాలను కూడా ప్రదర్శించడానికి ఇది సహాయపడుతుంది. ఉదాహరణకు క్రింది C++ ప్రోగ్రామ్‌లో లోపం ఉంది. మీరు పైథాన్ ద్వారా కంపైల్ చేసినప్పుడు ఏమి జరుగుతుందో చూద్దాం.

Example 14.8.1

```
// C++ program to print the message Hello
//Now select File→New in Notepad and type the C++ program
#include<iostream>
using namespace std;
int main()
{
      std::cout<<"hello"
      return 0;
}
// Save this file as hello.cpp

# Now select File→New in Notepad and type the Python program as main.py
# Program that compiles and executes a .cpp file
# Python main.py -i  hello
import sys, os, getopt
def main(argv):
   opts, args = getopt.getopt(argv, "i:")
   for o, a in opts:
      if o in "-i":
         run(a)
```


12th_CS_Telugu_Medium_Chapter_14.indd 278 28-03-2023 10:11:13

```
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

Output of the above program

```
C:\Users\Dell>python c:\pyprg\main.py -i c:\pyprg\hello
c:\pyprg\hello.cpp: In function 'int main()':
c:\pyprg\hello.cpp:6:21: error: expected ';' before 'return'
  std::cout<<"hello"
                    ^
                    ;
  return 0;
  ~~~~~~
'c:\pyprg\hello.exe' is not recognized as an internal or external command,
operable program or batch file.
```

గమనిక

పై ప్రోగ్రామ్‌లో పైథాన్ C++ లో లోపాలను ప్రదర్శించడానికి సహాయ పడుతుంది. దాని వాక్య సంఖ్యతో పాటు లోపం ప్రదర్శించబడుతుంది. వాక్య సంఖ్య పైథాన్ స్క్రిప్ట్ నుండి మొదలవుతుంది.

గుర్తించుకోవలసిన విషయములు:

- C++ కంపైలర్ ఆధారిత భాష అయితే పైథాన్ ఒక వ్యాఖ్యత ఆధారిత భాష.
- C++ స్థిరంగా సంకలనం (compiled) చేయబడుతుంది, అయితే పైథాన్ డైనమిక్‌గా వివరించబడుతుంది.
- C++ వంటి స్టాటిక్ టైప్ చేసిన భాషకు ప్రోగ్రామర్ ప్రతి "డేటా రకం" ఉపయోగించబోతుందో కంప్యూటర్‌కు స్పష్టంగా చెప్పాల్సిన అవసరం ఉంది.
- పైథాన్ వంటి డైనమిక్ టైప్ చేసిన భాష, డేటా కోసం డేటా రకాన్ని స్పష్టంగా ఇవ్వడం అవసరం లేదు. పైథాన్ విలువ యొక్క రకాన్ని బట్టి వేరియబుల్‌ను మార్చగలదు.
- స్క్రిప్టింగ్ భాష ఇతర ప్రోగ్రామింగ్ భాషలతో సమగ్రపరచడానికి మరియు కమ్యూనికేట్ చేయడానికి రూపొందించిన ప్రోగ్రామింగ్ భాష.
- MinGW రన్ టైమ్ హెడర్ ఫైళ్ళ సమితిని సూచిస్తుంది. ఇది విండోస్ ఆపరేటింగ్ సిస్టమ్‌లో అమలు చేయాల్సిన C, C++ మరియు ఫోర్ట్రాన్ కోడ్‌ను కంపైల్ చేయడానికి మరియు లింక్ చేయడానికి ఉపయోగపడుతుంది.
- దిగుమతి చేసుకున్న మాడ్యూల్ యొక్క విధులను యాక్సిన్ చేయడానికి డాట్ (.) ఆపరేటర్ ఉపయోగపడుతుంది.
- sys మాడ్యూల్ వ్యాఖ్యాత ఉపయోగించే కొన్ని వేరియబుల్స్ మరియు వ్యాఖ్యాతతో సంభాషించే ఫంక్షన్‌లకు యాక్సెస్ (access) ను అందిస్తుంది.
- పైథాన్‌లోని OS మాడ్యూల్ ఆపరేటింగ్ సిస్టమ్ ఆధారిత కార్యాచరణను ఉపయోగించుకునే మార్గాన్ని అందిస్తుంది.
- పైథాన్ యొక్క గెటాప్ట్ మాడ్యూల్ కమాండ్ లైన్ ఎంపికలు మరియు వాదనలను అన్వయించడము (split) నకు మీకు సహాయపడుతుంది.



అనుభవ విషయాలు

1. క్రింది వివరాలతో విద్యార్థి అనే class ని స్పష్టించడానికి C++ ప్రోగ్రామ్ వ్రాయండి.

   Protected member (సభ్యుడు)

   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

12th_CS_Telugu_Medium_Chapter_14.indd 280 28-03-2023 10:11:13

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

టెస్ట్ మరియు స్పోర్ట్స్ క్లాస్ నుండి పబ్లిక్‌గా తీసుకొనబడిన రిజల్ట్ అను క్లాస్ క్రింద ఇవ్వబడిన వివరాలను కలిగి ఉంటుంది.

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.

హైబ్రిడ్ అనే ఫైల్‌లో C++ ప్రోగ్రామ్‌ని సేవ్ చేయండి. hybrid.cpp ని ఎక్సిక్యూట్ చేయడానికి పైథాన్ ప్రోగ్రామ్‌ను వ్రాయండి.

2. Write a C++ program to print boundary elements of a matrix and name the file as Border.cpp. Write a python program to execute the Border.cpp

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. క్రింది వాటిలో ఏది స్క్రిప్టింగ్ భాష?

   అ) జావాస్క్రిప్ట్ ఆ) PHP ఇ) Perl (పెర్ల్) ఈ) HTML

2. పైథాన్ ప్రోగ్రామ్‌లో C++ ప్రోగ్రామ్‌ని దిగుమతి (Importing) చేసుకోవడం

   అ) వ్రాపింగ్ చేసుకోవడం ఆ) డైన్‌లోడ్ చేయడం

   ఇ) ఇంటర్ కనెక్టింగ్ ఈ) పార్సింగ్

BRJE6

3. API యొక్క విస్తరణ

అ) Application Programming Interpreter

ఆ) Application Programming Interface

ఇ) Application Performing Interface

ఈ) Application Programming Interlink

4. పైథాన్ మరియు C++ ప్రోగ్రామ్ ఇంటర్ పేసింగ్‌కు నిర్మాణం

అ) Ctypes ఆ) SWIG ఇ) Cython ఈ) Boost

5. ఈ క్రింది వానిలో ఏ సాఫ్ట్‌వేర్ డిజైన్ టెక్నిక్ మీ కోడ్ (code) రెండు వేర్వేరు భాగాలుగా విభజిస్తుంది?

అ) ఆబ్జక్ట్ ఓరియంటెడ్ ప్రోగ్రామింగ్ ఆ) మాడ్యులర్ ప్రోగ్రామింగ్

ఇ) లో లెవల్ ప్రొగ్రామింగ్ ఈ) ప్రొసిడ్యూర్ ఓరియంటెడ్ ప్రొగ్రామింగ్

6. విండోస్ ఆపరేటింగ్ సిస్టమ్‌తో ఇంటర్ ఫేస్ చేయడానికి మిమల్ని అనుమతించే మాడ్యూల్

అ) OS మాడ్యూల్ ఆ) sys మాడ్యూల్ ఇ) csv మాడ్యూల్ ఈ) getopt మాడ్యూల్

7. స్ట్రింగ్‌లను ఏ విధంగా విభజించినపుడు లోపం లేకుండా వున్నట్లయితే getopt() ఖాళీ శ్రేణిని తిరిగి పంపిస్తుంది.

అ) argv వేరియబుల్ ఆ) opt వేరియబుల్ ఇ) args వేరియబుల్
ఈ) ifile వేరియబుల్

8. క్రింది snippet ఫంక్షన్ కాల్ స్టేట్ మెంట్‌ను గుర్తించండి. if__name__=="__main__"

అ) main(sys.argv[1:]) ఆ) __name__ ఇ) __main__ ఈ) argv

9. ఈ క్రింది వాటిలో ఏది టెక్స్ట్ సంఖ్యలు ప్రతిబింబం మరియు సైన్స్‌కు సంబంధించిన పద్ధతులలో ఉపయోగిస్తారు.

అ) HTML ఆ) C ఇ) C++ ఈ) PYTHON

10. __name__ ఇది దేనిని కలిగి ఉంటుంది.

అ) c++ filename ఆ) main() name

ఇ) python filename ఈ) os module name

12th_CS_Telugu_Medium_Chapter_14.indd 282 28-03-2023 10:11:13

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

11. Scripting language ఇతర ప్రోగ్రామింగ్ లాంగ్వేజ్ కు మధ్యగల సిద్ధాంతపరమైన వ్యత్యాసమును వ్రాయండి?

2. compiler and interpreter లకు మధ్యగల బేధములను వ్రాయండి?

3. క్రింది వాటిని విపులీకరించండి (i) SWIG (ii) MinGW

4. మాడ్యూల్స్ యొక్క ఉపయోగము వ్రాయండి?

5. cd command యొక్క ఉపయోగమును ఉదాహరణతో తెలపండి.

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. PYTHON మరియు C++ మధ్యగల బేధములను వ్రాయండి?

2. స్క్రిప్టింగ్ లాంగ్వేజ్ యొక్క అనువర్తనాలను వ్రాయండి.

3. MinGW అనగానేమి? మరియు దాని ఉపయోగమును వ్రాయండి?

4. మాడ్యూల్ ఆపరేటర్‌ను గుర్తించి నిర్వచించండి క్రింది వాటి పేర్లు వ్రాయండి.

   welcome.display()

5. sys.argv అనగానేమి?

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. పైథాన్ యొక్క ఏదైనా లక్షణాలను వ్రాయండి?

2. క్రింది కమాండ్‌లను వివరించండి?

   Python <filename.py> -<i> <C++ filename without cpp extension>

3. Python లోని sys,os,getopt మాడ్యూల్ యొక్క ఉద్దేశ్యం ఏమిటి? వివరించండి.

4. కోసం getopt() కోసం syntax ను వ్రాసి దాని వాదనలు మరియు రిటర్న్ విలువలను వివరించండి.

5. క్రింది c++ కోడింగ్‌ను అమలు చేయడానికి పైథాన్ ప్రోగ్రామ్‌ను వ్రాయండి.

```
#include <iostream>
using namespace std;
int main()
{ cout<<“WELCOME”;
return(0);
}
```

The above C++ program is saved in a file welcome.cpp

REFERENCES

1. *Learn Python The Hard Way by Zed Shaw*
2. *Python Programming Advanced by Adam Stuart or Powerful Python by Aaron Maxwell*
3. *https://docs.python.org*

12th_CS_Telugu_Medium_Chapter_14.indd 284 28-03-2023 10:11:13

యూనిట్ V

అధ్యాయం 15

SQL ద్వారా డేటాను నిర్వహించుట

అభ్యసనా లక్ష్యములు



ఈ అధ్యాయము పూర్తి అయిన తర్వాత విద్యార్థులు పైథాన్ స్క్రిప్ట్‌ను ఉపయోగించి

- ఒక డేటాబేస్ పట్టికను తయారుచేసి దానిలో క్రొత్త వరుసలను డేటాబేస్‌లో చేర్చుట.
- ఒక డేటాబేస్ పట్టికలోని రికార్డును నవీనీకరించుట మరియు తొలగించుట.
- ఒక డేటా పట్టికను విశదీకరించుట.
- CSV ఫైల్‌ను విశదీకరించి వ్రాయుట.

15.1 పరిచయము

ఒక డేటాబేస్ అనేది ఏకీకృత డేటా సేకరణ అగును. 'డేటాబేస్' అనే పదము డేటాలను లేదా డేటాల నిర్వహణ అమరికను సూచించును. వినియోగదారులు మరియు డేటాబేస్ మధ్యగల పరస్పర చర్య కోసం సాఫ్ట్‌వేర్ అప్లికేషన్ డేటాబేస్ నిర్వహణ అవుతుంది. వినియోగదారులు మానవులుగా వుండనవసరంలేదు. ఇవి ఇతర ప్రోగ్రామ్‌లు మరియు అప్లికేషన్‌లు కూడా కావచ్చును. పైథాన్ ప్రోగ్రామ్ అనునది SQL డేటాబేస్‌తో ఎలా వినియోగదారుని చర్యనొందుటను ఇకపై నేర్చుకోగలము.

15.2 SQLite

SQLite అనునది ఒక సాధారణ సంబంధిత డేటాబేస్ వ్యవస్థీకృతం అగును. ఇది డేటాను సరియైన డేటా ఫైల్‌గా కంప్యూటర్ యొక్క అంతర్గత మెమరీలోనూ నిల్వ చేయబడుతుంది. ఇది My SQL లేక Oracle వంటి ప్రత్యేక డేటాబేస్ సర్వర్ ప్రోగ్రామ్ లేకుండా ఇంటిగ్రేటెడ్ అప్లికేషన్ రూపొందించబడినది అత్యంత పరీక్షించిన మరియు సౌకర్యవంతమైనందున SQLite తో పని చేయడము చాలా సులభం. SQLite కోసం పైథాన్‌లో గొప్ప లైబ్రరీని కలిగి ఉన్నది.

సోపానము: 1 sqlite3 ను అనుసందించుట.

సోపానము: 2 connect () క్రమ పద్ధతిని ఉపయోగించి అనుసంధానాన్ని సృష్టించి డేటాబేస్ పేరును పంపుట.

సోపానము: 3 cursor = connection. cursor () అను స్టేట్‌మెంట్‌ను ఉపయోగించి cursor () అను ఆబ్జెక్టను పంపడము.

- సోపానము 2 లో డేటాబేస్ సందించడం అంటే డేటాబేస్ పేరును పంపుట. ఈ విధంగా పంపునపుడు ఈ డేటాబేస్ ఇది వరకే వున్నట్లైతే అది చేర్చబడును లేకపోతే పైథాన్ ఇవ్వబడిన పేరులో ఒక క్రొత్త డేటాబేస్‌ను సృష్టించును

12th_CS_Telugu_Medium_Chapter_15.indd 285 28-03-2023 10:12:08

- సోపానము 3లో Cursor అనునది ఒక నియంత్రణ రూపం అగును. ఇది డేటాబేస్ నమోదులను అను సంధానించడానికి ఉపయోగపడుచున్నది.
- పైథాన్‌లో Cursor ఒక ముఖ్యపాత్ర వహించును. మొత్తము ఆదేశాలను Cursor ఆబ్జెక్ట్ ద్వారా మాత్రమే అమలుపరచబడును.

ఒక పట్టికను తయారుచేయుటకు డేటాబేస్‌లో ఒక ఆబ్జెక్ట్‌ను సృష్టించి దానికి తగిన SQL ఆదేశాలను వ్రాయవలెను.

ఉదాహరణ:- sql_comm = "SQL statement"

ఆదేశాలను నెరవేరుచుటకు cursor విధానాలను ఉపయోగించి sql ఆదేశాల పేరును పరామితిగా పంపవలెను. చాలా ఆదేశాలు sql_comm నిల్వచేయబడుతాయి మరియు అన్నీ చర్యలు పూర్తి చేసిన తరువాత మార్పులను ఫైల్‌లో భద్రపరచి తరువాత సంబంధించిన అనుసంధానమును తొలగించాలి.

15.3 SQLite ను ఉపయోగించి డేటాబేస్‌ను రూపొందించుట

క్రింది ఉదాహరణము sqlite3 ను ఉపయోగించి పైథాన్‌లోని ఒక డేటాబేస్‌ను ఎలా అనుసంధానించాలి అను విధానమును తెలియపరచును.

```
# Python code to demonstrate table creation and  insertions with SQL
# importing module
import sqlite3
# connecting to the database
connection = sqlite3.connect ("Academy.db")
# cursor
cursor = connection.cursor()
```

పై ఉదాహరణలో "Academy" అను పేరులో డేటాబేస్ సృష్టించబడినది. ఇది SQL యొక్క ("CREATE DATABASE) అను ఆదేశాలకు సమానమైనది పంపుటను ("sqlite3.connect ('Academy.db')") అను స్టెట్‌మెంట్‌ను ఉపయోగించినపుడు ముందే సృష్టించబడిన డేటాబేస్‌ను తెరుచుకొనును.

15.3.1 ఒక పట్టికను సృష్టించుట (Creating a Table)

ఒక ఖాళీ డేటాబేస్‌లను రూపొందిన తరువాత వీలైనంత వరకు ఒకటి లేదా అంతకంటే ఎక్కువైన పట్టికలను చేర్చవచ్చును. దీని ఆదేశ రూపము

CREATE TABLE Student (

RollnoINTEGER, SnameVARCHAR(20), GradeCHAR(1), gender CHAR(1),

Average float(5.2), birth_date DATE, PRIMARY KEY (Rollno));

12th_CS_Telugu_Medium_Chapter_15.indd 286 28-03-2023 10:12:09

పైన తెల్పిన విధానములో ఆదేశాలను అమలు పరుచు విదానముగానే ప్రత్యక్షముగా పైథాన్‌లో చేయవచ్చును. SQL లేదా SQLite కు ఆదేశాలను పంపుటకు cursor ఆబ్జెక్ట్ అవసరము సాధారణంగా SQL మరియు డేటాబేస్‌లోని cursor అనునది డేటాబేస్ రికార్డ్‌ను పొందుపరచబడిన నియంత్రణ రూపం అగును.

గమనిక

SQL యొక్క మొత్తము ఆదేశాలను అమలు చేయడానికి Cursor ఉపయోగపడుచున్నది.

అను సంధానానికి cursor() విధి విధానాలను అనుసరించి cursor ఆబ్జెక్ట్ రూపొందించవచ్చును. cursor ఫలిత సమితి నుండి రికార్డ్‌లను తీసుకొనుటకు ఉపయోగపడును. పైథాన్‌లో **SQL** ఆదేశాలను మూడు కోట్స్ బ్రాకెట్స్‌లలో నిర్వచింపబడును. మూడు కోట్‌ల వెనక కారణం కొన్ని సార్లు పట్టికలోని విలువలు సింగిల్ లేదా డబుల్ కోట్‌లను కలిగి ఉండవచ్చు

ఉదాహరణము: 15.3.1

```
sql_command = """
CREATE TABLE Student (
Rollno INTEGER PRIMARY KEY ,
Sname VARCHAR(20),
Grade CHAR(1),
gender CHAR(1),
Average DECIMAL(5,2),
birth_date DATE);"""
```

పై ఉదాహరణలో Roll no అను ఫీల్డ్ "INTEGER PRIMARY KEY" గా ఉన్నది. SQLite3 3లో ఈ విధంగా లేబుల్ చేయబడిన ఒక కాలమ్ దానంతటకు అదే స్వయం పెరుగుదల వుంటుంది. మరో విధంగా చెప్పాలంటే ఒక పట్టికలోని కాలమ్‌ను **INTEGER PRIMARY KEY** గా ప్రకటిస్తే ఎప్పుడైన ఒక **NULL** అనునది ఈ కాలమ్‌కు ఇన్‌పుట్‌గా ఉపయోగిస్తారు, **NULL** అనునది దానంతట అదే ఆ కాలమ్‌లో ఇప్పటి వరకు ఉపయోగించిన అత్యధిక విలువ కంటే ఒకటి ఎక్కువగా వుండే పూర్ణసంఖ్యగా మారుతుంది. పట్టిక శూన్యం అయితే విలువ 1 గా ఉపయోగించబడుతుంది.

15.3.2 రికార్డులను చేర్చుట (Adding Records)

SQLite పాపులేట్ చేయడానికి (రికార్డ్ జోడించడానికి) టేబుల్ "INSERT" కమాండ్ SQLite కి పంపబడుతుంది. ఎగ్జిక్యుట్ పద్ధతి కొంత చర్చను నిర్వహించడానికి SQL ఆదేశాన్ని అమలు చేస్తుంది. క్రింది ఉదాహరణ 15.3.2 పూర్తి పని ఉదాహరణ అమలు చేయడానికి ప్రోగ్రామ్ ఇప్పటికే అమలు చేయబడి ఉంది. మీరు SQL కమాండ్ "DROP TABLE" లైన్‌ను అన్ కామెంట్ చేయాలి.

ఉదాహరణము: 15.3.2 -1

```
import sqlite3
connection = sqlite3.connect ("Academy.db")
cursor = connection.cursor()
# if the table already exits then delete it using cursor.execute ("""DROP TABLE Student;""")
# where  Student is name of the table.
sql_command = """
CREATE TABLE Student (
Rollno INTEGER PRIMARY KEY , Sname VARCHAR(20), Grade CHAR(1),
gender CHAR(1), Average DECIMAL (5, 2), birth_date DATE);"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Student (Rollno, Sname, Grade, gender, Average,
    birth_date) VALUES (NULL, "Akshay", "B", "M","87.8", "2001-12-12");"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Student (Rollno, Sname, Grade, gender, Average,
    birth_date) VALUES (NULL, "Aravind", "A", "M","92.50","2000-08-17");"""
cursor.execute(sql_command)
# never forget this, if you want the changes to be saved:
connection.commit()
connection.close()
print("STUDENT TABLE CREATED")
```

అవుట్‌పుట్

STUDENT TABLE CREATED

సహజంగా చాలా సందర్భాలలో డేటాను SQL పట్టికలోకి చేర్చరు. కొన్ని పైథాన్ డేటారకం ఉదాహరణకు ఇన్‌సర్ట్ స్టేట్మెంట్ యొక్క ఇన్‌పుట్‌గా ఉపయోగించే ఒక నిఘంటువు లేక జాబిత లోపల ఇంచుమించు చాలా డేటా ఉంటుంది.

పై ప్రత్యక్ష ఉదాహరణలో Academy.db అనే డేటాబేస్ మరియు Student table ఇది వరకే వున్నదని అనుకొందుము. మన దగ్గర వున్న వ్యక్తుల యొక్క డేటాతో List అనునది Insert స్టేట్మెంట్‌లో ఉపయోగించబడుతుంది.

12th_CS_Telugu_Medium_Chapter_15.indd 288 28-03-2023 10:12:09

ఉదాహరణము: 15.3.2-2

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
student_data = [("BASKAR", "C", "M","75.2","1998-05-17"),
    ("SAJINI", "A", "F","95.6","2002-11-01"),
    ("VARUN", "B", "M","80.6","2001-03-14"),
    ("PRIYA", "A", "F","98.6","2002-01-01"),
    ("TARUN", "D", "M","62.3","1999-02-01") ]
for p in student_data:
    format_str = """INSERT INTO Student (Rollno, Sname, Grade, gender,Average,
    birth_date) VALUES (NULL,"{name}", "{gr}", "{gender}","{avg}","{birthdate}");"""
    sql_command = format_str.format(name=p[0], gr=p[1], gender=p[2],avg=p[3],
    birthdate = p[4])
cursor.execute(sql_command)
connection.commit()
connection.close()
print("RECORDS ADDED TO STUDENT TABLE ")
```

అవుట్‌పుట్

RECORDS ADDED TO STUDENT TABLE

In the above program {gr} is a place holder (variable) to get the value. format_str.format() is a function used to format the value to the required datatype.

15.4 పైథాన్ ఉపయోగించి SQL ను విశదీకరించుట

ఇప్పుడు "Student" పట్టికలో విశదీకరించుటను ఉపయోగించుటను నేర్చుకొనెదము. రికార్డు నుండి dataను పొందడం వాటిని చేర్చినంత సులభము అగును. Excute పద్ధతిలో SQL ఆదేశాలను ఉపయోగించి data base అన్నింటిని పట్టిక నుండి పొందవచ్చును.

15.4.1 SELECT విశదీకరించుట.

SQL విరివిగా ఉపయోగించు ఆదేశం **Select** స్టేట్‌మెంట్ అగును. SQL లోని **Select** స్టేట్‌మెంట్ డేటాబేస్‌లోని పట్టిక నుండి డేటాను తిరిగి పొందుటకు లేదా తీసుకురావడానికి ఉపయోగించెదరు. ఈ స్టేట్‌మెంట్‌ను ఉపయోగించడం కోసం వాక్యనిర్మాణం **"Select * from table_name"** మరియు మొత్తం పట్టిక డేటాను టుపుల్స్ జాబితా రూపంలో ఆబ్జెక్ట్‌లో పొందవచ్చు. మీరు ఈ Programe ను "sql_Academy_query.py" గా భద్రపరచినచో, వాస్తవ డేటాను బట్టి మీరు ఈ క్రింది ఫలితాన్ని పొందుతారు

మనము సృష్టించబడే database పైథాన్ ఫైల్ వలె అదే ఫోల్డర్‌లో ఉంటుందని గమనించుము. మనము file యొక్క మార్గాన్ని మార్చాలను కుంటే file ను తెలిచేటపుడు మార్గాన్ని మార్చండి.

ఉదాహరణము: 15.4.1-1

```
#save the file as "sql_Academy_query.py"
import sqlite3
connection = sqlite3.connect("Academy.db")
crsr = connection.cursor()
# execute the command to fetch all the data from the table Student
crsr.execute("SELECT * FROM Student")
# store all the fetched data in the ans variable
ans= crsr.fetchall()
# loop to print all the data
for i in ans:
        print(i)
```

15.4.1.1 fetchall() అను ఉపయోగించి రికార్డులను అన్నింటిని చూపెదము

fetchall() పద్ధతి వరుసల అన్నింటిని పట్టిక నుండి పొందుటకు ఉపయోగపడును.

ఉదాహరణము: 15.4.1.1-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetchall:")
result = cursor.fetchall()
for r in result:
    print(r)
```

అవుట్‌పుట్

```
fetchall:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'PRIYA', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```


12th_CS_Telugu_Medium_Chapter_15.indd 290 28-03-2023 10:12:09

గమనిక

cursor.fetchall() -fetchall () పద్ధతి database table లోని అడ్డు వరుసల అన్నింటిని పొందుటకు ఉపయోగపడును.

cursor.fetchone() - The fetchone () పద్ధతి ఫలిత సమితి నుండి తదుపరి అడ్డు వరుసను పొందడము (లేదా) వరుసలు ఏదియు లేనిచో None అను విలువను ఫలితముగా ఇచ్చును.

cursor.fetchmany() పద్ధతి ఫలిత సమితి నుండి మిగిలి ఉన్న వరుసల సంఖ్యలను ఇచ్చును.

15.4.1.2 fetchone() అను ప్రోగ్రామ్ ఉపయోగించి ఒక రికార్డును చూపడము

fetchone() అను పద్ధతి ఫలిత సమితిలోని తదుపరి వరుసను ఇచ్చును. (లేదా) ఏ వరుసను లేనిచో None అను విలువను ఇచ్చును.

ఉదాహరణము: 15.4.1.2-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("\nfetch one:")
res = cursor.fetchone()
print(res)
```

అవుట్‌పుట్

```
fetch one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
```

15.4.1.3 fetchone() ప్రోగ్రామ్‌ను ఉపయోగించి అన్ని రికార్డులను చూపడము.

while loop మరియు fetchone() పద్ధతి ఉపయోగించి పట్టికలోని అన్ని రికార్డులను చూడవచ్చును

ఉదాహరణము: 15.4.1.3 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching all records one by one:")
result = cursor.fetchone()
while result is not None:
    print(result)
    result = cursor.fetchone()
```

అవుట్‌పుట్

```
fetching all records one by one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'PRIYA', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```


12th_CS_Telugu_Medium_Chapter_15.indd 291 28-03-2023 10:12:09

15.4.1.4 fetchmany() ను ఉపయోగించి రికార్డులను చూపడము

ఒక ప్రత్యేక సంఖ్యను రికార్డులో చూపుటకు fetchmany() పద్ధతి ఉపయోగపడును. ఈ పద్ధతి ఫలిత సమితిలో మిగిలి ఉండు వరుసల సంఖ్యలను ఇచ్చును.

ఉదాహరణలు 15.4.1.4-1 fetchmany() ని ఉపయోగించి టుపుల్స్ విషయాలను చూపడము

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching first 3 records:")
result = cursor.fetchmany(3)
print(result)
```

అవుట్‌పుట్

```
fetching first 3 records:
[(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12'), (2, 'Aravind', 'A', 'M', 92.5, '2000-08-17'), (3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')]
```

ఉదాహరణము 15.4.1.4-2 loops ను ఉపయోగించకుండా టుపుల్స్ కంటెంట్‌ను స్క్రీన్‌లో ప్రదర్శించడానికి ప్రోగామ్.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching first 3 records:")
result = cursor.fetchmany(3)
print(*result,sep="\n") # * is used for unpacking a tuple.
```

అవుట్‌పుట్

```
fetching first 3 records:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
```

గమనిక

loop ఉపయోగించక విరామములో కూడిన పదాల పట్టికను ఒకే వరుసలో ముద్రించుటకు చిహ్నాలు = "\n" or sep= "," respectively ఉపయోగపడును.

12th_CS_Telugu_Medium_Chapter_15.indd 292 28-03-2023 10:12:09

15.4.2 SQL లోని క్లాజ్‌లు (CLAUSES)

SELECT statements ఉపయోగాల్లో SQL పలు విదములైన క్లాజ్‌లను అందిస్తోంది. SQLite లో సుమారు అన్ని రకములైన క్లాజ్‌లను ఉపయోగించుకొనుచున్నారు. అధికముగా ఉపయోగించు క్లాజ్‌లు క్రింది విధముగా

- DISTINCT
- WHERE
- GROUP BY
- ORDER BY.
- HAVING

15.4.2.1 SQL విభిన్న కీవర్డ్ (SQL DISTINCT KEYWORD):

ఒక ప్రత్యేకమైన నిలువ వరుస / పట్టికలోని నకిలీ విలువలను తొలగించడానికి ఒక ప్రత్యేకమైన కీవర్డ్ సహాయపడుతుంది. మనము ప్రత్యేక కీవర్డ్‌ను ఉపయోగించినపుడు ఏకైక విలువలు మాత్రమే పొందవచ్చును.

క్రింది ఉదాహరణములో “student table” అను పట్టిక నుండి విద్యార్థులు పొందు వివిధ గ్రేడ్‌లను చూపబడినది.

ఉదాహరణము: 15.4.2.1-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT (Grade) FROM student")
result = cursor.fetchall()
print(result)
```

అవుట్‌పుట్

```
[('B',), ('A',), ('C',), ('D',)]
```

పై ఉదాహరణలో “distinct” అను ప్రత్యేక పదమును ఉపయోగించని యడల 4 రికార్డులకు బదులుగా 7 రికార్డులు చూపబడి ఉండును. ఎందుకనగా ప్రధాన పట్టిక నందు ఒకే మాదిరిగా వుండే వాటితో కలిసి 7 రికార్డులు కల్గివుండును.

15.4.2.2 SQL WHERE క్లాజ్ (SQL WHERE CLAUSE)

ప్రత్యేక నియమాలను నెరవేర్చు రికార్డులను మాత్రము వేరు చేయుటకు WHERE క్లాజ్ ఉపయోగపడును. క్రింది ఉదాహరణములో “student table” డేటాబేస్ నుండి విద్యార్థుల గ్రేడ్‌లను మాత్రమే విభజించడం చూడగలము.

12th_CS_Telugu_Medium_Chapter_15.indd 293 28-03-2023 10:12:09

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")
result = cursor.fetchall()
print(*result,sep="\n")
```

అవుట్‌పుట్

```
('B',)
('A',)
('C',)
('D',)
```

15.4.2.3 SQL Group By క్లాజ్ (SQL Group By Clause)

ఒక ప్రత్యేకమైన రికార్డును క్లుప్తమైన వరుసలు కల్గిన సమూహముగా చేరును. ఇది వేర్వేరు సమూహమనకు ఒక రికార్డు ఇచ్చును. మూల్యాంక ప్రమేయము (COUNT, MAX, MIN, SUM, AVG) ఫలిత సమితిని ఒకటి లేక అంత కంటే ఎక్కువ నిలువ వరుస సమూహములుగా మార్చుటకు ఉపయోగపడును. క్రింది ఉదాహరణ student పట్టికలో నుండి స్త్రీ మరియు పురుషుల సంఖ్యలను లెక్కించి ఫలితమును చూపబడినది.

ఉదాహరణము: 15.4.2.3 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT gender,count(gender) FROM student Group BY gender")
result = cursor.fetchall()
print(*result,sep="\n")
```

అవుట్‌పుట్

```
('F', 2)
('M', 5)
```

15.4.2.4 SQL ORDER BY క్లాజ్ (SQL ORDER BY Clause)

క్లాజ్ ఒక నిర్ధిష్ట ప్రదేశాలలోని డేటా యొక్క వరుసను చూచుటకు SELECT statement లో చేర్చి ఉపయోగపడును ఫలిత సమితిని ఆరోహణ లేక అవరోహణ వరుసలో క్రమపరచుటకు ఉపయోగపడును క్రింది ఉదాహరణలో విద్యార్ధి పేరు మరియు వరుస సంఖ్య అనునవి అక్షర క్రమములో పేర్లు చూపబడినది.

12th_CS_Telugu_Medium_Chapter_15.indd 294 28-03-2023 10:12:09

ఉదాహరణము: 15.4.2.4 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname  FROM student Order BY sname")
result = cursor.fetchall()
print(*result,sep="\n")
```

అవుట్‌పుట్

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(6, 'PRIYA')
(4, 'SAJINI')
(7, 'TARUN')
(5, 'VARUN')
```

15.4.2.5 SQL HAVING క్లాజ్ (SQL HAVING Clause)

సమూహ ప్రమేయమును అనుసరించి డేటాను సూక్ష్మీకరించుటకు Having క్లాజ్ ఉపయోగపడును. ఇది WHERE నిబంధన వలే వుండును కాని సమూహ ప్రమేయాలతో ఉపయోగపడును. సమూహ ప్రమేయము WHERE క్లాజ్ లలో ఉపయోగించుట వీలుకాదు. కాని Having క్లాజ్‌లలో ఉపయోగించుటకు వీలు అగును.

ఉదాహరణము: 15.4.2.5 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT GENDER,COUNT(GENDER) FROM Student GROUP BY
GENDER HAVING COUNT(GENDER)>3")
result = cursor.fetchall()
co = [i[0] for i in cursor.description]
print(co)
print(result)
```

అవుట్‌పుట్

```
['gender', 'COUNT(GENDER)']
[('M', 5)]
```

15.5 SQL AND, OR మరియు NOT వాటి విధులు (క్రియలు)

WHERE క్లాజ్ AND, OR, మరియు NOT వాటి విధులుతో చేయు పనులు.

AND మరియు OR తో చేయు పనులు ఒకదాని కంటే ఎక్కువ నిబంధనలతో తొలగించడానికి ఉపయోగపడుచున్నది. ఉదాహరణకు students పట్టికలో నుండి 'A' మరియు 'B' గ్రేడ్‌ను పొందిన విద్యార్థి తప్ప ఇతరుల వివరాలను తెలియపరుచును.

ఉదాహరణకు WHERE క్లాజ్ తో NOT విధులు

ఉదాహరణము: 15.5 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student where NOT Grade='A' and NOT Grade='B'")
result = cursor.fetchall()
print(*result,sep="\n")
```

అవుట్‌పుట్

```
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

ఉదాహరణకు WHERE క్లాజ్ తో NOT విధులు

క్రింది ఇవ్వబడిన ఉదాహరణలలో సరాసరి 80 నుండి 90 శాతము (రెండు హద్దులను కలిపి) వచ్చిన విద్యార్థుల యొక్క పేరు అడ్మిషన్ సంఖ్య వారి యొక్క సరాసరిని తెలియపరుస్తుంది.

ఉదాహరణము: 15.5 -2

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,Same,Average FROM student WHERE (Average>=80 AND Average<=90)")
result = cursor.fetchall()
print(*result,sep="\n")
```

అవుట్‌పుట్

```
(1, 'Akshay', 87.8)
(5, 'VARUN', 80.6)
```

ఉదాహరణకు WHERE క్లాజ్ తో OR విధులు

ఈ క్రింది ఉదాహరణో 60 నుండి 70 శాతము మధ్య పొందని విద్యార్థుల సరాసరి యొక్క పేరు మరియు వరుస సంఖ్య తెలియపరుస్తుంది.

12th_CS_Telugu_Medium_Chapter_15.indd 296 28-03-2023 10:12:09

ఉదాహరణము: 15.5 -3

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname  FROM student WHERE (Average<60
OR  Average>70)")
result = cursor.fetchall()
print(*result,sep="\n")
```

అవుట్‌పుట్

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(4, 'SAJINI')
(5, 'VARUN')
(6, 'PRIYA')
```

15.6 విశదీకరించిన డేట్ (DATE) కాలమ్

క్రింది ఉదాహరణలో 2001 లో పుట్టిన విద్యార్థుల యొక్క పేరు మరియు ర్యాంక్‌ను తెలియపరుచును.

ఉదాహరణము: 15.6 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname FROM student
WHERE(Birth_date>='2001-01-01' AND Birth_date<='2001-12-01')")
result = cursor.fetchall()
print(*result,sep="\n")
```

అవుట్‌పుట్

```
(5, 'VARUN')
```

15.7 అగ్రిగేట్ ఫంక్షన్స్

నిలువ వరుసలలోని విలువలను తీసుకొని ప్రక్రియలను చేయడానికి ఈ ఫంక్షన్స్ ఉపయోగిస్తారు మరియు ఒకే ఒక విలువను తిరిగి పొందడము

- COUNT()
- AVG()
- SUM()
- MAX()
- MIN()

15.7.1 COUNT() ఫంక్షన్ (COUNT() function)

WHERE క్లాస్‌తో పేర్కొన్న నిబంధనను తృప్తి పరుస్తు పట్టికలోని అడ్డు వరుసల సంఖ్యను SQL COUNT() ఫంక్షన్ ఇస్తుంది. సరిపోలిన అడ్డువరుసలు లేనిచో COUNT() ను ఇస్తుంది.

ఉదాహరణము: 15.7.1-1

ఉదాహరణ 1: ఈ ఉదాహరణలో మనము మరిన్ని అడ్డు వరుసల సంఖ్యను లెక్కించపోతున్నాము.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(*)  FROM student ")
result = cursor.fetchall()
print(result)
```

అవుట్‌పుట్

```
[(7,)]
```

ఉదాహరణము: 15.7.1-2

ఉదాహరణ 2 : ఈ ఉదాహరణలో ఒక నిర్దేశిత నిలువ వరుసలోని రికార్డు సంఖ్యను లెక్కించపోతున్నాము.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result
```

అవుట్‌పుట్:

```
[(7,)]
```

గమనిక

NULL విలువలు లెక్కించబడవు. ఒక వేళ పై ఉదాహరణలో రికార్డులో NULL విలువ వుంటే అవుట్‌పుట్ 6 గా వుండును.

12th_CS_Telugu_Medium_Chapter_15.indd 298 28-03-2023 10:12:09

15.7.2AVG():

ఈ క్రింది SQL స్టేట్‌మెంట్ పైథాన్ ప్రోగ్రామ్ ద్వారా విద్యార్థుల సరాసరి మార్కులును కనుగొనవచ్చు.

ఉదాహరణము: 15.7.2-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT AVG(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

అవుట్‌పుట్

```
[(84.65714285714286,)]
```

గమనిక

NULL విలువలు పరిగణించబడదు.

15.7.3 SUM():

ఈ క్రింది SQL స్టేట్‌మెంట్ పైథాన్ ప్రోగ్రామ్ ద్వారా విద్యార్థుల విలువల గల పట్టికలో ప్రతి సరాసరి విలువలను కనుగొనవచ్చును.

ఉదాహరణము: 15.7.1-3

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT SUM(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

అవుట్‌పుట్

```
[(592.6,)]
```

గమనిక

NULL విలువలు పరిగణించబడదు.

15.7.4 MAX() మరియు MIN() ఫంక్షన్స్:

The MAX() ఫంక్షన్ ఒక నిలువ వరుసలోని ఎన్నుకొన్న అతి పెద్ద విలువను తెలియజేయును.

The MIN() ఫంక్షన్ ఒక నిలువ వరుసలోని ఎన్నుకొన్న అతి చిన్న విలువను తెలియజేయును.

ఈ ఉదాహరణ ద్వారా సరాసరి విలువ కల్గిన విద్యార్థి పేరును తెలియజేయును.

ఉదాహరణము: 15.7.4-1

```
import sqlite3
connection = sqlite3.connect("Organization.db")
cursor = connection.cursor()
print("Displaying the name of the Highest Average")
cursor.execute("SELECT sname,max(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
print("Displaying the name of the Least Average")
cursor.execute("SELECT sname,min(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

అవుట్‌పుట్

```
Displaying the name of the Highest Average
[('PRIYA', 98.6)]
Displaying the name of the Least Average
[('TARUN', 62.3)]
```

15.8 రికార్డును నవీనీకరించుటం

పైథాన్ స్క్రిప్ట్ ద్వారా పట్టికలో నమోదు చేసిన విలువలను నవీనీకరించవచ్చు. ఇచ్చిన ఉదాహరణలో విద్యార్థి పట్టికలో “Priya” అను పేరును “Priyanka” గా నవీనీకరించవచ్చు

12th_CS_Telugu_Medium_Chapter_15.indd 300 28-03-2023 10:12:09

ఉదాహరణము: 15.8 -1

```
# code for update operation
import sqlite3
# database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
# update the student record
conn.execute("UPDATE Student SET sname ='Priyanka' where Rollno='6'")
conn.commit()
print ("Total number of rows updated :", conn.total_changes)
cursor = conn.execute("SELECT * FROM Student")
for row in cursor:
        print (row)
conn.close()
```

అవుట్‌పుట్

```
Total number of rows updated : 1
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

గమనిక

ఈ పాఠ్యాంశము అంతయు SQL కొర్రీలు విద్యార్థి పట్టిక నుండి తీసుకొనబడినది. ఉదాహరణ 15.3.2.2లో విద్యార్థి పట్టికకు సంబంధించిన వాటిని పొందుపరచబడినది.

12th_CS_Telugu_Medium_Chapter_15.indd 301 28-03-2023 10:12:09

15.9 తొలగింపు (Deletion) ప్రక్రియ

Sql కమాండ్స్ (ఆదేశాలు) ద్వారా రికార్డులను తొలగించేలాగా పైథాన్ ద్వారా కూడా రికార్డులను తొలగించవచ్చును. క్రింది ఉదాహరణలో "student" పట్టిక నుండి Rollno 2 యొక్క విషయాన్ని తొలగించడాని చూడవచ్చును.

ఉదాహరణము: 15.9-1

```
# code for delete operation
import sqlite3
 # database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
# delete student record from database
conn.execute("DELETE from Student where Rollno='2'")
conn.commit()
print("Total number of rows deleted :", conn.total_changes)
cursor =conn.execute("SELECT * FROM Student")
for row in cursor:
        print(row)
conn.close()
```

అవుట్‌పుట్

```
Total number of rows deleted : 1
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```


12th_CS_Telugu_Medium_Chapter_15.indd 302 28-03-2023 10:12:09

15.10 ఉపయోగించేవారుచే డేటాను ఇన్‌పుట్ (నివేశపరచడం) చేయడం

ఈ ఉదాహరణలో ప్రోగ్రామ్ రన్ అయ్యేటప్పుడు మరియు Person అనే పట్టికలో రాసేటప్పుడు పైథాన్ input() ఆదేశాలను ఉపయోగించి డేటాను అంగీకరించబోతున్నాము.

ఉదాహరణము: 15.10 -1

```
# code for executing query using input data
import sqlite3
# creates a database in RAM
con =sqlite3.connect("Academy.db")
cur =con.cursor()
cur.execute("DROP Table person")
cur.execute("create table person (name, age, id)")
print("Enter 5 students names:")
who =[input() for i in range(5)]
print("Enter their ages respectively:")
age =[int(input()) for i in range(5)]
print("Enter their ids respectively:")
p_id =[int(input())for i in range(5)]
n =len(who)
for i in range(n):
# This is the q-mark style:
        cur.execute("insert into person values (?, ?, ?)", (who[i], age[i], p_id[i]))
cur.execute("select * from person")
# Fetches all entries from table
print("Displaying All the Records From Person Table")
print (*cur.fetchall(), sep='\n' )
```

అవుట్‌పుట్

```
Enter 5 students names:
        RAM
        KEERTHANA
        KRISHNA
        HARISH
        GIRISH
Enter their ages respectively:
        28
        12
        21
        18
        16
Enter their ids respectively:
        1
        2
        3
        4
        5
Displaying All the Records From Person Table
        ('RAM', 28, 1)
        ('KEERTHANA', 12, 2)
        ('KRISHNA', 21, 3)
        ('HARISH', 18, 4)
        ('GIRISH', 16, 5)
```

పై కోడింగ్‌ను ఉపయోగించుకొని Field Name తో తగిన మార్పులతో ఇది వరకే వున్న “Student” లాంటి పట్టికకు రికార్డులను సైతము జోడించవచ్చును.

గమనిక

అమలు చేయండి (sql[, parameters]) ఒకే SQL స్టేట్‌మెంట్‌ను అమలు చేస్తుంది. SQL స్టేట్మెంట్ పారామిటరైజ్ చేయబడవచ్చు. (SQL అక్షరాలకు బదులుగా ప్లేస్‌హోల్డర్‌లనుపయోగించండి) sqlite 3 మాడ్యుల్ రెండు రకాలు ప్లేస్ హోల్డర్‌లకు మద్దత్తు ఇస్తుంది. ప్రశ్న గుర్తులు(?) మరియు పేరు పెట్టబడింది ప్లేస్‌హోల్డర్లు ("name style").

12th_CS_Telugu_Medium_Chapter_15.indd 304 28-03-2023 10:12:09

15.11 బహుళ పట్టికను ఉపయోగించి ప్రశ్నించడం

పట్టికలను కలపడం ద్వారా ఒకటి కంటే ఎక్కువ పట్టికలను ప్రశ్నించడానికి పైథాన్ అనుమతిస్తుంది. క్రింది ఉదాహరణలో విద్యార్థికి సంబంధించిన వివరాలు (Rollno, Duty, Age) వంటి వివరాలు కల్గివున్న ఒక కొత్త పట్టిక "Appointments" రూపొందించబడినది. శీర్షికతో "student" మరియు "Appointments" పట్టికలను అనుసంధానము చేసి నిలువ వరుస శీర్షికలో కూడిన ఫలితాలను ప్రకటించడము జరిగినది.

ఉదాహరణము: 15.11-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("""DROP TABLE Appointment;""")
sql_command = """
CREATE TABLE Appointment(rollnointprimarkey,Dutyvarchar(10),age int)"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Appointment (Rollno,Duty ,age )
        VALUES ("1", "Prefect", "17");"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Appointment (Rollno, Duty, age)
        VALUES ("2", "Secretary", "16");"""
cursor.execute(sql_command)
#  never forget this, if you want the changes to be saved:
connection.commit()
cursor.execute ("SELECT student.rollno,student.sname,
        Appointment.Duty, Appointment.Age FROM student,Appointment
        where student.rollno=Appointment.rollno")
#print (cursor.description) to display the field names of the table
co = [i[0] for i in cursor.description]
print(co)
#  Field informations can be read from cursor.description.
result = cursor.fetchall()
for r in result:
        print(r)
```

అవుట్‌పుట్

```
['Rollno', 'Sname', 'Duty', 'age']
(1, 'Akshay', 'Prefect', 17)
(2, 'Aravind', 'Secretary', 16)
```

గమనిక

cursor description ఒక్కొక్క నిలువ వరుస శీర్షిక యొక్క వివరాలు కల్గివుంటుంది. ఇది టుపుల్ (tuple) గా నిల్వచేయబడును మరియు మొదటిది అనగా 0 (సున్న) చూచిక కాలము యొక్క పేరును తెలియజేస్తుంది. సూచిక 1 నుండి కాలము యొక్క విలువలు తెలియును. ఈ ఆదేశాన్ని (command) ఉపయోగించి పట్టిక యొక్క ఫీల్డ్ (field) పేర్లను చూపవచ్చును.

12th_CS_Telugu_Medium_Chapter_15.indd 305 28-03-2023 10:12:09

15.12 CSV ఫైల్‌తో ప్రశ్నను సమగ్ర పరచడము

మీరు query CSV ఫైల్‌లో కూడా నిల్వచేయవచ్చును. ఇది query పట్టిక రూపంలో ప్రదర్శించడానికి ఉపయోగపడుతుంది. ఈ క్రింది ఉదాహరణ (15.12.1)లో పైథాన్ స్క్రిప్ట్‌ను ఉపయోగించి విద్యార్థి పట్టిక "gender" ఆధారంగా అవరోహణ క్రమములో క్రమబదీకరించడం మరియు రికార్డులు అక్షర క్రమములో అమర్చబడినది "SQL.CSV", అనే CSV ఫైల్‌లో ఈ query యొక్క ఔట్‌పుట్ (output) వ్రాయబడుతుంది. మళ్ళీ CSV ఫైల్ నుండి విషయంను చదవడం మరియు ఫలితాలను ప్రదర్శించడము జరుగుతుంది.

ఉదాహరణము: 15.12 -1

```
import sqlite3
import io                                   #  to access replace()
import csv
                                        #  CREATING CSV FILE
d=open('c:/pyprg/sql.csv','w')
c=csv.writer(d)
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
                                        #  a=Connection.cursor()
cursor.execute("SELECT *  FROM student ORDER BY GENDER DESC,SNAME")
                                #  WRITING THE COLUMN HEADING
co = [i[0] for i in cursor.description]
c.writerow(co)
data=cursor.fetchall()
for item in data:
        c.writerow(item)
d.close()
#  Reading the CSV File
#  replace() is used to eliminate the newline character at the end of each row
with open('c:/pyprg/sql.csv', "r", newline=None) as fd:
        #  r = csv.reader(fd)
        for line in fd:
                line = line.replace("\n", "")
                print(line)
cursor.close()
connection.close()
```

అవుట్‌పుట్

```
Rollno,Sname,Grade,gender,Average,birth_date
1, Akshay, B, M, 87.8, 2001-12-12
2, Aravind, A, M, 92.5, 2000-08-17
3, BASKAR, C, M, 75.2, 1998-05-17
7, TARUN, D, M, 62.3, 1999-02-01
5, VARUN, B, M, 80.6, 2001-03-14
6, PRIYA, A, F, 98.6, 2002-01-01
4, SAJINI, A, F, 95.6, 2002-11-01
```


12th_CS_Telugu_Medium_Chapter_15.indd 306 28-03-2023 10:12:09

ఉదాహరణ 15.12 -2 MS-Excel ద్వారా ఫైల్ (“sqlexcel.csv”) ను తెరవడం మరియు ఫలితాన్ని (ఉదాహరణ 15.12 -1) స్క్రిప్ట్ లాంటి ప్రోగ్రామ్‌ను చూడవచ్చును.

```
import sqlite3
import io                              #to access replace()
import csv
# database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
print(“Content of the table before sorting and writing in CSV file”)
cursor = conn.execute("SELECT * FROM Student")
for row in cursor:
        print (row)
#  CREATING CSV FILE
d=open('c:\pyprg\sqlexcel.csv','w')
c=csv.writer(d)
cursor = conn.cursor()
cursor.execute("SELECT * FROM student ORDER BY GENDER DESC,SNAME")
#WRITING THE COLUMN HEADING
co = [i[0] for i in cursor.description]
c.writerow(co)
data=cursor.fetchall()
for item in data:
        c.writerow(item)
d.close()
print(”sqlexcel.csv File is created open by visiting c:\pyprg\sqlexcel.csv”)
conn.close()
```

గమనిక

పొరపొటున csv ఫైల్‌లో వ్రాసేటప్పుడు ప్రతి రికార్డు \n (newline) తో ముగుస్తుంది. csv ఫైల్‌ను చదివేటప్పుడు newline ను తొలగించడానికి replace () ను ఉపయోగిస్తాము.

12th_CS_Telugu_Medium_Chapter_15.indd 307 28-03-2023 10:12:09

అవుట్‌పుట్

Content of the table before sorting and writing in CSV file

(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')

(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')

(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')

(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')

(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')

(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')

(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')

sqlexcel.csv File is created open by visiting c:\pyprg\sqlexcel.csv

OUTPUT THROUGH EXCEL

	A	B	C	D	E	F
1	Rollno	Sname	Grade	gender	Average	birth_date
2						
3	1	Akshay	B	M	87.8	2001-12-12
4						
5	2	Aravind	A	M	92.5	2000-08-17
6						
7	3	BASKAR	C	M	75.2	1998-05-17
8						
9	7	TARUN	D	M	62.3	1999-02-01
10						
11	5	VARUN	B	M	80.6	2001-03-14
12						
13	6	PRIYA	A	F	98.6	2002-01-01
14						
15	4	SAJINI	A	F	95.6	200-11-01

15.3 పట్టిక జాబిత

డేటాబేస్‌లో రూపొందించిన పట్టికల జాబితాను ప్రదర్శించడానికి క్రింది ప్రోగ్రామ్ (ఉదాహరణ 15.3.1) ని ఉపయోగించవచ్చును.

ఉదాహరణము: 15.3 – 1

```
import sqlite3
con = sqlite3.connect('Academy.db')
cursor = con.cursor()
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
print(cursor.fetchall())
```

అవుట్‌పుట్

[('Student',), ('Appointment',), ('Person',)]

12th_CS_Telugu_Medium_Chapter_15.indd 308 28-03-2023 10:12:10

'Academy' డేటాబేస్‌లో రూపొందించిన అన్ని పట్టికల యొక్క పేర్లను పై ప్రోగ్రామ్ (ఉదాహరణ 15. 3. 1) ప్రదర్శిస్తుంది. మీ యొక్క డేటాబేస్ పట్టిక గురించి **master table** కీలక సమాచారాన్ని కల్గివుంటుంది మరియు దీనినే **sqlite_master** అందురు.

మీరు ఇప్పటి వరకు పైథాన్ స్క్రిప్ట్‌లో SQL ను ఉపయోగిస్తున్నారు. ఈ అధ్యాయములో అనేక ప్రాధమిక ఆదేశాలు పూర్తి చేయబడినది. పైథాన్ SQLite module దాదాపు అన్నీ SQL ఆదేశాల(commands)ను అమలుపరచగలదు. SQL అధ్యాయములో చర్చించిన ఇతర ఆదేశాలు గూర్చి కూడా మీరు ప్రయత్నించవచ్చు.

ముఖ్యాంశాలు గుర్తుకు తెచ్చుకుందాం

- డేటాబేస్ అనునది ఏకీకృత సేకరణ
- డేటాబేస్ అనునది వినియోగదారులు మానవ వినియోగదారులు గాని, ఇతర ప్రోగ్రామ్ లేదా అప్లికేషన్ కావచ్చు.
- SQLite అనునది ఒక సరళమైన రిలేషన్‌ల డేటాబేస్ వ్యవస్థ. ఇది సాధారణ డేటా ఫైల్స్‌లో దాని డేటాను భద్రపరుచును.
- Cursor అనేది డేటాబేస్ యొక్క రికార్డులను బదలి చేయడానికి మరియు తీసుకురావడానికి ఉపయోగించే నియంత్రణ నిర్మాణము. cursor object ను ఉపయోగించడం ద్వారా మాత్రమే అన్నీ SQL ఆదేశాలు అమలు చేయబడుతాయి.
- Table డేటా ఏక లేదా ద్వి కోట్స్‌ను కల్గివుండవచ్చు. కాబట్టి పైథాన్‌లోని SQL ఆదేశాలను ట్రిపుల్ కోట్‌డ్ string గా సూచిస్తారు.
- “SELECT” ఎంచుకోండి అనేది SQL లలో చాలా తరుచగా ఉపయోగించే స్టేట్మెంట్.
- డేటాబేస్‌లోని Table నుండి డేటాను తిరిగి పొందడానికి లేక తీసుకురావడానికి SQL లోని SELECT స్టేట్మెంట్‌ను ఉపయోగిస్తారు.
- GROUP BY క్లాస్ రికార్డులను సారాంశ వరుసల సమూహాముగా చేస్తుంది.
- నిర్ధిష్ట విభాగాల యొక్క డేటాను అక్షర క్రమములో ఉంచడానికి SELECT స్టేట్మెంట్‌తో బాటు ORDER BY క్లాజ్‌ను ఉపయోగిస్తాము.
- GROUP విధులు ఆధారంగా డేటాను filter చేయడానికి Having క్లాజ్‌ను ఉపయోగిస్తారు.
- GROUP BY తో పాటు WHERE క్లాజ్‌ను ఉపయోగించలేము.
- WHERE క్లాజ్‌ను AND, OR, మరియు NOT ప్రక్రియలలో కలపవచ్చును.
- ఒకటి కంటే ఎక్కువ నిబంధనల ఆధారంగా రికార్డులను వడపోత (filter) చేయడానికి ‘AND’ మరియు ‘OR’ ప్రక్రియలను ఉపయోగిస్తారు.
- నిలువ వరుసలోని విలువల నుండి ప్రక్రియలు చేయడానికి అగ్రిగేట్ ఫంక్షన్స్‌ను ఉపయోగిస్తారు మరియు ఒకే ఒక విలువను ఇస్తుంది.
- COUNT() ఫంక్షన్ పట్టికలోని అడ్డు వరుసల సంఖ్యను అందిస్తుంది.
- AVG() ఫంక్షన్ పట్టికలోని ఎంచుకొన్న నిలువ వరుసల సరాసరి తిరిగిపొందును.
- SUM() ఫంక్షన్ పట్టికలోని ఎంచుకొన్న నిలువ వరుసల మొత్తాన్ని తిరిగిపొందును.
- MAX() ఫంక్షన్ ఎంచుకొన్న కాలమ్ యొక్క అతి పెద్ద విలువను అందిస్తుంది.
- MIN() ఫంక్షన్ ఎంచుకొన్న కాలమ్ యొక్క అతి చిన్న విలువను అందిస్తుంది.
- sqlite_master అనేది మీ డేటాబేస్ పట్టికల గురించి కీలక సమాచారాన్ని కల్గివున్న మాస్టర్ పట్టిక.
- పైథాన్‌లో ఫైల్ యొక్క మార్గాన్ని ‘/’ లేక ‘\’ తో సూచిస్తారు. ఉదాహరణకు మార్గం అనేది 'c:/pyprg/sql.csv' (లేక) c:\pyprg\sql.csv’గా నిర్ధేశించబడినది.

12th_CS_Telugu_Medium_Chapter_15.indd 309 28-03-2023 10:12:10



ప్రాక్టికల్ ప్రోగ్రామ్స్

1. క్రింది ఇవ్వబడిన పట్టిక కోసం పైథాన్ ఉపయోగించి యుజర్ నుండి వివరాలను అంగీకరించటానికి మరియు csv ఫైల్‌లో నిల్వచేయడానికి ఒక ఇంటర్ యాక్టివ్ ప్రోగ్రామ్‌ను రూపొందించండి.

   డేటాబేస్ పేరు;- DB1

   పట్టిక పేరు : Customer

Cust_Id	Cust_Name	Address	Phone_no	City
C008	Sandeep	14/1 Pritam Pura	41206819	Delhi
C010	Anurag Basu	15A, Park Road	61281921	Kolkata
C012	Hrithik	7/2 Vasant Nagar	26121949	Delhi

2. GAMES అనే క్రింది పట్టికను తీసుకొందాము (i) నుండి (iv) ప్రశ్నల వరకు రికార్డులను ప్రదర్శించడానికి మరియు (v) నుండి (viii) వరకు ఔట్‌పుట్ ఇవ్వడానికి ఒక పైథాన్ ప్రోగ్రామ్‌ను వ్రాయండి.

   పటిక:: GAMES

Gcode	Name	GameName	Number	PrizeMoney	ScheduleDate
101	Padmaja	Carom Board	2	5000	01-23-2014
102	Vidhya	Badminton	2	12000	12-12-2013
103	Guru	Table Tennis	4	8000	02-14-2014
105	Keerthana	Carom Board	2	9000	01-01-2014
108	Krishna	Table Tennis	4	25000	03-19-2014

i. schedule date ల అవరోహణ క్రమములోని అన్నీ Games యొక్క పేర్లును వాటి Gcodes తో ప్రదర్శించడము.

ii. ప్రైజ్‌మనీ 7000 కంటే ఎక్కువగా కల్గివున్న games ల యొక్క వివరాలను ప్రదర్శించడము.

iii. Game పేరు యొక్క ఆరోహణ క్రమములో ఆటగాళ్ళు యొక్క పేర్లు మరియు వారి పేరును ప్రదర్శించడం.

iv. ఒక్కొక్క పాల్గొనే సమూహము (నిలువ వరుస 4లో చూపిన విధంగా) యొక్క ప్రైజ్‌మనీ మొత్తమును ప్రదర్శించడము.

v. Game పేరు ఆధారంగా అన్ని రికార్డులను ప్రదర్శించండి.

12th_CS_Telugu_Medium_Chapter_15.indd 310 28-03-2023 10:12:10

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. క్రింది వాటిలో ఏకీకృత డేటా సేకరణ ఏది?

   అ) డేటాబేస్ ఆ) DBMS ఇ) సమాచారం ఈ) రికార్డులు

2. SQLite ఏ డేటాబేస్ పద్ధతి క్రింద వస్తుంది?

   అ) చదునైన ఫైల్ డేటాబేస్ విధానం ఆ) సంబంధం గల డేటాబేస్ విధానం

   ఇ) క్రమానుగత డేటాబేస్ విధానం ఈ) ఆబ్జెక్ట్ ఒరియెంటెడ్ డేటాబేస్ విధానం

3. క్రింది వానిలో డేటాబేస్ యొక్క రికార్డు బదిలీ చేయడం మరియు తిరిగి పొందడానికి ఏ నియంత్రణ వ్యవస్థీను ఉపయోగిస్తారు?

   అ) పాయింటర్ ఆ) కీ ఇ) కర్సర్ ఈ) ఇన్‌సర్షన్ పాయింట్

4. రికార్డు యొక్క విలువలలో ఏ మార్పు చేయడానికైనా ఏ కమాండ్ చేత సేవ్ చేయబడాలి.

   అ) Save ఆ) Save As ఇ) Commit ఈ) Oblige

5. కొన్ని యాక్షన్ చేయడానికి క్రింది వానిలో SQL కమాండ్‌ను చేసేది ఏది?

   అ) execute() ఆ) key() ఇ) cursor() ఈ) run()

6. ఒక పట్టికలో ఎంపిక చేసిన నిలువ వరుసలోని అడ్డువరుసల సరాసరిని తిరిగి పొందే ఫంక్షన్ ఏది?

   అ) Add() ఆ) SUM() ఇ) AVG() ఈ) AVERAGE()

7. ఎంపిక కాబడిన నిలువ వరుసలోని అత్యధిక విలువను తిరిగి ఇచ్చే విధానము

   అ) MAX() ఆ) LARGE() ఇ) HIGH() ఈ) MAXIMUM()

8. క్రింది వాటిలో దేనిని మాస్టర్ పట్టిక అంటారు.

   అ) sqlite_master ఆ) sql_master ఇ) main_master ఈ) master_main

9. చాలా సాధారణంగా (ఉమ్మడిగా) ఉపయోగించే SQL ఏది?

   అ) cursor ఆ) select ఇ) execute ఈ) commit

10. Duplicate ను నివారించే కీ వర్డ్ ఏది?

    అ) Distinct ఆ) Remove ఇ) Where ఈ) GroupBy

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. డేటాబేస్‌ను ఉపయోగించే వారిని సూచించండి?
2. ఒక డేటాబేస్‌ను అనుసంధానం చేయడానికి ఏ పద్ధతిని ఉపయోగిస్తారు? ఒక ఉదాహరణ ఇవ్వండి.
3. ఒక కాలమ్‌ను “INTEGER PRIMARY KEY” గా ప్రకటించడం వల్ల ఉపయోగమేమి?
4. పట్టికలో రికార్డును పాపులేట్ చేయడానికి ఉపయోగించే కమాండ్‌ను వ్రాయండి. ఒక ఉదాహరణ ఇవ్వండి.
5. డేటాబేస్ పట్టిక నుంచి అన్ని రికార్డులను పొందడానికి ఏ పద్ధతిని ఉపయోగిస్తారు?

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. SQLite అనగానేమి? దాని ఉపయోగాలేవి?
2. fetchone() మరియు fetchmany() ల మధ్య బేధాన్ని చెప్పండి?
3. Where క్లాస్ యొక్క ఉపయోగమేమి? WHERE క్లాజ్‌ను ఉపయోగించి ఒక పైథాన్ స్టేట్‌మెంట్‌ను ఇవ్వండి.
4. క్రింది వివరాలను చదవండి. దాని ఆధారంగా డిపార్ట్‌మెంట్‌ల వారిగా రికార్డులను ప్రదర్శించడానికి ఒక పైథాన్ స్క్రిప్ట్ వ్రాయండి.

   డేటాబేస్ పేరు :- organization.db

   పట్టిక పేరు :- Employee

   పట్టికలోని నిలువ వరుసలు :- Eno, EmpName, Esal, Dept
5. క్రింది వివరాలను చదవండి దాని ఆధారంగా రికార్డులను వాటి యొక్క ENO అవరోహణ క్రమములో ప్రదర్శించడానికి ఒక పైథాన్ స్క్రిప్ట్‌ను వ్రాయండి.

   డేటాబేస్ పేరు :- organization.db

   పట్టిక పేరు :- Employee

   పట్టికలోని నిలువ వరుసలు :- Eno, EmpName, Esal, Dept

భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. SQLite మరియు దానిని ఉపయోగించడానికి ఉపయోగించే సోపానాలను గూర్చి క్లుప్తంగా వ్రాయండి.
2. Fetchmany() ను ఉపయోగించి క్రింది పట్టికలోని అన్ని రికార్డులను ప్రదర్శించడానికి పైథాన్ స్క్రిప్ట్‌ను వ్రాయండి.

ఐకోడ్	వస్తువు పేరు	ధర (రేటు)
1003	Scanner	10500
1004	Speaker	3000
1005	Printer	8000

12th_CS_Telugu_Medium_Chapter_15.indd 312 28-03-2023 10:12:10

1008	Monitor	15000
1010	Mouse	700

3. HAVING క్లాస్ యొక్క ఉపయోగము ఏమి ఉదాహరణలిమ్ము

4. క్రింది వివరాల ద్వారా ITEM అను పట్టికను తయారుచేయుటకు పైథాన్ స్క్రిప్ట్ వ్రాయుము.

పట్టికకు ఒక రికార్డును చేర్చుము

డేటాబేస్ పేరు :- ABC

పట్టిక పేరు :- Item

నిలువ వరుస పేరు మరియు వివరాలు :-

Icode	:-	integer and act as primary key
Item Name	:-	Character with length 25
Rate	:-	Integer
Record to be added	:-	1008, Monitor,15000

క్రింది Supplier మరియు item (i) పట్టికను గమనించి ఒకటి మరియు (ii) ప్రశ్నలకు పైథాన్ స్క్రిప్ట్‌ను వ్రాయుము

SUPPLIER				
Suppno	**Name**	**City**	**Icode**	**SuppQty**
S001	Prasad	Delhi	1008	100
S002	Anu	Bangalore	1010	200
S003	Shahid	Bangalore	1008	175
S004	Akila	Hydrabad	1005	195
S005	Girish	Hydrabad	1003	25
S006	Shylaja	Chennai	1008	180
S007	Lavanya	Mumbai	1005	325

i) డిల్లీలో నివసించని suppliers యొక్క Name, City మరియు Item పేర్లును డిస్‌ప్లే చేయుము.

ii) అకిలా యొక్క Suppqty యొక్క విలువతో 40ను అధికరించుము.

References

1. *The Definitive Guide to SQLite by Michael Owens*
2. *Programming for Beginners: 2 Manuscripts: SQL & Python by Byron Francis*
3. *Tutorialspoint.com*

యూనిట్ V

అధ్యాయం 16

డేటా విజువలైజేషన్: ఉపయోగించి రేఖా చిత్రము వృత్త రేఖాపటము మరియు బార్ చిత్రము

అభ్యసనా లక్ష్యములు

ఈ అధ్యాయము పూర్తి అయిన తర్వాత విద్యార్థి నేర్చుకొనే అంశాలు.

- డేటా విజువలైజేషన్ (Data Visualization) అను పదాన్ని నిర్వచించగలరు.
- సమాచార దృశీకరణ రకాలకు తెలుపుట
- సమాచార దృశీకరణ యొక్క ఉపయోగాలు జాబితా పరుచుట
- Matplotlib లో దృశీకరణ విధాన రకములుకు జాబితా పరుచుట
- Matplotlib సంస్థాపన అభ్యసించుట
- Matplotlib దిగుమతి చేసుకొనుటను అన్వేషించుట
- సమాచార దృశీకరణ వివరము వర్గీకరించుట
- Matplotlib ను ఉపయోగించి పలురకములైన రేఖాచిత్రాలను సృష్టించుట

16.1 సమాచార దృశీకరణ నిర్వచనము (Data Visualization Definition)

డేటా విజువలైజేషన్ అనేది సమాచారం మరియు డేటా యొక్క గ్రాఫికల్ ప్రాతినిధ్యం. డేటా విజువలైజేషన్ యొక్క లక్ష్యం సమాచారాన్ని దృశ్యమానంగా వినియోగదారులకు తెలియజేయడం. దీనికోసం డేటా విజువలైజేషన్ గణాంకగ్రాఫర్‌లకు ఉపయోగిస్తుంది. పరిమాణాత్మక సందేశాన్ని దృశ్యరూపంగా కమ్యూనికేట్ చేయడానికి న్యూమర్‌కల్ డేటా డౌట్స్, పంక్తులు లేదా బార్ (కమ్మి చిత్రాలను) ఉపయోగించి ఎన్‌కోడ్ చేయవచ్చు.

సాధార దృశ్య సమాచారంలోని రకాలు :

- చార్టులు (Charts)
- పట్టికలు (Tables)
- రేఖాచిత్రాలు (Graphs)
- పటాలు (Maps)
- దృశ్య సమాచారం (Infographics)
- డ్యాష్ బోర్డు (Dashboards)

డేటా విజువలైజేషన్ – ఉపయోగాలు

- డేటా విజువలైజేషన్ డేటాను సులభంగా విశ్లేషించడానికి మరియు వివరించడానికి వినియోగదారులకు ఉపయోగపడును.
- ఇది సంక్లిష్ట సమాచారాన్ని అర్థమయ్యేలా చేస్తుంది మరియు ఉపయోగిస్తుంది.

- విజువలైజేషన్‌లోని వివిధ చార్టులు ఒకటి లేదా అంతకంటే ఎక్కువ చరరాశుల కోసం డేటాలో సంబంధాన్ని చూపించడానికి సహాయపడుతుంది.

మీకు తెలుసా?

ఇన్ఫోగ్రాఫిక్ చేయండి → ఇన్ఫోగ్రాఫిక్(ఇన్ఫర్మేషన్ గ్రాఫిక్) అనేది గ్రాఫిక్ ఆకృతిలో సమాచారం యొక్క ప్రాతినిధ్యం.

డ్యాష్‌బోర్డు → డ్యాష్‌బోర్డ్ అనేది ఒక ఏకీకృత దృశ్యప్రదర్శనను సృష్టించడానికి సమావేశమైన వనరుల సమాహారం. డేటా విజువలైజేషన్లు మరియు డ్యాష్‌బోర్డ్‌లు సంక్లిష్టమైన ఆలోచనలు మరియు భావనలను సాధారణ దృశ్య ఆకృతిలోకి అనువదిస్తాయి. Textలో గుర్తించలేని నమూనాలు మరియు సంబంధాలు డ్యాషబోర్డ్ ఉపయోగించి ఒక చూపులో గుర్తించబడతాయి.

మ్యాట్‌ప్లాట్‌లిబ్ (Matplotlib) పరిచయం – పైథాన్‌లో డేటా విజువలైజేషన్

పైథాన్‌లో అత్యంత ప్రాచుర్యం పొందిన డేటా విజువలైజేషన్ లైబ్రరీ మ్యాట్‌ప్లాట్‌లిబ్ కోడ్ యొక్క కొన్ని పంక్తులలో చార్టులను సృష్టించడానికి మిమ్మల్ని అనుమతిస్తుంది.

Types of Visualizations in Matplotlib

మ్యాట్‌ప్లాట్‌లిబ్‌లో విజువలైజేషన్ల రకాలు మ్యాట్‌ప్లాట్‌లిబ్ క్రింద అనేక రకాలు విజువలైజేషన్‌లు ఉన్నాయి. వాటిలో కొన్ని :

- లైన్‌ప్లాట్ (Line plot)
- స్కాటర్ (Scatter plot)
- సోపాన చిత్రము (Histogram)
- బాక్స్‌ప్లాట్ (Box plot)
- బార్ చిత్రము (Bar chart)
- వృత్త లేఖాచిత్రము (Pie chart)

మీకు తెలుసా?

స్కాటర్‌ప్లాట్ : స్కాటర్ ప్లాట్ అనేది డేటాను పాయింట్ల సేకరణగా చూపించే ఒక రకమైన ప్లాట్. ఒక బిందువు యొక్క స్థానం దాని రెండు డైమన్సనాలో విలువపై ఆధారపడి ఉంటుంది. ఇక్కడ ప్రతి విలువ క్షితిజ సమాంతర లేదా నిలువు పరిమాణంపై ఉంటుంది.

బాక్స్‌ప్లాట్: బాక్స్‌ప్లాట్ అనేది ఐదు సంఖ్యల సారాంశం ఆధారంగా డేటా పంపిణీని ప్రదర్శించే ప్రామాణిక మార్గం; కనిష్ట, మొదటి క్వార్టైల్, మధ్యస్థ మూడవ క్వార్డైల్ మరియు గరిష్ట.

మ్యాట్‌ప్లాట్‌లిబ్‌ను (Matplotlib) ఇన్‌స్టాల్ చేస్తోంది.

మీరు (pip) ను ఉపయోగించి మ్యాట్‌ప్లాట్‌లిబ్‌ను ఇన్‌స్టాల్ చేయవచ్చు. (pip) పైప్ పైథాన్ ప్యాకేజీలను వ్యవస్థాపించడానికి నిర్వహణ సాఫ్ట్‌వేర్.

12th_CS_Telugu_Medium_Chapter_16.indd 315 28-03-2023 10:13:40

గమనిక

pip అనుబంధంలో ఇచ్చిన వివరణాత్మక విధానాలు

16.2 ప్రారంభించడం

మ్యాట్‌ప్లాట్‌లిబ్‌ను ఇన్‌స్టాల్ చేసిన తర్వాత, మ్యాట్‌ప్లాట్‌లిబ్‌ను దిగుమతి చేయడం ద్వారా కోడింగ్ ప్రారంభిస్తాము.

(import matplotlib.pyplot as plt)

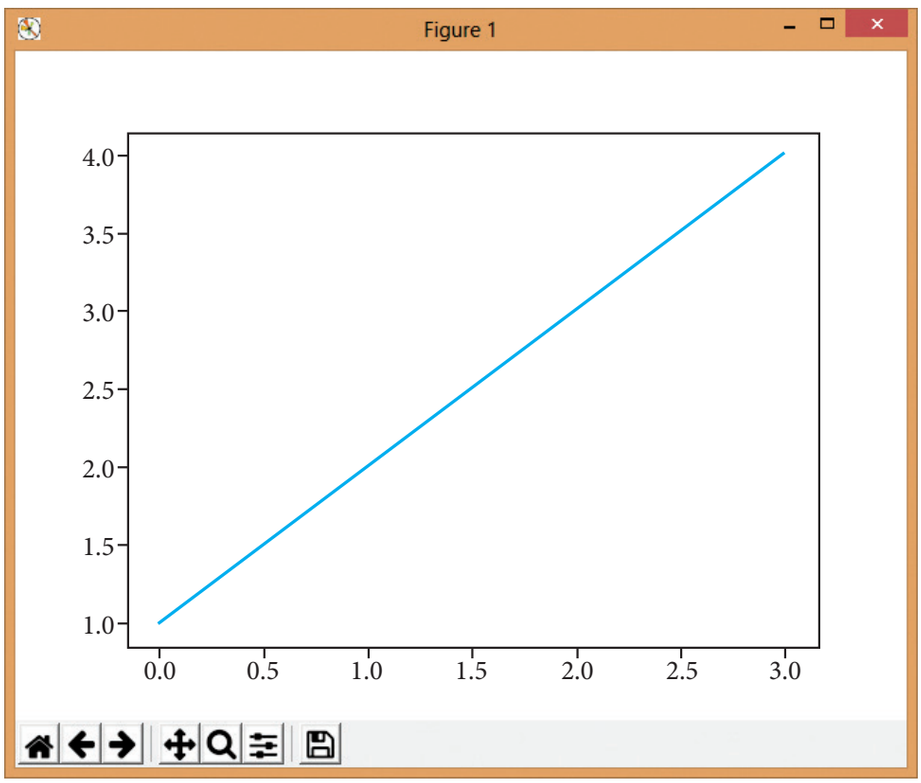
మీరు ప్లాట్లను ప్రదర్శించాలి. పైథాన్ స్క్రిప్ట్ నుండి మ్యాట్‌ప్లాట్‌లిబ్‌ను ఉపయోగించి, మీ ప్లాట్‌ను ప్రదర్శించడానికి మీరు పైల్ లోపల plt.show() పద్ధతిని జోడించాలి.

ఉదాహరణ

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.show()
```

అవుట్‌పుట్

This window is a matplotlib window, which allows you to see your graph. You can hover the graph and see the coordinates in the bottom right.



పటము 16.1

x-అక్షం 0–3 నుండి y-అక్షం 1–4 నుండి ఎందుకు ఉంటుంది అని మీరు ఆలోచిస్తూ ఉండవచ్చు. ప్లాట్ ( ) ఆదేశానికి మీరు ఒకే జాబితా లేదా శ్రేణిని అందిస్తే, మ్యాట్‌ప్లాట్‌లిబ్ ఇది y విలువల శ్రేణికి ఊహిస్తుంది మరియు మీకోసం x విలువలను స్వయంచాలకంగా ఉత్పత్తి చేస్తుంది. పైథాన్ పరిధులు 0తో మొదలవుతాయి. కాబట్టి డిఫాల్ట్ x వెక్టర్ y కి సమానమైన పొడవును కలిగి ఉంటుంది. కానీ 0 తో మొదలవుతుంది. అందువల్ల x డేటా [0, 1, 2, 3].

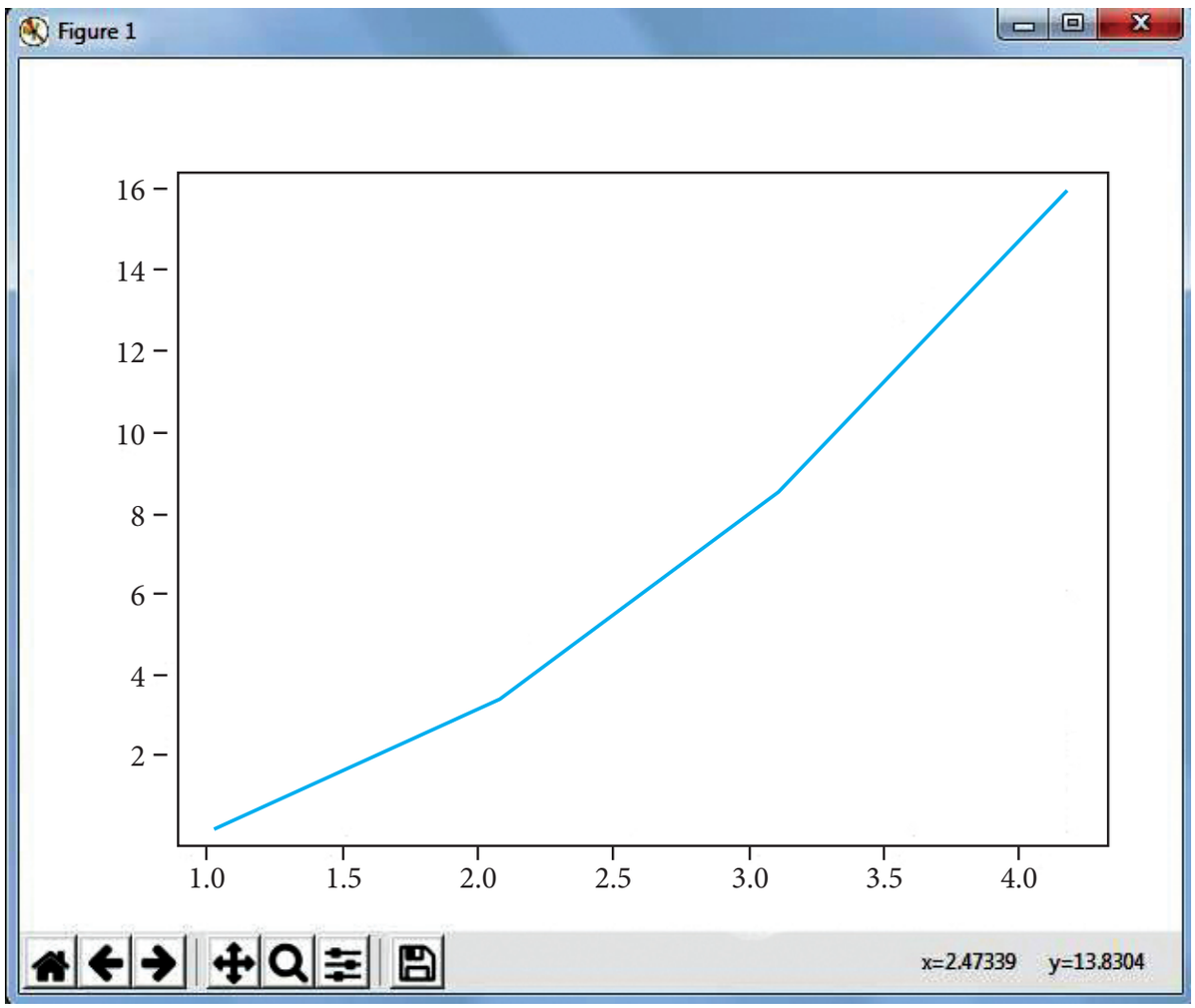
ప్లాట్ ( ) ఒక బహుముఖ ఆదేశం, మరియు ఏకపక్ష సంఖ్యలో వాదనలు పడుతుంది.

ప్రోగ్రామ్

For example, to plot x versus y, you can issue the command:

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16])
plt.show()
```

ఈ ప్లాట్ చాలా పారామితులను తీసుకుంటుంది. అయితే ఇక్కడ మొదటి రెండు 'x' మరియు 'y' నిరూపకాలు. దీని అర్థం ఈ జాబితాల ప్రకారం మీకు 4 కో-ఆర్డినేట్లు ఉన్నాయి: (1,1), (2,4), (3,9) మరియు (4,16).



పటము 16.2

12th_CS_Telugu_Medium_Chapter_16.indd 317 28-03-2023 10:13:40

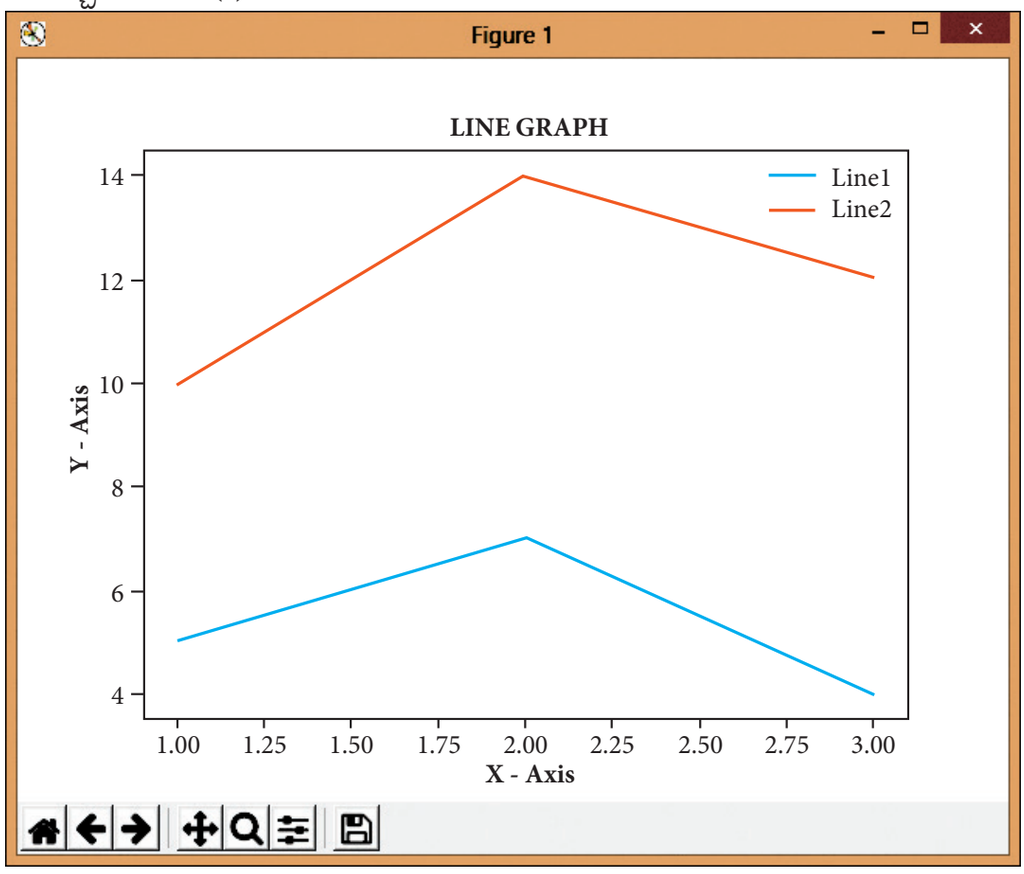
రెండు పంక్తులను ప్లాట్ చేయడానికి

రెండు పంక్తులను ప్లాట్ చేయడానికి, క్రింది కోడ్‌ను ఉపయోగించండి:

```
import matplotlib.pyplot as plt
    x = [1,2,3]
    y = [5,7,4]
    x2 = [1,2,3]
    y2 = [10,14,12]
    plt.plot(x, y, label='Line 1')
    plt.plot(x2, y2, label='Line 2')
    plt.xlabel('X-Axis')
    plt.ylabel('Y-Axis')
    plt.title('LINE GRAPH')
    plt.legend()
    plt.show()
```

అవుట్‌పుట్ (Output)

plt.xlabel మరియు plt.ylabel తో, మీరు సంబంధిత అక్షరానికి లేబుళ్ళను కేటాయించవచ్చు. తర్వాత మీరు ప్లాట్ యొక్క శీర్షికను plt.title తో కేటాయించవచ్చు. ఆ పై మీరు డిపాల్ట్ లెజెండ్‌ను plt. తో ప్రారంభించవచ్చు. లెజెండ్ ( )

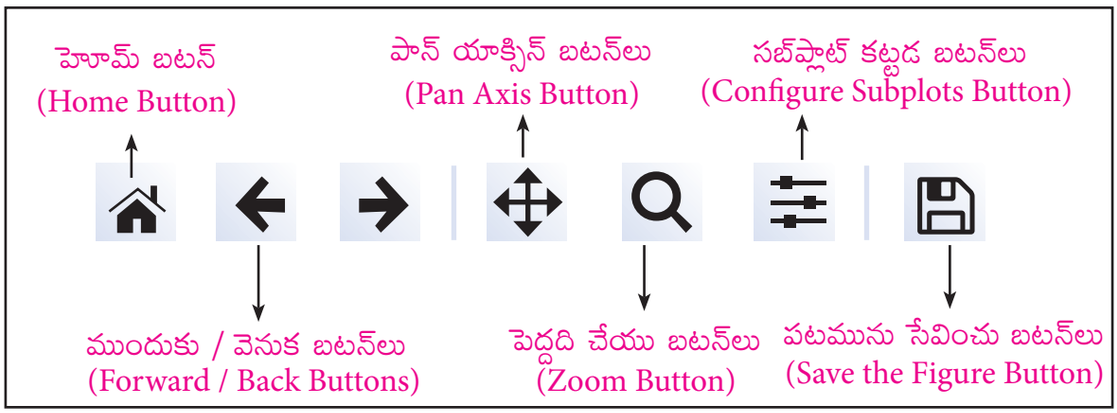


పటము 16.3

12th_CS_Telugu_Medium_Chapter_16.indd 318 28-03-2023 10:13:40

అవుట్‌పుట్‌లో ఉండే బటన్‌లు:

అవుట్‌పుట్ పటంలో మీరు కొన్ని బటన్‌లను ఎడమవైపు చివర చూడవచ్చు. మనము వాటి ఉపయోగాలు తెలుసుకుందాం.



పటము 16.4

హోమ్ బటన్ (Home Button) → మీరు మీ చార్టుని నావిగేట్ చేయడం ప్రారంభించిన తర్వాత హోమ్ బటన్ సహాయం చేస్తుంది. మీరు ఎప్పుడైన అసలు వీక్షణకు తిరిగి రావాలనుకుంటే మీరు దీనిపై క్లిక్ చేయవచ్చు.
ముందు / వెనుక బటన్‌లు (Forward/Back buttons) → ఈ బటన్‌లను మీ బ్రౌజర్‌లోని పార్వర్డ్ / బ్యాక్ బటన్లు వలె ఉపయోగించవచ్చు. మీరు ఉన్న మునుపటి స్థానానికి తిరిగి వెళ్ళడానికి లేదా మళ్ళీ పార్వార్డ్ చేయడానికి వీటిని క్లిక్ చేయవచ్చు.
పాన్ యాక్సిస్ బటన్‌లు (Pan Axis) → ఈ (+) క్రాస్ లుకింగ్ బటన్ దాన్ని క్లిక్ చేయడానికి మిమ్మల్ని అనుమతిస్తుంది. ఆ పై మీ గ్రాఫ్‌ను క్లిక్ చేసి లాగండి.
జూమ్ (Zoom) → జూమ్ బటన్ దానిపై క్లిక్ చేసి ఆపై మీరు ప్రత్యేకంగా జూమ్ చేయాలనుకుంటున్న స్క్వేర్‌ను క్లిక్ చేసి లాగండి. జూమ్ చేయడానికి ఎడమ క్లిక్ మరియు లాగడం అవసరం. మీరు ప్రత్యామ్నాయంగా కుడి క్లిక్‌తో జూమ్ అవుట్ చేసి లాగవచ్చు.
సబ్‌ప్లాట్‌లను కాన్ఫిగర్ చేయుట (Configure Subplots) → సబ్‌ప్లాట్‌లను కాన్ఫిగర్ చేయండి. ఈ బటన్ మీ ఫిగర్ మరియు ప్లాట్‌తో వివిధ అంతరాల ఎంపికలను ఆకృతీకరించుటకు మిమ్మల్ని అనుమతిస్తుంది.
ఆకృతిని సేవ్ చేయుట (Save Figure) → ఈ బటన్ మీ ఫిగర్‌ను వివిధ రూపాల్లో సేవ్ చేయడానికి అనుమతిస్తుంది.

16.3 ప్రత్యేక రేఖాచిత్ర (Plot) రకములు (Special Plot Types)

హిస్టోగ్రామ్స్ మరియు స్కాటర్ ప్లాట్ నుండి కమ్మి చిత్రాలు (Bar graphs) మరియు బార్ చార్ట్‌ల వరకు వివిధ రకాల ప్లాట్లను సృష్టించడానికి మ్యాట్‌ప్లాట్‌లిబ్ మిమ్మల్ని అనుమతిస్తుంది.

రేఖా పటము (Line Chart)

లైన్ చార్ట్ అనునది ఒక రకమైన చార్ట్, ఇది సరళరేఖ విభాగాలతో అనుసందానించబడిన 'మార్కర్స్'అని పిలువబడే డేటా పాయింట్ల శ్రేణిగా సమాచారాన్ని ప్రదర్శిస్తుంది. సమయ వ్యవధిలో సమయశ్రేణిలో డేటాలోని ధోరణిని దృశ్యమానం చేయడానికి ఒక లైన్ చార్ట్ తరుచుగా ఉపయోగించబడుతుంది. అందువలన పంక్తి తరుచుగా కాలక్రమానుసారం డ్రా అవుతుంది.

12th_CS_Telugu_Medium_Chapter_16.indd 319 28-03-2023 10:13:40

ఉదాహరణ : లైన్ ప్లాట్ (Line plot)

```
import matplotlib.pyplot as plt
years = [2014, 2015, 2016, 2017, 2018]
total_populations = [8939007, 8954518, 8960387, 8956741, 8943721]
plt.plot (years, total_populations)
plt.title ("Year vs Population in India")
plt.xlabel ("Year")
plt.ylabel ("Total Population")
plt.show()
```

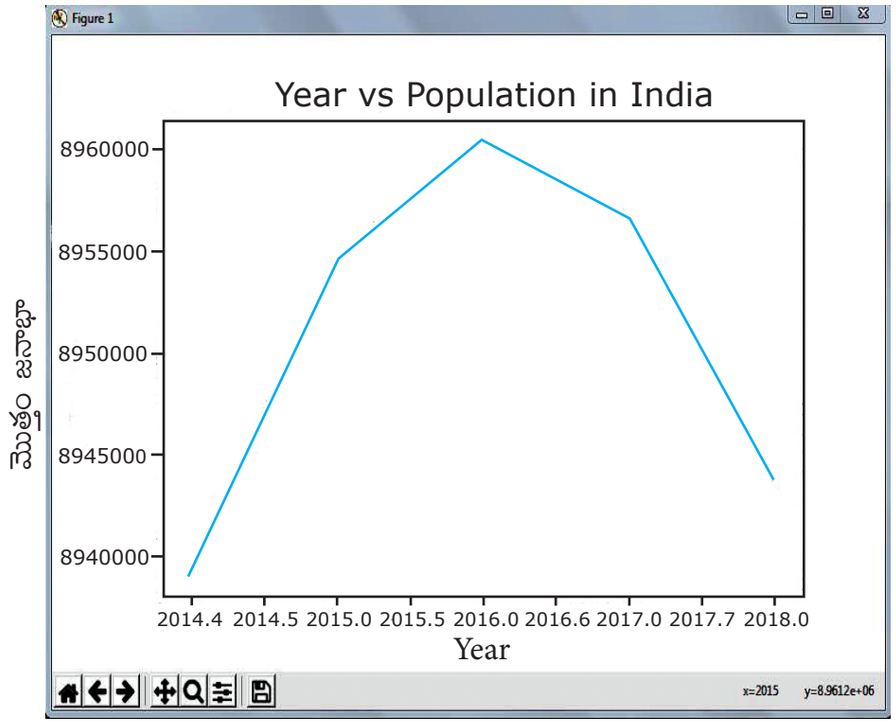
ఈ ప్రోగ్రామ్‌లో

Plt.title() → రేఖాచిత్రపు పేరు(శీర్షిక)ను సూచించును

Plt.xlabel() → X-అక్షమును తెలియజేయును

Plt.ylabel() → Y-అక్షపు పేరును తెలియజేయును.

అవుట్‌పుట్



పటము *16.5*

బార్ పటాలు(Bar Chart)

బార్ చార్టు ప్లాట్ (లేదా బార్ చార్ట్) అనేది చాలా సాధారణమైన ప్లాట్లలో ఒకటి. ఇది సంఖ్యాపరమైన చరరాశుల, వర్గీకరణ చలరాశుల మధ్యసంబంధాన్ని చూపుతుంది.

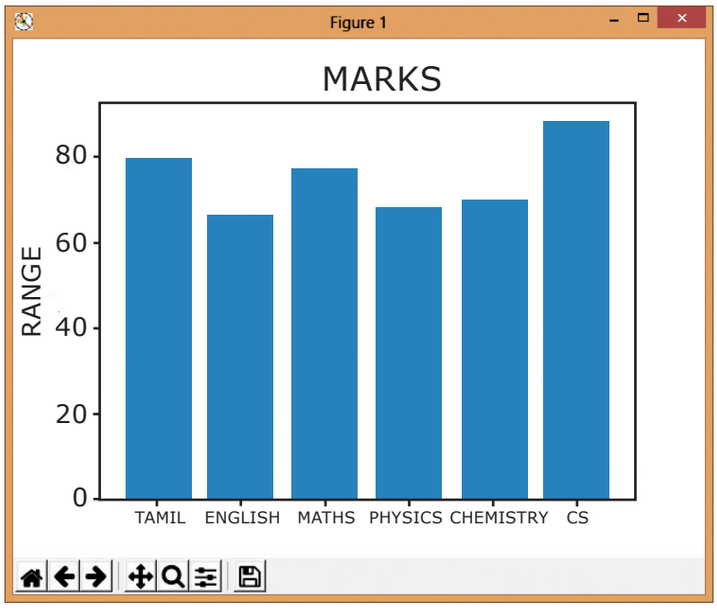
బార్‌చార్టు దీర్ఘచతురస్రాకార బార్లతో వర్గీకరణ డేటాను సూచిస్తుంది. ప్రతి బార్‌కు (కమ్మీ రేఖకు) ఒక ఎత్తు ఉంటుంది. అది సూచించే విలువకు అనుగుణంగా ఉంటుంది. బార్లును నిలువుగా లేదా అడ్డంగా ప్లాట్ చేయవచ్చును. మనం ఎప్పుడైనా సంఖ్యాపరమైన విలువలను పోల్చవలసినపుడు ఇది ఉపయోగంగా ఉండును. మనం మ్యాట్‌ప్లాట్‌లిబ్ తో బార్ చార్టు చేయడానికి మనం plt.bar() ఫంక్షన్‌ను ఉపయోగించవచ్చు.

12th_CS_Telugu_Medium_Chapter_16.indd 320 28-03-2023 10:13:40

ఉదాహరణ

```
import matplotlib.pyplot as plt
# Our data
labels = ["TAMIL", "ENGLISH", "MATHS", "PHYSICS", "CHEMISTRY", "CS"]
usage = [79.8, 67.3, 77.8, 68.4, 70.2, 88.5]
# Generating the y positions. Later, we'll use them to replace them with labels.
y_positions = range (len(labels))
# Creating our bar plot
plt.bar (y_positions, usage)
plt.xticks (y_positions, labels)
plt.ylabel ("RANGE")
plt.title ("MARKS")
plt.show()
```

అవుట్‌పుట్



పటము 16.6

పైన చూపించిన క్రింది విషయములను సూచించును.

Labels → కమ్మీరేఖా (బార్) చిత్రపు శీర్షికను తెలియజేయును.

Usgae → కమ్మీరేఖా (బార్) చిత్రపు విలువను తెలియజేయును.

Xticks → అక్షములో టిక్ గుర్తు నియమిత విలువను చూపించును. తర్వాత ఒక్కొక్క టిక్ చూపించే పేరును గుర్తించును.

Range → సంఖ్య క్రమమును తయారగును.

12th_CS_Telugu_Medium_Chapter_16.indd 321 28-03-2023 10:13:40

మీకు తెలుసా?

కమ్మీ రేఖా చిత్రములు (Bar Graph) మరియు (Histogram) సోపాన చిత్రము రెండు విధానాలలో సమాచార వివరణను పటం రూపంలో చూపించవచ్చు.

హిస్టోగ్రామ్ మరియు బార్‌గ్రాప్ మధ్య కీలక తేడాలు :

హిస్టోగ్రామ్ మరియు బార్‌చార్టుల మధ్య తేడాలు క్రింది విధంగా ప్రస్తావించవచ్చు.

1. హిస్టోగ్రామ్ గ్రాఫికల్ ప్రాతినిధ్యాన్ని సూచిస్తుంది. ఇది బార్ల ద్వారా డేటాను ప్రదర్శిస్తుంది. కమ్మీ (బార్‌గ్రాఫ్) రేఖాచిత్రము నమూనా అది ఇచ్చిన వివరణలోని వివిధ రకాల విషయాలను సూచిస్తుంది.
2. హిస్టోగ్రామ్ (సోపాన చిత్రము) అనునది నిరంతర పౌన:పున్య విభజన చరరాశి విలువలను సూచిస్తుంది. బార్‌గ్రాఫ్ అనునది వియుక్త వేరియబుల్స్ వివిధ రకాల బార్ చిత్రాల చరరాశుల విలువలను పోల్చి చూపించును.
3. హిస్టోగ్రామ్ సంఖ్య డేటాను సూచిస్తుంది. అయితే బార్‌గ్రాఫ్ వర్గీకరణను సూచిస్తుంది.
4. హిస్టోగ్రామునందు మధ్య అంతరములు ఉండవు కాని బార్‌గ్రాఫ్‌ల మధ్య స్థిరమైన అంతరములు సూచించబడును.
5. హిస్టోగ్రామ్ యొక్క అంశాలు డేటా శ్రేణులను సూచించడానికి సంఖ్యలుతో కలిసి వర్గీకరించబడును. కాని బార్‌గ్రాఫ్‌లో వివరణలు వ్యక్తిగత వివరణలు తెలిపే విధంగా ఉంటాయి.
6. బార్‌గ్రాఫ్‌లోని వివరణలను ఎగువశ్రేణి నుండి దిగువ శ్రేణిలోని విలువలను మార్పుచేయుటకు వీలుపడుతుంది. కాని హిస్టోగ్రాము నందు వివరాలు వరుసలో ఉండుట వలన ఇది వీలు పడదు.
7. హిస్టోగ్రామ్ నందు చతురస్రాకార దిమ్మెలు ఒకే కొలతతో ఉండలేక పోవచ్చు కాని కమ్మీరేఖా చిత్రాలయందు వెడల్పులు ఒకే విధంగా ఉంటాయి.

వృత్తరేఖా చిత్రము (Pie Chart)

బహుశా వృత్తరేఖాచిత్రము అనునది చాలా సాధారణమైన చార్టులలో ఒకటి. ఇది వృత్తాకార విచిత్రము ఇది సంఖ్యా నిష్పత్తిని వివరించడానికి భాగాలుగా విభజించబడుతుంది. వృత్తకార పటంలోని ఈ భాగాలు ఆవృత్తకారపు విలువలుగా పరిగణించిబడును.

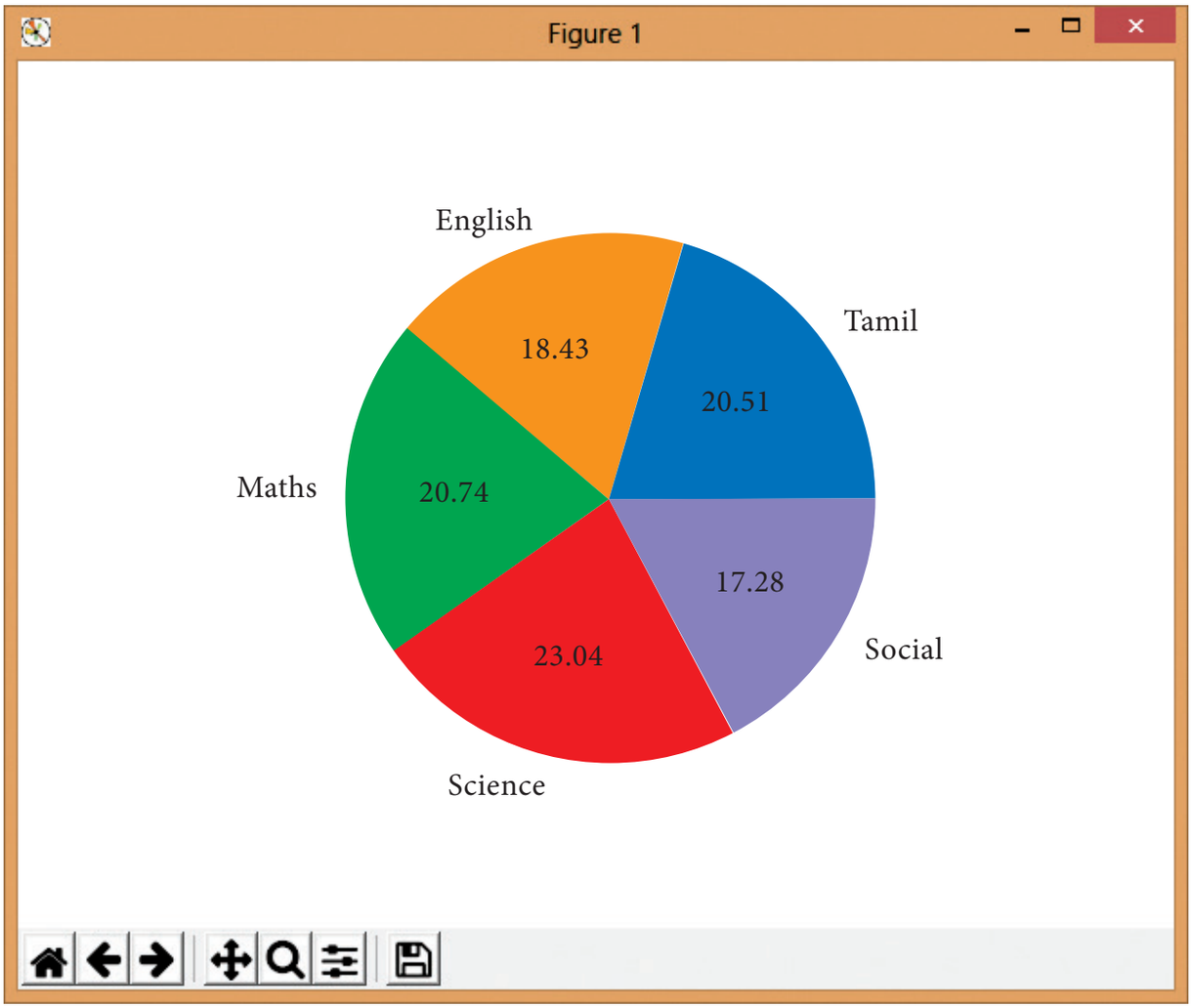
Matplotlib ను ఉపయోగించి వృత్తరేఖాచిత్రం తయారికి మనం ***plt.pie()*** అను ఫంక్షన్‌ను ఉపయోగిస్తాము. autopct పరిమితిని ఉపయోగించి ఆ పటం విలువలను శాతములో తెలుపును.

ఉదాహరణ

```
import matplotlib.pyplot as plt
sizes = [89, 80, 90, 100, 75]
labels = ["Tamil", "English", "Maths", "Science", "Social"]
plt.pie (sizes, labels = labels, autopct = "%.2f")
```

```
plt.axes().set_aspect ("equal")
plt.show()
```

అవుట్‌పుట్ (Output)



పటము 16.7

ప్రయత్నించి చూచుట

1. శీర్షిక *x* మరియు *y* అక్షములతో కూడిన రేఖాచిత్రమును తయారుచేయండి.
2. ఇటీవల జరిగిన పరీక్షలో మీ మార్కుల జాబితాను తీసుకొని ఒక వృత్త రేఖా చిత్రాన్ని తయారుచేయండి?
3. గత 10 సంవత్సరాలలో 12వ తరగతి విద్యార్థుల కంప్యూటర్ సైన్స్‌లో విద్యా పనితీరుపై వివరణను ఇచ్చు రేఖాపటంను తయారుచేయండి?
4. ఒక వారంలో గల కంప్యూటర్ సైన్స్ పిరియడ్స్‌ను బార్‌చార్ట్ ద్వారా చూపించండి?

12th_CS_Telugu_Medium_Chapter_16.indd 323 28-03-2023 10:13:40

మూల్యాంకనం

భాగము - I

సరియైన జవాబును ఎన్నుకొనుము: **(1 మార్కులు)**

1. 2D గ్రాఫిక్స్ కోసం ఉపయోగించే పైథాన్ ప్యాకేజ్ ఏది?

   a. matplotlib.pyplot b. matplotlib.pip

   c. matplotlib.numpy d. matplotlib.plt

2. ఈ క్రింది వానిలో పైథాన్ ప్యాకేజ్‌ని నిర్వహించే మాడ్యుల్స్ కొరకు ఉపయోగించు నది?

   a. Matplotlib b. PIP

   c. plt.show() d. python package

3. ఈ క్రింది కోడ్‌ను చదివండి. ఈ కోడ్ యొక్క ఉపయోగం మరియు సరైన సమాధానంను ఎన్నుకొనుము.

   C:\Users\YourName\AppData\Local\Programs\Python\Python36-32\Scripts>pip –version

   a. Check if PIP is Installed b. Install PIP

   c. Download a Package d. Check PIP version

4. ఈ క్రింది కోడ్‌ను చదవండి. ఈ కోడ్ యొక్క ఉపయోగం మరియు సరైన సమాధానంను ఎన్నుకొనుము.

   C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip list

   a. List installed packages b. list command

   c. Install PIP d. packages installed

5. డేటా మరియు సమాచారాన్ని దృశ్యమానం చేయడానికి క్రింది మాడ్యులోలో ఏది దిగుమతి చేయబడాలి.

   a. csv b. getopt

   c. mysql d. matplotlib

6. ..... అనేది ఒక రకమైన చార్ట్ ఇది కనిక్ట్ చేయబడిన డేటా పాయింట్ల శ్రేణిగా సమాచారన్ని పదర్శిస్తుంది.

   a. csv b. Pie chart

   c. Bar chart d. All the above

7. క్రింద ఇచ్చిన కోడ్‌ను చదవండి.

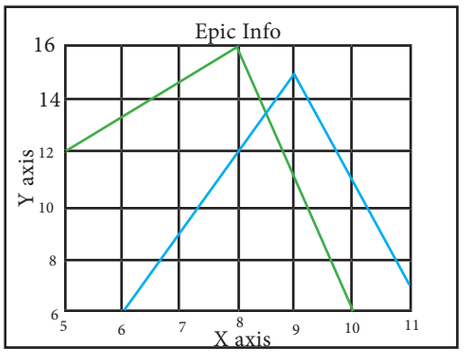
a. import matplotlib.pyplot as plt

b. plt.plot(3,2)

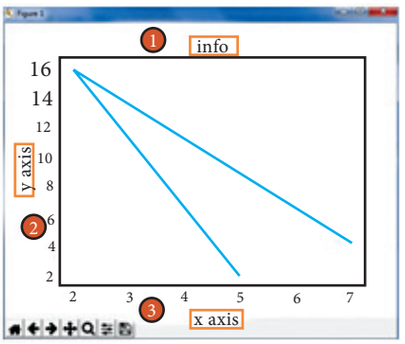
c. plt.show()

పైన ఇచ్చిన కోడింగ్‌లో ఔట్‌పుట్‌ను గుర్తించండి.

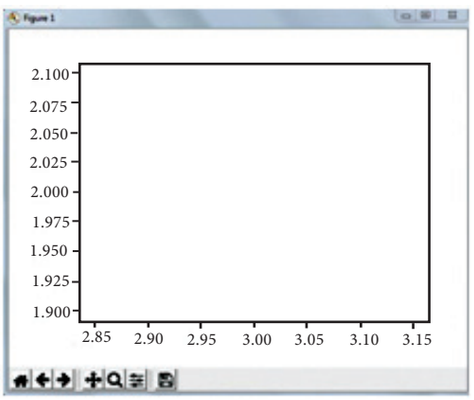
a.



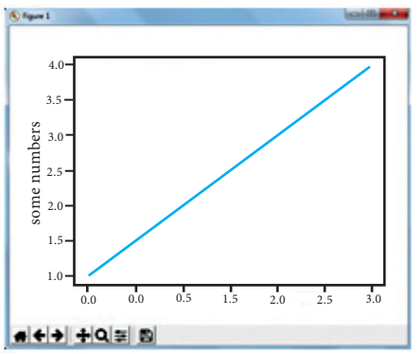
b.



c.



d.



8. ఈ క్రింది వివరణలను ఆధారంగా సరియైన చార్టును ఎన్నుకొండి.

సూచన 1 : ఈ చార్టు తరచుగా సమయ వ్యవధిలో డేటాలోని దోరణిని దృశ్యమానం చేయడానికి ఉపయోగించబడుతుంది.

సూచన 2 : ఈ రకమైన చార్టులోని రేఖ తరచుగా కాలక్రమానుసారు (chronologically) గీస్తారు.

a. Line chart b. Bar chart c. Pie chart d. Scatter plot

9. క్రింది ప్రవచనమును చదవండి. ఇచ్చిన వృత్తరేఖాచిత్రంలోని సరైన సమాధానాన్ని ఎన్నుకొనుము.

ప్రవచనము A: plt.pie() ప్రమేయాన్ని ఉపయోగించి Matplotlib లో వృత్తరేఖా పటమును గీయవచ్చు.

ప్రవచనము B: పైథాన్ స్ట్రింగ్ ఫార్మాటింగ్‌ను ఉపయోగించి శాతము విలువను ప్రదర్శించడానికి autopct పరామితి అనుమతిస్తుంది.

a. ప్రవచనము A సరి b. ప్రవచనము B సరి

c. రెండు ప్రవచనములు సత్యం d. రెండు ప్రవచనాలు అసత్యం

12th_CS_Telugu_Medium_Chapter_16.indd 325 28-03-2023 10:13:41

భాగము - II

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(2 మార్కులు)**

1. సమాచార దృశీకరణము (Data visualization) అనగానేను?
2. సమాచార దృశీకరణములోని సాధారణ రకములను పట్టీకరించండి.
3. మ్యాట్‌ప్లాట్‌లిబ్ (Matplotlib) లోని దృశీకరణ విధానములను పట్టీకరించండి.
4. మ్యాట్‌ప్లాట్‌లిబ్‌ను ఇన్‌స్టాల్ చేయడం ఎలా?
5. క్రింది ప్రమేయములలోని తేడాలను వ్రాయండి: plt.plot([1,2,3,4]), plt.plot([1,2,3,4], [1,4,9,16]).

భాగము - III

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(3 మార్కులు)**

1. క్రింద ఇవ్వబడిన ఔట్‌పుట్ సమాచార దృశీకరణకు ప్లాట్‌ను గీయండి.

```
import matplotlib.pyplot as plt
plt.bar([1,3,5,7,9],[5,2,7,8,2], label="Example one")
plt.bar([2,4,6,8,10],[8,6,2,5,6], label="Example two", color='g')
plt.legend()
plt.xlabel('bar number')
plt.ylabel('bar height')
plt.title('Epic Graph\nAnother Line! Whoa')
plt.show()
```

2. సమాచార దృశీకరణం (డేటా విజువలైజేషన్) యొక్క ఏదేని మూడు ఉపయోగాలు వ్రాయండి.
3. ఈ క్రింది వాటికి కోడింగ్‌ను వ్రాయండి.

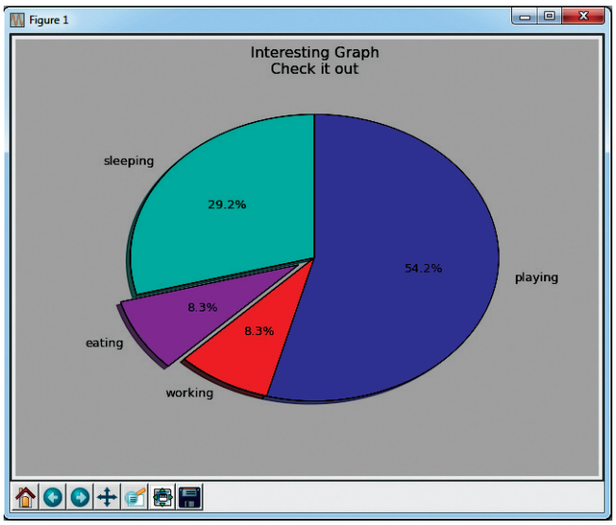
   a. మీ కంప్యూటర్‌లో PIP ను ఇన్‌స్టాల్ చేయడానికి

   b. మీ యొక్క కంప్యూటర్‌లో ఇన్‌స్టాల్ చేయబడిన PIP యొక్క వెర్షన్‌ను తనిఖీ చేయుము.

   c. మీ సిస్టమ్‌లోని ప్యాకేజీలను జాబితా చేయడానికి

12th_CS_Telugu_Medium_Chapter_16.indd 326 28-03-2023 10:13:41

4. ఈ క్రింది ఇవ్వబడిన వృత్తరేఖా చిత్రము యొక్క ప్లాట్‌కు ఔట్‌పుట్‌ను వ్రాయండి.



భాగము - IV

క్రింది ప్రశ్నలకు సమాధానములను తెలుపుము : **(5 మార్కులు)**

1. Matplotlib ను ఉపయోగించి pyplots యొక్క రకాలను సవివరంగా వివరించుము.
2. Matplotlib lib window నందు ఉపయోగించు వివిధ రకాలైన buttons వివరించుము.
3. ఈ క్రింది ప్రమేయముల ఉపయోగాలను వివరించుము.
   a. plt.xlabel
   b. plt.ylabel
   c. plt.title
   d. plt.legend()
   e. plt.show()

Reference

1. https://towards datascience.com / data - science - with - python - intro -to- data -visualization-and-matplotlib-5f799b7c6d82.
2. https://heartbeat.fritz.ai/introduction-to-matplotlib-data-visualization-in-python-d9143287ae39.
3. https://python programming.net / legends - titles - labels - matplotlib - tutorial/?completed=/matplotlib-intro-tutorial/.
4. https://keydifferences.com/difference-between-histogram-and-bar-graph.html.

12th_CS_Telugu_Medium_Chapter_16.indd 327 28-03-2023 10:13:41



పదకోశం

పరభాష (Terminology)	అర్థాలు (Meaning)
Access control	కంప్యూటింగ్ వాతావరణంలో వనరులను ఎవరు లేదా ఏమి చూడవచ్చు లేదా ఉపయోగించవచ్చు నియంత్రించే భాద్రత సాంకేతికత.
Access modifiers	ప్రైవేట్, రక్షిత మరియు పబ్లిక్.
Alternative	రెండు ఎంపికలలో ఒకటి.
append()	జాబితాలో ఒక మూలకాన్ని చేర్చడానికి ఉపయోగిస్తారు.
Argument	ఈ వేరియబుల్ యొక్క వాస్తవ విలువ ఆర్గ్‌మెంట్ ఫంక్షన్‌కు ఆమోదించింది.
argv	కమాండ్ లైన్ ఆర్గ్యుమెంట్ ద్వారా పంపబడిన విలువలను కలిగి ఉన్న శ్రేణి.
Attribute	ఒక వస్తువును తయారుచేసే డేటా అంశాలు.
Authorization	అనుమతి ఇవ్వబడడం లేదా యాక్సస్ ఇవ్వడం.
Block	ప్రకటనల సమితి
Boolean	తార్కికం అని అర్థం.
break	నియంత్రణ ఉనికిలో ఉంటుంది.
c = sqlite3.connect('test. db')	***SQLite***డేటాబేస్ ***'test.db'.***కు డేటాబేస్ కనెక్షన్ను సృష్టించండి. మీరు ప్రత్యేక పేరు ***RAM*** : మెమరీ లో డేటాబేస్ సృష్టించడానికి కూడా సరఫరా చేయవచ్చు.
c.close()	డేటాబేస్ యొక్క కనెక్షన్ను విడుదల చేయడానికి
c.commit()	పట్టికలో చేసిన మార్పులను (***save***) సేవ్ చేయడానికి
c.execute()	అన్ని ***SQL*** ఆదేశాన్ని అమలు చేయండి. రెండు రకాల ***placeholder*** లను అంగీకరించండి ప్రశ్న మార్క్ (“మార్కెట్ శైలి”) మరియు పేరున్న ***placeholder*** : పేరు (‘పేరుశైలి’)
Cartesian product	రెండు సంబందాల నుండి నిలువు వరుసలను విలీనం చేయడానికి కార్డీషియన్ ఆపరేషన్ సహాయపడుతుంది.
cd	***cd command*** డైరెక్టర్‌ని మార్చడానికి సూచిస్తుంది.
Class	వస్తువులను సృష్టించే డెంప్లేట్
Class variable	తరగతి లోపల ప్రకటించిన సాధారణ వేరియబుల్
cls	కమాండ్ విండోలో స్క్రిన్ క్లియర్ చేయడానికి

12th_CS_Telugu_Medium_Glossary.indd 328 30-03-2023 13:11:03

Comma(,)	***csv*** ఫైల్‌లో ప్రతి డేటాను వేరు చేయడానికి ***Comma*** ఉపయోగపడును.
compiler	మొత్తం ప్రోగ్రామ్‌ను స్కాన్ చేసి మొత్తంగా మెషన్ కోడ్‌లోకి అనువదిస్తుంది. ఇది మొత్తం ప్రోగ్రామ్‌ను స్కాన్ చేసిన తర్వాతే దోష సందేశాన్ని ఉత్పత్తి చేస్తుంది. అందువల్ల డీబగ్ చాలా కష్టం.
Conjunction	సమ్మతి, యాదృశ్చికము
Constraint	పరిమితి లేదా పరిమితి
Constructor	ఒక ఆబ్జక్ట్ పరిధిలోకి ప్రవేశించినపుడు ఒక ప్రత్యేక ఫంక్షన్ స్వయంచాలకంగా అమలు అగును.
continue	మిగిలిన భాగంను దాటవేయడానికి మరియు తదుపరి పునరావృతంలో ప్రారంభించడానికి
CRUD	***Create, Read, Update*** మరియు ***Delete***
csv.reader()	***Reader*** ఫంక్షన్ ఫైల్ యొక్క ప్రతి పంక్తిని తీసుకొని అన్ని నిలువ వరుస జాబితాను రూపొందించడానికి తయారుచేయబడినది.
csv.register_dialect()	ఒక మాండలికం చదవలసిన ***csv*** ఫైల్ యొక్క ఆకృతిని వివరిస్తుంది.
CsvQuote All	కోటింగ్ అన్ని ***csvquote*** కు సెట్ చేయబడితే అపుడు ***writerow()*** అని
cur = c.cursor()	***field*** లను కోడ్ చేస్తుంది. ***cursor object*** ను తయారు చేయడం.
cur.fetchall()	సరిపోయే అడ్డువరుసల జాబితాను పొందడానికి
cur.fetchmany()	ఫలిత సమితి యొక్క తదుపరి సంఖ్యల వరుసలను ***(n)*** తిరిగి ఇచ్చే పద్ధతి
cur.fetchone()	ఒకే సరిపోలిన అడ్డువరుసను తిరిగి పొందడానికి
CWI	***Centrum Wiskunde & Informatica***
DBA	***DataBase Adminstrator***
DBMS	***Database Management System***
def	ఫంక్షన్‌ను నిర్వచించడానికి ***def keyword*** ను ఉపయోగిస్తారు.
Destructor	ఒక ***object*** దాని పరిధి నుండి నిష్క్రమించినపుడు ఒక ప్రత్యేక ఫంక్షన్ స్వయంచాలకంగా అమలు అగును.
dict()	***orderdict*** లేకుండా డేటాను డిక్షనరి ఫార్మట్‌లో ప్రింట్ చేయడానికి ఇది ఉపయోగపడుతుంది.
Dictionary	***Key*** విలువ జతల సేకరణ
DictReader()	***CSV*** యొక్క మొదటి పంక్తులో చదవడం ద్వారా మరియు ఈ పంక్తిలోని ప్రతి ***column*** కమాతో వేరు చేయబడిన విలువను నిఘంటువు ***key***గా ఉపయోగించడం ద్వారా పనిచేస్తుంది.
Dictwriter()	***Dictionary*** డేటాను వేరే ***CSV*** ఫైల్‌లో వ్రాయండి.

12th_CS_Telugu_Medium_Glossary.indd 329 30-03-2023 13:11:03

elif	***else…if***
Embedded	గట్టిగా జతచేయబడినవి.
Enter key	***csv*** ఫైల్‌లో అడ్డు వరుసలను సృష్టించడానికి ***Enter key*** లేదా ***newline*** ఉపయోగపడుతుంది.
eval()	ఈ ఫంక్షన్ స్ట్రింగ్ విలువను అంచనా వేయడానికి ఉపయోగించబడుతుంది.
Father of Relational Database	***Dr. Edgar Frank Codd***
g++	***c++*** ప్రోగ్రామ్‌ను కంపైల్ చేయడానికి కంపైలర్
GIS	***Geographic Information System***
global Scope	గ్లోబల్ స్కోప్‌తో కూడిన వేరియబుల్ ప్రోగ్రామ్‌లో ఎక్కడైనా ఉపయోగించవచ్చు.
Glue language	మీరు భాషలో పూర్తి అనువర్తనాన్ని వ్రాయరు, బదులుగా మీరు ఇతర (జిగురు భాష) భాషలలో (బహుశా చాలా భిన్నమైన) వ్రాసిన మాడ్యుళ్ళను ఆర్కెస్ట్రేట్ చేయడానికి (నిర్వహించడానికి) భాషను ఉపయోగిస్తున్నారు, తద్వారా అవి అప్లికేషన్‌ను రూపొందించడానికి కలిసి పనిచేస్తాయి. జిగురు భాష దీనిని సులభతరం చేస్తుంది. (అనుకూలమైన వాక్య నిర్మాణం, ఇంటర్ – ప్రాసెస్ కమ్యూనికేషన్ మరియు డేటా నిర్వహణ కోసం మంచి మద్దతు, సంకలనదశ మొదలగునవి)
id ()	ఇవ్వబడిన ఆబ్జెక్ట్ యొక్క మెమరీ అడ్రస్‌ను తిరిగి ఇస్తుంది.
IDLE	***Integrated Development Environment***
immutable	మార్చుటకు వీలుకానిది
Implementation	***interfac*** లో ప్రకటించిన ఆపరేషన్‌లను నిర్వహిస్తుంది.
import	పైథాన్‌లో ***Import C++ #header_file*** ఫైల్‌లో చేర్చండి. పైథాన్ గుణకాలు ***"import"*** స్టేట్‌మెంట్‌ను ఉపయోగించి ఫైల్ / ఫంక్షన్ ***import*** చేయడం ద్వారా మరొక మాడ్యుల్ నుండి కోడ్‌కు ప్రాప్యతను పొందవచ్చును.
Impure Functions	ఇంప్యూర్ ఫంక్షన్స్ దీని యొక్క వాదనలో ఒకదాని యొక్క అంతర్గత స్థితిని లేదా కొన్ని బాహ్య వేరియబుల్ యొక్క విలువ యొక్క విలువను మార్చే ఏదైనా ఒక ఫంక్షన్‌ను ఇంప్యూర్ ఫంక్షన్ అందురు.
Instantiation	ఇది ఒక ***object*** ను సృష్టించే ప్రక్రియ
Integrity	మొత్తము మరియు అవిభక్త ***(undivided)***
Interactive Mode	ప్రాంప్ట్ వద్ద కమాండ్ మరియు వ్యక్తీకరణలను టైప్ చేయడం ద్వారా పైథాన్ వ్యాఖ్యతను ఉపయోగించే మార్గం.
Interface	ఇంటర్‌ఫేస్ ఒక వస్తువు ఏమి చేయగలదో నిర్వచిస్తుంది, కాని వాస్తవానికి ఇది చేయదు.

12th_CS_Telugu_Medium_Glossary.indd 330 30-03-2023 13:11:04

interpreter		ప్రోగ్రామ్ ఒక సమయంలో ఒక ప్రకటనను అనువదిస్తుంది. మొదటి లోపం నెరవేరే వరకు ఇది ప్రోగ్రామ్‌ను అనువదించడం కొనసాగిస్తుంది, ఈ సందర్భంలో అది ఆగిపోతుంది. అందువలన ***debugging*** సులభం.
Intersection		ఖండన ***A*** మరియు ***B*** రెండింటిలో ఉన్న అన్ని ***tuple*** ల సమితి కలిగి ఉన్న ఒక సంబంధాన్ని నిర్వచిస్తుంది.
Key		ఇది నిఘంటువులో విలువకు మ్యాప్ చేయబడింది.
lambda		***Lambda*** ఫంక్షన్‌ను ఎక్కువుగా చిన్న మరియు ***one- time*** అనామిక్ ఫంక్షన్ సృష్టించడానికి ఉపయోగిస్తారు.
LEGB rule		***Local → Enclosed → Global → Built-in scope*** **స్థానిక → పరివేష్టిత → గ్లోబల్ → అంతర్నిర్మిత పరిధి**
List		***Mutable*** ఆర్డర్ చేసిన విలువల సేకరణ
local Scope		ఫంక్షన్ యొక్క ***body*** లోపల లేదా ***block*** లో ప్రకటించబడిన వేరియబుల్‌ను స్థానిక స్కోప్ అంటారు.
Looping		పునరావృతం
Mapping		వేరియబుల్ పేరును ఒక ***object*** తో బందించే ప్రక్రియ
Method		ఒక తరగతి లోపల ప్రకటించిన లేదా నిర్వహించబడిన ఫంక్షన్
module		మాడ్యుల్ ఫైథాన్ నిర్వచనాలు మరియు స్టేట్‌మెంట్‌లను కలిగి ఉన్న ఫైల్. ఫైల్ పేరు ***py*** అనుబంధంతో ఉన్న మాడ్యుల్ పేరు, మాడ్యుల్ లోపల మాడ్యుల్‌లో పేరు గ్లోబల్ వేరియబుల్ పేరు యొక్క విలువగా లభిస్తుంది.
Namespaces		***objects*** లకు వేరియబుల్స్ పేర్లును మ్యాపింగ్ చేయడానికి
Nested Block		కంటైనర్‌లు ఒక బ్లాక్‌లోని బ్లాక్‌ను నెస్‌డెడ్ బ్లాక్ అంటారు.
next()		తదుపరి ( ) ఫంక్షన్ తదుపరి అంశాన్ని ఇరేటర్ నుండి తిరిగి ఇస్తుంది. ***csv*** ఫైల్ కలెక్షన్ ఆఫ్ డేటా మరియు ఫంక్షన్‌ల వరుసను దాటవేయడానికి దీనిని ఉపయోగించవచ్చు.
Object		డేటా మరియు విధుల సేకరణ
Object Oriented Programming		వాస్తవ ప్రపంచ ఆధారంగా కంప్యూటర్ ప్రోగ్రామ్ యొక్క భావన
operator.itemgetter(col_ no)		ఒక ***csv*** ఫైల్ నుండి ఒకటి కంటే ఎక్కువ నిలువ వరుసల ద్వారా క్రమబద్ధీకరించడం.
os.system()		***system*** ఆదేశమును అమలుచేయడానికి ఉపయోగిస్తారు మరియు ఫైథాన్ ప్రోగ్రామ్‌లో ***c++*** ప్రోగ్రామ్‌ను ***g++*** గా ఉపయోగించి కంపైల్ చేయడానికి ఉపయోగిస్తారు.
Parameter		ఫంక్షన్ నిర్వచనం యొక్క డిక్లరేషన్‌లో పారామితి వేరియబుల్
parse		ఇన్‌పుట్‌ను సులభంగా నిల్వచేయగల లేదా మార్చగల డేటాను ముక్కలుగా విభజించడానికి

12th_CS_Telugu_Medium_Glossary.indd 331 30-03-2023 13:11:04

pass		ఫంక్షన్ మరియు లూప్‌లలో ***placeholder*** గా ఉపయోగించవచ్చు.
Projection (π)		***projection*** ఇన్‌పుట్ రిలేషన్ యొక్క అన్ని లక్షణాలను తొలగిస్తుంది.
Prompt		జాబితాలో పేర్కొన్న వాటిని అక్షర (<<<) వినియోగదారు నుండి ఇన్‌పుట్ తీసుకోవడానికి సిద్ధంగా ఉందని సూచించడానికి వ్యాఖ్యాత ప్రదర్శిస్తుంది.
Pure Functions		***Pure functions*** ఎల్లప్పుడు అదే ఫలితాన్ని ఇస్తే అదే వాదనలు లోపలికి పంపబడుతుంది.
Python prompt		>>>
Queue		***Queue*** అనేది ఒక నైరూప్య (***abstract***) డేటా నిర్మాణం, ఇది స్టాక్స్‌తో సమానంగా ఉంటుంది. స్టాక్‌ల మాదిరిగా కాకుండా ***Queue*** దీని రెండు చివరలలో తెరిచి ఉంటుంది. డేటాను ***(enqueue)*** చొప్పించడానికి ఒక చివర ఎల్లప్పుడు ఉపయోగించబడుతుంది. మరియు మరొకటి డేటాను ***(dequeue)*** తొలగించడానికి ఉపయోగిస్తారు. ***Queue*** ఫస్ట్-ఇన్-ఫస్ట్-అవుట్ పద్ధతిని అనుసరిస్తుంది. అనగా మొదటి నిల్వచేసిన డేటా అంశం మొదటి అమలు చేయబడుతుంది.
RDBMS		***Relational Database Management System***
recursion		ఒక ఫంక్షన్ తనను తాను పిలిచినపుడు పునరావృతం అంటారు.
Redundant		డేటా యొక్క నకిలీ
Routines		నిత్య కృత్యాలను ఫంక్షన్‌లు లేదా పద్ధతులు అంటారు. ఫైథాన్‌లో దీనిని నిర్వచనం అని పిలుస్తారు.
Schema		నిర్మాణం లేదా నమూనా
Scope		ఒక ***program*** యొక్క ఒక భాగంలో వేరియబుల్స్, పారామితులు మరియు ఫంక్షన్‌ల యొక్క దృశ్యమానత అదే ప్రోగ్రామ్ యొక్క మరియొక భాగానికి.
Script		ఫైథాన్ ప్రోగ్రామ్ ఫైల్‌లో నిల్వచేయబడుతుంది.
Script Mode		***prompt*** వద్ద ఆదేశం మరియు వ్యక్తీకరణలను టైప్ చేయడం ద్వారా ఫైథాన్ వ్యాఖ్యాతను ఉపయోగించు మార్గం
Select (σ)		ఎంపిక యొక్క ఉపసమితిని ఎంచుకోవడానికి ***SELECT*** ఆపరేషన్ ఉపయోగపడుతుంది.
Selectors		డేటా రకము నుండి సమాచారమును తిరిగి పొందే విధులు
Sequential		ఒకదాని తర్వాత ఒకటి
Set difference(-)		(–) సూచిక దీనిని సూచిస్తుంది. ***A-B*** యొక్క ఫలితము ***A*** లో ఉన్న ***B*** లో లేని అన్ని టుపుల్స్‌ను కలిగి ఉన్న ఒక సంబంధం.

12th_CS_Telugu_Medium_Glossary.indd 332 30-03-2023 13:11:04

skipinitialspace=true		నిజమైనపుడు డెలీమీటర్‌ను అనుసరించే వైట్ ***space*** వెంటనే విస్మరించబడుతుంది. డీఫాల్ట్ అనునది తప్పు.
slicing		విభజన (***cut***)
Stack		ఒక స్టాక్ (కొన్ని సార్లు "పుష్-డాన్-స్టాక్" అని పిలుస్తారు) అనేది ఆర్డర్ చేసిన వస్తువుల సేకరణ, ఇక్కడ క్రొత్త వస్తువులను చేర్చడం మరియు ఉన్న వస్తువులను తీసివేయడం ఎల్లప్పుడు ఒకే చివరలో జరిగే వస్తువల ఆర్డర్. ఈ ముగింపును సాధారణంగా "టాప్" ఉన్న అని పిలుస్తారు. పై భాగానికి ఎదురుగా ఉన్న ముగింపును "బేస్" అని పిలుస్తారు. ఈ ఆడరింగ్ సూత్రంను కొన్ని సార్లు ***LIFO*** అని పిలుస్తారు. లాస్ట్-ఇన్‌ఫస్ట్-అవుట్
stride		ఒక పెద్ద మెట్టు (***long step***)
string		అక్షరాలు, సంఖ్యలు లేదా చిహ్నాల వరుసలు
subscript		ఒక సూచిక సంఖ్య
Syntax		ప్రోగ్రామ్ యొక్క నిర్మాణము
Syntax Error		అన్వయించడానికి అసాధ్యమైన ప్రోగ్రామ్‌లోని లోపము
Token		ప్రోగ్రామ్ యొక్క వాక్యనిర్మాణము యొక్క ప్రాథమిక అంశాలలో ఒకటి.
Tuple		ఇది మార్పులేని ***objects*** ల క్రమం. ***Tuples*** జాబితాల మాదిరిగానే సన్నివేశాలు. కుండలీకరణాలు ( ) మధ్యలో ***Tuples*** నిర్వచించబడ్డాయి.
Union operation(U)		యూనియన్ చిహ్నంగా చూచిస్తుంది ఇందులో అన్ని ***A*** లేదా ***B*** పట్టికలో ఉన్నాయి.
variable		విలువలను నిల్వచేయడానికి ***Memory Box***
writer ow()		ఫైల్‌లో ఒకే వరుస డేటాను వ్రాసే విధానం
writer ows()		ఒకే ఫైల్‌లో బహుళ వరుసల డేటాను వ్రాసే విధానం
FALSE		తార్కిక విలువ 0
TRUE		తార్కిక విలువ 1

12th_CS_Telugu_Medium_Glossary.indd 333 30-03-2023 13:11:04

అనుబంధం -1

పైథాన్ జాబితాలోని విధులు

1. అంతర్నిర్మిత విధులు

ఫంక్షన్ (విధులు)	వివరణ
abs()	సంఖ్య యొక్క సంపూర్ణ విలువను అందిస్తుంది.
all()	పునరుత్పాదకములోని Iterable అన్ని అంశాలు ప్రయత్నించినపుడు నిజమవుతుంది.
any()	పునరుత్పాదక మూలకము ఏదైన నిజమైతే తనిఖీ చేస్తుంది.
ascii ()	ముద్రించదగిన ప్రాతినిత్యాలను కలిగి ఉన్న స్ట్రింగ్‌ను అందిస్తుంది.
bin()	పూర్ణాంకమును బైనరీ స్ట్రింగ్‌కు మారుస్తుంది.
bool ()	విలువను బూలియన్‌గా మారుస్తుంది.
bytearray ()	ఇచ్చిన Boolean పరిమాణం యొక్క శ్రేణిని అందిస్తుంది.
bytes()	మార్పులేని Bytes ల ఆబ్జెక్ట్‌లను తిరిగి యిస్తుంది.
callable()	Immutable పిలవదగినదా అని తనిఖీ చేయునది
chr ()	పూర్ణాంకము నుండి అక్షరాన్ని అందిస్తుంది.
classmethod ()	ఇవ్వబడిన ఫంక్షన్ నుండి తరగతి పద్ధతిని అందిస్తుంది.
compile()	పైథాన్ కోడ్ Function ను అందిస్తుంది.
complex()	సంక్లిష్ట సంఖ్యను సృష్టించడం.
delattr ()	Object నుండి లక్షణాలను తొలగిస్తుంది.
dir()	Object లక్షణాలను అందించడానికి ప్రయత్నిస్తుంది.
divmod()	Tuple యొక్క భాగఫలము మరియు శేషంలను అందిస్తుంది.
enumerate()	గణన Object తిరిగి యిస్తుంది.
eval ()	ప్రోగ్రామ్‌లో పైథాన్ కోడ్‌ను నడుపుతుంది.
exec()	డైనమిక్‌గా సృష్టించిన ప్రోగ్రామ్‌ను అమలుచేస్తుంది.
filter()	నిజమైన మూలకాల నుండి ఇటరేటర్‌ను నిర్మస్తుంది.
float()	సంఖ్య స్ట్రింగ్ నుండి Floating పాయింట్ సంఖ్యను అందిస్తుంది.
format()	ఫార్మెట్ చేసిన రిటర్న్స్ విలువ యొక్క ప్రాతినిధ్యం
getattr ()	Object యొక్క పేరు పెట్టబడిన అక్షరం యొక్క విలువను తిరిగి యిస్తుంది.
globals()	ప్రస్తుత Global చిహ్నము టేబుల్ యొక్క నిఘంటువును అందిస్తుంది.
hasattr ()	వస్తువు లక్షణానికి పేరు పెట్టిందో లేదో తిరిగి యిస్తుంది.
hash()	Object యొక్క hash విలువని తిరిగి యిస్తుంది.
help()	అంతర్నిర్మిత సహాయ వ్యవస్థను ప్రారంభిస్తుంది.
hex()	పూర్ణాంకము నుండి హెక్సా డిసిమల్‌కు మారుస్తుంది.
id()	రిటర్న్స్ ఒక Object యొక్క గుర్తింపు
isinstance ()	ఒక Object తరగతి యొక్క ఉదాహరణ కాదా అని తనిఖీ చేస్తుంది.
issubclass()	ఒక Object తరగతి యొక్క ఉపవర్గం కాదా అని తనిఖీ చేస్తుంది.

12th_CS_Telugu_Medium_Annexure.indd 334 30-03-2023 13:10:49

iter ()	ఒక Subclass కోసం ఇటరేటర్‌ను తిరిగి యిస్తుంది.
len()	Object యొక్క పొడవును తిరిగి యిస్తుంది.
locals()	ప్రస్తుత స్థానిక గుర్తు పట్టిక నిఘంటువును అందిస్తుంది.
map()	ఫంక్షన్ వర్తిస్తుంది మరియు జాబితాను అందిస్తుంది.
max()	అతిపెద్ద మూలకాన్ని అందిస్తుంది.
memoryview()	వాదన యొక్క Memory వీక్షణను అందిస్తుంది.
min()	అతిచిన్న మూలకాన్ని అందిస్తుంది.
next()	ఇటరేటర్ నుండి తదుపరి మూలకాన్ని తిరిగి పొందుతుంది.
object()	లక్షణం లేని Object ను సృష్టిస్తుంది.
oct()	పూర్ణాంకమును అష్టానికి (Octal) మారుస్తుంది.
open()	ఫైల్ ఆబ్జెక్ట్‌ను అందిస్తుంది.
ord()	యూనికోడ్ అక్షరం కోసం యూనికోడ్ పాయింట్‌ను అందిస్తుంది.
pow()	X యొక్క ఘాతం Y గా తిరిగి ఇస్తుంది.
print()	ఇచ్చిన వస్తువును ముద్రిస్తుంది.
property()	ప్రాపర్టీ లక్షణాన్ని అందిస్తుంది.
range()	స్టార్ట్ మరియు స్టాప్ మధ్య పూర్ణాంకాల శ్రేణిని అందిస్తుంది.
repr ()	ఆబ్జెక్ట్ యొక్క ప్రింటబుల్ రెప్రజెంటేషన్ అందిస్తుంది.
reversed()	సీక్వెన్స్ యొక్క రిజర్వు చేసిన యుటిరేటర్ ను తిరిగి ఇస్తుంది.
round()	n డిజిట్ ప్లేసస్ నుండి ఫ్లోటింగ్ పాయింట్ నంబర్‌ను రౌండ్ చేస్తుంది.
set()	ఫైథాన్ సెట్‌ను తిరిగి ఇస్తుంది.
setattr ()	Object యొక్క అట్రిబ్యూట్ విలువలను సెట్ చేస్తుంది.
slice()	Range () ద్వారా పేర్కొనబడిన స్లైస్ ఆబ్జెక్ట్‌ను సృష్టిస్తుంది.
sorted()	ఇవ్వబడిన ఇటర్‌బుల్ నుండి sorted list ను అందిస్తుంది.
staticmethod()	ఫంక్షన్ నుండి స్టాటిక్ మెథడ్‌ను సృష్టిస్తుంది.
str ()	ఆబ్జెక్ట్ యొక్క ఇన్‌ఫార్మల్ రెప్రజంటేషన్‌ను అందిస్తుంది.
sum()	ఇటరబుల్ అంశాలను Add చేస్తుంది.
super()	పేరంట్ క్లాస్‌ను సూపర్ ద్వారా సూచించడానికి మిమ్మల్ని అందిస్తుంది.
type()	ఆబ్జెక్ట్ యొక్క రకాన్ని అందిస్తుంది.
vars()	క్లాస్ యొక్క dict - attribute ను అందిస్తుంది.
__import__()	అడ్వాన్స్‌డ్ ఫంక్షన్‌ను __import__ ద్వారా పిలువబడుతుంది.

2. స్ట్రింగ్ ఫంక్షన్స్

ఫంక్షన్	వివరణ
capitalize()	ఫస్ట్ క్యారక్టర్‌ను కేపిటల్ లెటర్‌లోనికి మారుస్తుంది.
casefold()	casefolded స్ట్రింగ్స్‌గా మారుస్తుంది.
center()	పేర్కొన్న అక్షరాలతో specified character pad స్ట్రింగ్స్

12th_CS_Telugu_Medium_Annexure.indd 335 30-03-2023 13:10:49

count()	స్ట్రింగ్‌లోని సబ్ స్ట్రింగ్‌ల సంఘటనలను అందిస్తుంది.
encode()	ఇవ్వబడిన స్ట్రింగ్ యొక్క ఎన్‌కోడ్ చేసిన స్ట్రింగ్‌ను అందిస్తుంది.
endswitch()	స్ట్రింగ్ Specified Suffix తో ముగుస్తుందో లేదో అని చెక్ చేస్తుంది.
expandtabs()	Tab క్యారెక్టర్‌ని స్పేస్‌తో భర్తీ చేయండి.
find()	Sub స్ట్రింగ్ యొక్క మొదటి సంఘటన యొక్క ఇండెక్స్‌ను అందిస్తుంది.
format()	మంచి అవుట్‌పుట్ కొరకు స్ట్రింగ్ ఫార్మట్
format_map ()	డిక్షనరిని ఉపయోగించి స్ట్రింగ్‌ను రూపొందిస్తుంది.
index()	సబ్‌స్ట్రింగ్ యొక్క ఇండెక్స్‌ను తిరిగి యిస్తుంది.
input()	స్ట్రింగ్ యొక్క లైన్‌ను చదువుతుంది మరియు తిరిగి యిస్తుంది.
int()	నంబర్ లేదా స్ట్రింగ్ నుండి ఇంటీజర్‌ను అందిస్తుంది.
isalnum()	Alphanumeric అక్షరాలను తనిఖీ చేస్తుంది.
isalpha()	అన్ని అక్షరాలు Alphabets గా ఉన్నాయో లేదో తనిఖీ చేస్తుంది.
isdecimal ()	డెసిమల్ క్యారెక్టర్స్‌ను తనిఖీ చేస్తుంది.
isdigit ()	డిజిటల్ క్యారెక్టర్స్‌ను తనిఖీ చేస్తుంది.
isidentifier()	చెల్లుబాటు Identifier కోసం తనిఖీ చేస్తుంది.
islower ()	స్ట్రింగ్‌లోని అన్ని Alphabets lower case అవుతుందా అని తనిఖీ చేస్తుంది.
isnumeric ()	Numeric క్యారెక్టర్స్‌ను తనిఖీ చేస్తుంది.
isprintable()	Printable క్యారెక్టర్స్‌ను తనిఖీ చేస్తుంది.
isspace()	వైట్‌స్పేస్ క్యారెక్టర్‌ను తనిఖీ చేస్తుంది.
istitle ()	Titlecased స్ట్రింగ్ కోసం తనిఖీ చేస్తుంది.
isupper()	అన్ని క్యారెక్టర్స్ uppercase క్యారెక్టర్స్ అయితే తిరిగివస్తుంది.
join ()	కన్ క్యాటినేటడ్ స్ట్రింగ్‌ను అందిస్తుంది.
ljust ()	ఇవ్వబడిన విడ్త్ సమర్థించబడిన ఎడమవైపుకు తిరిగి వస్తుంది.
lower()	లోయర్ కేస్ స్ట్రింగ్‌ను తిరిగి యిస్తుంది.
lstrip ()	లీడింగ్ క్యారెక్టర్స్‌ను తొలగిస్తుంది.
maketrans()	ట్రాన్స్‌లేషన్ టేబుల్‌ను తిరిగి యిస్తుంది.
partition()	ఒక టుపుల్‌ను తిరిగి యిస్తుంది.
replace()	లోపల సబ్ స్ట్రింగ్‌ను భర్తీ చేస్తుంది.
rfind ()	సబ్ స్ట్రింగ్ యొక్క Highest Index ను అందిస్తుంది.
rindex()	ఒక సబ్‌స్ట్రింగ్ యొక్క Highest Index ను అందిస్తుంది.
rjust ()	ఇవ్వబడిన విడ్త్ సమర్థించబడితే స్ట్రింగ్‌ను కుడివైపుకు అందిస్తుంది.
rpartition ()	ఒక టుపుల్‌ను తిరిగి ఇస్తుంది.
rsplit ()	కుడివైపు నుండి స్ట్రింగ్‌ను విభజిస్తుంది.
rstrip ()	Trailing క్యారెక్టర్స్‌ను తొలగిస్తుంది.
slice()	range () ద్వారా పేర్కొన్న slice object ను సృష్టిస్తుంది.

12th_CS_Telugu_Medium_Annexure.indd 336 30-03-2023 13:10:49

split()	ఎడమవైపు నుండి స్ట్రింగ్‌ను విభజిస్తుంది.
splitlines()	Boundaries వద్ద స్ట్రింగ్‌ను విభజిస్తుంది.
startswith()	Specified స్ట్రింగ్‌తో స్ట్రింగ్ ప్రారంభమైతే తనిఖీ చేస్తుంది.
strip()	Leading మరియు Trailing క్యారెక్టర్స్ రెండింటిని తొలగిస్తుంది.
swapcase()	lowercase ను uppercase గాను మరియు uppercase ను lowercase గాను మార్చును.
title()	Title Case స్ట్రింగ్‌ను తిరిగి ఇస్తుంది.
translate()	mapped charactered స్ట్రింగ్‌ను తిరిగి ఇస్తుంది.
upper()	uppercased స్ట్రింగ్‌ను తిరిగి ఇస్తుంది.
zfill()	స్ట్రింగ్ యొక్క Copy ని Zero తో Pad చేస్తుంది.

3. List ఫంక్షన్స్

ఫంక్షన్	వివరణ
append()	List లో ఒక మూలకాన్ని Add చేస్తుంది.
clear()	List లో అన్ని అంశాలను తొలగిస్తుంది.
copy()	List యొక్క Shallow copy ని అందిస్తుంది.
count()	List లోని మూలకాల యొక్క occurrences ను అందిస్తుంది.
extend()	ఒక List లోని మూలకాలను వేరొక List లోనికి Add చేస్తుంది.
index()	List లోని మూలకం యొక్క smallest index ను తిరిగి ఇస్తుంది.
insert()	List లోకి మూలకాలను Insert చేస్తుంది.
list() Function	పైథాన్‌లో List ను సృష్టించడం.
pop()	ఇవ్వబడిన Index నుండి మూలకాలను తొలగిస్తుంది.
remove()	List నుండి మూలకాలను తొలగించడం.
reverse()	List ను తలక్రిందులుగా చేయును.
slice()	Range () పేర్కొనబడిన Slice ఆబ్జెక్ట్‌ను సృష్టించును.
sort()	List లోని మూలకాలను Sort చేయును.

4. Tuple ఫంక్షన్స్

ఫంక్షన్	వివరణ
count()	tuple లోని మూలకము యొక్క Occurrences ని తిరిగి ఇస్తుంది.
index()	tuple లోని element యొక్క అతిచిన్న Index ను అందిస్తుంది.
slice()	range() ఫంక్షన్‌లో పేర్కొనబడిన slice object ను సృష్టించుట
tuple()Function	ఒక Tuple ను సృష్టించుట
zip()	Tuple యొక్క ఇటిరేటర్‌ను అందిస్తుంది.

5. Set ఫంక్షన్స్

ఫంక్షన్	వివరణ
add()	సెట్‌కు మూలకాలను Add చేయడం.
clear()	సెట్‌లోని అన్ని మూలకాలను తొలగించడం

12th_CS_Telugu_Medium_Annexure.indd 337 30-03-2023 13:10:49

copy()	సెట్ యొక్క Shallow copy ని అందిస్తుంది.
difference()	రెండు సెట్ల యొక్క భేదాన్ని అందిస్తుంది.
difference_update()	సెట్ యొక్క ఇంటర్ సెక్షన్‌లో అప్‌డేట్ కాలింగ్ సెట్
discard()	సెట్ నుండి Element ను తొలగించుట.
frozenset()	మార్చుటకు వీలుకాని frozenset object ను తిరిగి ఇస్తుంది.
intersection()	రెండు లేకా అంతకన్నా ఎక్కువ సెట్స్ యొక్క ఇంటర్ సెక్షన్‌ను (ఖండనము) తిరిగి ఇస్తుంది.
intersection_update()	సెట్ యొక్క ఇంటర్ సెక్షన్‌తో అప్‌డేట్స్ కాలింగ్ సెట్స్
isdisjoint()	Disjoint set ను తనిఖీ చేస్తుంది.
issubset()	ఒక సెట్ మరొక సెట్‌కు సబ్‌సెట్ అగునా అని తనిఖీ చేస్తుంది.
issuperset()	ఒక సెట్ మరొక సెట్‌కు సూపర్ సెట్ అవుతుందా అని తనిఖీ చేయుట
pop()	Arbitrary మూలకాన్ని తొలగిస్తుంది.
remove()	సెట్ నుండి Element ను తొలగిస్తుంది.
set()	ఫ్రెథాన్ సెట్‌ను అందిస్తుంది.
symmetric_difference()	symmetric భేదమును అందిస్తుంది.
symmetric_difference_update()	symmetric భేదముతో సెట్‌ను అప్‌డేట్ చేస్తుంది.
union()	సెట్స్ యొక్క సమ్మేళనమును అందిస్తుంది.
update()	సెట్‌కు మూలకాలను Add చేస్తుంది.

6. Dictionary ఫంక్షన్స్

ఫంక్షన్	వివరణ
clear()	అన్ని అంశాలను తొలగిస్తుంది.
copy()	ఒక డిక్షనరీ యొక్క Shallow copy ని అందిస్తుంది
dict()	ఒక డిక్షనరీని సృష్టిస్తుంది.
fromkeys()	ఇవ్వబడిన వరుస నుండి డిక్షనరిని సృష్టిస్తుంది.
get()	కీ యొక్క విలువను అందిస్తుంది.
items()	డిక్షనరి Pair యొక్క విక్షణను అందిస్తుంది (కీ, విలువలు)
keys()	అన్ని కీ ల యొక్క Object వీక్షణను అందిస్తుంది.
pop()	కీ ఇచ్చిన అంశాలను తీసివేస్తుంది మరయు తిరిగి ఇస్తుంది.
popitem()	డిక్షనరీ నుండి Elements ను తీసివేస్తుంది మరియు తిరిగి ఇస్తుంది.
setdefault()	కీ లేకపోతే విలువతో కీ ని ఇన్‌సెట్ చేయండి.
update()	డిక్షనరీని అప్‌డేట్ చేస్తుంది.
values()	అన్ని విలువల యొక్క వీక్షణను డిక్షనరిలో అందిస్తుంది.

12th_CS_Telugu_Medium_Annexure.indd 338 30-03-2023 13:10:49

ప్రయోగభ్యాసములు

12th Computer Science_EM Practical.indd 339 30-03-2023 13:14:23

పరిష్కారంతో కూడిన ప్రయోగాభ్యాసములు

ప్రయోగ కరదపిక

సూచనలు:

1. పైథాన్ నుండి పైథాన్ 8 అభ్యాసములు MySQL నుండి రెండు అభ్యాసములు ప్రాక్టికల్స్ తరగతులో సాధన చేయబడినది.
2. ఇంటర్నల్ ఎంపికలో ఒక ప్రశ్న.
3. మార్కుల పంపిణీ.

I. ఇంటర్నల్ అసెస్మెంట్	5 మార్కులు
రికార్డ్ బుక్	5 మార్కులు

II. ఎక్స్టర్నల్ అసెస్మెంట్	15 మార్కులు
అ) పైథాన్ ప్రోగ్రాం కోడింగ్	10 మార్కులు
ఆ) ఎగ్జిక్యూషన్, అవుట్ పుట్	5 మార్కులు
మొత్తం	**20 మార్కులు**

12th Computer Science_EM Practical.indd 340 30-03-2023 13:14:23

విషయ సూచిక

వరుస సంఖ్య	ప్రశ్న సంఖ్య	ప్రోగ్రాం పేరు	పుట సంఖ్య
1	PY1	(అ) పాక్టోరియల్ లెక్కింపు (ఆ) సిరీస్ మొత్తం	342
2	PY2	(అ) బేసి లేదా సరి (ఆ) స్ట్రింగ్ ని రివర్స్ చేయండి	343
3	PY3	విలువలను రూపొందించి బేసి సంఖ్యలను తీసివేయండి	344
4	PY4	ప్రధాన సంఖ్యలను రూపొందించి ఆపరేషన్లను సెట్ చేయండి	345
5	PY5	క్లాసును ఉపయోగించి స్ట్రింగ్ ఎలిమెంట్లను ప్రదర్శించండి	346
6	DB6	MySQL - ఉద్యోగుల పట్టిక	348
7	DB7	MySQL - విద్యార్థుల పట్టిక	352
8	PY8	CSV తో పైథాన్	357
9	PY9	SQL తో పైథాన్	359
10	PY10	PIP తో పైథాన్ గ్రాఫిక్స్	361

12th_CS_Telugu_Medium_Practical Folder.indd 341 11-04-2023 16:58:48

PY1 (అ) – పాక్టోరియల్ లెక్కింపు

1(అ) **లూప్‌ని ఉపయోగించి ఇవ్వబడిన సంఖ్యకు పాక్టోరియల్ లెక్కించుటకు ఒక ప్రోగ్రామును వ్రాయుము. (అంతర్నిమిత పాక్టోరియల్ ఫంక్షన్‌ను ఉపయోగించకండి).**

ఉద్దేశము: లూప్‌ని ఉపయోగించి ఇవ్వబడిన సంఖ్యకు పాక్టోరియల్ లెక్కించుట.

కోడింగ్:

```
num = int(input('Enter a Number: '))
fact = 1
for i in range(1,num+1):
        fact = fact * i
print("Factorial of ", num, " is ", fact)
```

అవుట్‌పుట్:

```
Enter a Number: 5
Factorial of  5  is  120
```

ఫలితము:

ఫ్యాక్టోరియల్ను లెక్కించుటకైనా పైథాన్ ప్రోగ్రాం చేయబడి అవుట్‌పుట్ ధ్రువీకరించబడినది.

PY1 (ఆ) - సిరీస్ మొత్తము

1(ఆ) **$1^1/1 + 2^2/2 + 3^3/3 + \ldots\ldots n^n/n$ సిరీస్ మొత్తమును లెక్కించుటకు ప్రోగ్రామును వ్రాయుము**

ఉద్దేశము: $1^1/1 + 2^2/2 + 3^3/3 + \ldots\ldots n^n/n$ సిరీస్ మొత్తం లెక్కించుట

కోడింగ్:

```
n = int(input("Enter a value of n: "))
s = 0
for i in range(1,n+1):
    a= (i**i)/i
    s=s+a
print("The sum of the series is ", s)
```

అవుట్‌పుట్:

```
Enter a value of n: 4
సిరీస్ మొత్తం  76
```


12th Computer Science_EM Practical.indd 342 30-03-2023 13:14:23

PY2 (అ) – బేసి లేదా సరి

2(అ) ఇవ్వబడిన సంఖ్య బేసి లేదా సరి అని తనిఖీ చేయుటకు ఫంక్షన్లను ఉపయోగించి ఒక ప్రోగ్రామును వ్రాయుము

ఉద్దేశము: డిఫైన్డ్ ఫంక్షన్ ఉపయోగించి ఇవ్వబడిన సంఖ్య బేసి లేదా సరి అని తనిఖీ చేయుట

కోడింగ్:

```
def oddeven(a):
    if (a%2==0):
            return "Even"
    else:
            return "Odd"
 num = int(input("Enter a number: "))
 print("The given number is ", oddeven(num))
```

అవుట్‌పుట్:

```
Enter a number: 7
The given number is Odd
Enter a number: 6
The given number is Even
```

ఫలితము:

ఇవ్వబడిన సంఖ్య బేసి లేదా సరి అని తెలుసుకునుటకు పైథాన్ ప్రోగ్రాం చేయబడి అవుట్‌పుట్ ధ్రువీకరించబడినది.

PY2 (ఆ) – స్ట్రింగ్‌ను రివర్స్ చేయుట

2(ఆ) ఇవ్వబడిన స్ట్రింగ్ రివర్స్ చేయుటకు ఒక ప్రోగ్రాం వ్రాయుము. ఉదాహరణకు, "wel" = "lew". (స్లైడ్ ఆపరేషన్తో కూడిన స్ట్రింగ్ స్లైస్‌ను ఉపయోగించకండి)

ఉద్దేశము: ఇవ్వబడిన స్ట్రింగ్‌ను రివర్స్ చేయుట

కోడింగ్:

```
def reverse(str1):
    str2=''
    for i in str1:
            str2 = i + str2
            return str2
word = input("\n Enter a String: ")
print("\n The reverse of the given string is: ", reverse(word))
```


12th Computer Science_EM Practical.indd 343 30-03-2023 13:14:23

అవుట్‌పుట్:

Enter a String: school

The reverse of the given string is: loohcs

ఫలితము:

ఇవ్వబడిన స్ట్రింగ్‌ను రివర్స్ చేయుటకుగాను పైథాన్ ప్రోగ్రాం చేయబడి అవుట్‌పుట్ ధ్రువీకరించబడినది.

PY3 – విలువలను రూపొందించి బేసి సంఖ్యలను తీసివేయండి.

3 **ఒకటి నుండి పది వరకు గల సంఖ్యల విలువలను రూపొందించుటకు ఒక ప్రోగ్రామును రాసిన పిదప జాబితా నుండి అన్ని సంఖ్యలను తీసివేయండి**

ఉద్దేశము: ఒకటి నుండి పది వరకు గల సంఖ్యల విలువలను రూపొందించుట మరియు జాబితా నుండి అన్ని సంఖ్యలను తీసివేయుట.

కోడింగ్:

```
num = list(range(1,11))
print("Numbers from 1 to 10.....\n",num)
for i in num:
    if(i%2 == 1):
        num.remove(i)
print("The values after removing odd numbers.....\n",num)
```

అవుట్‌పుట్:

Numbers from 1 to 10.....

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

The values after removed odd numbers.....

[2, 4, 6, 8, 10]

ఫలితము:

ఒకటి నుండి పది వరకు గల సంఖ్యల విలువలను రూపొందించుటకు మరియు జాబితా నుండి బేసి సంఖ్యలను తీసివేయుటకు పైథాన్ ప్రోగ్రాం చేయబడి అవుట్‌పుట్ ధ్రువీకరించబడినది.

12th Computer Science_EM Practical.indd 344 30-03-2023 13:14:24

PY4 – ప్రధాన సంఖ్యలను రూపొందించి ఆపరేషన్లను సెట్ చేయండి.

4. ప్రధాన సంఖ్యలు మరియు బేసి సంఖ్యల సమితిని రూపొందించే ప్రోగ్రామును వ్రాయుము. యూనియన్ ఇంటర్‌సెక్షన్ మరియు సిమెట్రిక్ ఆపరేషన్ల వ్యత్యాశములను ప్రదర్శించుము.

ఉద్దేశము: ప్రధాన సంఖ్యలు మరియు బేసి సంఖ్యల సమితి రూపకల్పన యూనియన్, ఇంటర్ సెక్షన్ మరియు సిమెట్రిక్ ఆపరేషన్‌లో వ్యత్యాసముల ప్రదర్శన.

కోడింగ్:

```
odd = set(range (1,10,2))
prime=set()
for i in range(2,10):
        for j in range (2,i):
            if (i%j==0):
                break
            else:
                prime.add(i)
print("Odd Numbers: ", odd)
print("Prime Numbers: ", prime)
print("Union: ", odd.union(prime))
print("Intersection: ", odd.intersection(prime))
print("Difference: ", odd.difference(prime))
print("Symmetric Difference: ", odd.symmetric_difference(prime))
```

అవుట్‌పుట్:

```
Odd Numbers:  {1, 3, 5, 7, 9}
Prime Numbers:  {2, 3, 5, 7}
Union:  {1, 2, 3, 5, 7, 9}
Intersection:  {3, 5, 7}
Difference:  {1, 9}
Symmetric Difference:  {1, 2, 9}
```

ఫలితము:

ప్రధాన సంఖ్యలను రూపొందించుటకు మరియు ఆపరేషన్ లను సెట్ చేయుటకు పైథాన్.

12th Computer Science_EM Practical.indd 345 30-03-2023 13:14:24

PY5 – క్లాస్‌ను ఉపయోగించి స్ట్రింగ్ ఎలిమెంట్లను ప్రదర్శించండి.

5. క్లాస్‌ను ఉపయోగించి ఇవ్వబడిన స్ట్రింగ్ లో స్ట్రింగ్ ను ఆమోదించుటకు ఒక ప్రోగ్రాం రాసి అప్పర్ కేస్ లోయర్ కేసు, అచ్చులు, హల్లులు మరియు ఖాళీలను సంఖ్యను ముద్రించుము.

ఉద్దేశము: స్ట్రింగ్ ఆమోదించుటకు క్లాస్‌ను ఉపయోగించి ఇవ్వబడిన స్టీమ్ లో అప్పటికే లోయర్ కేసు, అచ్చులు, హల్లులు మరియు ఖాళీల సంఖ్యను ముద్రించుట.

కోడింగ్:

```
class String:
    def __init__(self):
        self.upper=0
        self.lower=0
        self.vowel=0
        self.consonant=0
        self.space=0
        self.string=""
    def getstr(self):
        self.string=str(input("Enter a String: "))
    def count(self):
        for ch in self.string:
            if (ch.isupper()):
                self.uppercase+=1
            if (ch.islower()):
                self.lowercase+=1
            if (ch in ('AEIOUaeiou')):
                self.vowel+=1
            if (ch==""):
                self.spaces+=1
    self.consonant=self.upper+self.lower-self.vowel
    def display(self):
        print("The given string contains...")
        print("%d Uppercase letters"%self.upper)
        print("%d Lowercase letters"%self.lower)
        print("%d Vowels"%self.vowel)
        print("%d Consonants"%self.consonant)
        print("%d Spaces"%self.space)
```


12th Computer Science_EM Practical.indd 346 30-03-2023 13:14:24

```
S = String()
S.getstr()
S.execute()
S.display()
```

అవుట్‌పుట్:

```
Enter a String: Welcome to Computer Science
The given string contains...
3 Uppercase letters
21 Lowercase letters
10 Vowels
14 Consonants
3 Spaces
```

ఫలితము:

క్లాస్‌ను ఉపయోగించి స్ట్రింగ్ ఎలిమెంట్లను ప్రదర్శించుటకు గాను పైథాన్ ప్రోగ్రాం.

12th Computer Science_EM Practical.indd 347 30-03-2023 13:14:24

6. Empno, Empname, Desig, Dept, Age మొదలగు అంశాలతో కూడిన ఉద్యోగుల పట్టికను రూపొందించుము. ఆ పట్టికలో ఐదు రికార్డులను పొందుపరచుము
పట్టికకు ఇంకను రెండు రికార్డులను జోడించుము
చేరిన తేదీ అనే అని మరో అంశమును జోడించడం ద్వారా పట్టిక నిర్మాణాన్ని సవరించండి.
ఏదైనా రికార్డు యొక్క చేరిన తేదీలో శూన్య విలువను తనిఖీ చేయండి.
01/01/2018 తరువాత చేరిన ఉద్యోగులను జాబితా చేయండి.

ఉద్దేశము:

(1) Empno, Empname, Desig, Dept, Age మొదలగు అంశాలతో కూడిన ఉద్యోగుల పట్టికను రూపొందించుము.

(2) ఆ పట్టికలో ఐదు రికార్డులను పొందుపరచుము

(3) పట్టికకు ఇంకను రెండు రికార్డులను జోడించుము

(4) పట్టిక నిర్మాణాన్ని సవరించండి

(5) శూన్య విలువను తనిఖీ చేయండి

(6) ఉద్యోగులను జాబితా చేయండి

SQL QUERIES AND OUTPUT:

(i) Creating Table Employee

```
mysql> Create table Employee (Empno integer(4) primary key,
       Empname varchar(20), Desig varchar(10), Dept varchar(10),
       Age integer(2), Place varchar(10));
```

(ii) View Table Structure:

```
mysql> Desc Employee;
```

Field	Type	Null	Key	Default	Extra
Empno	int(4)	NO	PRI	NULL	
Empname	varchar(20)	YES		NULL	
Desig	varchar(10)	YES		NULL	
Dept	varchar(10)	YES		NULL	
Age	int(2)	YES		NULL	
Place	varchar(10)	YES		NULL	

6 rows in set (0.00 sec)

12th Computer Science_EM Practical.indd 348 30-03-2023 13:14:25

(iii) Inserting Data into Table:

```
mysql> Insert into employee values(1221, 'Sidharth', 'Officer', 'Accounts', 45, 'Salem');
mysql> Insert into employee values(1222, 'Naveen', 'Manager', 'Admin', 32, 'Erode');
mysql> Insert into employee values(1223, 'Ramesh', 'Clerk', 'Accounts', 33, 'Ambathur');
mysql> Insert into employee values(1224, 'Abinaya', 'Manager', 'Admin', 28, 'Anna Nagar');
mysql> Insert into employee values(1225, 'Rahul', 'Officer', 'Accounts', 31, 'Anna Nagar');
```

(iv) Select all the record:

```
mysql> select * from Employee;
```

Empno	Empname	Desig	Dept	Age	Place
1221	Sidharth	Officer	Accounts	45	Salem
1222	Naveen	Manager	Admin	32	Erode
1223	Ramesh	Clerk	Accounts	33	Ambathur
1224	Abinaya	Manager	Admin	28	Anna Nagar
1225	Rahul	Officer	Accounts	31	Anna Nagar

5 rows in set (0.00 sec)

(v) Adding two more records:

```
mysql> Insert into employee values(3226, 'Sona', 'Manager', 'Accounts', 42, 'Erode');
mysql> Insert into employee values(3227, 'Rekha', 'Officer', 'Admin', 34, 'Salem');
```

```
mysql> select * from Employee;
```

Empno	Empname	Desig	Dept	Age	Place
1221	Sidharth	Officer	Accounts	45	Salem
1222	Naveen	Manager	Admin	32	Erode
1223	Ramesh	Clerk	Accounts	33	Ambathur
1224	Abinaya	Manager	Admin	28	Anna Nagar
1225	Rahul	Officer	Accounts	31	Anna Nagar
3226	Sona	Manager	Accounts	42	Erode
3227	Rekha	Officer	Admin	34	Salem

7 rows in set (0.00 sec)

12th Computer Science_EM Practical.indd 349 30-03-2023 13:14:25

(vi) Adding one more Field:

```
mysql> Alter table employee add(doj date);
```

```
mysql> desc employee;
```

Field	Type	Null	Key	Default	Extra
Empno	int(4)	NO	PRI	NULL	
Empname	varchar(20)	YES		NULL	
Desig	varchar(10)	YES		NULL	
Dept	varchar(10)	YES		NULL	
Age	int(2)	YES		NULL	
Place	varchar(10)	YES		NULL	
doj	date	YES		NULL	

7 rows in set (0.00 sec)

(vii) Inserting date of joining to each employee:

```
mysql> update employee set doj = '21-03-2010' where empno=1221;
mysql> update employee set doj = '13-05-2012' where empno=1222;
mysql> update employee set doj = '25-10-2017' where empno=1223;
mysql> update employee set doj = '17-06-2018' where empno=1224;
mysql> update employee set doj = '02-01-2018' where empno=1225;
mysql> update employee set doj = '31-12-2017' where empno=3226;
mysql> update employee set doj = '16-08-2015' where empno=3227;
```

```
mysql> select * from Employee;
```

Empno	Empname	Desig	Dept	Age	Place	doj
1221	Sidharth	Officer	Accounts	45	Salem	2010-03-21
1222	Naveen	Manager	Admin	32	Erode	2012-05-13
1223	Ramesh	Clerk	Accounts	33	Ambathur	2017-10-25
1224	Abinaya	Manager	Admin	28	Anna Nagar	2018-06-17
1225	Rahul	Officer	Accounts	31	Anna Nagar	2018-01-02
3226	Sona	Manager	Accounts	42	Erode	2017-12-31
3227	Rekha	Officer	Admin	34	Salem	2015-08-16

7 rows in set (0.00 sec)

12th Computer Science_EM Practical.indd 350 30-03-2023 13:14:25

(viii) Checking null value in doj

```
mysql>  select * from emp where empno is null;
```

Empty set (0.00 sec)

(ix) List the employees who joined after 01/01/2018.

```
mysql>  Select * from emp where doj > '01-01-2018';
```

Empno	Empname	Desig	Dept	Age	Place	doj
1224	Abinaya	Manager	Admin	28	Anna Nagar	2018-06-17
1225	Rahul	Officer	Accounts	31	Anna Nagar	2018-01-02

2 rows in set (0.00 sec)

12th Computer Science_EM Practical.indd 351 30-03-2023 13:14:25

7 దిగువ ఇవ్వబడిన అంశాలతో ఒక పట్టికను రూపొందించి క్రింది పట్టికలో ఇచ్చిన విధముగా దత్తాంశములను నమోదు చేయుము.

Field Name	Type	Size
Reg_No	char	5
Sname	varchar	15
Age	int	2
Dept	varchar	10
Class	char	3

Data to be entered :

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
M1002	Akash	20	ME	ME2
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2
E1001	Ravi	20	ECE	EC1
E1002	Leena	21	EEE	EE1
E1003	Rose	20	ECE	EC2

Then, query the followings :

(i) List the students whose department is “Computer Science”.

(ii) List all the students of age 20 and more in Mechanical department.

(iii) List the students department wise.

(iv) Modify the class M2 to M1.

(v) Check for the uniqueness of Register no.

12th Computer Science_EM Practical.indd 352 30-03-2023 13:14:25

SQL QUERIES AND OUTPUT

(1) Creating Table - Student

```
mysql> Create table Student(Reg_Nochar(5), Sname varchar(20), Age integer(2),
       Dept varchar(10), Class char(3));
```

Query OK, 0 rows affected (0.51 sec)

View table structure:

```
mysql> desc Student;
```

Field	Type	Null	Key	Default	Extra
Reg_No	char(5)	YES		NULL	
Sname	varchar(20)	YES		NULL	
Age	int(2)	YES		NULL	
Dept	varchar(10)	YES		NULL	
Class	char(3)	YES		NULL	

5 rows in set (0.02 sec)

(2) Inserting Data into table:

```
mysql> Insert into Student values ('M1001', 'Harish', 19, 'ME', 'ME1');
mysql> Insert into Student values ('M1002', 'Akash', 20, 'ME', 'ME2');
mysql> Insert into Student values ('C1001', 'Sneha', 20, 'CSE', 'CS1');
mysql> Insert into Student values ('C1002', 'Lithya', 19, 'CSE', 'CS2');
mysql> Insert into Student values ('E1001', 'Ravi', 20, 'ECE', 'EC1');
mysql> Insert into Student values ('E1002', 'Leena', 21, 'EEE', 'EE1');
mysql> Insert into Student values ('E1003', 'Rose', 20, 'ECE', 'EC2');
```

View all records :

```
mysql> select * from Student;
```


12th Computer Science_EM Practical.indd 353 30-03-2023 13:14:26

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
M1002	Akash	20	ME	ME2
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2
E1001	Ravi	20	ECE	EC1
E1002	Leena	21	EEE	EE1
E1003	Rose	20	ECE	EC2

7 rows in set (0.00 sec)

(3) Other Queries:

(i) List the students whose department is "CSE":

```
mysql> Select * from Student where Dept='CSE';
```

Reg_No	Sname	Age	Dept	Class
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2

2 rows in set (0.03 sec)

(ii) List all the students of age 20 and more in ME department:

```
mysql> Select * from Student where Age >=20 and Dept='ME';
```

Reg_No	Sname	Age	Dept	Class
M1002	Akash	20	ME	ME2

1 row in set (0.02 sec)

12th Computer Science_EM Practical.indd 354 30-03-2023 13:14:26

(iii) List the students department wise:

```
mysql>  Select * from Student Group by Dept Order by Sname;
```

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
E1002	Leena	21	CSE	EE1
E1001	Ravi	20	ECE	EC1
C1001	Sneha	20	EEE	CS1

4 rows in set (0.00 sec)

(iv) Modify the class M2 to M1:

```
mysql>  Update Student set Class='ME1' where Class='ME2';
        Query OK, 1 row affected (0.11 sec)
        Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql>  select * from Student;
```

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
M1002	Akash	20	ME	ME2
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2
E1001	Ravi	20	ECE	EC1
E1002	Leena	21	EEE	EE1
E1003	Rose	20	ECE	EC2

7 rows in set (0.00 sec)

12th Computer Science_EM Practical.indd 355 30-03-2023 13:14:26

(v) Check for the uniqueness of Register no.

```
mysql>  Select Distinct Reg_No from Student;
```

Reg_No
M1001
M1002
C1001
C1002
E1001
E1002
E1003

7 rows in set (0.02 sec)

12th Computer Science_EM Practical.indd 356 30-03-2023 13:14:26

8 పదిమంది ఆటగాళ్లు పేర్లను మరియు వారి స్కోర్లను పొందుటకు పైథాన్ ఉపయోగించి ఒక ప్రోగ్రామును వ్రాయుము ఇన్ పుట్ ఫైల్లో వ్రాయుము. పైథాన్ ఉపయోగించి ఆటగాడి పేరును అంగీకరించుము. పేరు మరియు స్కోరును ప్రదర్శించుటకు csv ఫైల్ ను చదువుము. ఆటగాడి పేరును కనుగొనినచో తగిన సందేశాన్ని ఇవ్వుము.

ఉద్దేశము: ఆటగాళ్ల పేరు వారి స్కోరు పొందడానికి ఇన్ పుట్ పైల్లో వ్రాయండి పైతాన్ను ఉపయోగించి ఆటగాడి పేరును అంగీకరించండి పేరు మరియు కోరును ప్రదర్శించడానికి ఫైల్ ను చదవండి ఆటగాడి పేరు కనుగొనబడకపోతే తగిన సందేశాన్ని ఇవ్వండి.

కోడింగ్:

```
import csv
with open('c:\\pyprg\\player.csv', 'w', newline=” ) as f:
        w = csv.writer(f)
        n=1
        while (n<=10):
                name = input("Player Name?:" )
                score = int(input("Score: "))
                w.writerow([name,score])
                n+=1
print("Player File created")
f.close()
searchname=input("Enter the name to be searched ")
f=open('c:\\pyprg\\player.csv','r')
reader =csv.reader(f)
lst=[]
for row in reader:
        lst.append(row)
q=0
for row in lst:
        if searchname in row:
```


12th Computer Science_EM Practical.indd 357 30-03-2023 13:14:26

```
            print(row)
            q+=1
            if(q==0):
                  print("string not found")
f.close()
f.close()
```

అవుట్‌పుట్:

```
Player Name?:Rohit Sharma
    Score: 264
Player Name?:VirenderSehwag
    Score: 219
Player Name?:Sachin Tendulkar
    Score: 200
Player Name?:Dhoni
    Score: 190
Player Name?:Sachin Tendulkar
    Score: 250
Player Name?:ViratKohli
    Score: 148
Player Name?:Ganguly
    Score: 158
Player Name?:KapilDev
    Score: 175
Player Name?:Amarnath
    Score: 148
Player Name?:SunilGavaskar
    Score: 200
Player File created
Enter the name to be searched Sachin Tendulkar
['Sachin Tendulkar', '200']
['Sachin Tendulkar', '250']
```


12th Computer Science_EM Practical.indd 358 30-03-2023 13:14:26

9

పైథాన్ ఉపయోగించి SQL పట్టికను రూపొందించుము. 10 మంది పేర్లను మరియు వయస్సును అంగీకరించుము. వయస్సు ఆధారితంగా అవరోహణ క్రమములో ప్రదర్శించుము.

ఉద్దేశము: పైథాన్ ఉపయోగించి SQL పట్టికను రూపొందించుము. 10 మంది పేర్లను మరియు వయస్సును అంగీకరించుము. వయస్సు ఆధారితంగా అవరోహణ క్రమములో ప్రదర్శించుట.

కోడింగ్:

```
import sqlite3

connection = sqlite3.connect("info.db")

cursor = connection.cursor()

cursor.execute("create table student(name, age)")

print("Enter 10 students names and their ages respectively:")

for i in range(10):

    who =[input("Enter Name:")]

    age =[int(input("Enter Age:"))]

    n =len(who)

    for i in range(n):

        cursor.execute("insert into student values (?, ?)", (who[i],age[i]))

cursor.execute("select * from student order by age desc")

print("Displaying All the Records From student Table in Descending order of age")

print (*cursor.fetchall(),sep='\n' )
```


12th Computer Science_EM Practical.indd 359 30-03-2023 13:14:26

అవుట్‌పుట్:

Enter 10 students names and their ages respectively:

Enter Name:Annamalai

Enter Age:17

Enter Name:Aashik Mathew

Enter Age:23

Enter Name:Kumaran

Enter Age:30

Enter Name:Sivasakthiya

Enter Age:28

Enter Name:Leena

Enter Age:45

Enter Name:Meena

Enter Age:65

Enter Name:Kamalakannan

Enter Age:35

Enter Name:Sowmyaa

Enter Age:20

Enter Name:Ramaa

Enter Age:70

Enter Name:Melvin

Enter Age:35

Displaying All the Records From student Table in Descending order of age

('Ramaa', 70)

('Meena', 65)

('Leena', 45)

('Kamalakannan', 35)

('Melvin', 35)

('Kumaran', 30)

('Sivasakthiya', 28)

('Aashik Mathew', 23)

('Sowmyaa', 20)

('Annamalai', 17)

12th Computer Science_EM Practical.indd 360 30-03-2023 13:14:26

10 పై చార్ట్ లో జాబితా మరియు ప్రదర్శనలను ఉపయోగించి ఐదు మార్కులను పొందుటకు ఒక ప్రోగ్రామును వ్రాయుము.

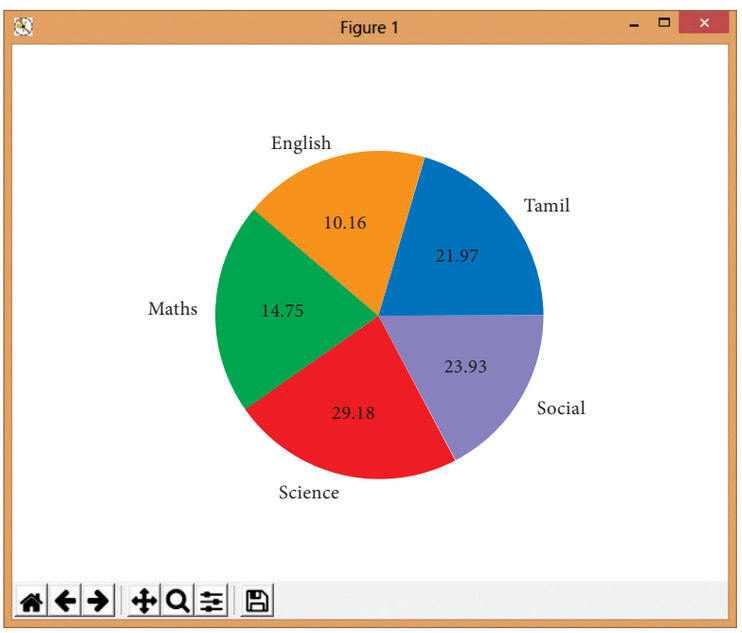
ఉద్దేశము: ఐదు మార్కులను పొందుటకు గాను పై చార్ట్ లోని జాబితా మరియు ప్రదర్శనను మెట్ ప్లాట్స్ ఉపయోగించడం.

కోడింగ్:

```
importmatplotlib.pyplot as plt
marks=[]
i=0
subjects = ["Tamil", "English", "Maths", "Science", "Social"]
while i<5:
        marks.append(int(input("Enter Mark = ")))
        i+=1
for j in range(len(marks)):
        print("{}.{} Mark = {}".format(j+1, subjects[j],marks[j]))
plt.pie (marks, labels = subjects, autopct = "%.2f ")
plt.show()
```

అవుట్‌పుట్:

```
Enter Mark = 67
Enter Mark = 31
Enter Mark = 45
Enter Mark = 89
Enter Mark = 73
    1.Tamil Mark = 67
    2.English Mark = 31
    3.Maths Mark = 45
    4.Science Mark = 89
    5.Social Mark = 73
```



12th Computer Science_EM Practical.indd 361 30-03-2023 13:14:26

Higher Secondary - Class XII - Computer Science
List of Reviewers & Translators
TELUGU MEDIUM

Academic Coordinators

Dr. M.Ethirajulu,
Senior Lecturer,
SCERT,
Chennai-6.

K.Chandrasekaran,
Lecturer, DIET,
Chennai.

Reviewers

P.M. Suresh,
Conputer Instructor,
G.B.H.S.S. Pothatturpeta,
Thiruvallur District

Translators

S. Mouli,
P.G.T. Assistant (Maths)
G.H.S.S, Athimanjeripet
Thiruvallur District.

V.V. Neeraja,
B.T. Assistant (Social Science)
C.H.S, Nammalwarpet,
Chennai-12.

K. Ramesh kumar,
B.T. Assistant (Maths)
G.H.S, Keechalam,
Thiruvallur District.

M. Devarajulu,
B.T. Assistant (Maths)
G.H.S.S, Athimanjeripet
Thiruvallur District.

K. Chokkalingam,
B.T. Assistant (Maths)
G.H.S, Nediyam,
Thiruvallur District.

M. Somasekar,
B.T. Assistant (Maths)
G.H.S, Karimbedu,
Thiruvallur District

K. Jamuna,
B.T. Assistant (Maths)
Dr.R.K.B.H.S.S, Thiruthani,
Thiruvallur District

S. Yashodha,
B.T. Assistant (Maths)
G.H.S, Nediyam,
Thiruvallur District.

P. Balakrishna,
B.T. Assistant (Maths)
G.B.H.S.S, Paradharami,
Vellor District.

S. Girija kumari,
B.T. Assistant (Maths)
G.H.S, Viligaram,
Thiruvallur District.

K.V. Nithyanandam,
B.T. Assistant (Msc, BED)
G.H.S, Nediyam,
Thiruvallur District.

Venkataiah,
B.T. Assistant
G.G.H.S.S, Tiruttani,
Thiruvallur District.

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

Wrapper Design

Kathir Arumugam

In-House QC

Ashok kumar
Daniel
Yuvaraj
Sridhar

Co-ordinator

Ramesh Munisamy

This book has been printed on 80 G.S.M.
Elegant Maplitho paper.

Printed by offset at:

12th Computer Science_EM Practical.indd 362 30-03-2023 13:14:26