GOVERNMENT OF TAMILNADU

ہائرسکنڈری سالِ دوّم

HIGHER SECONDARY SECOND YEAR

کمپیوٹر سائنس

COMPUTER SCIENCE

(URDU MEDIUM)

Department of School Education

Untouchability is Inhuman and a Crime

100
95
75
25
5
0

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:24 PM

Government of Tamil Nadu

First Edition - 2021
Revised Edition - 2023
Reprint - 2022, 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

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:24 PM

دیباچہ
PREFACE

کمپیوٹر اور کمپیوٹنگ ٹکنالوجی کا زبردست اثر انسانوں کی بہتری کے لئے جدید معاشرے کی تشکیل میں پڑا ہے۔ انسانی تہذیب نے نشوونما کے ساتھ کمپیوٹر کے ذریعہ اعلی ترین ترقی حاصل کی، جو "انٹرنیٹ ایرا" کے نام سے مشہور ہے۔

Python ایک اعلی سطح کی زبان ہونے کی وجہ سے ابتدائی زبان کے لئے آسان ترکیب اور میموری کی طاقتور نظم ونسق کی وجہ سے سیکھنا بہتر ہے۔ کسی بھی انٹرنٹ اپلی کیشنز کے لیے یہ سیکھنا اچھا ہے کیونکہ python بہتر انتخاب ہے۔

اس نصابی کتاب کے استعمال کرنے والے گیارہویں جماعت میں C++ سے واقفیت حاصل ہے اس لیے Python کی زبان کا مطالعہ کرنے میں کوئی پریشانی نہیں ہوگی۔ کوئی خاطر خواہ علم اور نہ ہی تجربہ درکار ہے، کیوں کہ کتاب میں بیان کردہ مثالوں پر عمل کرنا آسان ہے۔

اس کتاب کا استعمال کیسے کریں؟
HOW TO USE THE BOOK

- اس کتاب میں کمپیوٹر ٹکنالوجی میں پیشگی معلومات کی ضرورت نہیں ہے۔
- ہر یونٹ سادہ سرگرمیوں اور مظاہروں پر مشتمل ہے جو اساتذہ اور طلبا آسانی سے حل کر سکتے ہیں۔
- تکنیکی تفہیمات کو آسانی سے سمجھنے کے لیے لغات اس کی وضاحت کر دی گئی ہے۔
- کیا تم جانتے ہو؟ بکس اضافی معلومات کے ساتھ قاری کے علم کو تقویت بخشتے ہیں۔
- سافٹ ویئر اپلی کیشنز کا استعمال کرتے ہوئے مشقوں کو حل کرنے کے لئے ورک شاپس متعارف کروائی گئیں۔
- QR code کو اضافی مدد سے جوڑنے کے لئے استعمال کیا جاتا ہے۔

HOW TO GET CONNECTED TO QR CODE

QR code سے کیسے منسلک ہوں گے؟



- اپنے اسمارٹ فون میں گوگل پلے اسٹور/ ایپل ایپ اسٹور سے کیو آر کوڈ اسکینر ڈاؤن لوڈ کریں۔
- کوڈ اسکینر ایپ کو کھولیں.
- ایک بار جب درخواست میں اسکینر کے بٹن پر کلک کیا جاتا ہے، کیمرا کھل جاتا ہے اور پھر اسے درسی کتاب میں QR کوڈ کے قریب لاتا ہے۔
- ایک بار جب کیمرا QR کوڈ کا پتہ لگاتا ہے تو، سکرین میں ایک URL ظاہر ہوتا ہے۔
- URL پر کلک کریں اور مشمولات کے صفحے پر جائیں۔

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:25 PM

Computer Knowledge Hub

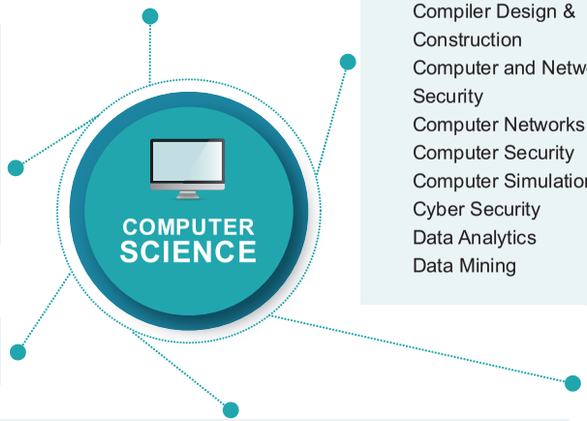
- Centre for Development of Advanced Computing https://www.cdac.in/
- Indian Computing Olympiad https://www.iarcs.org.in/inoi
- International Olympiad of Informatics https://ioinformatics.org
- Microsoft certification exams https://www.microsoft.com/en-in/learning/exam-list.aspx
- National Cyber Olympiad 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



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

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

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 Std - CS EM Introduction Pages.indd 4 23-12-2022 13:19:00

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

12th Std - CS EM Introduction Pages.indd 5 23-12-2022 13:19:01

Table of Contents فہرست مضامین

Computer Science-II Year کمپیوٹر سائنس سال دوم

اکائی نمبر	ابواب	عنوان	Title	صفحہ نمبر	مہینہ
اکائی-I مسئلہ حل کرنے کی تکنیک	1	فنکشن	(Function)	1	جون
	2	ڈیٹا کو حاصل کرنا	(Data Abstraction)	11	جون
	3	اسکوپنگ	(Scoping)	21	جون
	4	الگورتھمی حکمت عملی	(Algorithmic Strategies)	31	جون
اکائی II Core	5	پائیتھن - متغیرات اور آپریٹرز	(Python-Variables and operators)	47	جولائی
	6	کنٹرول ساختیں	(Control Structures)	67	جولائی
	7	پائیتھن فنکشنس	**(Python Functions)**	89	جولائی
	8	اسٹرنگز اور اسٹرنگ کو برتنا	(Strings and String manipulation)	114	اگست
اکائی III ماڈولریٹی اور OOPS	9	فہرستیں، ٹوپلس، سیٹ اور لغت	(Lists, Tuples, Sets and Dictionary)	132	اگست
	10	پائیتھن کلاسس اور آبجکٹس (اشیاء)	(Python classes and objects)	170	اگست
اکائی-IV ڈیٹا بیس کے نظریات اور MySQL	11	ڈیٹا بیس کے تصورات	(Database concepts)	182	اکتوبر
	12	اسٹرکچرڈ کوری لینگویج (SQL)	(Structured Query Language) (SQL)	198	اکتوبر
	13	پائیتھن اور CSV فائلیں	(Python and CSV files)	224	اکتوبر
اکائی-V MySQL اور C++	14	پائیتھن میں C++ پروگرام درآمد کرنا	(Importing C++ programs in Python)	259	اکتوبر
	15	SQL کے ذریعے ڈیٹا کو برتنا	**(Data manipulation through SQL)**	278	نومبر
	16	Pyplot کا استعمال کرتے ہوئے ڈیٹا کو دیکھنا خطی خاکے، پئی خاکے اور ستونی خاکے	(Data visualization using Pyplot: Line chart, Pie chart and bar chart)	307	نومبر
		لغت	**Glossary**	321	
ضمیمہ		Python کے افعال کی فہرست		327	
		عملی مشقیں		331	

E - book

Assessment

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:25 PM

I اکائی

CHAPTER 1 باب نمبر 1

FUNCTION فنکشن

سیکھنے کے مقاصد

اس باب کی تکمیل کے بعد، طالب علم:

- فنکشن (افعال) کی تفصیلات کو سمجھیں۔
- پیرامیٹر (اور دلائل)
- انٹرفیس بمقابلہ عمل functions خالص افعال۔
- ضمنی اثرات-(غیر خالص افعال)۔

1.1 تعارف

الگورتھم لکھنے اور اندازہ کرنے کا سب سے اہم معیار یہ ہے کہ کسی کام کو مکمل کرنے میں وقت لگتا ہے۔ الگورتھم کے معنی خیز موازنہ کرنے کے لئے، حساب کرنے کی مدت، پروگرامنگ زبان، مرتب، استعمال میں آنے والے کمپیوٹر وغیرہ پر منحصر ہے۔ جیسا کہ آپ جانتے ہو کہ الگورتھم کا اظہار پروگرامنگ کی زبان کے بیانات کے ذریعے کیا جاتا ہے۔ اگر بہت سارے بیانات دہرائے جائیں تو کام کو ختم کرنے کے لئے subroutines کا استعمال کیا جائے گا۔

subroutines کمپیوٹر پروگراموں کی بنیادی عمارتیں ہیں۔ subroutines کوڈ کے چھوٹے حصے ہیں جو کسی خاص کام کو انجام دینے کے لئے استعمال ہوتے ہیں جو بار بار استعمال کیے جاسکتے ہیں۔ پروگرامنگ زبانوں میں ان subroutines کو functions کہا جاتا ہے۔

1.2 پروگرامنگ زبان کے حوالے سے فنکشن (تفاعل)

(Function with respect to Programming language)

ایک فنکشن کوڈ کی ایک اکائی ہوتی ہے جس کی وضاحت اکثر کوڈ ڈھانچے میں ہوتی ہے۔ خاص طور پر، ایک فنکشن میں کوڈ کا ایک سیٹ ہوتا ہے جو متعدد قسم کے آدانوں، جیسے مختلف حالتوں، تاثرات پر کام کرتا ہے اور ٹھوس آؤٹ پٹ تیار کرتا ہے۔

1.2.1 فنکشن کی تفصیلات

(Function Specification)

آیئے ایک مثال پر غور کریں $a:=(24)$. $a:=(24)$ اس میں ایک جملہ ہے لیکن (24) خود ایک جملہ نہیں ہے۔ بلکہ، یہ ایک فنکشن کی توضیح (Function definition) ہے۔ توضیحات اقدار کو ناموں سے منسلک کرتی ہیں، اس معاملے میں 24 کی قیمت کو 'a' نام سے باندھا کیا جاتا ہے۔ توضیحات اظہار نہیں ہوتی، اسی وقت اظہار کو بھی توضیح کے طور پر نہیں سمجھا جاتا ہے۔ توضیحات منفرد syntactic بلاکس ہیں۔ تعریفیں ان کے اندر اظہار خیال کر سکتی ہیں اور اس کے برعکس بھی۔

1.2.2 پیرامیٹرز (اور آزاد متغیر)

(Parameters (and arguments))

پیرامیٹرز ایک فعل کی توضیح کے متغیرات ہیں اور آزاد متغیر (arguments) وہ قدریں ہیں جو کسی فعل کی توضیح میں منتقل کی جاتی ہیں

1۔ پیرامیٹر بغیر کسی قسم کے (Parameter without type)

آیئے ایک فنکشن کے توضیح (function definition) کی ایک مثال دیکھیں۔

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:26 PM

```
(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)
```

مذکورہ تفاعل (فنکشن) کی توضیح (function definition) میں متغیر 'b' پیرامیٹر ہے اور وہ قدر جو متغیر کو منتقل کی جاتی ہے 'b' آزاد متغیر (argument) ہے۔ تفاعل کی شرط اول (requires) اور کنڈیشن (returns) دی گئی ہے۔ غور کیجئیکہ ہم نے کسی قسم کا ذکر نہیں کیا ہے (ڈیٹا کی قسمیں) ۔کچھ زبان مرتب کرنے والے (language compilers) اس نوعیت (ڈیٹا ٹائپ) کے مسئلہ کو الگورتھم کے ذریعہ حل کرتے ہیں، لیکن کچھ کو اس کے قسم کے ذکر کرنے کی ضرورت ہوتی ہے۔

مذکورہ بالا تفاعل کی توضیح میں اگر جملہ اس کی شاخ میں 1 واپس آسکتا ہے تو، ظاہر ہے کہ Typing کے قانون کے مطابق اگر پورے جملہ میں 'int' کی قسم موجود ہوگا۔ تفاعل کا جملہ اگر 'int' قسم کا ہے تو، تفاعل کا return کی 'int' قسم کا ہوگا۔ مساوات آپریٹر کے ساتھ b کو 0 سے موازنہ کیا جاتا ہے۔ لہذا 'b' بھی 'int' کی ایک قسم ہے۔ چونکہ a کسی اور جملہ کے ساتھ * آپریٹر کا استعمال کرتے ہوئے ضرب ہوتا ہے، لہذا 'a' لازمی طور پر 'int' ہونا چاہئے۔

2۔ قسم کے ساتھ پیرامیٹر (Parameter with type)

اب ہم کچھ وجوہات کی بنا پر اقسام کے ساتھ ایک ہی تفاعل کی توضیح لکھیں گے۔

```
(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 b (a-1)
```

'a' اور 'b' کے لئے قوسین ضروری ہیں۔ عام طور پر ہم ان تشریحات کو چھوڑ سکتے ہیں، کیونکہ ان کو مرتب کرنے والے کو آسانی سے سمجھانے کے لئے اس طرح لکھا جاتا ہے۔ بعض اوقات ہمیں واضح طور پر اقسام کو لکھنا پڑتا ہے۔ یہ ایسے اوقات میں مفید ہے جب آپ کو مرتب کرنے والے سے کسی قسم کی غلطی (error) حاصل ہو جاتی ہے جس کا کوئی مطلب نہیں ہوتا۔ اقسام کو واضح طور پر بیان کرنا اس طرح کے غلط پیغام کو debugging کرنے میں مددگار ثابت ہوسکتا ہے۔

افعال کی توضیح کرنے کا syntax ریاضی کے استعمال میں بہت قریب ہے: اس توضیح کو کلیدی لفظ let کے ذریعہ متعارف کرایا گیا ہے، اس کے بعد اس فعل کا n a m e اور اس کے آزاد متغیر (arguements)۔ پھر وہ فارمولا جو آزاد متغیر کو محسوب کرتا ہے، = علامت کے بعد لکھا جاتا ہے۔ اگر آپ ایک بار بار آنے والے تفاعل کی وضاحت کرنا چاہتے ہیں تو: "let" کی بجائے "let rec" استعمال کریں۔

Syntax : تفاعل کی توضیح کے لئے syntax یہ ہے۔

```
let rec fna1 a2 ... an := k
```

یہاں fn متغیر ہے جس کی نشاندہی کرنے والے ایک شناخت کار کو بطور فنکشن نام استعمال کیا جارہا ہے۔ an کے لئے a1 نام متغیر ہیں جو پیرامیٹرز کے بطور استعمال ہونے والے شناخت کاروں کی نشاندہی کرتے ہیں۔ کلیدی لفظ rec کی ضرورت اس وقت ہوتی ہے، جب fn بار بار کام کرنے والا فنکشن ہے۔ ورنہ اسے چھوڑ دیا جاسکتا ہے۔

نوٹ:

ایک فعلی توضیح (function definition) اپنے آپ کو ایک مکرر (بار بار آنے والی) توضیح (recursive function) بھی کہلاتی ہے۔

مثال کے طور پر: آیئے یہ جانچنے کے لئے ایک مثال دیکھیں کہ داخل کردہ نمبر جفت ہے یا طاق۔

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:26 PM

```
(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'
```

فنکشن کی قسم کے لئے syntax یہ ہے۔

$$x \rightarrow y$$
$$x1 \rightarrow x2 \rightarrow y$$
$$x1 \rightarrow ... \rightarrow xn \rightarrow y$$

'x' اور 'y' متغیرات کی نشاندہی کرنے والی اقسام ہیں۔ قسم $x \rightarrow y$ فنکشن کی قسم ہے جو قسم 'x' کی ایک ان پٹ لیتی ہے اور 'y' قسم کی آؤٹ پٹ واپس کرتی ہے۔ جبکہ $x1 \rightarrow x2 \rightarrow y$ فنکشن کی ایک قسم ہے جو دو input لیتا ہے، پہلا ان پٹ 'x1' قسم کا ہوتا ہے اور دوسرا ان پٹ 'x2' قسم کا ہوتا ہے، اور قسم 'y' کی آؤٹ پٹ واپس کرتا ہے ۔ اسی طرح $x1 \rightarrow x...2 \rightarrow xn \rightarrow y$ میں n آزاد متغیرات (arguments) ان پٹ کے طور پر اور آؤٹ پٹ کے طور پر 'y' ٹائپ ہے۔

نوٹ:

تمام تفاعلات جامد توضیحات (static definitions) ہیں۔ کوئی بھی متحرک تفاعل کی توضیح (dynamic function definitions) نہیں ہے۔

1.3 انٹرفیس بمقابلہ عمل Interface Vs Implementation

انٹرفیس عمل کا ایک مجموعہ ہوتا ہے جو کوئی شے کر سکتا ہے۔ مثال کے طور پر جب آپ لائٹ سوئچ دبائیں تو روشنی آجاتی ہے، آپ کو اس کی پرواہ نہیں ہوگی کہ اس نے روشنی کیسے پھیلائی۔

Object Oriented Programming Language میں، انٹرفیس ان تمام افعال کی تفصیل ہے جو ایک کلاس میں لازمی طور پر ہونا ضروری ہے۔ ہماری مثال میں کوئی بھی شئے جو ایک بلب کی طرح "ACTS LIKE" (کی طرح) عمل کرتا ہے، اس کے تفاعل کی توضیحات جیسے turn_on() اور ایک turn_off() ہونی چاہئیں۔ انٹرفیس کا مقصد یہ ہے کہ کمپیوٹر کو TYPE T (انٹرفیس چاہے کچھ بھی ہو) اس کی کلاس کی خصوصیات کو نافذ کرنے کی اجازت دے، انہیں X، Y، Z، وغیرہ نامی تفاعلات (فنکشن) ہونا چاہئے۔

ایک کلاس کے اعلامیہ میں بیرونی انٹرفیس (اس کی مقامی حالت) کو اس انٹرفیس کے نفاذ کے ساتھ جوڑتا ہے (طرز عمل کو انجام دینے والا کوڈ) ۔ ایک شے کلاس سے تیار کی گئی ایک مثال ہے۔

انٹرفیس بیرونی دنیا کے لئے کسی شئے کو دیکھنے کی وضاحت کرتا ہے۔ انٹرفیس اور عمل درآمد کے درمیان فرق یہ ہے۔

انٹرفیس (Interface)	عمل درآمد (Implementation)
انٹرفیس صرف یہ وضاحت کرتا ہے کہ ایک شئے کیا کر سکتی ہے، حقیقت میں یہ ایسا نہیں کرتا۔	انٹرفیس میں موجود واضح کردہ ہدایات پر عمل کرتا ہے۔ اسے implementation کہتے ہیں۔

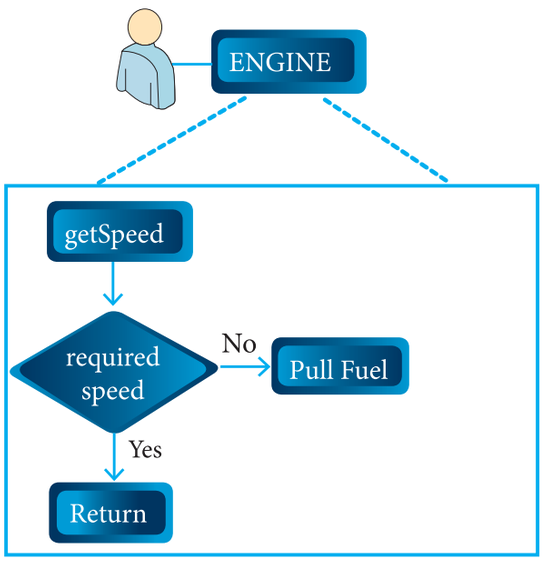
آبجیکٹ پر مبنی پروگرام کے کلاس میں انٹرفیس ہوتے ہیں اور اس شے کو کس طرح پروسس کیا جاتا ہے اس کو انجام دیتے ہیں۔

1.3.1 انٹرفیس کی خصوصیات (Characteristics of interface)

- کلاس ٹیمپلیٹ انٹرفیس کی وضاحت کرتا ہے تاکہ کسی شے کو مناسب طریقے سے تیار کر کے اسے چلانے کے قابل بنایا جا سکے۔
- کسی شے کی خصوصیات اور طرز عمل پر اعتراض کو افعال بھیج کر کنٹرول کیا جاتا ہے۔

مثال کے طور پر، کار کی رفتار بڑھانے کی مثال لیتے ہیں۔

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:27 PM



جوشخص گاڑی چلاتا ہے اسے اندرونی کام کرنے کا کوئی پرواہ نہیں ہوتی۔ کار کی رفتار بڑھانے کے لئے وہ مطلوبہ رفتار حاصل کرنے کے لئے بس ایکسلریٹر کو دباتا ہے۔ یہاں ڈرائیور (Calling/invoking object) اور انجن (the called object) کے درمیان ایکسلریٹر ایک انٹرفیس ہے۔

اس صورت میں، فنکشن کال کی رفتار :(70) ہوگی۔
یہ انٹرفیس ہے۔

اندرونی طور پر، کار کا انجن تمام چیزیں کر رہا ہے۔ اسی جگہ پر ایندھن، ہوا، دباؤ اور بجلی مل کر گاڑی کو منتقل کرنے کی طاقت پیدا کرتی ہے۔ یہ ساری حرکتیں ڈرائیور سے الگ کر دی گئی ہیں، جو صرف تیز رفتار سے جانا چاہتا ہے۔ اس طرح ہم عمل کو انٹرفیس سے الگ کرتے ہیں۔

آیئے ہم ایک سادہ سی مثال دیکھیں، کسی فنکشن کے درج ذیل نفاذ پر غور کریں، جس میں اس کے تین آزاد متغیرات (arguments) میں سے کم از کم پتہ چل سکے:

```
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 خالص تفاعل (Pure functions)

خالص تفاعل وہ فنکشن ہیں، جو ایک ہی آزاد متغیر کے منظور ہونے پر عین نتیجہ پیش کریں گے۔ مثال کے طور پر ریاضی کی تقریب گناہ (0) ہمیشہ نتیجہ اخذ کرتا ہے۔ اس کا مطلب یہ ہے کہ جب بھی آپ ایک ہی argument کے ساتھ فنکشن کو call کریں گے، آپ کو ہمیشہ ایک ہی نتیجہ ملے گا۔ ایک فنکشن ایک خالص فنکشن ہوسکتا ہے بشرطیکہ اس میں کوئی بیرونی متغیر نہ ہونا پڑے جو اس متغیر کے طرز عمل کو تبدیل کر دے۔

آیئے ایک مثال دیکھتے ہیں۔

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

مذکورہ بالا فنکشن مربع ایک خالص فنکشن ہے کیونکہ یہ ایک ہی ان پٹ کے مختلف نتائج نہیں دے گا۔

خالص فنکشن رکھنے کے مختلف نظریاتی فوائد ہیں۔ ایک فائدہ یہ ہے کہ اگر کوئی فنکشن خالص ہے، پھر اگر اسے ایک ہی argument کے ساتھ کئی بار کہا جاتا ہے تو، مرتب کرنے والے کو صرف اس فعل کو ایک بار call کرنے کی ضرورت ہے۔

ایک مثال دیکھیں۔

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


12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:27 PM

اگر یہ مرتب کیا گیا ہے تو، ہر بار *Strlin (s)* کہا جاتا ہے اور پورے *Strlin (s)* پر تکرار کرنے کی ضرورت ہوتی ہے۔ اگر تالیف کرنے والا(compiler) یہ جانکاری رکھتا ہے کہ *Strlin* ایک خالص فنکشن ہے اور اس لوپ میں 's' اپ ڈیٹ نہیں ہوتا ہے، تو یہ *Strlin* کرنے کے لئے فالتو اضافی کالوں کو ختم کرسکتا ہے اور صرف ایک وقت پر عمل کرنے کیلئے لوپ بنا سکتا ہے۔ ان سے ہم کیا سمجھ سکتے ہیں، سٹرلین ایک خالص فنکشن ہے کیونکہ فنکشن ایک پیرامیٹر کے طور پر ایک متغیر لیتا ہے، اور اس کی لمبائی تلاش کرنے کے لئے اس تک رسائی حاصل کرتا ہے۔ یہ فنکشن بیرونی میموری کو پڑھتا ہے لیکن اس میں کوئی تبدیلی نہیں آتی ہے، اور لوٹی ہوئی قیمت بیرونی میموری سے حاصل ہوتی ہے۔

نوٹ:

خالص فنکشن کی تشخیص سے اس کے آؤٹ پٹ پر کوئی ضمنی اثر نہیں پڑتا۔

1.4.1 غیر خالص فنکشن (Impure functions)

فنکشن کے اندر استعمال ہونے والے تغیرات ضمنی اثرات کا سبب بن سکتے ہیں اگرچہ وہ افعال جو کسی آزاد متغیر argument کے ساتھ منظور نہیں ہوئے ہیں۔ ایسے معاملات میں فنکشن کو غیر خالص فنکشن کہا جاتا ہے۔ جب کوئی فنکشن اس کے ڈیفنس بلاک سے باہر متغیرات یا افعال پر انحصار کرتا ہے تو، آپ کبھی بھی اس بات کا یقین نہیں کرسکتے ہیں کہ جب بھی اس فنکشن کو call کیا گیا ہے، اس کے ساتھ وہی سلوک کرے گا۔ مثال کے طور پر ریاضی کی فنکشن بے ترتیب () ایک ہی فنکشن کال کے لئے مختلف آؤٹ پٹ دے گی۔

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

یہاں فنکشن غیر خالص ہے کیونکہ یہ یقینی نہیں ہے کہ جب ہم فنکشن کو call کریں گے تو اس کا کیا نتیجہ نکلے گا۔

1.4.2 ضمنی اثرات (غیر خالص فنکشن) (Side effects) (Impure functions)

چونکہ آپ کو معلوم ہے کہ جب بیرونی دنیا کے ساتھ قابل عمل تعامل ہوتا ہے تو اس کے فنکشن پر مضر اثرات پڑتے ہیں۔ ایسے حالات ہیں کہ ہمارے فنکشن غیر خالص ہوسکتے ہیں، حالانکہ ہمارا مقصد اپنے فنکشن کو خالص بنانا ہے۔ صرف یہ سمجھنے کے لئے کہ ضمنی اثر ضروری خراب چیز نہیں ہے۔ کچھ اوقات وہ کارآمد ہیں (خاص طور پر فنکشنل پروگرامنگ کے نمونے سے ماہر)۔

کسی فنکشن کے باہر متغیر کو ترمیم کرنا (Modify variable outside a function)

ایک تفاعل سے باہر متغیر کو تبدیل کرنا، ضمنی اثرات کے سب سے اہم گروہوں میں سے ہے۔

مثال کے طور پر

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

مندرجہ بالا مثال کے طور پر y کی قیمت فنکشن کی تعریف کے اندر تبدیل ہوجاتی ہے جس کی وجہ سے نتیجہ ہر بار بدل جائے گا۔ inc() فنکشن کا ضمنی اثر یہ ہے کہ یہ بیرونی مرئی بدلنے والے متغیر 'y' کے ڈیٹا کو تبدیل کر رہا ہے۔ جیسا کہ آپ دیکھ سکتے ہیں کہ کچھ ضمنی اثرات تلاش کرنا کافی آسان ہیں اور ان میں سے کچھ مشکل ہوسکتے ہیں۔ ایک اچھی علامت یہ ہے کہ ہمارے فنکشن سے نجات (ضمنی اثر) یہ ہے کہ یہ کوئی argument نہیں لیتا ہے اور اس سے کوئی قیمت واپس نہیں ہوتی ہے

ان تمام مثالوں اور تعریفوں سے جو ہم خالص اور غیر خالص فنکشن کے مابین اہم اختلافات کے بارے میں سمجھ سکتے ہیں۔

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:28 PM

خالص فنکشن	غیر خالص فنکشن
خالص فنکشن کی واپسی کی قیمت صرف اس کے آزاد متغیرات (arguments) پر منحصر ہے۔ لہذا، اگر آپ arguments کے ایک ہی سیٹ کے ساتھ خالص فنکشن کو call کرتے ہیں تو، آپ کو ہمیشہ وہی واپسی کے اقدار ملیں گے۔ ان کے کوئی ضمنی اثرات نہیں ہیں۔	غیر خالص فنکشن کی واپسی کی قیمت مکمل طور پر اس کے منظور شدہ arguments پر منحصر نہیں ہے۔ لہذا، اگر آپ انہی arguments کے ایک ہی سیٹ کے ساتھ غیر خالص فنکشن کو call کرتے ہیں تو، آپ کو واپسی کی مختلف اقدار مل سکتی ہیں، مثال کے طور پر، random( )، Date()
جو arguments انہیں بھیجے جاتے ہیں، ان کی وہ ترمیم نہیں کرتے۔	جو arguments انہیں بھیجے جاتے ہیں، ان کی وہ ترمیم کر سکتے ہیں۔

آیئے اب ایک خالص فنکشن کی مثال دیکھتے ہیں، جس میں دو مثبت سالم اعداد کی مقسوم علیہ اعظم (gcd) کا تعین کیا جا سکتا ہے۔

```
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
```

مذکورہ بالا مثال کے پروگرام میں gcd اس فنکشن کا نام ہے جو متغیر 'b' بننے مکرر واقع ہوتا ہے یہاں تک کہ وہ '0' بن جاتا ہے۔ یاد رکھئے کہ b اور (a mod b) دو arguments ہیں جو gcd فنکشن کے 'a' اور 'b' کو بھیجتے ہیں۔

1.4.3 فنکشن کو استعمال کرتے ہوئے کروم لینڈ کے گرگٹوں کا مسئلہ

(Chameleons of Chromeland problem using function)

Chromeland کے گرگٹوں کے مسئلہ کو یاد کریں، جسے آپ نے گیارھویں جماعت میں پڑھا تھا۔ فرض کریں کہ گرگٹ کی دو قسمیں تعداد میں برابر ہیں۔ ایک الگورتھم بنائیں جو ان دونوں اقسام کے مابین ملاقاتوں کا اہتمام کرے تاکہ وہ اپنا رنگ تیسری قسم میں تبدیل کر دیں۔ آخر میں، سب کو ایک ہی رنگ کا مظاہرہ کرنا چاہئے۔

آیئے ہم ہر قسم کے گرگٹ کی تعداد کو متغیر a، b اور c فرض کرتے ہیں اور ان کے ابتدائی اقدار بالترتیب A، B اور C کی نمائندگی کرتے ہیں۔ a = b ان کی input property فرض کریں۔

ان کے اِن پٹ - آؤٹ پٹ کا رشتہ a = b = 0 اور c=A+B+C ہے۔ آیئے الگورتھم میں ایک رنگ بھریں۔ الگورتھم کو اس طرح بیان کیا جا سکتا ہے۔

-- inputs	:	a = A, b = B, c = C, a = b
-- outputs	:	a = b = 0, c = A+B+C

ہر تکراری اقدام میں، دونوں اقسام کے دو گرگٹ (تعداد میں برابر) ملتے ہیں اور اپنے رنگوں کو تیسرے میں تبدیل کرتے ہیں۔ مثال کے طور پر، اگر A,B,C = 4, 4, 6 تو پھر ملاقات کے سلسلے کا نتیجہ اس طرح ہوگا۔

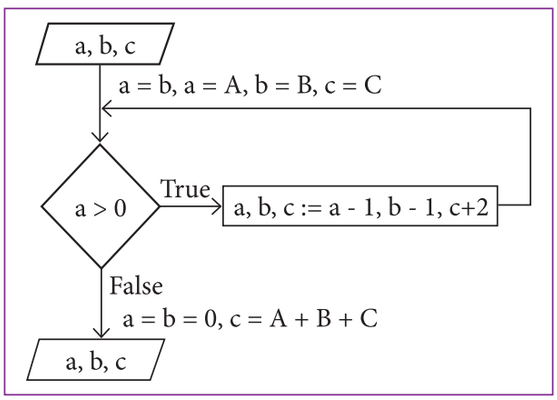
تکرار iteration	a	b	c
0	4	4	4
1	3	3	8
2	2	2	10
3	1	1	12
4	0	0	14

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:28 PM

ہر میٹنگ میں، a اور b میں سے ہر ایک میں 1 کی کمی واقع ہوتی ہے، اور c میں 2 کا اضافہ ہوتا ہے۔ اس حل کا تکراری الگورتھم کے طور پر کیا جا سکتا ہے۔

```
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
```

الگورتھم کو ذیل کے فلوچارٹ میں دکھایا گیا ہے.



اب ہم فنکشن کا استعمال کرتے ہوئے یہ الگورتھم لکھتے ہیں۔

```
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
```

یاد رکھنے کے نکات:

- الگورتھم کا اظہار پروگرامنگ کی زبان کے بیانات کے ذریعے کیا جاتا ہے۔
- subroutine کوڈ کے چھوٹے حصے ہیں جو کسی خاص کام کو انجام دینے کے لئے استعمال ہوتے ہیں جو بار بار استعمال کیے جا سکتے ہیں۔
- ایک فنکشن کوڈ کی ایک اکائی ہے جس کی وضاحت اکثر کوڈ کی ساخت میں ہوتی ہے۔
- کسی فنکشن میں کوڈ کا ایک سیٹ ہوتا ہے جو کئی طرح کے input پر کام کرتا ہے اور ٹھوس output تیار کرتا ہے۔
- توضیحات الگ syntactic بلاکس ہیں۔
- پیرامیٹرز ایک فعل کی توضیح کے متغیرات ہیں اور آزاد متغیر (arguments) وہ قدریں ہیں جو کسی فعل کی توضیح میں منتقل کی جاتی ہیں
- جب آپ definition annotation لکھتے ہیں تو قوسین میں فنکشن کی توضیح لکھی جاتی ہے۔
- انٹرفیس عمل کا ایک مجموعہ ہوتا ہے جو کوئی شے کر سکتی ہے۔
- انٹرفیس صرف اس بات کی وضاحت کرتی ہے کہ کوئی شئے کیا کر سکتی ہے، لیکن اصل میں یہ نہیں کرے گی۔
- عمل میں انٹرفیس میں بیان کردہ ہدایات پر عمل درآمد ہوتا ہے۔
- خالص فنکشن وہ فنکشن ہیں جو ایک ہی arguments کے گزرنے پر بالکل درست نتیجہ پیش کریں گے۔
- فنکشن کے اندر استعمال ہونے والے متغیرات ضمنی اثرات کا سبب بن سکتے ہیں اگرچہ وہ فنکشن جو کسی دلائل کے ساتھ منظور نہیں ہوئے ہیں۔ ایسے معاملات میں فنکشن کو غیر خالص فنکشن کہا جاتا ہے۔

100
95
75
25
5
0
12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:28 PM

Hands on Practice عملی مشق

1۔ کم ازکم 3 نمبر تلاش کرنے کے لئے الگورتھمک فنکشن کی توضیحات لکھیں۔

2۔ طبیعی اعداد n کے حاصل جمع معلوم کرنے کے لئے الگورتھمک مکرر فنکشن کی توضیحات (recursive function definition) لکھیں۔

S2LUO

Evaluation محاسبہ

حصہ - I

صحیح جواب منتخب کریں۔ (1 مارک)

1۔ کوڈ کے چھوٹے چھوٹے حصے جو کسی خاص کام کو انجام دینے کے لئے استعمال ہوتے ہیں۔

(A) سبروٹائنز (Subroutines) (B) فائلیں (C) سیوڈوکوڈ (Pseudo code) (D) ماڈیولز (Modules)

2۔ مندرجہ ذیل میں سے کون سا کوڈ کی ایک اکائی ہے جس کی وضاحت اکثر کوڈ کی ساخت میں ہوتی ہے؟

(A) سبروٹائنز (Subroutines) (B) فنکشن (function) (C) فائلس (Files) (D) ماڈیولز (Modules)

3۔ مندرجہ ذیل میں سے کون سا منفرد syntactic بلاک ہے؟

(A) سبروٹائنز (Subroutines) (B) فنکشن (function) (C) توضیحات (Definition) (D) ماڈیولز

4۔ ایک فنکشن کی توضیح میں متغیر کو یہ کہا جاتا ہے۔

(A) سبروٹائنز (Subroutines) (B) فنکشن (function) (C) توضیحات (Definition) (D) پیرامیٹرز

5۔ وہ اقدار جو فنکشن کی توضیح کی طرف بھیجی جاتی ہیں، یہ ہیں۔

(A) آزاد متغیر (Arguments) (B) سبروٹائنز (Subroutines) (C) فنکشن (D) توضیح (Definition)

6۔ فنکشن کی توضیح میں درج ذیل annotation میں سے کونسا لکھا جاتا ہے ؟

(A) خمیدہ قوسین (B) عام قوسین (C) مربع قوسین (D) حاشیہ

7۔ مندرجہ ذیل میں سے کون سا یہ وضاحت کرتا ہے کہ کوئی شے کیا کرسکتی ہے ؟

(A) آپریٹنگ سسٹم (B) مرتب (Compiler) (C) انٹرفیس (Interface) (D) Interpreter ترجمان

8۔ مندرجہ ذیل میں سے کون سا انٹرفیس میں بیان کردہ ہدایات پر عمل پیرا ہے؟

(A) آپریٹنگ سسٹم (B) مرتب (compiler) (C) عمل درآمد (Implementation) (D) ترجمان

9۔ وہ فنکشن جب ایک جیسے arguments بھیجے جائیں گے تو درست نتیجہ پیش کریں گے۔

(A) غیر خالص افعال (B) جزوی افعال (C) متحرک افعال (D) خالص افعال

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:29 PM

10۔ وہ فنکشن جب arguments بھیجنے پر ضمنی اثرات رکھتے ہیں، ان کو یہ کہتے ہیں۔

(A) غیر خالص فنکشن (B) جزوی فنکشن (C) متحرک فنکشن (D) خالص فنکشن

حصہ - II

ذیل کے سوالوں کا جواب دیجیے۔ (2 نمبر)

1۔ ایک سبروٹین (Subroutine) کیا ہے؟

2۔ پروگرامنگ کی زبان کو مدنظر رکھتے ہوئے فنکشن کی وضاحت کریں۔

3۔ X:=(78) سے حاصل ہونے والا نتیجہ لکھیں۔

4۔ انٹرفیس (Interface) اور عمل درآمد (implementation) میں فرق کیجئے۔

5۔ مندرجہ ذیل میں سے کون سا عام فنکشن (normal function) کی توضیح ہے اور کونسا مکرر فنکشن کی توضیح (recursive function definition) ہے۔

```
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
```

حصہ - III

درج ذیل سوالات کے جوابات دیجئے۔ (3 نمبرات)

1۔ انٹرفیس کی خصوصیات کا ذکر کریں.

2۔ کیوں strlen کو ایک خالص فنکشن کہا جاتا ہے؟

3۔ غیر خالص فنکشن کا ضمنی اثر کیا ہے؟ مثال دیں۔

4۔ خالص اور غیر خالص تقریب میں فرق کریں۔

5۔ اگر آپ فنکشن سے باہر کسی متغیر میں ترمیم کرتے ہیں تو کیا ہوتا ہے؟ ایک مثال دیں۔

100 95 75 25 5 0

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:29 PM

حصہ - IV

درج ذیل سوالوں کے جوابات دیجئے۔ (5 نشانات)

1۔ پیرامیٹر کسے کہتے ہیں اوران پر ایک نوٹ لکھیں۔

(i) قسم کے بغیر پیرامیٹر (ii) قسم کے ساتھ پیرامیٹر

2۔ مندرجہ ذیل پروگرام کی شناخت کریں۔

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

i) فنکشن کا نام

ii) اس بیان کی نشاندہی کریں جو بتاتا ہے کہ یہ تکرار کرنے والا فنکشن ہے۔

iii) آزاد متغیر (argument variable) کا نام

iv) بیان جس میں فنکشن کو مکرر طور پر (recursion) پکارا جاتا ہے۔

v) بیان جو تکرار ختم کرتا ہے۔

3۔ ایک مثال کے ساتھ خالص اور غیر خالص افعال کی وضاحت کریں۔

4۔ ایک مثال کے ساتھ انٹرفیس اور عمل درآمد (implementation) کی وضاحت کریں۔

حوالہ جات

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*

12 CS UNIT 1 CDR Set
Sunday, April 16, 2023 9:34:29 PM

سیکھنے کے مقاصد

اس باب کی تکمیل کے بعد، طالب علم یہ سمجھنے کے قابل ہو جائے گا

- ڈیٹا کو حاصل کرنے کی ساخت کیا ہے ؟
- ڈیٹا کو حاصل کرنے کی قسم
- حقیقی طور پر عمل میں لانے اور حاصل کرنے کے عمل کے درمیان فرق کرنا
- جوڑیاں
- ساخت میں ڈیٹا کو ڈھالنا

2.1 ڈیٹا حاصل کرنا - تعارف
Data Abstraction- Introduction

ڈیٹا کو حاصل کرنا، کمپیوٹر سائنس کا ایک طاقتور تصور ہے، جو پروگرامرز کو کوڈ کو اشیاء کی طرح سمجھنے میں مدد کرتا ہے۔ مثال کے طور پر، کار جیسی شئے، پنسل جیسی شئے، لوگوں جیسی اشیاء وغیرہ۔ پروگرامرز کو کوڈ کو لاگو کرنے کے طریقہ کار کے بارے میں فکر کرنے کی ضرورت نہیں ہے- انہیں صرف یہ جاننا ہوگا کہ یہ کیا ہے اور کیا کام کرتا ہے۔

یہ خاص طور پر اس وقت اہم ہے جب متعدد افراد پروجیکٹ کر رہے ہوں۔ یہاں پروجیکٹ سے مراد پروگرامنگ ہی ہے۔ ڈیٹا حاصل کرنے کے ساتھ، آپ کے گروپ کے ممبروں کو سمجھنے کے لئے آپ کے کوڈ کی ہر لائن کو پڑھنے کی ضرورت نہیں ہے۔ وہ صرف یہ فرض کر سکتے ہیں کہ یہ کام کرتا ہے۔

ڈیٹا حاصل کرنے میں ماڈولریٹی فراہم کرتا ہے (ماڈیولریٹی کا مطلب ہے کسی پروگرام کو کئی ماڈیولز میں تقسیم کرنا)۔ کلاس (ساخت) "ڈیٹا کو حاصل کرنے کی اقسام"(Abstract Data Types) (ADT) کی نمائندگی کرتی ہیں۔

2.2 ڈیٹا کو حاصل کرنے کی اقسام
Abstract Data Types

Abstract Data Types (ADT) اشیاء کے لئے (ایک قسم (یا کلاس) ہے جس کے طرز عمل کی قیمت قدر کے ایک سیٹ اور آپریشنز کے ایک سیٹ سے ہوتی ہے۔

ADT کی تعریف میں صرف اس بات کا تذکرہ کیا گیا ہے کہ کونسا عمل (operation) کرنا ہے؟ لیکن یہ نہیں کہ ان کارروائیوں کو کس طرح نافذ کیا جائے گا۔ اس میں یہ واضح نہیں کیا گیا ہے کہ ڈیٹا کو میموری میں کس طرح منظّم کیا جائے گا اور کارروائیوں کو عملی جامہ پہنانے کے لئے کون سے الگورتھم استعمال کیے جائیں گے۔ اسے حاصل کرنا یا نکالنا (abstract) کہا جاتا ہے کیونکہ یہ ایک آزاد نظریہ دیتا ہے۔ صرف ضروری اشیاء فراہم کر کے تفصیلات کو چھپانے کو حاصل کرنا یا نکالنا (abstraction) کہا جاتا ہے۔

آپ دیکھ سکتے ہیں کہ ان تعریفوں میں یہ واضح نہیں کیا گیا ہے کہ ان ADT کی نمائندگی کس طرح کی جائے گی اور کاروائیاں کیسے عمل میں آئیں گی۔ ADT کو نافذ کرنے کے مختلف طریقے ہو سکتے ہیں، مثال کے طور پر، لسٹ ADT اکیلے منسلک فہرست یا دگنا طور پر جڑی ہوئی فہرست کا استعمال کرتے ہوئے لاگو کیا جا سکتا ہے۔ اسی طرح، stack ADT اور Queue ADT کو فہرستوں کا استعمال کر کے لاگو کیا جا سکتا ہے۔

ڈیٹا حاصل کرنے کا عمل، یہ دوہراتا ہے کہ ہم دنیا کی چیزوں کے بارے میں کیا سوچتے ہیں۔ مثال کے طور پر، جب آپ کار چلانا چاہتے ہیں تو آپ کو یہ جاننے کی ضرورت نہیں ہوگی کہ انجن کیسے بنایا گیا تھا یا ٹائر کس طرح کی شئے سے بنے ہوئے ہیں۔ آپ کو صرف یہ جاننا ہوگا کہ پہیے کو کس طرح موڑنا اور اکسلریٹر کس طرح دبانا ہے۔ ڈیٹا کو ختم کرنے میں آسانی کے لئے، آپ کو دو طرح کے افعال تخلیق کرنے کی ضرورت ہوگی: Constructors اور selectors۔

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:47 PM

2.3 تعمیر کرنے والے اور انتخاب کرنے والے
constructors and selectors

کنسٹرکٹرز ایسے افعال ہوتے ہیں جو حاصل کرنے والے ڈیٹا کی تعمیر کرتے ہیں۔ سلیکٹرز وہ افعال ہوتے ہیں جو ڈیٹا کی قسم سے معلومات بازیافت (retrieve) کرتے ہیں۔

مثال کے طور پر، کہتے ہیں کہ آپ کے پاس شہر نامی ایک حاصل کردہ ڈیٹا کی قسم city ہے۔اس city object میں شہر کا نام اور اس کا طول البلد اور عرض البلد ہوگا۔کسی city کی تشکیل کے لئے، آپ اس طرح کے فنکشن استعمال کریں گے۔

city = makecity (name, lat, lon)

کسی شہر کے بارے میں معلومات نکالنے کے لئے آپ اس طرح کے فنکشن استعمال کریں گے

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

مندرجہ ذیل کاذب کوڈ (pseudo code) دو شہروں کے اشیاء کے مابین فاصلے کا حساب لگائے گا:

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

مندرجہ بالا کوڈ میں, distance()، getlat()، اور getlan() کو فنکشن کی طرح پڑھیں اور کہ lt، عرض البلد ہے، اور lg طول البلد ہے۔

Read := as "assigned as"

اسے اس طرح assigned، یا یہ اس طرح (becomes) پڑھیں۔

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

اس کو اس طرح پڑھیں lt 1 کی قیمت getlat(city1) اور lg 1 کی قیمت getlon(city1) کی ہوگی۔

نوٹ کریں کہ آپ کو یہ جاننے کی ضرورت نہیں ہے کہ ان افعال کو کس طرح نافذ کیا گیا تھا۔آپ یہ فرض کر رہے ہیں کہ ہمارے لئے کسی اور نے ان کی تعریف کی ہے۔

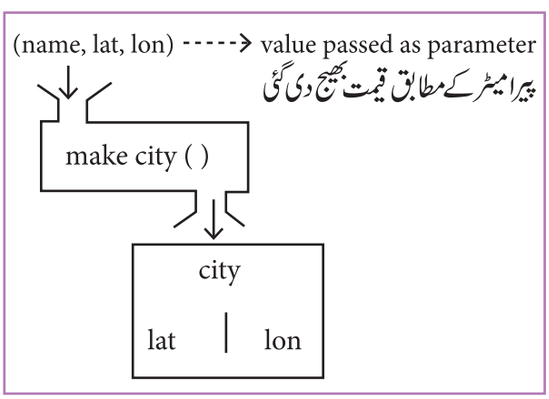
یہ ٹھیک ہے۔آخری صارف (end user) نہیں جانتا ہے کہ فنکشن کیسے نافذ کیے گئے ہیں۔تاہم، فنکشن کو ابھی بھی کسی کے ذریعہ بیان کرنا ہے۔

آیئے مذکورہ کوڈ میں کنسٹرکٹرز اور سلیکٹرز کی شناخت کریں۔

جیسا کہ آپ پہلے ہی جانتے ہیں کہ تعمیر کرنے والے وہ فنکشن ہوتے ہیں جو حاصل کردہ ڈیٹا کے قسم کی تعمیر کرتے ہیں۔مذکورہ کاذب کوڈ میں وہ فنکشن جس سے city کا مقصد پیدا ہوتا ہے وہ کنسٹرکٹر ہے۔

city = makecity (name, lat, lon)

یہاں makecity (name, lat, lon) تعمیر کرنے والے ہیں جو object city کی تعمیر کرتا ہے۔



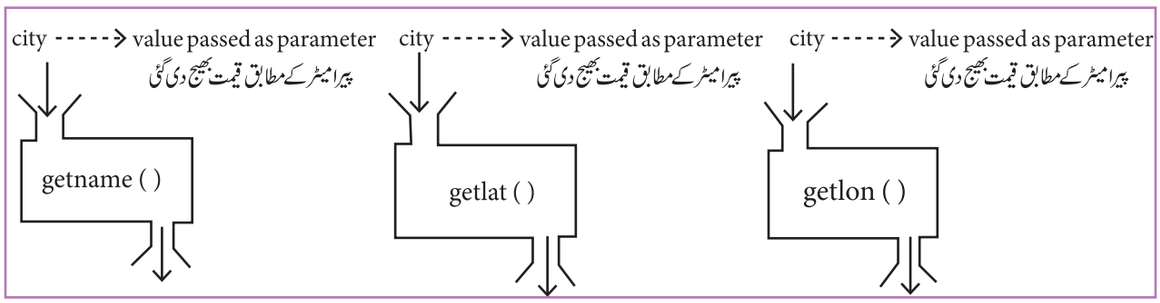
خاکہ 1 تعمیر کرنے والے

سلیکٹرز ڈیٹا کی قسم سے معلومات کو بازیافت کرنے والوں کے سوا اور کچھ نہیں ہوتے ہیں۔لہذا مذکورہ کوڈ میں

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

سلیکٹرز ہیں کیونکہ یہ افعال شہر کے آبجیکٹ کی معلومات نکالتے ہیں۔

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:47 PM



آیئے ہم ایک میلان(slope) کے لئے کنسٹرکٹر اور سلیکٹر کی شناخت کے لئے ایک اور مثال پر غور کریں۔

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

نوٹ:

ڈیٹا کو حاصل کرنے کی قسم (ADT) کی وضاحت کرنے سے ڈیٹا کو حاصل کرنے کی تائید ہوتی ہے، جو تعمیر کرنے والے اور انتخاب کرنے والے کا مجموعہ ہے۔ تعمیر کرنے والے ایک خاکہ تیار کرتے ہیں، مل کر معلومات کے مختلف ٹکڑوں کو جمع کرتے ہیں، جبکہ سلیکٹرز اس چیز سے معلومات کے انفرادی ٹکڑے نکالتے ہیں۔

2.4 ناطق اعداد کو استعمال کرتے ہوئے ڈیٹا کے قسم کو حاصل کرنے کی نمائندگی
Representation of Abstract datatype using Rational numbers

ڈیٹا کو حاصل کرنے کا بنیادی خیال پروگراموں کی تشکیل کا ہے تا کہ وہ ڈیٹا کو حاصل کرنے کا کام کریں۔ یعنی، ہمارے پروگراموں میں ڈیٹا کو اس طرح استعمال کرنا چاہئے، تا کہ جتنا ممکن ہو سکے، اتنا ڈیٹا کے بارے میں کچھ مفروضے بنا سکیں۔ اسی وقت، ایک ٹھوس ڈیٹا نمائندگی پروگرام کے ایک آزاد حصے کے طور پر بیان کی جاتی ہے۔

نوٹ:

حقیقی ڈیٹا (Concrete data) کی نمائندگی میں، ہر فنکشن کی توضیح معلوم ہوتی ہے۔

کوئی بھی پروگرام دو حصوں پر مشتمل ہوتا ہے۔ ایک پروگرام کے دو حصے ہیں، وہ حصہ جو ڈیٹا کو حاصل کرنے کے طور پر کام کرتا ہے اور وہ حصہ جو حقیقی ڈیٹا کی نمائندگی کی وضاحت کرتا ہے، وہ افعال کے ایک چھوٹے سے سیٹ کے ذریعہ منسلک ہوتا ہے جو حقیقی ڈیٹا کی نمائندگی کے لحاظ سے ڈیٹا حاصل کرنے کے عمل کو نافذ کرتا ہے۔ اس تکنیک کی وضاحت کرنے کے لئے، آیئے ناطق اعداد کو محسوب کرنے کے لئے فنکشن کا ایک سیٹ ڈیزائن کرنے والی ایک مثال پر غور کریں۔

مثال

ناطق اعداد، سالم اعداد کی ایک نسبت ہے، اور ناطق اعداد میں حقیقی اعداد کی ایک اہم ذیلی کلاس بھی پائی جاتی ہے۔ ناطق اعداد جیسا کہ 8/3 یا 19/23 کو اس طرح لکھا جاتا ہے۔ شمار کنندہ/نسب نما

<numerator>/<denominator>

جہاں دونوں <numerator> اور <denominator> سالمی عددی اقدار کیلئے مقام رکھتے ہیں۔ ناطق اعداد کی قیمت متعین کرنے کے لئے دونوں حصوں کی ضرورت ہے۔ دراصل سالم اعداد کو تقسیم کرنے سے ایک درمیانی تقریبی عدد حاصل ہوتا ہے، جس سے سالم اعداد کی سا لمیت ختم ہو جاتی ہے۔

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:48 PM

8/3 = 2.6666666666666665

تاہم، آپ شمار کنندہ اور نسب نما کو ایک ساتھ جوڑ کر ناطق اعداد کے لئے قطعی نمائندگی تشکیل دے سکتے ہیں۔

جیسا کہ ہم جانتے ہیں کہ فنکشن کو حاصل کرنے کے استعمال کرنے سے، اور ہمارے پروگرام کے کچھ حصوں پر عمل درآمد ہونے سے پہلے ہم اچھے طریقے سے پروگرامنگ شروع کر سکتے ہیں۔ آیئے یہ فرض کرتے ہوئے شروع کریں کہ آپ کے پاس پہلے سے ہی شمار کنندہ اور نسب نما کے ذریعہ ناطق عدد کرنے کا ایک طریقہ ہے۔ آپ یہ بھی فرض کر سکتے ہیں کہ، ایک ناطق عدد دیئے جانے پر، آپ کے پاس اس کے شمار کنندہ اور نسب نما کے جزو پائے جاتے ہیں۔

آیئے یہ بھی فرض کریں کہ کنسٹرکٹر اور سلکٹر بھی دستیاب ہیں۔

ہم یہاں پروگراموں کی تشکیل کے لئے ایک بہترین حکمت عملی استعمال کر رہے ہیں: خواہش مندانہ سوچ (Wishful thinking) ہم نے ابھی تک یہ نہیں کہا ہے کہ ناطق اعداد کی نمائندگی کس طرح کی جاتی ہے، یا کنسٹرکٹر اور سلیکٹرز کو کیسے نافذ کیا جانا چاہئے۔

نوٹ:

wishful thinking عقائد کی تشکیل اور اس کے مطابق فیصلے کرنا ہے جو حقیقت کی طرف راغب کرنے کی بجائے تصور کرنے میں خوش ہو سکتی ہے۔

مثال : ناطق اعداد کے لئے ایک ADT Example: An ADT for rational numbers

- - 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

اب آپ کے پاس ناطق اعداد پر حسابی اعمال کرنا ہے، جس کی توضیح سلکٹر فنکشن numer اور denom کی طرح کی گئی ہے۔ مگر آپ نے ابھی تک ان فنکشنوں کی وضاحت نہیں کی ہے۔ آپ کو یہ کرنا ہے کہ شمار کنندہ اور نسب نما کو جوڑ کر ایک مرکب قیمت میں تبدیل کرنا ہے۔

مندرجہ بالا کنسٹرکٹر اور سلیکٹر کا استعمال کرتے ہوئے ناطق اعداد کی نمائندگی کے لئے کاذب کوڈ (pseudo code) یہ ہے۔

```
x,y:=8,3
rational(n,d)
  numer(x)/numer(y)
- - output :  2.6666666666666665
```

2.5 فہرستیں، ٹوپلس (Lists, Tuples)

ہمارے ڈیٹا کو حاصل کرنے کی حقیقی سطح کو نافذ کرنے کے قابل بنانے کے لئے، کچھ زبانیں جیسے python وغیرہ ایک مرکب ڈھانچہ فراہم کرتی ہیں جسے جوڑی Pair کہا جاتا ہے، جو ایک فہرست list یا Tuple سے بنا ہوا ہے۔ Pairs کو عمل میں لانے کے لئے سب سے پہلا طریقہ فہرست کی تعمیر (List construct) ہے۔

2.5.1 لِسٹ (فہرست) (List)

فہرست کو کوما کے ذریعہ الگ کر کے مربع بریکٹ کے اندر اظہار خیالات رکھ کر تیار کیا گیا ہے۔ اس طرح کے اظہار کو لسٹ لفظی کہا جاتا ہے۔ فہرست متعدد اقدار کو محفوظ کر سکتی ہے۔ ہر قیمت کسی بھی قسم کی ہو سکتی ہے اور یہاں تک کہ ایک اور فہرست ہو سکتی ہے۔

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:48 PM

فہرست (List) کے لئے مثال [10,20] ہے۔

فہرست کے عناصر کو دو طریقوں سے حاصل کیا جاسکتا ہے۔ پہلا راستہ ہمارے متعدد تفویضات کے واقف طریقہ کے ذریعہ ہے، جو فہرست کو اپنے عناصر میں پیک کرتا ہے اور ہر عنصر کو ایک مختلف نام سے باندھ دیتا ہے۔

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

مذکورہ بالا مثال میں x , 10 اور y , 20 ہوجائے گا۔

ایک فہرست میں عناصر تک رسائی حاصل کرنے کا دوسرا طریقہ عنصر سلیکشن آپریٹر کے ذریعہ ہے، جس کا اظہار بھی مربع بریکٹ [square brackets] کے ذریعے کیا گیا ہے۔ اس کے برعکس، مربع بریکٹ کا اظہار براہ راست کسی دوسرے اظہار کی پیروی کرتے ہوئے فہرست کی قیمت کا اندازہ نہیں کرتا، بلکہ اس کے بجائے پہلے والے اظہار کی قدر سے ایک عنصر کا انتخاب کرتا ہے۔

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

مذکورہ دونوں مثالوں میں حسابی طور پر ہم کسی سیٹ کی طرح فہرست کی نمائندگی کر سکتے ہیں۔

جس میں - lst[(0, 10), (1, 20)]



دو قیمتوں کو ایک ساتھ باندھ کر یکجا کرنے کو جوڑی (pair) کہا جاتا ہے۔ اس طرح کرنے کا عام طریقہ List ہے۔ چنانچہ list کو pair بھی کہا جاسکتا ہے۔

List کو استعمال کر کے ناطق اعداد کی نمائندگی

آپ کاذب کوڈ (pseudo code) سے ایک ناطق عدد کو دو سالم اعداد کی جوڑی کی طرح نمائندگی کر سکتے ہیں۔ یعنی شمار کنندہ اور نسب نما

code : a numerator and a denominator.

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

2.5.2 ٹوپل (Tuple)

یاد رکھنا، ایک جوڑی ایک مرکب ڈیٹا کی ایک قسم ہے جس میں ڈیٹا کے دیگر دو ٹکڑے رکھتا ہے۔ اب تک، ہم نے آپ کو جوڑے کے ڈیٹا کی نمائندگی کرنے کے دو طریقے بتائے ہیں۔ پہلا طریقہ لسٹ کنسٹریکٹ کے استعمال سے اور دوسرا طریقہ جوڑی کو لاگو کر کے ٹوپل کنسٹریکٹ کا۔

ایک Tuple میں قیمتوں کو کاما (,) کے ساتھ جدا کر کے انہیں ایک قوسین کے اندر رکھا جاتا ہے۔ Tuple ایک فہرست ہی کی طرح ہے۔ ان دونوں کے مابین فرق یہ ہے کہ آپ ایک مرتبہ tuple کے عناصر کو تبدیل نہیں کر سکتے ہیں جب اسے تفویض کیا جاتا ہے جبکہ فہرست list میں، عناصر کو تبدیل کیا جاسکتا ہے۔

Example colour= ('red', 'blue', 'Green')

ایک جوڑی کے طور پر Tuple کی نمائندگی.

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

نوٹ کریں کہ مربع بریکٹ کی ترقیم کو جوڑی میں محفوظ کردہ ڈیٹا تک رسائی کے لئے استعمال کیا جاتا ہے۔ ڈیٹا کو صفر اشاریہ دیا جاتا ہے، مطلب ہے کہ آپ نے پہلے عنصر کو اعداد [0] اور دوسرے عدد [1] کی رسائی (access) حاصل کی ہے۔

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:49 PM

2.6 ڈیٹا کو ساخت کی شکل میں حاصل کرنا (Data abstraction in structure)

جیسا کہ آپ پہلے ہی جان چکے ہیں کہ فہرست (list) اعداد و شمار کو خلاصہ کرنے کی اجازت دیتی ہے جس میں آپ یادداشت کے خانوں (memory cells) کے ایک سیٹ کو نام دے سکتے ہیں۔ مثال کے طور پر، گیم mastermind میں، آپ کو چار رنگوں کی فہرست کا ٹریک رکھنا ہوگا جس کا کھلاڑی اندازہ کرتا ہے۔

چار الگ الگ متغیرات (color1، color2، color3، اور color4) استعمال کرنے کے بجائے آپ ایک متغیر 'predict' استعمال کر سکتے ہیں۔ جیسے،

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

ہمیں جو فہرستیں کرنے کی اجازت نہیں دیتا ہے وہ ایک کثیر آئٹم آبجیکٹ کے مختلف حصوں کا نام ہے۔predict کی صورت میں، آپ کو واقعی حصوں کا نام لینے کی ضرورت نہیں ہے۔

ہر رنگ کے لئے ایک انڈیکس کا استعمال کرتے ہیں۔

لیکن کسی پیچیدہ چیز کی صورت میں جیسے کسی شخص کی طرح، ہمارے پاس ایک کثیر آئٹم آبجیکٹ موجود ہے جہاں ہر 'item' ایک نامزد شے ہے: جیسے first name، last name، the id اور email وغیرہ۔ کسی شخص کی نمائندگی کے لئے ہر کوئی ایک فہرست استعمال کر سکتا ہے۔

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

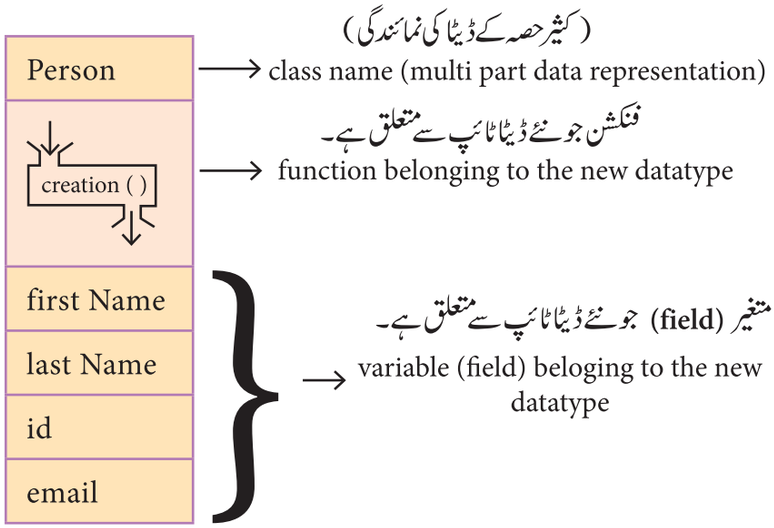
لیکن اس طرح کی نمائندگی واضح طور پر یہ وضاحت نہیں کرتی ہے کہ ہر ایک حصہ کی نمائندگی کرتا ہے۔

اس مسئلے کے لئے لسٹ استعمال کرنے کے بجائے، آپ structure construct (ساختی تعمیر) استعمال کر سکتے ہیں۔ (OOPS کی زبان میں اسے class construct کہا جاتا ہے)۔

کثیر حصے کی اشیاء کی نمائندگی کرنے کے لئے جہاں ہر حصے کا نام ہے (ایک نام دیا گیا ہے)۔ درج ذیل کاذب کوڈ (pseudo code) پر غور کریں:

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

نئے ڈیٹا ٹائپ میں ایک شخص کو تصویری طور پر اس طرح نمائندگی کی گئی ہے۔



12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:49 PM

فرض کیجئے کہ () main میں یہ ہیں۔ Let main() contains	
p1:=Person()	بیان ایک شئے کو create کرتا ہے۔
firstName := " Padmashri "	ایک فیلڈ سیٹ کرتے ہیں جس کے first name میں قیمت Padmashri سیٹ کیا جاتا ہے۔
lastName :="Baskar"	ایک فیلڈ سیٹ کرتے ہیں، جس کے last name میں قیمت Baskar سیٹ کیا جاتا ہے۔
id :="994-222-1234"	ایک فیلڈ سیٹ کرتے ہیں، جس کے id value میں 994-222-1234 سیٹ کیا جاتا ہے۔
email="compsci@gmail.com"	ایک فیلڈ سیٹ کرتے ہیں، جس کے email میں compsci@gmail.com سیٹ کیا جاتا ہے۔
- - output of firstName : Padmashri	

class structure (کلاس کی ساخت) کثیر حصے کی اشیاء کے لئے فارم کی وضاحت کرتا ہے جو کسی شخص کی نمائندگی کرتا ہے۔اس کی توضیح میں ایک نئی ڈیٹا ٹائپ کا اضافہ ہوتا ہے۔ اس معاملے میں person نامی ایک type نیا بنتا ہے۔ایک بار واضح ہونے کے بعد، ہم اس قسم کے نئے متغیر (instance) تشکیل دے سکتے ہیں۔اس مثال میں فرد کو کلاس یا ایک قسم کہا جاتا ہے، جبکہ P1 کو کسی شے یا مثال کے طور پر کہا جاتا ہے۔آپ ایک class person کو ایک cookie cutter کی طرح اور P1 کو ایک اور خاص cookie کی طرح سمجھ سکتے ہیں۔ cookie cutter کو استعمال کر کے تم کئی cookies بنا سکتے ہیں۔ اسی طرح class کو استعمال کر کے تم اسی type کے کئی objects بنا سکتے ہیں۔

اب تک، آپ دیکھ چکے ہیں کہ کس طرح ایک کلاس متعلقہ ڈیٹا آئٹموں کی گروپ بندی کر کے ڈیٹا کو حاصل کرنے کی وضاحت کرتی ہے۔ایک کلاس میں صرف ڈیٹا نہیں ہوتے، بلکہ اس کے اندر فنکشن کی وضاحت بھی ہوتی ہے۔ ہم یہ کہہ سکتے ہیں کہ اس طرح کے افعال کلاس کے ماتحت ہوتے ہیں، کیونکہ ان کا کام کلاس کے ڈیٹا کے ساتھ ہوتا ہے، جیسے کسی شخص (person object) کے ڈیٹا میں ترمیم یا تجزیہ کرنا۔

لہذا ہم ایک کلاس کی وضاحت باندھے ہوئے ایک ڈیٹا کی طرح کرتے ہیں، جس میں فنکشن (functions) اس ڈیٹا پر عمل کرتے ہیں۔ مذکورہ بالا ساری مثال اور وضاحت سے کوئی اس نتیجہ پر پہنچ سکتا ہے کہ کس طرح ڈیٹا کو حاصل کیا جا سکتا ہے اور پیچیدہ ڈیٹا کو کیسے بہت ہی آسان طریقے سے ہم برت سکتے ہیں۔

یاد رکھنے کے نکات: Points to remember:

- Abstract Data Types (ADT) اشیاء کے لئے (ایک قسم (یا کلاس) ہے جس کے طرز عمل کی قیمت قدر کے ایک سیٹ اور آپریشنز کے ایک سیٹ سے ہوتی ہے۔
- ADT کی تعریف میں صرف اس بات کا تذکرہ کیا گیا ہے کہ کونسا عمل (operation) کرنا ہے؟ لیکن یہ نہیں کہ ان کارروائیوں کو کس طرح نافذ کیا جائے گا۔
- ADT میں یہ واضح نہیں کیا گیا ہے کہ ڈیٹا کو میموری میں کس طرح منظّم کیا جائے گا اور کارروائیوں کو عملی جامہ پہنانے کے لئے کون سے الگورتھم استعمال کیے جائیں گے۔
- تعمیر کار وہ افعال ہوتے ہیں جو abstract data types کی تشکیل کرتے ہیں۔
- سلیکٹرز وہ افعال ہوتے ہیں جو data type سے معلومات بازیافت کرتے ہیں۔
- حقیقی ڈیٹا کی اقسام یا ساختیں (CDT's) نسبتاً simple آسان تصور کا براہ راست عمل ہے۔
- حاصل کردہ اعداد و شمار کی اقسام (ADT's) کسی تصور کا اعلیٰ سطحی نظریہ (اور استعمال) پیش کرتے ہیں، جو اس کے نفاذ پر غیر منحصر ہوتا ہے۔

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:50 PM

یادرکھنے کے نکات: Points to remember:

- حقیقی ڈیٹا کی قسم ایک ڈیٹا کی قسم ہوتی ہے جس کی نمائندگی معلوم ہوتی ہے اور حاصل کردہ ڈیٹا میں ڈیٹا کی قسم کی نمائندگی معلوم نہیں ہوتی ہے۔
- جوڑی (Pair) ایک مرکب ساخت ہے جو list یا tuple سے بنا ہوا ہے۔
- list کو کوما کے ذریعہ الگ کر کے مربع بریکٹ کے اندر جملوں (expressions) کو رکھ کر تیار کیا گیا ہے۔
- list کے عناصر کو دو طریقوں سے حاصل کیا جا سکتا ہے۔ پہلا طریقہ متعدد تفویض کے ذریعہ ہے اور دوسرا طریقہ عنصر سلیکشن آپریٹر کے ذریعہ ہے۔
- دو اقدار کو ایک ساتھ بنڈل کرنا ایک جوڑی کے طور پر سمجھا جاتا ہے۔
- list کسی ملٹی آئٹم آبجیکٹ کے مختلف حصوں کا نام لینے کی اجازت نہیں دیتی ہے۔

NK14H

Evaluation محاسبہ

حصہ - 1

صحیح جواب کا انتخاب کریں۔ (1 Mark)

1۔ مندرجہ ذیل میں سے کون سا فنکشن جو abstract data type تیار کرتا ہے؟

(A) تعمیر کرنے والے (Constructors) (B) تباہ کرنے والے (Destructors)

(C) مکرر (recursive) (D) نیسٹڈ (Nested)

2۔ مندرجہ ذیل میں سے کون سا فنکشن جو ڈیٹا ٹائپ سے معلومات کو بازیافت کرتا ہے؟

(A) تعمیر کرنے والے (Constructors) (B) انتخاب کرنے والے (Selectors)

(C) مکرر (recursive) (D) نیسٹڈ (Nested)

3۔ ڈیٹا کی وہ ساخت جو عناصر کو ایک قابل تبدیل متواتر ترتیب (mutable ordered sequence) میں لاتی ہے، اسے یہ کہتے ہیں۔

(A) بلٹ ان (built in) (B) فہرست (list) (C) ٹپل (Tuple) (D) اخذ کردہ ڈیٹا (Derived data)

4۔ غیر قابل تبدیل اشیاء کے ایک تواتر (sequence of immutable objects) کو یہ کہا جاتا ہے۔

(A) بلٹ ان (built in) (B) فہرست (list) (C) ٹپل (Tuple) (D) اخذ کردہ ڈیٹا (Derived data)

5۔ ڈیٹا کی قسم جس کی نمائندگی، معلوم شدہ ہے، اسے یہ کہا جاتا ہے۔

(A) بلٹ ان ڈیٹا ٹائپ (built in datatype) (B) اخذ کردہ ڈیٹا ٹائپ (Derived datatype)

(C) حقیقی ڈیٹا ٹائپ (Concrete datatype) (D) حاصل کردہ ڈیٹا ٹائپ (Abstract datatype)

6۔ ڈیٹا کی قسم، جس کی نمائندگی نامعلوم ہے، اسے یہ کہا جاتا ہے۔

(A) بلٹ ان ڈیٹا ٹائپ (built in datatype) (B) اخذ کردہ ڈیٹا ٹائپ (Derived datatype)

(C) حقیقی ڈیٹا ٹائپ (Concrete datatype) (D) حاصل کردہ ڈیٹا ٹائپ (Abstract datatype)

7۔ مندرجہ ذیل میں سے کونسی ایک مرکب ساخت ہے؟

(A) جوڑی (Pair) (B) ٹرپلٹ (Triplet) (C) اکہری (single) (D) چوکور (quadrat)

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:50 PM

8۔ دو قیمتوں کو ایک ساتھ باندھنے کو یہ کہا جاتا ہے۔

(A) جوڑی (Pair)　(B) ٹرپلٹ (Triplet)　(C) اکہری (single)　(D) چوکور (quadrat)

9۔ مندرجہ ذیل میں سے کون کسی ملٹی آئٹم، آبجیکٹ کے مختلف حصوں کا نام لینے کی اجازت دیتا ہے؟

(A) Tuples　(B) فہرستیں (Lists)　(C) طبقات (Classes)　(D) چوکور (quadrats)

10۔ مربع قوسین کے اندر جملوں (expressions) رکھ کر مندرجہ ذیل میں سے کون سا تعمیر کیا جاتا ہے؟

(A) Tuples　(B) فہرستیں (Lists)　(C) طبقات (Classes)　(D) چوکور (quadrats)

حصہ - II

مندرجہ ذیل سوالات کے جوابات دیجئے۔　(2 نمبرات)

1۔ حاصل کردہ ڈیٹا ٹائپ (Abstract datatype) قسم کیا ہے؟

2۔ مختلف تعمیر کنندگان اور سلیکٹرز.

3۔ جوڑی کیا ہے؟ ایک مثال دیں۔

4۔ ایک فہرست کیا ہے؟ ایک مثال دیں۔

5۔ ایک Tuple کیا ہے؟ ایک مثال دیں۔

حصہ - III

درج ذیل سوالات کے جوابات دیجئے۔　(3 نمبرات)

1۔ حقیقی ڈیٹا ٹائپ (Concrete data type) اور حاصل کردہ ڈیٹا ٹائپ (Abstract datatype) میں فرق کریں۔

2۔ پروگرام ڈیزائننگ کے لئے کون سی حکمت عملی استعمال کی جاتی ہے؟ اس حکمت عملی کی وضاحت کریں۔

3۔ شناخت کریں کہ مندرجہ ذیل میں سے کون کنسٹرکٹر اور سلیکٹر ہیں؟

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

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

(f) display()

4۔ فہرست کے عناصر تک رسائی کے مختلف طریقے کیا ہیں؟ مثال دیں۔

5۔ شناخت کریں کہ مندرجہ ذیل میں سے کون فہرست (list)، ٹوپل (tuple) اور کلاس (Class) ہیں؟

(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]

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:51 PM

حصہ - IV

درج ذیل سوالات کے جوابات دیں (5 مارکس)

1۔ آپ کس طرح ڈیٹا کو حاصل (Data abstraction) کریں گے؟ مناسب مثال کے ساتھ اس کی وضاحت کریں۔

2۔ ایک فہرست (list) کیا ہے؟ فہرست (list) کو جوڑی (pairs) کیوں کہا جاسکتا ہے۔ مناسب مثال کے ساتھ بیان کریں۔

3۔ آپ ملٹی آئٹم تک کیسے رسائی حاصل کریں گے ؟ مثال کے ساتھ بیان کریں۔

Reference Books کتب حوالہ

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*

12 CS UNIT 2 CDR Set
Sunday, April 16, 2023 9:34:51 PM

اکائی I

باب 3

اسکوپنگ SCOPING

I7SCG

سیکھنے کے مقاصد

- اس باب کی تکمیل کے بعد، طالب علم اس قابل ہو جائے گا۔
- سمجھنا کہ اسکوپنگ کیا ہے۔
- LEGB قواعد کو نافذ کرنے کے قابل بنانا
- سمجھنا کہ ماڈیول کیا ہے
- پروگرامنگ کی زبان میں ایکسیس کنٹرول کے نفاذ کو سمجھنا۔

3.1 تعارف

اسکوپ سے مراد کسی پروگرام کے ایک حصے میں متغیرات، پیرامیٹرز اور تفاعل کو اسی پروگرام کے دوسرے حصے سے دیکھنا یا انہیں دائرہ عمل میں لانا ہے۔ دوسرے لفظوں میں، آپ کے پروگرام کے کون سے حصے سے انہیں دائرہ عمل میں لا سکتے یا استعمال کر سکتے ہیں۔ عام طور پر، پروگرام میں بیان کردہ ہر متغیر کی عالمی گنجائش ہوتی ہے۔ ایک بار واضح ہونے کے بعد، آپ کے پروگرام کا ہر حصہ اس متغیر تک رسائی حاصل کر سکتا ہے۔ لیکن متغیر کے دائرہ کار کو کسی ایک توضیح تک محدود رکھنا ایک اچھا عمل ہے۔ اس طرح، فنکشن کے اندر ہونے والی تبدیلیاں غیر متوقع طریقوں سے فنکشن کے متغیر کو متاثر نہیں کر سکتی ہیں۔

3.2 متغیر دائرہ کار Variable Scope

پروگرامنگ زبان میں تغیرات کے دائرہ کار کو سمجھنے کے لئے، اس کے بارے میں جاننا ضروری ہے کہ متغیرات واقعتاً کیا ہیں۔ بنیادی طور پر، وہ میموری میں کسی شے کے پتے (حوالہ جات یا اشارے) ہیں۔

جب آپ متغیر کو =: اس کے ساتھ تفویض کرتے ہیں۔ مثال کے طور پر شئے کے ساتھ متغیر کو باندھ رہے ہیں یا (نقشہ سازی) (mapping) کر رہے ہیں۔ کثیر متغیرات کو اسی مثال کے ساتھ نقشہ سازی کی جا سکتی ہے۔

نوٹ

متغیر نام کو کسی شے کے ساتھ باندھنے کے عمل کو میپنگ کہا جاتا ہے۔ =: (کولن اور مساوی کی علامت) متغیر اور آبجیکٹ کی نقشہ سازی کے لئے پروگرامنگ زبانوں میں استعمال کیا جاتا ہے۔

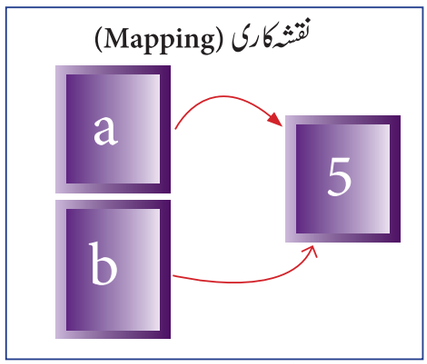
پروگرامنگ زبانیں نام کی جگہوں (name space) کے ساتھ ان تمام نقشے کو ٹریک کرتی رہتی ہیں۔ name space میں متغیر کے ناموں کو اشیاء میں نقش کرنے کے لئے کنٹینر (مرتبان) ہیں۔ آپ ان کو ایک لغت (dictionary) کی طرح سوچ سکتے ہیں، جس میں الفاظ کی فہرست اور اس کے معنی شامل ہیں۔ الفاظ کو لغت میں اس کے معنی کے ساتھ نقش کیا گیا ہے، جبکہ پروگرامنگ کی زبان میں ناموں کو اشیاء (name:=object) کے ساتھ نقش کیا گیا ہے۔ اس سے آپ ان ناموں کے ذریعہ اشیاء تک رسائی کی اجازت دیتے ہیں جو آپ انہیں تفویض کرتے ہیں۔

مندرجہ ذیل مثال میں، a کو سالم عدد 5 کے ساتھ نقش کیا گیا ہے۔ a متغیر نام ہے، جب کہ سالم عدد 5 ایک شئے ہے۔

پھر، b کے مساوی a کیا گیا ہے۔ اصل معنی یہ ہے کہ b اسی سالم عدد a کے ساتھ پابند ہے، جو 5 ہے۔

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


12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:05 PM

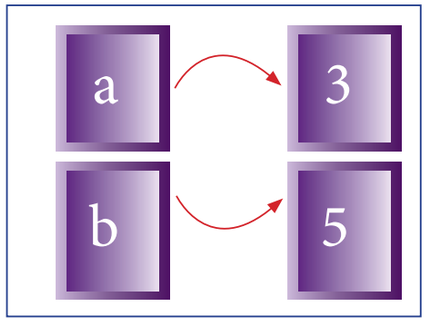


اب مذکورہ بالا بیانات کی مندرجہ ذیل توسیع (extension) پر غور کریں ۔

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

اگر آپ پھر a کو 3 کے مساوی کرتے ہیں تو، ایک اچھے پروگرامر سے توقع کی جاسکتی ہے کہ b بھی 3 کے مساوی ہو، لیکن ایسا نہیں ہے۔ b کی عددی قیمت میں ابھی بھی سالم عدد 5 کے ساتھ نقشہ کاری کی ہوئی (یا اشارہ کیا گیا) ہے۔ صرف ایک چیز جو تبدیل ہوئی ہے، وہ a ہے۔ جو اب سالمی عدد کی قیمت 3 میں میپ ہوجاتی ہے۔

a کی قیمت بدلنے کے بعد نقشہ سازی کرنا



متغیر کو دائرہ کار میں لانے کا مقصد یہ ہے کہ کوڈ کا وہ حصہ ہے، جہاں وہ دکھائی دے۔ دراصل، اس کا حوالہ دینے کے لئے، آپ کو اس کے بعد کوئی بھی سابقہ استعمال کرنے کی ضرورت نہیں ہے۔ آیئے ایک مثال لیتے ہیں،

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

جب آپ a کی قیمت کو ظاہر کرنے کی کوشش کرتے ہیں تو پروگرام میں error کے لئے یہ جھنڈا (flag) دکھائی دیتا ہے۔ "name 'a' is not defined"۔ اس کی وجہ یہ ہے کہ متغیر کی مدت حیات (life time of the variable) صرف طریقہ کار کے اختتام تک ہوتی ہے۔ نیز جس مدت کے لئے متغیر زندہ ہے، اسے اس کا مدت حیات (life time) کہا جاتا ہے۔

3.3 LEGB کا قاعدہ

دائرہ عمل (scope) اس ترتیب کی بھی وضاحت کرتا ہے کہ کس ترتیب میں متغیرات کو نقش کرنا پڑتا ہے۔ آیئے ایک سادہ مثال کے ذریعہ میں ذیل کو سیکھیں۔

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

جب مذکورہ بالا بیانات پر عمل درآمد کیا جاتا ہے، تو بیان (4) اور (5) نتیجہ کو ظاہر کرتے ہیں۔

Output آؤٹ پٹ

بیرونی × متغیر (outer × variable)

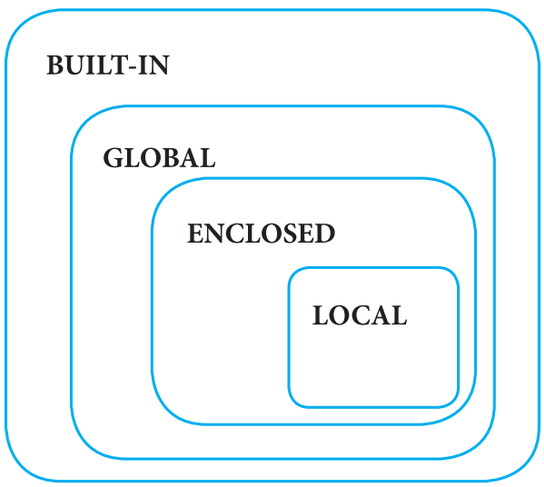
اندرونی × متغیر (inner × variable)

مذکورہ بالا بیانات مختلف آؤٹ پٹ دیتے ہیں کیونکہ ایک ہی متغیر نام x مختلف اسکوپس میں رہتا ہے، ایک فنکشن ڈسپلے کے اندر () اور دوسرا اوپری سطح میں۔ قیمت 'outer value' اس وقت print ہوگا جب x کو فنکشن کی وضاحت کے باہر کا حوالہ دیا جائے۔ جب display() عمل درآمد ہوتا ہے، تو 'inner x variable' print ہوگا، جب x کی قیمت فنکشن کی وضاحت کے اندر کا حوالہ دیا جائے۔ مذکورہ بالا مثال سے، ہم اندازہ کرسکتے ہیں کہ ایک قاعدہ عمل میں لایا گیا ہے، تاکہ فیصلہ کرنے کے لئے کہ کون سے دائرہ کار کو منتخب کیا جائے۔

LEGB کا قاعدہ اس ترتیب کا فیصلہ کرنے کے لئے استعمال کیا جاتا ہے جس میں دائرہ کار کے حل کے لئے کونسے اسکوپ (scope) کو تلاش کرنا ہے۔ دائرہ کار (اعلی سے کم سے کم) کے لحاظ سے دائرہ کار درج ذیل کی طرح فہرست کیا گیا ہے۔

12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:05 PM

Local(**L**)	فنکشن یا کلاس کے اندر وضاحت کی جاتی ہے
Enclosed(**E**)	بند فنکشن کے اندر وضاحت کی جاتی ہے۔ (Nested function concept)
Global(**G**)	اعلیٰ ترین سطح پر وضاحت کی جاتی ہے۔
Built-in (**B**)	بلٹ ان فنکشن (ماڈیولز) میں محفوظ مقام



3.4 متغیر دائرہ کار کی اقسام Types of Variable Scope

متغیر اسکوپ کی 4 قسمیں ہیں، آیئے ان پر ایک ایک کر کے ان پر تبادلہ خیال کریں۔

3.4.1 مقامی اسکوپ (Local scope)

مقامی دائرہ کار سے مراد موجودہ فعل میں متعین متغیرات ہیں۔ ہمیشہ، ایک فنکشن سب سے پہلے اپنے مقامی دائرہ کار میں متغیر نام تلاش کرے گا۔ صرف اس صورت میں جب یہ اسے نہیں مل پاتا ہے، بیرونی اسکوپس (دائرہ کار) کی جانچ کی جاتی ہے۔

اس مثال کو دیکھیں۔

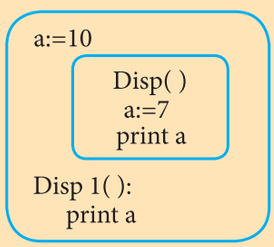
1. **Disp():** 2. a:=7 3. print a 4. Disp()	Entire program Disp( ): a:=7 print a Disp ( )	Output of the Program 7

مذکورہ کوڈ کے نفاذ پر متغیر a قیمت 7 کو ظاہر کرتا ہے، کیونکہ اس کی وضاحت، مقامی دائرہ کار میں دستیاب ہے۔

100
95
75
25
5
0
12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:05 PM

3.4.2 عالمی دائرہ کار (Global scope)

ایک متغیر جس کو کسی پروگرام میں تمام افعال سے باہر قرار دیا جاتا ہے اسے عالمی متغیر (global variable) کے نام سے جانا جاتا ہے۔اس کا مطلب یہ ہے کہ، پروگرام میں تمام افعال کے اندر یا اس سے باہر عالمی متغیر تک رسائی حاصل کی جاسکتی ہے۔مندرجہ ذیل مثال پر غور کریں۔

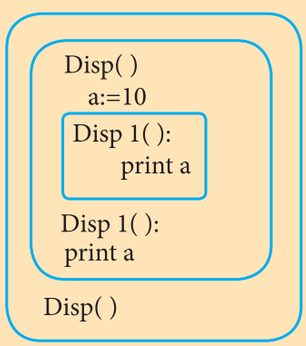
1. a:=10 2. **Disp()**: 3. a:=7 4. print a 5. Disp() 6. print a	Entire program 	Output of the Program 7 10

مندرجہ بالا کوڈ کے نفاذ پر، متغیر a جو فعل کے اندر بیان ہوتا ہے، function call Disp() کے لئے 7 کی قیمت دکھاتا ہے اور پھر یہ 10 دکھاتا ہے، کیونکہ عالمی سطح پر a کی وضاحت کی گئی ہے۔

3.4.3 منسلک دائرہ کار (Enclosed Scope)

پروگرامنگ کی تمام زبانیں افعال کو nest کرنے کی اجازت دیتی ہیں۔کسی فنکشن (طریقہ) کے ساتھ ایک اور فنکشن کو نیسٹ فنکشن (Nested function) کہا جاتا ہے۔ایک متغیر جو کسی فنکشن کے اندر اعلان کیا جاتا ہے جس میں کسی اور فنکشن کی وضاحت ہوتی ہے، اندرونی فنکشن بھی بیرونی فنکشن کے متغیر تک رسائی حاصل کرسکتا ہے۔ اس دائرہ کار کو منسلک دائرہ کار (enclosed scope) کہا جاتا ہے۔

جب ایک مرتب کرنے والا (Compiler) یا ترجمان (interpreter) کسی پروگرام میں متغیر کی تلاش کرتا ہے، تو وہ پہلے مقامی (local) کو تلاش کرتا ہے، اور پھر منسلک اسکوپس (enclosed scope) کو تلاش کرتا ہے۔مندرجہ ذیل مثال پر غور کریں۔

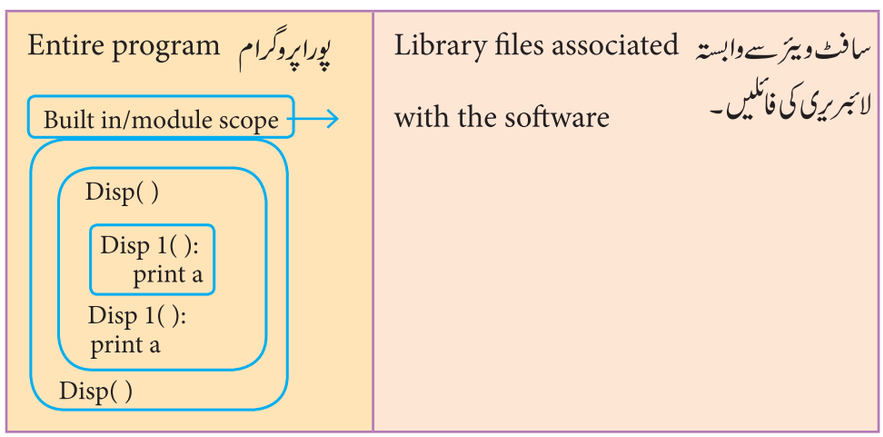
1. Disp(): 2. a:=10 3. Disp1(): 4. print a 5. Disp1() 6. print a 7. Disp()	Entire program 	Output of the Program 10 10

12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:06 PM

مندرجہ بالا مثال میں Disp1() کی وضاحت Disp() میں کی گئی ہے۔ Disp() میں متعین متغیر 'a' حتی کہ Disp1() بھی استعمال کرسکتا ہے کیونکہ یہ Disp() کا ممبر بھی ہے۔

3.4.4 بلٹ ان اسکوپ (Built-in scope)

آخر میں، ہم وسیع تر دائرہ کار کے بارے میں بات کرتے ہیں۔ جب ہم مرتب کرنے والے (compiler) یا ترجمان (interpreter) کو شروع کرتے ہیں تو بلٹ ان اسکوپ میں تمام نام ہوتے ہیں جو پہلے سے بنی ہوئی اسکوپ پروگرام میں شامل ہوتے ہیں۔ کوئی بھی متغیر یا ماڈیول جو کسی پروگرامنگ زبان کی لائبریری کے افعال میں متعین ہوتا ہے اس میں بلٹ ان یا ماڈیول اسکوپ ہوتا ہے۔ جیسے ہی لائبریری کی فائلوں کو پروگرام میں درآمد کیا جاتا ہے وہ بھر (load) جاتے ہیں۔



مطور پر صرف افعال یا ماڈیول سوفٹ ویئر کے ساتھ، بطور پیکج آتے ہیں۔ لہذا وہ بلٹ ان اسکوپ میں آئیں گے۔

3.5 ماڈیول (Module)

ماڈیول ایک پروگرام کا ایک حصہ ہے۔ پروگرام ایک یا زیادہ آزادانہ طور پر تیار کردہ ماڈیولز پر مشتمل ہوتے ہیں۔ ایک ہی ماڈیول میں ایک یا متعدد بیانات (statements) ایک دوسرے سے تعلق پایا جاتا ہے۔ ماڈیول انفرادی سطح پر مکمل طور پر کام کرتے ہیں اور دوسرے ماڈیولز کے ساتھ مل سکتے ہیں۔ پروگرامنگ اور ڈیبگنگ کے کام کو آسان بنانے کے لئے ایک سافٹ ویئر پروگرام ماڈیول میں تقسیم کیا جاتا ہے۔ ایک پروگرام کو چھوٹے فنکشنل ماڈیولز میں تقسیم کیا جاتا ہے، جو output حاصل کرنے کے لئے مل کر کام کرتے ہیں۔ کمپیوٹر پروگرام کو علیحدہ سب پروگراموں میں تقسیم کرنے کے عمل کو ماڈیولر پروگرامنگ کہتے ہیں۔ ماڈیولر پروگرامنگ پروگرامرز کو پروگرام کے کام کو تقسیم کرنے اور ڈیبگ (Debug) کرنے کے قابل بناتا ہے آزادانہ طور پر ماڈیولز کی مثالیں procedures، subroutines اور functions ہیں

3.5.1 ماڈیول کی خصوصیات (Characteristics of Modules)

ایک ماڈیول کی مطلوبہ خصوصیات درج ذیل ہیں۔

1۔ ماڈیولز ہدایات، پروسیسنگ منطق، اور ڈیٹا پر مشتمل ہیں۔
2۔ ماڈیول الگ الگ لائبریری میں مرتب اور اسٹور کیے جاسکتے ہیں۔
3۔ ماڈیول ایک پروگرام میں شامل کیا جاسکتا ہے.
4۔ ماڈیول کے segments کو نام اور کچھ پیرامیٹرز کی مدد سے استعمال کیا جاسکتا ہے۔
5۔ ماڈیول کے segments دوسرے ماڈیولز کے ذریعہ استعمال ہوسکتے ہیں۔

12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:06 PM

3.5.2 ماڈیولر پروگرامنگ کے استعمال کے فوائد میں یہ شامل ہیں (The benefits of using modular programming include)

- کم کوڈ لکھا جائے۔
- کوڈ کو کئی بار ٹائپ کرنے کی ضرورت کو ختم کرتے ہوئے دوبارہ استعمال کے لئے ایک ہی طریقہ کار تیار کیا جائے۔
- پروگراموں کو زیادہ آسانی سے ڈیزائن کیا جاسکتا ہے کیونکہ ایک چھوٹی ٹیم پورے کوڈ کے صرف ایک چھوٹے سے حصے کے ساتھ کام کرتی ہے۔
- ماڈیولر پروگرامنگ بہت سے پروگرامرز کو اسی application پر تعاون کرنے کی اجازت دیتا ہے۔
- کوڈ متعدد فائلوں میں محفوظ ہے۔
- کوڈ مختصر، آسان اور سمجھنے میں آسان ہے۔
- غلطیوں کی نشاندہی آسانی سے کی جاسکتی ہے، کیونکہ وہ سبروٹین یا فنکشن میں محدود کردیا جاتا ہے۔
- ایک ہی کوڈ کو بہت سے ایپلی کیشنز میں استعمال کیا جاسکتا ہے۔
- متغیر کی اسکوپنگ آسانی سے کنٹرول کی جاسکتی ہے۔

3.5.3 رسائی کنٹرول (Access Control)

رسائی کنٹرول ایک حفاظتی تکنیک ہے جو کمپیوٹنگ ماحول میں یہ دیکھتا ہے کہ وسائل کو کون استعمال کرسکتا ہے، اور اس کو کنٹرول کرتا ہے۔ یہ سیکیورٹی میں ایک بنیادی تصور ہے جو شے خطرہ سے محفوظ رکھتا ہے۔ دوسرے لفظوں میں access control پر قابو پانا data تک رسائی کی منتخب پابندی ہے۔ آبجیکٹ اوریئنٹڈ پروگرامنگ لینگویج (OOPS) (Object Oriented Programming Languages) میں، اس کو لاگو کیا جاتا ہے۔ کلاسیکی آبجیکٹ پر مبنی زبانیں، جیسے C++ اور Java، طبقاتی ممبروں تک عوامی، نجی اور محفوظ الفاظ کے ذریعہ رسائی کو کنٹرول کرتی ہیں۔

کلاس کے نجی ممبروں (Private members) کو کلاس کے باہر سے آنے کی اجازت نہیں ہے۔ انہیں صرف کلاس کے اندر ہی access کیا جاسکتا ہے۔

عوامی ممبران (public members) (عام طور پر کلاس میں اعلان کردہ طریقے) کلاس سے باہر access کرنے کے قابل ہوتے ہیں۔ ایک ہی طبقے کے مقصد کے لئے عوامی طریقہ کار کی ضرورت ہے۔ نجی مثال کے متغیرات اور عوامی طریقوں کا یہ اہتمام ڈیٹا کو ایک خول میں بند کیا جانے (encapsulation) کے اصول کو یقینی بناتا ہے۔

ایک کلاس کے محفوظ ممبر (protected member) طبقے کے اندر ہی قابل رسائی ہیں اور اس کی ذیلی کلاسوں کے لئے بھی دستیاب ہیں۔ کسی اور عمل کو اس تک رسائی کی اجازت نہیں ہے۔ یہ آبائی کلاس کے مخصوص وسائل کو چائلڈ کلاس کے ذریعے وراثت میں قابل بناتا ہے۔

Python میں کوئی میکانزم نہیں ہے، جو مؤثر طریقے سے کسی بھی صورت میں متغیر یا طریقہ تک رسائی پر پابندی لگائے۔ python نے متغیر یا طریقہ کار کے نام کو کسی اور واحد یا ڈبل انڈر سکور (single or double underscore) کے ساتھ متعین کرنے کی ایک آسانی تجویز کی ہے تاکہ محفوظ اور نجی رسائی سے متعلق افراد کے طرز عمل پر عمل پیرا ہوسکے۔

python کلاس میں شامل تمام ممبران ڈیفالٹ طور پر پبلک ہوتے ہیں جبکہ C++ اور Java میں وہ نجی ہوتے ہیں۔ python میں طبقاتی ماحول سے باہر کسی بھی ممبر تک رسائی حاصل کی جاسکتی ہے جو C++ اور جاوا میں ممکن نہیں ہے۔

12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:07 PM

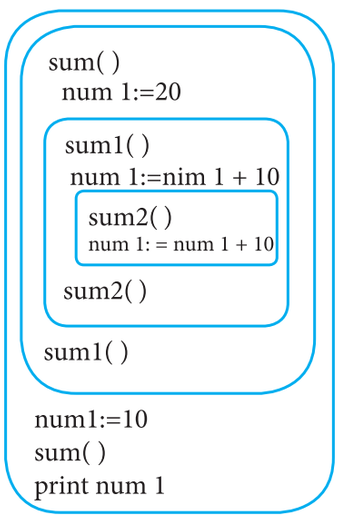
یاد رکھنے کے نکات

- اسکوپ سے مراد کسی پروگرام کے ایک حصے میں متغیرات، پیرامیٹرز اور تفاعل کو اسی پروگرام کے دوسرے حصے سے دیکھنا یا انہیں دائرہ عمل میں لانا ہے۔
- متغیر نام کو کسی شے کے ساتھ باندھنے کے عمل کو میپنگ کہا جاتا ہے۔=:(کولن اور مساوی کی علامت) متغیر اور آبجیکٹ کی نقشہ سازی کے لئے پروگرامنگ زبانوں میں استعمال کیا جاتا ہے۔
- name space میں متغیر کے ناموں کو اشیاء میں نقش کرنے کے لئے کنٹینر (مرتبان) ہیں۔
- متغیر کو دائرہ کار میں لانے کا مقصد یہ ہے کہ کوڈ کا وہ حصہ ہے، جہاں وہ دکھائی دے۔LEGB کا قاعدہ اس ترتیب کا فیصلہ کرنے کے لئے استعمال کیا جاتا ہے جس میں دائرہ کار کے حل کے لئے کونسے اسکوپ(scope) کو تلاش کرنا ہے۔
- مقامی دائرہ کار سے مراد موجودہ فعل میں متعین متغیرات ہیں۔
- ایک متغیر جس کو کسی پروگرام میں تمام افعال سے باہر قرار دیا جاتا ہے اسے عالمی متغیر (global variable) کے نام سے جانا جاتا ہے۔
- کسی فنکشن (طریقہ) کے ساتھ ایک اور فنکشن کو نیسٹ فنکشن(Nested function) کہا جاتا ہے۔
- ایک متغیر جو کسی فنکشن کے اندر اعلان کیا جاتا ہے جس میں کسی اور فنکشن کی وضاحت ہوتی ہے، اندرونی فنکشن بھی بیرونی فنکشن کے متغیر تک رسائی حاصل کرسکتا ہے۔ اس دائرہ کار کو منسلک دائرہ کار (enclosed scope) کہا جاتا ہے۔
- جب ہم مرتب کرنے والے(compiler) یا ترجمان (interpreter) کو شروع کرتے ہیں تو بلٹ ان اسکوپ میں تمام نام ہوتے ہیں جو پہلے سے بنی ہوئی اسکوپ پروگرام میں شامل ہوتے ہیں۔
- ماڈیول ایک پروگرام کا ایک حصہ ہے۔ پروگرام ایک یا زیادہ آزادانہ طور پر تیار کردہ ماڈیولز پر مشتمل ہوتے ہیں۔
- ایک پروگرام کو چھوٹے فنکشنل ماڈیولز میں تقسیم کیا جاتا ہے، جو output حاصل کرنے کے لئے مل کر کام کرتے ہیں۔کمپیوٹر پروگرام کو علیحدہ سب پروگراموں میں تقسیم کرنے کے عمل کو ماڈیولر پروگرامنگ کہتے ہیں۔
- رسائی کنٹرول (access control) ایک حفاظتی تکنیک ہے جو کمپیوٹنگ ماحول میں یہ دیکھتا ہے کہ وسائل کو کون استعمال کرسکتا ہے، اور اس کو کنٹرول کرتا ہے۔
- عوامی ممبران(public members)(عام طور پر کلاس میں اعلان کردہ طریقے) کلاس سے باہر access کرنے کے قابل ہوتے ہیں
- ایک کلاس کے محفوظ ممبر (protected member) طبقے کے اندر ہی قابل رسائی ہیں اور اس کی ذیلی کلاسوں کے لئے بھی دستیاب ہیں
- کلاس کے نجی ممبروں (Private members) کو کلاس کے باہر سے آنے کی اجازت نہیں ہے۔ انہیں صرف کلاس کے اندر ہی access کیا جاسکتا ہے۔
- python نے متغیر یا طریقہ کار کے نام کو کسی اور واحد یا ڈبل انڈر سکور(single or double underscore) کے ساتھ متعین کرنے کی ایک آسانی تجویز کی ہے تاکہ محفوظ اور نجی رسائی سے متعلق افراد کے طرز عمل پر عمل پیرا ہوسکے۔
- python کلاس میں شامل تمام ممبران ڈیفالٹ طور پر پبلک ہوتے ہیں جبکہ C++ اور Java میں وہ نجی ہوتے ہیں۔
- python میں طبقاتی ماحول سے باہر کسی بھی ممبر تک رسائی حاصل کی جاسکتی ہے جو C++ اور جاوا میں ممکن نہیں ہے۔

100
95
75
25
5
0
12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:07 PM

عملی مشق (Hands on practice)

1۔ مندرجہ ذیل خاکہ کا مشاہدہ کریں اور مندرجہ ذیل کے لئے کاذب کوڈ (pseudo code) لکھیں.



IPVLM

محاسبہ Evaluation

حصہ-I

صحیح جواب کا انتخاب کریں۔ (ایک نمبر)

1۔ مندرجہ ذیل میں سے کون سا پروگرام کے ایک حصے میں متغیرات کو دیکھنے کے لئے اسی پروگرام کے دوسرے حصے کی طرف اشارہ کرتا ہے۔

(A) دائرہ کار (Scope) (B) میموری (Memory) (C) پتہ (Address) (D) قابل رسائی (Accessibility)

2۔ کسی شے کے ساتھ متغیر نام کو باندھنے کا عمل یہ کہا جاتا ہے۔

(A) دائرہ کار (Scope) (B) نقش کاری (Mapping)

(C) دیر سے باندھنا (Late binding) (D) فوری باندھنا (Early binding)

3۔ متغیر (Variable) اور آبجیکٹ (object) کی نقشہ سازی (mapping) کے لئے پروگرامنگ کی زبانوں میں مندرجہ ذیل میں سے کون سا استعمال ہوتا ہے ؟

(A) :: (B) := (C) = (D) ==

4۔ اشیاء کو متغیر کے ناموں کی نقشہ سازی کے لئے استعمال ہونے والے مرتبان (containers) یہ کہلاتے ہیں۔

(A) دائرہ کار (scope) (B) نقشہ سازی (Mapping)

(C) باندھنا (binding) (D) نام کی جگہیں (Name spaces)

5۔ کون سا دائرہ کار (scope) موجودہ فنکشن میں متغیرات کی وضاحت کرتا ہے؟

(A) مقامی دائرہ کار (local scope) (B) عالمی دائرہ کار (Global scope)

(C) ماڈیول اسکوپ (Module scope) (D) فنکشن اسکوپ (Function scope)

12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:08 PM

6۔ ایک کمپیوٹر کے پروگرام کو علیحدہ سب پروگراموں میں تقسیم کرنے کا عمل یہ کہلاتا ہے۔

(A) پروسیڈیورل پروگرامنگ (Procedural programming)

(B) ماڈیولر پروگرامنگ (Modular programming)

(C) ایوینٹ ڈرائیون پروگرامنگ (Event driven programming)

(D) آبجیکٹ اورینٹڈ پروگرامنگ (Object oriented programming)

7۔ مندرجہ ذیل میں سے کون سی حفاظتی تکنیک ہے جو کمپیوٹنگ ماحول میں وسائل کو استعمال کر سکتی ہے اور ان کو منظّم کرتی ہے ؟

(A) پاس ورڈ (password)　　(B) توثیق (Authentication)

(C) رسائی کنٹرول (access control)　　(D) سند (Certification)

8۔ کلاس کے درج ذیل ممبروں میں سے کس کو کلاس کے اندر سے ہی access کیا جا سکتا ہے ؟

(A) عوامی ارکان (Public members)　　(B) محفوظ ممبر (Protected member)

(C) تحفظ شدہ ممبر (secured member)　　(D) نجی ممبر (Private member)

9۔ کلاس کے باہر سے کون سے ممبر قابل رسائی ہیں؟

(A) عوامی ارکان (Public members)　　(B) محفوظ ممبر (Protected member)

(C) تحفظ شدہ ممبر (secured member)　　(D) نجی ممبر (Private member)

10۔ ان ممبران کو جو کلاس کے اندر سے قابلِ رسائی ہیں اور اس کی ذیلی کلاسوں کے لئے بھی دستیاب ہیں، انہیں یہ کہا جاتا ہے۔

(A) عوامی ارکان (Public members)　　(B) محفوظ ممبر (Protected member)

(C) تحفظ شدہ ممبر (secured member)　　(D) نجی ممبر (Private member)

حصہ - II

درج ذیل سوالات کے جوابات دیجئے۔ (2 نمبرات)

1۔ دائرہ کار (scope) کیا ہے ؟

2۔ تغیر کے لئے کیوں دائرہ کار (scope) کا استعمال کیا جانا چاہئے۔ وجہ بتائیں۔

3۔ نقشہ نگاری (mapping) کیا ہے ؟

4۔ نام کی جگہوں (name spaces) سے آپ کا کیا مطلب ہے ؟

5۔ python کیسے نجی اور محفوظ رسائی والے افراد کی نمائندگی کرتے ہیں ؟

حصہ - III

درج ذیل سوالات کے جوابات دیجئے۔ (3 نمبرات)

1۔ ایک مثال کے ساتھ مقامی دائرہ کار (local scope) کی وضاحت کریں۔

2۔ ایک مثال کے ساتھ عالمی دائرہ کار (global scope) کی وضاحت کریں۔

3۔ ایک مثال کے ساتھ منسلک دائرہ کار (enclosed scope) کی وضاحت کریں۔

12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:08 PM

4۔ رسائی کنٹرول (access control) کی ضرورت کیوں ہے؟

5۔ درج ذیل کاذب کوڈ (pseudo code) میں متغیرات کے دائرہ کار (scope of the variables) کی نشاندہی کریں اور اس کا output لکھیں۔

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

حصہ - IV

درج ذیل سوالات کے جوابات دیجئے۔ (5 مارکس)

1۔ مثال کے ساتھ متغیر یا LEGB کے اصول کے لئے دائرہ کار (scope) کے اقسام کی وضاحت کریں۔

2۔ ماڈیول (module) کی کوئی پانچ خصوصیات لکھیں۔

3۔ ماڈیولر پروگرامنگ کو استعمال کرکے اس کے کوئی پانچ فوائد لکھیں۔

REFERENCES حوالہ جات

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*

12 CS UNIT 3 CDR Set
Sunday, April 16, 2023 9:35:08 PM

باب 4

الگورتھمی حکمت عملی ALGORITHMIC STRATEGIES

IGYL4

سیکھنے کے مقاصد Learning Objectives

اس باب کے آخر میں طلبہ اس قابل ہو جائیں گے۔

- الگورتھم کی بنیادی باتوں اور تکنیکی نقطہ نظر کو جان لیں گے۔
- الگورتھم کی کارکردگی، وقت اور جگہ کی پیچیدگی کو سمجھ لیں گے۔
- تلاش اور ترتیب دینے کے لیے الگورتھم تیار کریں اور ان کا تجزیہ کریں گے۔
- الگورتھمی نقطہ نظر کے ذریعے متحرک پروگرامنگ کے بارے میں جان لیں گے۔

4.1 الگورتھمی حکمت عملی کا تعارف
Introduction to Algorithmic strategies

ایک الگورتھم کسی خاص کام کو پورا کرنے کے لئے ہدایات کا ایک محدود سیٹ ہے۔ یہ کسی دیئے گئے مسئلے کو حل کرنے کے لئے ایک قدم بہ قدم طریقہ کار ہے۔ کسی بھی مناسب پروگرامنگ زبان میں الگورتھم نافذ کیا جاسکتا ہے۔

الگورتھم کے پاس ان پٹ، آؤٹ پٹ ہونا ضروری ہے اور انھیں مندرجہ ذیل خصوصیات کو مطمئن کرنا چاہئے جیسے قطعیت (definiteness)، درستگی (Correctness) اور تاثیر (effectiveness)۔ ڈیٹا کی ساخت کے ذریعہ ڈیٹا کو برقرار اور مؤثر طریقے سے جوڑ دیا جاتا ہے۔ اس طرح کے ڈیٹا کیساخت سے ڈیٹا کو اسٹور کرنے، جوڑ توڑ اور بازیافت کرنے کیلئے الگورتھم تیار کیا جاسکتا ہے۔ اعداد و شمار کے ڈھانچے کی مثالیں arrays، Structures، List، tuples، dictionary وغیرہ ہیں۔

تلاش کرنا Search	خطی اور بائنری تلاش کا استعمال کرتے ہوئے ڈیٹا کی ساخت میں کسی شئے کی تلاش کرنا۔
ترتیب دینا Sort	طریقوں کو استعمال کرتے ہوئے کسی خاص ترتیب میں اشیاء کو ترتیب دینے کے لئے جیسے bubble sort، insertion sort، selection sort وغیرہ۔
داخل کرنا Insert	ڈیٹا سٹرکچر میں کسی آئٹم کو داخل کرنے کے لئے۔
Update	ڈیٹا سٹرکچر میں موجود آئٹم کو اپ ڈیٹ کرنے کے لئے کرنا
حذف کریں Delete	ڈیٹا سٹرکچر میں موجود آئٹم کو حذف کرنا۔

الگورتھم کی تعریف کرنے کا طریقہ الگورتھمی حکمت عملی کہلاتا ہے۔ مثال کے طور پر دیئے گئے n قدر کے لیے فیکٹوریئل (factorial) کا حساب لگانا۔ تو پھر یہ عمل تفاعل کی وضاحت کرتے ہوئے ایک بار تکرار-1 کے لئے فیکٹوریل کا حساب لگانے کے لئے کیا جاسکتا ہے پھر جب تک مطلوبہ تکرار کی تعداد نہ پہنچے اس کو مکرر کیا جاسکتا ہے۔

کیا تم جانتے ہو؟

لفظ الگورتھم ایک فارسی مصنف، ابو جعفر محمد ابن موسی الخوارزمی (سن 825 عیسوی (CE)) سے موسوم ہے، جس نے ریاضی پر ایک درسی کتاب لکھی۔ کسی مسئلے کو حل کرنے کے طریقہ کے لئے لفظ الگورتھم استعمال ہوا ہے۔

4.1.1 الگورتھم کی خصوصیات

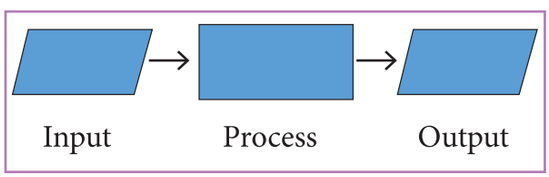
الگورتھم میں درج ذیل خصوصیات ہونی چاہیے۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:22 PM

ان پٹ Input	صفر یا اس سے زیادہ مقدار فراہم کرنی چاہئے۔
آؤٹ پٹ Output	کم از کم ایک مقدار تیار کی جائے۔
محدودیت Finiteness	ایک محدود مرحلوں کے بعد الگورتھم ختم ہوجاتا ہے۔
قطعیت Definiteness	تمام کارروائیاں اچھی طرح بیان ہونی چاہئے۔ مثال کے طور پر صفر کے ساتھ تقسیم کرنا یا منفی تعداد کے لئے جذر المربع بنانا ناقابل قبول ہے۔
تاثیر Effectiveness	ہر ہدایت کو موثر طریقہ سے کرنا لازمی ہے۔
درستگی Correctness	الگورتھم غلطی سے پاک ہونا چاہئے۔
سادگی Simplicity	لاگو کرنے کے لئے آسان ہونا چاہئے۔
قابل شناخت Unambiguous	الگورتھم واضح اور قابل شناخت ہونا چاہئے۔ اس کے ہر اقدام اور ان کے نتائج/نتائج کو واضح ہونا چاہئے اور اس کا مقصد بننا چاہئے۔ صرف ایک معنی ہونا چاہئے۔
قابل عمل Feasibility	دستیاب وسائل کے ساتھ ممکن قابل عمل ہونا چاہئے۔
حرکت پذیر Portable	ایک الگورتھم عمومی، کسی بھی پروگرامنگ زبان یا کسی آپریٹنگ سسٹم سے غیر منحصر ہونا چاہئے جو input کی تمام وسعتوں (range) کو سنبھال سکتا ہو۔
آزاد Independent	ایک الگورتھم میں مرحلہ در مرحلہ ہدایات ہونا چاہئے، جو کسی بھی پروگرامنگ کوڈ پر غیر منحصر ہونا چاہئے۔

4.1.2 الگورتھم لکھنا (Writing an Algorithm)

الگورتھم عام ہیں اور صرف کمپیوٹر تک محدود نہیں۔ اسے مختلف ریئل ٹائم سرگرمیوں میں بھی استعمال کیا جاسکتا ہے۔ جان بوجھ کر یا نادانستہ طور پر ہم اپنی روزمرہ کی زندگی میں بہت سے الگورتھم انجام دیتے ہیں جیسے اسکول بیگ میں کتابیں جوڑنا، کسی جگہ کو جانے کے لئے مختصر ترین راستہ تلاش کرنا، روزانہ کی سرگرمیاں طے کرنا، امتحانات کی تیاری وغیرہ۔ جیسا کہ ہم جانتے ہیں کہ تمام پروگرامنگ زبانیں بنیادی کوڈ استعمال کرتی ہیں۔ ضوابط اور تکرار وغیرہ کو کوڈ کی تشکیل کرنے کے لئے الگورتھم لکھنے استعمال کی جاسکتی ہے۔ مندرجہ ذیل شکل 4.1 میں ایک عام الگورتھم دکھایا گیا ہے۔



ایک مثالی الگورتھم

مثال

کافی کی تیاری کی مثال پر غور کریں۔ کافی بنانے کے لئے ہمیں مندرجہ ذیل اجزاء رکھنے کی ضرورت ہے: پانی، دودھ، کافی پاؤڈر اور شکر۔ یہ اجزاء الگورتھم کے آلے ہیں۔ ایک کپ کافی تیار کرنا عمل (process) کہلاتا ہے۔ اس عمل کی پیداوار (output) کافی ہے۔

کافی کی تیاری کا طریقہ کار مندرجہ ذیل ہے۔

1۔ کافی پاؤڈر کے ساتھ ایک کٹورا لیں۔
2۔ پانی ابالیں اور کٹوری میں ڈالیں۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:22 PM

3۔ اسے فلٹر کریں۔

4۔ دودھ ابالیں۔

5۔ ابلے ہوئے دودھ کے ساتھ چینی اور فلٹر شدہ کافی ملا دیں۔

6۔ کپ میں کافی میں ڈال کر پیش کریں۔

الگورتھم کا استعمال کرتے ہوئے اس طرح کے طریقہ کار کی نمائندگی کی جاسکتی ہے۔اس طرح، الگورتھم مرحلہ وار ہدایات پر مشتمل ہے جو کسی کام کو پورا کرنے کے لئے ضروری ہے اور پروگرامر کو پروگرام تیار کرنے میں مدد کرتا ہے۔

مسئلہ :

دئے گئے عدد کا مربع معلوم کرنے اور نتیجہ ظاہر کرنے کے لئے الگورتھم تیار کریں۔

الگورتھم کو اس طرح لکھا جاسکتا ہے:

Step 1 – start the process

Step 2 – get the input x

Step 3 –calculate the square by multiplying the input value ie., square ← x* x

Step 4 – display the result square

Step 5 – stop

الگورتھم کسی دیئے گئے مسئلے کا حل نکالنے کے لئے ڈیزائن کیا جاسکتا ہے۔ایک مسئلہ بہت سے طریقوں سے حل کیا جاسکتا ہے۔کئی الگورتھموں سے ایک کو استعمال کے لئے لیا جاسکتا ہے۔

کیا تم جانتے ہو؟

ایک الگورتھم جو ایک درست ان پٹ کے لئے متوقع پیداوار **output** فراہم کرتا ہے، اسے الگورتھمک حل کہا جاتا ہے۔

الگورتھم Algorithm	پروگرام Program
• الگورتھم دیئے گئے مسئلے کو منطقی طور پر حل کرنے میں مدد کرتا ہے۔ اسے پروگرام سے موازنہ کیا جاسکتا ہے۔	• پروگرام ایک پروگرامنگ زبان میں الگورتھم کا اظہار ہوتا ہے۔
• الگورتھم کو ان کے نفاذ کے طریقوں، ڈیزائن تکنیک وغیرہ کی بنیاد پر درجہ بندی کی جاسکتی ہے۔	• الگورتھم کی ساخت یا آبجیکٹ پر مبنی پروگرامنگ کے ذریعہ عمل درآمد کیا جاسکتا ہے۔
• الگورتھم لکھنے کے لئے کوئی خاص قواعد موجود نہیں ہیں لیکن کچھ ہدایات پر عمل کرنا چاہئے۔	• منتخب زبان کے ساتھ مخصوص syntax کے ساتھ پروگرام لکھنا چاہئے۔
• الگورتھم کا ذب کوڈ (Pseudo code) سے مشابہت رکھتا ہے، جسے کسی بھی زبان میں لاگو کیا جا سکتا ہے۔	• پروگرام، پروگرامنگ کی زبان سے زیادہ مخصوص ہوتا ہے۔

جدول 4.1 الگورتھم بمقابلہ پروگرام

4.1.3 الگورتھم کا تجزیہ (Analysis of Algorithm)

کمپیوٹر کے وسائل محدود ہیں۔الگورتھم کی استعداد کی وضاحت وقت اور جگہ کی پیچیدگی کے استعمال کے مطابق ہوتی ہے۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:23 PM

الگورتھم کا تجزیہ عام طور پر اس میں شامل مختلف آپریشنوں کے چلانے اور عملدرآمد کے وقت سے متعلق ہوتا ہے۔ کسی آپریشن کے چلنے کے وقت کا حساب لگایا جاتا ہے کہ فی آپریشن کتنے پروگرامنگ ہدایات پر عمل درآمد ہوتا ہے۔

الگورتھم کا تجزیہ اور کارکردگی کی تشخیص کو دو مختلف مراحل میں تقسیم کیا جاسکتا ہے۔

1۔ ایک پریوری کا اندازہ (A priori estimates):

یہ الگورتھم کا نظریاتی کارکردگی کا تجزیہ ہے۔ بیرونی عوامل کو فرض کرتے ہوئے الگورتھم کے کارکردگی کی پیائش کی جاتی ہے۔

2۔ پوسٹروری ٹیسٹ (A posteriori testing) :

اسے کارکردگی کی پیائش کہا جاتا ہے۔ اس تجزیے میں، الگورتھم کو چلانے کا وقت اور عمل درآمد جیسے اصل اعداد و شمار جمع کیے جاتے ہیں۔

کیا تم جانتے ہو

مختلف ان پٹ سائز کے لیے الگورتھم کے وقت اور جگہ کی پیچیدگیوں کا اندازہ الگورتھم کا تجزیہ (Algorithm analysis) کہلاتا ہے۔

4.2 الگورتھم کی پیچیدگی (Complexity of an algorithm)

فرض کریں کہ A الگورتھم ہے اور n ان پٹ ڈیٹا کا سائز ہے، الگورتھم A کے ذریعہ استعمال ہونے والا وقت اور جگہ دو اہم عوامل ہیں، جو A کی کارکردگی کا فیصلہ کرتے ہیں۔

وقت کا عامل - (Time factor)

چھانٹے جانے والے الگورتھم کے اہم آپریشنوں کی گنتی کے ذریعہ وقت کی پیائش کی جاتی ہے۔

جگہ کا عامل- (Space factor)

الگورتھم کے لئے مطلوبہ زیادہ سے زیادہ میموری کی جگہ سے پیائش کی جاتی ہے۔

الگورتھم کی پیچیدگی f(n) ان پٹ ڈیٹا کے سائز کے مطابق الگورتھم کے ذریعہ چلنے کا وقت اور/ یا اسٹوریج کی جگہ فراہم کرتا ہے۔

4.2.1 وقت کی پیچیدگی (Time complexity)

الگورتھم کے وقت کی پیچیدگی عمل کو مکمل کرنے کے لئے الگورتھم کے ذریعہ اٹھائے گئے مرحلوں کی تعداد کے ذریعہ دی جاتی ہے۔

4.2.2 جگہ کی پیچیدگی (space complexity)

الگورتھم میں جگہ کی پیچیدگی اس کی تکمیل تک چلانے کے لئے درکار میموری کی مقدار ہے۔ الگورتھم کے ذریعہ مطلوبہ جگہ مندرجہ ذیل دو اجزاء کے حاصل جمع کے برابر ہے۔

ایک مقررہ حصہ کی تعریف، الگورتھم کے لئے کچھ اعداد و شمار اور متغیرات کو ذخیرہ کرنے کے لئے درکار کل جگہ کی حیثیت سے کی جاتی ہے۔ مثال کے طور پر، الگورتھم میں استعمال ہونے والے سادہ متغیر اور مستقلہ۔

متغیر والے حصے کی تعریف، متغیر کے ذریعہ مطلوبہ کل جگہ کے طور پر کی جاتی ہے، جس کا سائز مسئلہ اور اس کی تکرار پر منحصر ہوتا ہے۔ مثال کے طور پر: تکرار کسی دی گئی قدر کے فیکٹوریل کا حساب لگانے کے لئے استعمال ہوتا ہے۔

4.3 الگورتھم کی کارکردگی Efficiency of an algorithm

کمپیوٹر کے وسائل محدود ہیں جن کا موثر استعمال کیا جانا چاہئے۔ الگورتھم کی کارکردگی کو الگورتھم کے ذریعہ محسوب وسائل (computational resources) کی تعداد کے طور پر استعمال کیا جاتا ہے۔ اس کے وسائل کے استعمال کا تعین کرنے کیلئے الگورتھم کا تجزیہ کرنا ضروری ہے۔ مختلف وسائل کے استعمال کی بنیاد پر الگورتھم کی کارکردگی کی پیائش کی جاسکتی ہے۔

الگورتھم کی زیادہ سے زیادہ کارکردگی کیلئے ہم وسائل کے استعمال کو کم سے کم کرنا چاہتے ہیں۔ وقت اور جگہ کی پیچیدگی جیسے اہم وسائل کا براہ راست موازنہ نہیں کیا جاسکتا، لہذا الگورتھمک کارکردگی کے لئے وقت اور جگہ کی پیچیدگی پر غور کیا جاسکتا ہے۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:23 PM

4.3.1 کارکردگی کو تعین کرنے کا طریقہ
(Method for determining Efficiency)

الگورتھم کی کارکردگی کا انحصار اس بات پر ہے کہ وہ وقت اور میموری کی جگہ کو کس حد تک موثر طریقے سے استعمال کرتا ہے۔

الگورتھم کی وقت کی کارکردگی مختلف عوامل سے پیائش کی جاتی ہے۔ مثال کے طور پر، ایک الگورتھم کے لئے ایک پروگرام لکھیں، پروگرامنگ کی کوئی زبان استعمال کر کے اس پر عمل کریں، اور چلنے میں کل وقت کی پیائش کریں۔ اس کو چلاتے وقت جو پیائش کرتے ہیں، اس کا انحصار متعدد عوامل پر ہوگا، جیسے:

- مشین کی رفتار
- مرتب کرنے والے (compiler) اور دوسرے سسٹم سافٹ ویئر ٹولز۔
- آپریٹنگ سسٹم۔
- استعمال ہونے والے پروگرامنگ کی زبان۔
- درکار ڈیٹا کا حجم (جسامت)۔

تاہم، اس بات کا تعین کرنے کے لئے کہ الگورتھم کسی دیئے گئے مسئلے کو کس حد تک موثر طریقے سے حل کرتا ہے، آپ یہ طے کرنا چاہیں گے کہ الگورتھم کی نوعیت سے اس کو چلانے کا وقت کس طرح متاثر ہوتا ہے۔ لہذا، ہمیں بنیادی قوانین تیار کرنے کی ضرورت ہے جو بنیادی پروگرام الگورتھم کی نوعیت کے لحاظ سے کسی پروگرام کی کارکردگی کو متعین کرتے ہیں۔

کیا تم جانتے ہو

الگورتھم ڈیزائن کرنے کا ایک طریقہ الگورتھمک حکمت عملی کہلاتا ہے۔

4.3.2 اسپیس ٹائم ٹریڈ آف (Space-Time tradeoff)

اسپیس ٹائم یا ٹائم میموری ٹریڈ آف زیادہ اسٹوریج کی جگہ کا استعمال کر کے یا زیادہ وقت گزار کر بہت کم جگہ میں دیئے گئے الگورتھم کو حل کر کے کم وقت میں حل کرنے کا ایک طریقہ ہے۔

کسی دیئے گئے پروگرامنگ کے مسئلے کو حل کرنے کے لئے، بہت سے مختلف الگورتھم استعمال ہو سکتے ہیں۔ ہوسکتا ہے کہ ان میں سے کچھ الگورتھم انتہائی وقت کے اعتبار سے کارکردہ اور بعض جگہ (space) کے اعتبار سے کارکردہ ہو سکتے ہیں۔

ٹائم/ اسپیس ٹریڈ آف سے مراد ایسی صورتحال ہے، جہاں آپ پروگرام کو آہستہ چلا کر میموری کے استعمال کو کم کر سکتے ہیں، یا میموری کے استعمال میں اضافہ کر کے چلنے کے وقت کو کم کر سکتے ہیں۔

کیا تم جانتے ہو

کسی دیئے گئے مسئلے کو حل کرنے کے لئے بہترین الگورتھم وہ ہے جس کو میموری میں کم جگہ کی ضرورت ہوتی ہے اور آؤٹ پٹ پیدا کرنے کے لئے، اس کی ہدایت پر عمل کرنے میں کم وقت لگتا ہے۔

4.3.3 اسمپ ٹوٹک (علامتی) ترقیم
(Asymptotic Notations)

Asymptotic Notations ایسی زبانیں ہیں جو وقت اور جگہ کی پیچیدگی کے بارے میں معنی خیز بیانات کا استعمال کرتی ہیں۔ مندرجہ ذیل تین اسمپ ٹوٹک اشارے زیادہ تر الگورتھم کے وقت کی پیچیدگی کی نمائندگی کرنے کے لئے استعمال ہوتے ہیں۔

(i) بگ او O (Big O)

بگ O اکثر ایک الگورتھم کے بدترین صورت حال کی وضاحت کرنے کے لئے استعمال ہوتا ہے۔

(ii) بڑا (اومیگا) Ω (Big Ω)

بگ اومیگا Ω ، بگ او کا اُلٹ ہے، اگر بگ O کو کسی اسمپ ٹوٹک فنکشن کے اوپری حد (بدترین-صورت) کی وضاحت کرنے کے لئے استعمال کیا جاتا ہے تو، بگ اومیگا Ω نچلے حد (بہترین معاملہ) کی وضاحت کرنے کے لئے استعمال ہوتا ہے۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:24 PM

(iii) بڑا Θ (ٹیٹا) (Big Θ)

جب الگورتھم میں نچلی حد = اوپری حد کے ساتھ کوئی پیچیدگی ہوتی ہے، تو یوں کہیے کہ الگورتھم میں O (n log n) ہے اور Ω (n log n) کی پیچیدگی ہوتی ہے، اس کی اصل میں پیچیدگی ہوتی یہ ہے Θ (n log n)، جس کا مطلب اس الگورتھم کے چلنے کا وقت بدترین صورت اور بہترین صورت میں ہمیشہ n log n تک آتا ہے۔

4.3.4 بہترین، بدترین اور اوسط کے صورت حال کی کارکردگی (Best, Worst and Average case efficienty)

آیئے ایک صف (array) میں محفوظ کردہ قیمتوں کے اعداد کی ایک فہرست کو فرض کریں۔ فرض کریں کہ اگر ہم اس لسٹ میں کسی خاص عنصر کو تلاش کرنا چاہتے ہیں تو، الگورتھم جو فہرست کے کلیدی عنصر کو فہرست میں موجود ہر عنصر کے ساتھ ترتیب کے ساتھ n عناصر کے مابین موازنہ کر کے تلاش کرنا ہے۔

سب سے بہتر معاملہ یہ ہوگا کہ اگر فہرست میں پہلا عنصر کلیدی عنصر سے ملتا ہو جو عناصر کی فہرست میں تلاش کیا جائے۔ اس معاملے میں کارکردگی کا اظہار O(1) کے طور پر کیا جائے گا کیونکہ صرف ایک موازنہ ہی کافی ہے۔

اسی طرح، ایسی صورت میں سب سے خراب صورت حال ہوگی اگر پوری فہرست کو تلاش کیا جائے اور عنصر فہرست کے آخر میں ہی مل جائے یا فہرست میں نہ ملے۔ اس معاملے میں الگورتھم کی کارکردگی کا اظہار O(n) کے طور پر کیا جائے گا کیونکہ تلاش کو مکمل کرنے کے لئے n موازنہ کی ضرورت ہے۔

نیچے دیئے گئے موازنہ کی اوسط تعداد معلوم کر کے الگورتھم کی اوسط صورت حال کی کارکردگی حاصل کی جاسکتی ہے۔

1 = کم سے کم موازنوں کی تعداد

n = زیادہ سے زیادہ موازنوں کی تعداد

n = اگر عنصر نہیں موجود ہو تو زیادہ سے زیادہ موازنوں کی تعداد

(n+1)/2 = لہذا، موازنہ کی اوسط تعداد

لہذا اوسط کیس کی کارکردگی کا اظہار O(n) کے طور پر کیا جائے گا۔

4.4 تلاش کرنے کی ٹکنیک کے لئے الگورتھم Algorithm for Searching Techniques

4.4.1 خطی تلاش (Linear search)

خطی تلاش کو تواتری تلاش (sequential search) بھی کہا جاتا ہے جو کسی فہرست میں کسی خاص قدر کی تلاش کے لئے ایک ترتیب والا طریقہ ہے۔ یہ طریقہ مطلوبہ عنصر کے ملنے یا فہرست ختم ہونے تک ترتیب میں ہر عنصر کے ساتھ تلاش کے عنصر کی جانچ کرتا ہے۔ اس تلاشی الگورتھم میں، فہرست آرڈر کرنے کی ضرورت نہیں ہے۔

کاذب کوڈ (Pseudo code)

1۔ لوپ کو استعمال کرتے وقت array سے گزریں۔

2۔ ہر تکرار میں، فہرست کی موجودہ قیمت کے ساتھ ہدف کی تلاش کی کلیدی قیمت کا موازنہ کریں۔

- اگر قیمتیں جڑتے ہیں تو موجودہ اشاریہ اور array کے قدر کو ڈسپلے کرتا ہے۔
- اگر قیمتیں جڑتے نہیں ہیں تو، اگلے array بڑھیں۔

3۔ اگر کوئی قیمت نہیں جڑتی ہے تو، یہ دکھائی دیتا ہے کہ

search element not found

ذیل میں دیئے گئے array میں عدد 25 تلاش کرنے کے لئے، خطی سرچ مرحلہ وار ترتیب میں تواتر کے ساتھ پہلے عنصر سے ترتیب دی جائے گی۔ اگر اس اشاریہ (Index) میں تلاش کیا گیا عدد پایا گیا تو اشاریہ کو لوٹا دیا جائے گا، ورنہ آخری عدد تک تلاش جاری رکھی جائے گی۔ اس مثال میں عدد 25 ، اشاریہ 3 میں پایا گیا ہے۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:25 PM

index	0	1	2	3	4
values	10	12	20	25	30

مثال 1 :

Input: values[] = {5, 34, 65, 12, 77, 35}
target = 77
Output: 4
Example 2:
Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}
target = 200
Output: -1 (not found)

4.4.2 ثنائی تلاش (Binary search)

ثنائی تلاش کو نصف وقفہ سے متعلق سرچ الگورتھم (half-interval search algorithm) بھی کہتے ہیں۔ یہ ایک ترتیب شدہ array میں تلاش کرنے والے عنصر کے مقام کو معلوم کرتا ہے۔ ثنائی تلاش الگورتھم کو divide and conquer search algorithm (تقسیم اور فتح تلاشی الگورتھم) کے طور پر بھی چلایا جا سکتا ہے اور لوگارتمی وقت (logarithmic time) میں اسے عمل درآمد کیا جاتا ہے۔

ثنائی تلاش کے لئے کاذب کوڈ
(Pseudo code for Binary search)

1۔ درمیانی عنصر کے ساتھ شروع کریں:

- اگر تلاش کیا جانے والا عنصر صف (array) کے درمیانی عنصر کے برابر ہو یعنی،

  درمیانی قیمت = صف میں موجود عناصر کی تعداد/2

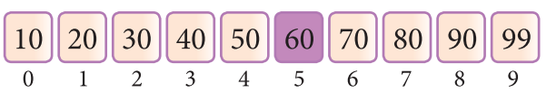
  ہو تو وسط عنصر کا اشاریہ واپس کریں۔
- اگر نہیں ہے تو، پھر درمیانی عنصر کی تلاش کی قیمت کے ساتھ موازنہ کریں۔
- اگر تلاش کیا جانے والا عنصر درمیانی اشاریہ کے عدد سے بڑا ہو تو، دائیں جانب کے عناصر کو منتخب کریں، اور مرحلہ 1 پر جائیں۔
- اگر تلاش کیا جانے والا عنصر درمیانی اشاریہ کے عدد سے کم ہے تو، بائیں جانب والے عناصر کو منتخب کریں اور مرحلہ 1 سے شروع کریں۔

2۔ جب کوئی جوڑ مل جاتا ہے تو جوڑ (match) کے عنصر کی فہرست کے ساتھ success message کا پیغام ڈسپلے کرتا ہے۔

3۔ اگر تمام موازنہ کے لئے کوئی مماثلت نہیں ملتی ہے، تو ناکام پیغام (unsuccessful message) دکھاتا ہے۔

ثنائی تلاش کے کام کرنے کے اصول
(Binary Search Working Principles)

ثنائی کی تلاش کے لئے ایک صف میں موجود عناصر کی فہرست کی پہلے ترتیب دی جانی چاہئے۔ مندرجہ ذیل مثال ثنائی تلاش کے مرحلہ وار آپریشن کی وضاحت کرتی ہے۔ مندرجہ ذیل عناصر کی صف (array) پر غور کریں، صف کو ترتیب دیا جا رہا ہے۔ لہذا یہ بائنری سرچ الگورتھم کرنے کے قابل بناتا ہے۔ آیئے ہم فرض کریں کہ تلاش کیا جانے والا عنصر 60 ہے اور ہمیں بائنری تلاش کا استعمال کرتے ہوئے تلاش کئے جانے والے عنصر 60 کے مقام یا فہرست کو تلاش کرنا ہوگا۔

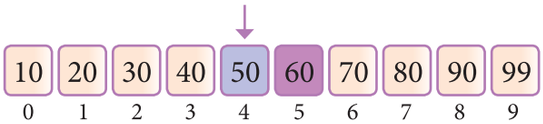


پہلے، ہم اس فارمولے کا استعمال کر کے صف کے درمیانی عنصر کا اشاریہ تلاش کرتے ہیں۔

درمیانی عدد =(زیادہ-کم)/2+کم

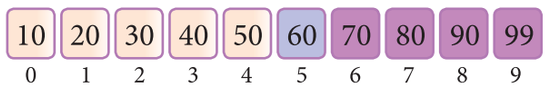
mid = low + (high - low) / 2

یہ اس طرح سے ہے، 4 = 2 /(9-0) + 0 (اعشاریہ کے حصہ کو نظر انداز کیا گیا ہے) تو، صف کی درمیانی قیمت 4 ہے۔



12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:25 PM

اب تلاش کیا جانے والے عنصر کا موازنہ درمیانی قیمت والے مقام پر ذخیرہ شدہ قیمت سے کریں۔ مقام یا اشاریہ 4 میں ذخیرہ شدہ قیمت 50 ہے، جو تلاش کے عنصر سے مماثل نہیں ہے۔ چونکہ تلاش کی جانے والی قیمت 60 ہے جو 50 سے زیادہ ہے۔

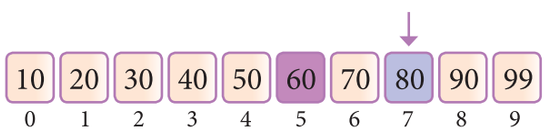


ہم اپنی نچلی سطح کو وسط + 1 میں تبدیل کرتے ہیں اور فارمولے کا استعمال کر کے ایک بار پھر نئی درمیانی قیمت تلاش کرتے ہیں۔

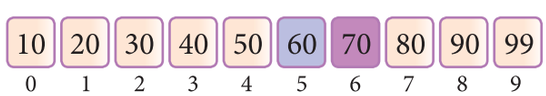
1 + درمیانی = کم

2/(زیادہ-کم)+کم = درمیانی عدد

ہمارا نیا وسط اب 7 ہے۔ ہم مقام 7 پر ذخیرہ شدہ قیمت کو اپنے ہدف کی قیمت 31 سے موازنہ کرتے ہیں۔



مقام یا انڈیکس 7 میں ذخیرہ شدہ قیمت تلاش کے عنصر سے مماثل نہیں ہے، بلکہ اس سے کہیں زیادہ ہے جس کی ہم تلاش کر رہے ہیں۔ لہذا، تلاش کیا جانے والا عنصر موجودہ درمیانی قیمت والے مقام سے نچلے حصے میں ہونا چاہئے۔

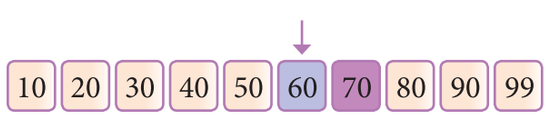


تلاش کیا جانے والا عنصر ابھی تک نہیں ملا۔ لہذا، ہم نے فارمولہ استعمال کر کے درمیانی عدد کا دوبارہ حساب کیا۔

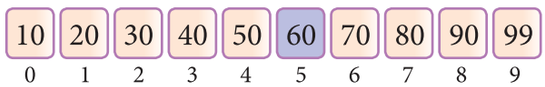
1 - درمیانی = زیادہ

2/(زیادہ-کم)+کم = درمیانی عدد

اب درمیانی قیمت 5 ہے۔



اب ہم مقام 5 پر ذخیرہ شدہ قیمت کو اپنے تلاش کے عنصر سے موازنہ کرتے ہیں۔ ہم نے پایا کہ یہ ایک جوڑ (match) ہے۔



ہم یہ نتیجہ اخذ کر سکتے ہیں کہ تلاش کیا جانے والے عنصر 60، انڈیکس یا مقام 5 پر پایا جاتا ہے۔ مثال کے طور پر اگر ہم تلاش کئے جانے والے عنصر کو 95 کے طور پر لیں تو، اس قیمت کے لئے یہ بائنری سرچ الگورتھم میں unsuccessful result نتیجہ دیتا ہے۔

4.5 چھانٹنے کی تکنیک Sorting Techniques

4.5.1 ببل سارٹ الگورتھم (Bubble sort algorithm)

bubble sort ایک آسان الگورتھم ہے۔ الگورتھم ایک صف میں محفوظ کردہ قیمتوں کی فہرست کے آغاز سے شروع ہوتا ہے۔ اس سے متصل عناصر کے ہر جوڑے کا موازنہ کیا جاتا ہے اور اگر وہ چھانٹے ہوئے نہیں ہوں تو ان کو بدل دیتا ہے۔ یہ موازنہ اور اس وقت تک جاری رکھا جاتا ہے، جب تک کہ کسی ادل بدل کرنے (swap) کی ضرورت نہیں ہوتی ہے، جو یہ اشارہ کرتا ہے کہ کسی صف میں محفوظ کردہ قیمت کی فہرست کو ترتیب دیا گیا ہے۔ الگورتھم ایک موازنہ کی طرح ہے، جس کے لئے فہرست کے اوپری حصے تک چھوٹے عناصر bubble کے لئے نامزد کر دئے گئے ہیں۔ اگرچہ الگورتھم بہت آسان ہے، لیکن ترتیب اور دوسرے چھانٹنے کے طریقوں کے مقابلے میں یہ بہت ہی آہستہ اور کم کارکردگی والا ہے۔

فرض کریں کہ ایک صف (array) میں n عناصر کی فہرست (list) ہے۔ تبادلہ کا فنکشن، دئے گئے صف کے عناصر کی قیمتوں کو ادل بدل (swap) کرتا ہے۔

کاذب کوڈ (Pseudo code)

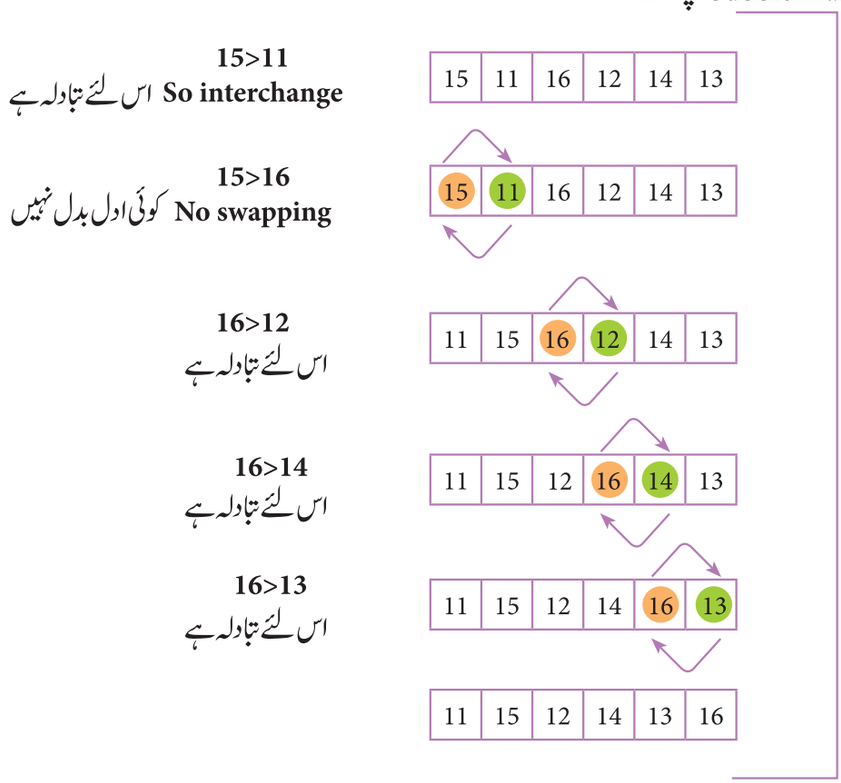
1۔ پہلے عنصر سے شروع کریں یعنی انڈیکس = 0، موجودہ عنصر کا سرنی کے اگلے عنصر سے موازنہ کریں۔

2۔ اگر موجودہ عنصر صف کے اگلے عنصر سے زیادہ ہے تو، ان کو تبدیل کریں۔

3۔ اگر موجودہ عنصر عنصر کے اگلے یا دائیں طرف سے کم ہو تو، اگلے عنصر میں چلے جائیں۔ مرحلہ 1 پر جائیں اور انڈیکس کا اختتام ہونے تک دہرائیں۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:26 PM

آیئے، {15, 11, 16, 12, 14, 13} قیمتوں والی ایک صف (array) پر غور کریں۔ ذیل میں، ہمارے پاس اس کی ایک تصویری نمائندگی ہے کہ کس طرح دئے گئے array کو bubble چھانٹے گا۔



مذکورہ بالا تصویری مثال تکرار-1 کے لئے ہے ۔اسی طرح، باقی تکرار (iteration) بھی کیا جا سکتا ہے۔
آخری تکرار چھانٹی ہوئی array دے گی۔
تمام تکرار کے اختتام پر ہم چھانٹی ہوئی قیمتوں کو ایک array میں دیں گے، جیسا کہ ذیل میں دیا گیا ہے:

11	12	13	14	15	16

4.5.2 چھانٹنے کا انتخاب (Selection sort)

چھانٹنے کا انتخاب ایک آسان چھانٹنے والا الگورتھم (simple sorting algorithm) ہے، جو فہرست میں ہر پاس (pass) کے لئے صرف ایک ہی تبادلہ کر کے bubble sort کی کارکردگی میں بہتری لاتا ہے۔ یہ الگورتھم صف میں پہلے سب سے چھوٹے عنصر کو پائے گا اور اسی صف کی پہلے مقام میں موجود عنصر کے ساتھ اس کو ادل بدل (swap) کرے گا۔ پھر اسے دوسرا سب سے چھوٹا عنصر لے گا اور اس عنصر کو دوسرے مقام میں ادل بدل کرے گا، اور یہ اس وقت تک جاری رہے گا، جب تک کہ پوری صف کو متعلقہ ترتیب میں ترتیب نہ دے۔
یہ الگورتھم بار بار اگلے سب سے چھوٹے عنصر کو منتخب کرتا ہے اور ہر پاس کے لئے صحیح جگہ پر ادل بدل (swap) کرتا ہے۔ لہذا اسے چھانٹنے کا انتخاب (selection sort) کہتے ہیں۔

کاذب کوڈ (Pseudo code)

1۔ پہلے عنصر یعنی اشاریہ-0 (Index-0) سے شروع کریں، ہم صف میں سب سے چھوٹے عنصر کی تلاش کرتے ہیں اور اسے پہلے مقام کے عنصر سے بدل دیتے ہیں۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:26 PM

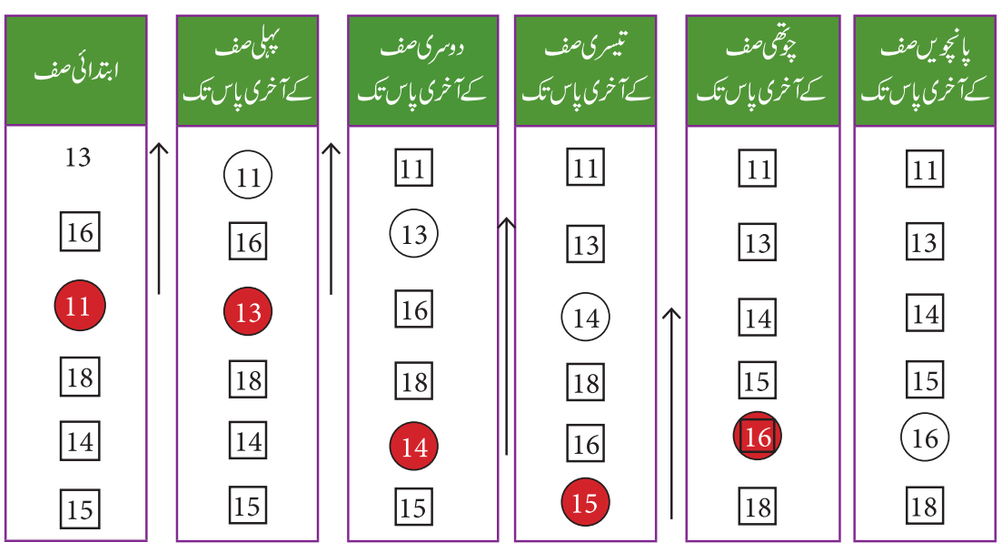
2۔ اب ہم دوسرے عنصر کے مقام کی طرف بڑھتے ہیں، اور ذیلی صف میں موجود سب سے چھوٹے عنصر کی تلاش کرتے ہیں، انڈیکس شروع کرنے سے لے کر ذیلی صف کے آخری اشاریہ تک۔

3۔ اب دوسرا سب سے چھوٹا عنصر، جس کی شناخت مرحلہ نمبر 2 میں کی گئی ہے اسے دوسری پوزیشن پر یا اصلی صف میں تبدیل کریں۔ اس کو تختی صف (sub array) کی پہلی پوزیشن بھی کہا جاتا ہے۔

4۔ اس کو دہرایا جاتا ہے، یہاں تک کہ صف مکمل طور پر ترتیب نہ ہو جائے۔

آیئے ایک صف (array) کو فرض کریں، جس کی قیمتیں اس طرح ہیں۔ {13, 16, 11,18, 14, 15}

ذیل میں، ہمارے پاس اس کی ایک تصویری نمائندگی ہے کہ چھانٹنے کا انتخاب کس طرح دی جانے والی صف کو ترتیب دے گی۔



پہلی پاس میں، سب سے چھوٹا عنصر 11 ہوگا، لہذا اسے پہلی پوزیشن پر رکھا جائے گا۔

اس کے بعد، اگلے سب سے چھوٹے عنصر کو ایک صف سے تلاش کیا جائے گا۔ اب ہم سب سے چھوٹے کے طور پر 13 حاصل کریں گے، لہذا اسے پھر دوسری پوزیشن پر رکھا جائے گا۔

پھر پہلا عنصر چھوڑ کر، باقی عناصر سے، اگلا سب سے چھوٹا عنصر تلاش کیا جائے گا۔ ہم سب سے چھوٹے کے طور پر 13 حاصل کریں گے، لہذا اس کے بعد اسے دوسری پوزیشن پر رکھا جائے گا۔

اس کے بعد 11 اور 13 کو چھوڑ دیں کیونکہ وہ صحیح پوزیشن پر ہیں، ہم باقی عناصر سے اگلے سب سے چھوٹے عنصر کو تلاش کریں گے اور اسے تیسری پوزیشن پر رکھیں گے اور اس وقت تک یہ کرتے رہیں گے جب تک کہ صف (array) مکمل طور پر چھانٹی نہیں جاتی (sorted)۔

آخر کار ہمیں pass کے آخر میں چھانٹی ہوئی صف، اس طرح حاصل ہوگی، جیسا کہ اوپر دکھایا گیا ہے۔

4.5.3 داخلہ کو چھانٹنا (Insertion sort)

داخلہ کو چھانٹنا ایک سادہ چھانٹنے کا الگورتھم ہے۔ یہ عناصر کو ایک ایک کر کے فہرست سے نکال کر اور پھر ان کی صحیح پوزیشن میں ایک نئی ترتیب شدہ فہرست میں داخل کرنے کا کام کرتا ہے۔ یہ الگورتھم اختتام پر حتمی چھانٹی ہوئی صف بناتا ہے۔ جیسا کہ ہم نے بحث کیا ہے، یہ الگورتھم n–1 پاس کی تعداد کو استعمال کر کے حتمی چھانٹی ہوئی فہرست بناتا ہے جو اگلی الگورتھم کے مطابق ہوتی ہے، جس کے بارے میں ہم نے تبادلہ خیال کیا تھا۔

داخلہ کی چھانٹ کے لئے کاذب کوڈ (Pseudo for Insertion sort)

مرحلہ 1 - اگر یہ پہلا عنصر ہے تو، یہ پہلے ہی چھانٹا ہوا ہے۔

مرحلہ 2 - اگلے عنصر کو چُنیں (pick)۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:26 PM

مرحلہ 3 - ترتیب شدہ ذیلی فہرست(sub-list) میں شامل تمام عناصر کے ساتھ موازنہ کریں۔

مرحلہ 4 - ترتیب شدہ ذیلی فہرست میں شامل تمام عناصر کو شفٹ کریں جو چھانٹنے کی قیمت سے زیادہ ہے۔

مرحلہ 5 - قیمت داخل کریں۔

مرحلہ 6 - فہرست مکمل طور پر چھانٹے جانے تک دہرائیں۔

فرض کیجئے کہ 44 چھانٹا گیا ہے
پہلے آئٹم کی فہرست

44	16	83	07	67	21	34	45	10
16	44	83	07	67	21	34	45	10
16	44	83	07	67	21	34	45	10
07	16	44	83	67	21	34	45	10
07	16	44	67	83	21	34	45	10
07	16	21	44	67	83	34	45	10
07	16	21	34	44	67	83	45	10
07	16	21	34	44	45	67	83	10
07	10	16	21	34	44	45	67	83

16 کو داخل کیا گیا
83 کو داخل کیا گیا
07 کو داخل کیا گیا
67 کو داخل کیا گیا
21 کو داخل کیا گیا
34 کو داخل کیا گیا
45 کو داخل کیا گیا
10 کو داخل کیا گیا

پاس (pass) کے اختتام پر داخلہ کی چھانٹ کا الگورتھم (insertion sort algorithm)، نزولی ترتیب کے مطابق چھانٹا ہوا output دیتا ہے، جیسا کہ ذیل میں دکھایا گیا ہے:

07	10	16	21	34	44	45	67	83

4.6. متحرک پروگرامنگ
Dynamic programming

متحرک پروگرامنگ ایک الگورتھمی ڈیزائن کا طریقہ ہے، جب کسی مسئلے کے حل کو فیصلوں کی ترتیب کے نتیجے میں دیکھا جاسکتا ہے۔ متحرک پروگرامنگ کا نقطہ نظر تقسیم اور فتح (divide and conquer) کے مشابہ ہے۔ دیئے گئے مسئلے کو چھوٹے اور ابھی تک چھوٹے چھوٹے ذیلی مسئلوں (sub-problems) میں تقسیم کیا گیا ہے۔

جب بھی مسائل کو اسی طرح کے ذیلی مسئلوں میں تقسیم کیا جاسکتا ہے تو متحرک پروگرامنگ استعمال کی جاتی ہے۔ تاکہ عمل کو مکمل کرنے کے لئے ان کے نتائج کو دوبارہ استعمال کیا جاسکے۔ متحرک پروگرامنگ کے طریق کار کو بہتر طریقے سے حل معلوم کرنے کے لئے استعمال کیا جاتا ہے۔ ہر اندرونی ذیلی مسئلوں کے لئے، متحرک الگورتھم پہلے حل شدہ ذیلی مسئلوں کے نتائج کو جانچنے کی کوشش کرے گا۔ بہتر حل حاصل کرنے کے لئے منطبق شدہ ذیلی مسئلوں (overlapped sub-problems) کے حل اکٹھے کیے گئے ہیں۔

متحرک پروگرامنگ کرنے کے اقدامات

- دیئے گئے مسئلے کو چھوٹے انطباق کرنے والے ذیلی مسئلوں میں تقسیم کیا جائے گا۔
- دیئے گئے مسئلے کا ایک زیادہ سے زیادہ حل چھوٹے ذیلی مسئلے کا نتیجہ استعمال کرکے حاصل کیا جاسکتا ہے۔
- متحرک الگورتھم، یادداشت کو استعمال کرتے ہیں۔

نوٹ:

یادداشت (memorization) ایک اصطلاح ہے جو مکمل طور پر استعمال کر کے کمپیوٹر کے پروگراموں کو تیز تر کرنے اور بنیادی طور پر مہنگے فنکشن کالوں کے نتائج کو اسٹور کرکے اور ان کے cached نتیجوں کے اگر input یکساں ہوں تو انہیں دوبارہ حاصل کراتا ہے۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:27 PM

4.6.1۔ فیبوناسی سیریز - ایک مثال
(Fibonacci Series - An example)

فبوناسی سیریز دو پہلے اعداد کو جمع کر کے بعد میں اس کے مطابق عدد تیار کرتی ہے۔ Fibonacci سیریز دو اعداد سے شروع ہوتی ہے۔ Fib 0 اور Fib 1۔ Fib 0 اور Fib 1 کے ابتدائی قیمتوں کو 0 اور 1 کے طور پر لیا جاسکتا ہے۔

فبوناسی سیریز مندرجہ ذیل شرائط کو پورا کرتی ہے۔

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

لہذا، n کی قیمت 8 کے لئے ایک فیبوناسی سیریز اس طرح دکھائی دیتی ہے۔

$$Fib_8 = 0\ 1\ 1\ 2\ 3\ 5\ 8\ 1\ 3$$

4.6.2 متحرک پروگرامنگ کے نقطہ نظر کے ساتھ فیبوناسی کا دوہرائے جانے والا الگورتھم
(Fibonacci Iterative Algorithm with Dynamic Programming approach)

مندرجہ ذیل مثال میں فبوناسی سیریز تیار کرنے کے لئے متحرک پروگرامنگ کے نقطہ نظر کا ایک آسان طریقہ دکھاتا ہے۔

شروع کریں : f0=0, f1=1

مرحلہ 1 : فبوناسی f0 اور f1 کے ابتدائی قیمتوں کو پرنٹ کریں۔

مرحلہ 2 : فبوناسی fib ← f0 + f1 کو محسوب کریں۔

مرحلہ 3 : f0 ← f1, f1 ← fib کو تفویض کریں۔

مرحلہ 4 : فیبوناسی fib کی اگلی قیمت پرنٹ کریں۔

مرحلہ 5 : مرحلہ نمبر 2 کو جائیں اور رقموں کی مخصوص تعداد تیار ہونے تک دہرائیں۔

مثال کے طور پر اگر ہم 10 ہندسوں تک فبوناسی سیریز تیار کرتے ہیں تو، الگورتھم سیریز کو اس طرح تیار کرے گا:

فبوناسی سیریز یہ ہے: 0 1 1 2 3 5 8 13 21 34 55

یاد رکھنے کے نکات :Points to remember

- ایک الگورتھم کسی خاص کام کو پورا کرنے کے لئے ہدایات کا ایک محدود سیٹ ہے۔
- الگورتھم مرحلہ وار ہدایات پر مشتمل ہے جو کسی کام کو پورا کرنے کے لئے ضروری ہے اور پروگرامر کو پروگرام تیار کرنے میں مدد کرتا ہے۔
- ایک پروگرامنگ زبان میں الگورتھم کا اظہار ایک پروگرام کہلاتا ہے۔
- الگورتھم کا تجزیہ عام طور پر اس میں شامل مختلف آپریشنوں کے چلانے اور عمل درآمد کے وقت سے متعلق ہوتا ہے۔
- الگورتھم میں جگہ کی پیچیدگی اس کی تکمیل تک چلانے کے لئے درکار میموری کی مقدار ہے۔
- بگ O اکثر ایک الگورتھم کے بدترین صورت حال کی وضاحت کرنے کے لئے استعمال ہوتا ہے۔
- بگ اومیگا Ω نچلے حد (بہترین معاملہ) کی وضاحت کرنے کے لئے استعمال ہوتا ہے۔
- الگورتھم کے وقت کی پیچیدگی عمل کو مکمل کرنے کے لئے الگورتھم کے ذریعہ اٹھائے گئے مرحلوں کی تعداد کے ذریعہ دی جاتی ہے۔
- الگورتھم کی کارکردگی کو الگورتھم کے ذریعہ محسوب وسائل (computational resources) کی تعداد کے طور پر استعمال کیا جاتا ہے۔
- الگورتھم ڈیزائن کرنے کا ایک طریقہ الگورتھم حکمت عملی کہلاتا ہے۔
- اسپیس ٹائم یا ٹائم میموری ٹریڈ آف زیادہ اسٹوریج کی جگہ کا استعمال کر کے یا زیادہ وقت گزار کر بہت کم جگہ میں دیئے گئے الگورتھم کو حل کر کے کم وقت میں حل کرنے کا ایک طریقہ ہے۔
- Asymptotic Notations ایسی زبانیں ہیں جو وقت اور جگہ کی پیچیدگی کے بارے میں معنی خیز بیانات کا استعمال کرتی ہیں۔
- bubble sort ایک آسان الگورتھم ہے۔ الگورتھم ایک صف میں محفوظ کردہ قیمتوں کی فہرست کے آغاز سے شروع ہوتا ہے۔ اس سے متصل عناصر کے ہر جوڑے کا موازنہ کیا جاتا ہے اور اگر وہ چھانٹے ہوئے نہیں ہوں تو ان کو بدل دیتا ہے۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:28 PM

Points to remember: یادرکھنے کے نکات

- چھانٹنے کا انتخاب (Selection sort) ایک آسان چھانٹنے والا الگورتھم (simple sorting algorithm) ہے، جو فہرست میں ہر پاس (pass) کے لئے صرف ایک ہی تبادلہ کرکے bubble sort کی کارکردگی میں بہتری لاتا ہے۔
- داخلہ کو چھانٹنا (Insertion sort) ایک سادہ چھانٹنے کا الگورتھم ہے۔ یہ عناصر کو ایک ایک کرکے فہرست سے نکال کر اور پھر ان کی صحیح پوزیشن میں ایک نئی ترتیب شدہ فہرست میں داخل کرنے کا کام کرتا ہے۔
- متحرک پروگرامنگ ایک الگورتھمی ڈیزائن کا طریقہ ہے، جب کسی مسئلے کے حل کو فیصلوں کی ترتیب کے نتیجے میں دیکھا جاسکتا ہے۔

39Q05

Evaluation محاسبہ

حصہ 1

صحیح جواب کا انتخاب کریں: (1 نمبر)

1۔ یہ لفظ ایک فارسی ریاضی دان ابوجعفر محمد ابن موسیٰ موسیٰ الخوارزمی کے نام سے موسوم ہے۔
(A) فلوچارٹ (B) فلو (C) الگورتھم (D) ترکیب

2۔ مندرجہ ذیل چھانٹنے والے الگورتھم میں سے کونسے الگورتھم میں کم سے کم ادل بدل (swaps) کی ضرورت ہے؟
(A) ببل چھانٹ (bubble sort) (B) فوری چھانٹ (quick sort)
(C) ضم چھانٹ (merge sort) (D) چھانٹنے کا انتخاب (selection sort)

3۔ الگورتھم کی کارکردگی کے لئے دو اہم اقدامات یہ ہیں۔
(A) پروسیسر اور میموری (Processor and memory) (B) پیچیدگی اور صلاحیت (complexity and capacity)
(C) وقت اور جگہ (time and space) (D) ڈیٹا اور اسپیس (Data and space)

4۔ خطی سرچ الگورتھم کی پیچیدگی (complexity of linear search algorithm) یہ ہے۔
O(n) (A) O(log n) (B) O(n2) (C) O(n log n) (D)

5۔ درج ذیل چھانٹنے والے الگورتھموں میں سے کس میں بدترین پیچیدہ معاملہ ہے ؟
(A) ببل چھانٹ (bubble sort) (B) فوری چھانٹ (quick sort)
(C) ضم چھانٹ (merge sort) (D) چھانٹنے کا انتخاب (selection sort)

6۔ مندرجہ ذیل میں سے کون سی قسم مستحکم (قیام پذیر) چھانٹنے والا الگورتھم نہیں ہے؟
(A) داخلہ کی چھانٹ (insertion sort) (B) چھانٹنے کا انتخاب (selection sort)
(C) ببل چھانٹ (bubble sort) (D) ضم چھانٹ (merge sort)

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:28 PM

7۔ ببل چھانٹ کی بہترین ترتیب میں وقت کی پیچیدگی یہ ہے۔

θ (n log n)2) (D)　　θ (n2) (C)　　θ (nlog n) (B)　　θ (n) (A)

8۔ asymptotic تشخیص میں Θ کی علامت یہ نمائندگی کرتا ہے۔

(A) بنیادی صورت حال (B) اوسط صورت حال (C) بدترین صورت حال (D) NULL صورت حال

9۔ اگر کسی مسئلہ کو کئی ذیلی مسئلوں میں بانٹ دیا جائے، جس کو کئی بار دوبارہ استعمال کیا جاسکتا ہے، تو وہ مسئلہ کونسی خصوصیت کو ظاہر کرتا ہے ؟

(A) منطبق شدہ ذیلی مسئلے (overlapping subproblems)

(B) استعمال کی جانے والی ذیلی ساخت (Optimal substructure)

(C) یادداشت (memorization) (D) لالچی (greedy)

10۔ متحرک پروگرامنگ میں، پہلے سے حساب شدہ قیمتوں کو کو ذخیرہ (محفوظ) کرنے کی تکنیک کو یہ کہا جاتا ہے؟

(A) قیمت کو بچانے کی خصوصیت (saving value property)

(B) قیمت کو محفوظ کرنے کی خصوصیت (Storing value property)

(C) یادداشت (memorization) (D) نقشہ سازی (Mapping)

حصہ-II

مندرجہ ذیل سوالات کے جوابات دیجئے۔ (2 نمبر)

1۔ الگورتھم کیا ہے ؟

2۔ کاذب کوڈ (pseudo code) کی وضاحت کریں۔

3۔ الگورسٹ (algorist) کون ہے؟

4۔ چھانٹنا (sorting) کیا ہے؟

5۔ تلاش (سرچنگ) (searching) کسے کہتے ہیں؟ اس کی اقسام لکھیں۔

حصہ-III

درج ذیل سوالات کے جوابات دیجئے۔ (3 نمبر)

1۔ الگورتھم کی خصوصیات کی فہرست بنائیں۔

2۔ الگورتھم پیچیدگی اور اس کی اقسام کے بارے میں تبادلہ خیال کریں۔

3۔ وہ کون سے عوامل ہیں جو وقت اور جگہ کی پیچیدگی کو متاثر کرتے ہیں؟

4۔ Asymptotic ترقیم پر ایک نوٹ لکھیں۔

5۔ متحرک پروگرامنگ (dynamic programming) کے بارے میں آپ کیا جانتے ہیں ؟

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:28 PM

حصہ - IV

درج ذیل سوالات کے جوابات لکھئے۔ (5 مارکس)

1۔ الگورتھم کی خصوصیات کی وضاحت کریں۔

2۔ خطی سرچ الگورتھم (linear search algorithm) کے بارے میں تبادلہ خیال کریں۔

3۔ ثنائی تلاش (binary search) کیا ہے؟ مثال کے ساتھ بحث کریں۔

4۔ مثال کے ساتھ ببل چھانٹ الگورتھم (bubble sort algorithm) کی وضاحت کریں۔

5۔ متحرک پروگرامنگ (dynamic programming) کے تصور کی مناسب مثال کے ساتھ وضاحت کریں۔

کتب حوالہ جات

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*

ویب حوالہ جات Web References

www.wickipedia.org

فردی مطالعہ (کیس اسٹڈی)/ طالب علمی کی سرگرمی CASE STUDY/ STUDENT'S ACTIVITY

1۔ تمام ضروری شرائط کو پورا کر کے اپنے کلاس طالب علموں کے سہ ماہی امتحانات کے نمبروں کے گریڈنگ سسٹم کے الگورتھم بنائیں۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:29 PM

بارہویں کے نصاب میں پائیتھن (Python) کی کیا ضرورت ہے؟

Why Python in standard XII curriculum?

یہ وہ سوال ہے جو اساتذہ اور طلبہ کے ذہنوں کو جھنجھوڑ رہا ہے۔

موجودہ کتاب کو اس طرح ترتیب دیا گیا ہے کہ نوزائیدہ پڑھنے والا بھی پائیتھن کے پروگرامنگ کو سمجھ کر اس میں کام کرسکتا ہے۔

شہادتیں (Testimonies)

• "شروع سے ہی پائیتھن گوگل کا ایک اہم حصہ رہا ہے اور اب بھی اسی طرح قائم ہے جیسے جیسے یہ نظام بڑھتا اور تیار ہوتا ہے۔ آج گوگل کے درجنوں انجینئر پائیتھن کا استعمال کرتے ہیں اور ہم اس زبان میں مہارت رکھنے والے مزید افراد کی تلاش کر رہے ہیں۔" - پیٹر نورویگ، گوگل، انکارپوریشن میں سرچ کوالٹی کے ڈائریکٹر۔

• "پائیتھن ہماری سائٹ کے لئے تیز رفتار ہے اور کم سے کم ڈویلپرز کے ساتھ، ریکارڈ اوقات میں ہمیں برقرار رکھنے والی خصوصیات تیار کرنے کی اجازت دیتا ہے۔"

-کوونگ ڈو، سافٹ ویئر آرکیٹیکٹ، یوٹیوب ڈاٹ کام۔

پائیتھن کی مقبولیت نے حالیہ برسوں میں مستحکم اور غیر منقولہ نمو دیکھا ہے۔ آج، پائیتھن کے ساتھ واقفیت ہر پروگرامر کے لئے ایک فائدہ ہے، کیونکہ پائیتھن نے ہر جگہ اپنی منفرد بنالی ہے اور کسی بھی سوفٹ ویئر حل میں ادا کرنے کے لئے مفید کردار ادا کرتا ہے۔

دوسری زبانوں کے مقابلے میں پائیتھن نے متاثر کن ترقی کا تجربہ کیا ہے۔ نیویارک کے انسٹی ٹیوٹ آف الیکٹریکل اینڈ الیکٹرانکس انجینئرز کے ذریعہ شائع ہونے والا آئی ای ای اسپیکٹرم میگزین، لگاتار دوسرے سال بھی 2018 کے لئے اعزاز کی فہرست میں ہے۔

مذکورہ بالا شماریاتی ڈیٹا نے پائیتھن 100 اسکورنگ کے ساتھ اپنی گرفت برقرار رکھی ہے، اور C++ زبان 99.7 کے اسکور کے ساتھ دوسرے نمبر پر ہے۔ پائیتھن کے مقبول ہونے کی وجہ سے متعدد اعلیٰ تکنیکی ادارے، جیسے google، Instagram، Pinterest، Yahoo، Disney، IBM، Nokia ، وغیرہ۔

1. Python	100.0
2. C++	99.7
3. Java	97.5
4. C	96.7
5. C#	89.4
6. PHP	84.9
7. R	82.8
8. JavaScript	82.6
9. Go	76.4
10. Assembly	74.1

بہت سے تجارتی اداروں کو مندرجہ ذیل وجوہات کی بنا پر پائیتھن کا انتخاب کرنے کا مشورہ دیا جاتا ہے۔

- آسان syntax اور پڑھنے کی اہلیت
- OOPS کے ساتھ اعلیٰ سطحی اسکرپٹنگ زبان
- بہت زیادہ افعال کے لئے مشہور، ماڈیولز، لائبریریوں، فریم ورکز اور ٹول کٹس کے اضافے کے لئے۔
- بلٹ ان افعال سائنسی کمپیوٹنگ کی حمایت کرتا ہے۔

کمپیوٹرز کی آمد کے ساتھ، تقریبا تمام شعبوں میں ہمارے کام کرنے کے انداز میں نمایاں تبدیلیاں آئی ہیں۔ کمپیوٹرائزیشن نے ہر تنظیم میں پیداواری صلاحیت کو بہتر بنانے اور فیصلہ سازی کو تیز کرنے میں مدد فراہم کی ہے۔ یہاں تک کہ افراد کے لئے، چاہے وہ انجینئر ہوں، ڈاکٹر ہوں، چارٹرڈ اکاؤنٹنٹ ہوں یا گھریلو ساز، کام کرنے کا انداز یکسر بدل گیا ہے۔ اس طرح جب ہم جاتے ہیں، آیئے تبدیلی کو قبول کرتے ہیں اور ایک روشن دن کی طرف آگے بڑھتے ہیں۔

12 CS UNIT 4 CDR Set
Sunday, April 16, 2023 9:35:29 PM

U86IT

سیکھنے کے مقاصد

اس سبق کا مطالعہ کرنے کے بعد، طلباء،

- پائیتھن کے پروگرام بنانے کے لئے گرافیکل یوزر انٹرفیس (GUI) اور انٹیگریٹڈ ڈویلپمنٹ ماحول (IDE) کے استعمال کی تعریف کریں گے۔
- پروگرامنگ کے لئے انٹرایکٹو اور اسکرپٹ وضع میں کام کریں گے۔
- متغیر کے لئے اقدار تشکیل دیں اور تفویض کریں گے۔
- پائیتھن میں ڈیٹا کی مختلف اقسام کے تصور اور استعمال کو سمجھیں گے۔
- مختلف قسم کے آپریٹرز (ریاضی، متعلقہ اور منطقی) کی اہمیت اور استعمال کی تعریف کریں گے۔
- پائیتھن اظہار (زبانیں) اور بیانات تخلیق کرنا۔

5.1 تعارف Introduction



پائیتھن ایک عمومی مقصد کی پروگرامنگ زبان ہے جو گائڈو وان روسم (Guido Van Rossum نے CWI (Centrum Wiskunde & Informatica) سے تشکیل دی ہے جو نیدرلینڈ میں ریاضی اور کمپیوٹر سائنس کے لئے قومی تحقیقاتی ادارہ ہے۔ یہ زبان 1991 میں جاری کی گئی تھی۔ پیتھون کو ستر کی دہائی سے BBC کی ایک مزاحیہ سیریز - "Monty Python's Flying Circus" سے اس کا نام ملا۔ پائیتھن Procedural اور آبجیکٹ اورینٹڈ پروگرامنگ (Object Oriented Programming) دونوں طریقوں کی حمایت کرتا ہے۔

5.2 پائیتھن کی اہم خصوصیات Key features of Python

- ✓ یہ عام مقصد کی پروگرامنگ زبان ہے جو سائنسی اور غیر سائنسی پروگرامنگ دونوں کے لئے استعمال کی جاسکتی ہے۔
- ✓ یہ ایک پلیٹ فارم کی آزاد پروگرامنگ زبان ہے۔
- ✓ پائیتھن میں لکھے گئے پروگرام آسانی سے پڑھنے کے قابل اور قابل فہم ہیں۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:41 PM

پائیتھن IDLE (Integrated Development Learning Environment) کا ورژن 3.x استعمال کرتا ہے اور پائیتھن کوڈ کو چلانے کے لئے استعمال کیا جاتا ہے۔ اسے ویب وسائل www.python.org سے ڈاؤن لوڈ کیا جاسکتا ہے۔

5.3 پائیتھن میں پروگرامنگ
Programming in Python

پائیتھن میں، پروگراموں کو دو طریقوں سے لکھا جاسکتا ہے یعنی انٹرایکٹو وضع (Interactive mode) اور اسکرپٹ وضع (Script mode)۔ انٹرایکٹو موڈ ہمیں python command prompt (>>>) میں کوڈ لکھنے کی اجازت دیتا ہے جبکہ اسکرپٹ موڈ میں پروگراموں کو توسیع .py کے ساتھ الگ فائل کے طور پر لکھا اور اسٹور کیا اور چلایا جاسکتا ہے۔ اسکرپٹ وضع کو پائیتھن سورس فائل بنانے اور ایڈٹ کرنے کے لئے استعمال کیا جاتا ہے۔

5.3.1 انٹرایکٹو وضع پروگرامنگ (Interactive mode programming)

انٹرایکٹو موڈ میں پائیتھن کوڈ کو براہ راست ٹائپ کیا جاسکتا ہے اور مترجم (interpreter) فوری طور پر نتیجہ (نمائش) ظاہر کرتا ہے۔ انٹرایکٹو موڈ کو ایک سادہ کیلکولیٹر کے طور پر بھی استعمال کیا جاسکتا ہے۔

(i) پائیتھن IDLE کو طلب کرنا

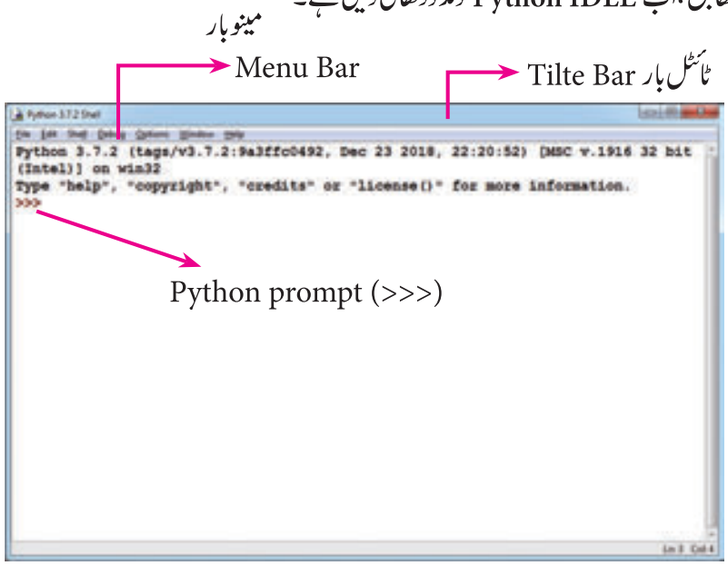
مندرجہ ذیل کمانڈ کے ذریعہ Windows OS سے پائیتھن IDLE کو طلب کیا جاسکتا ہے۔

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

(یا)

اگر دستیاب ہو تو ڈیسک ٹاپ پر پائیتھن آئیکن  پر کلک کریں۔

خاکہ 5.1 میں دکھائے مطابق، اب Python IDLE ونڈو دکھائی دیتی ہے۔



Python IDLE Window

پرامپٹ (>>>) اشارہ کرتا ہے کہ مترجم ہدایات قبول کرنے کے لئے تیار ہے۔ لہذا، اسکرین پر پرامپٹ کا مطلب ہے کہ IDLE انٹرایکٹو موڈ میں کام کر رہا ہے۔ اب ہم ریاضی کے ایک سادہ جملہ کو استعمال کر کے ایک سادہ کیلکولیٹر کی حیثیت سے اسے چلانے کی کوشش کریں۔

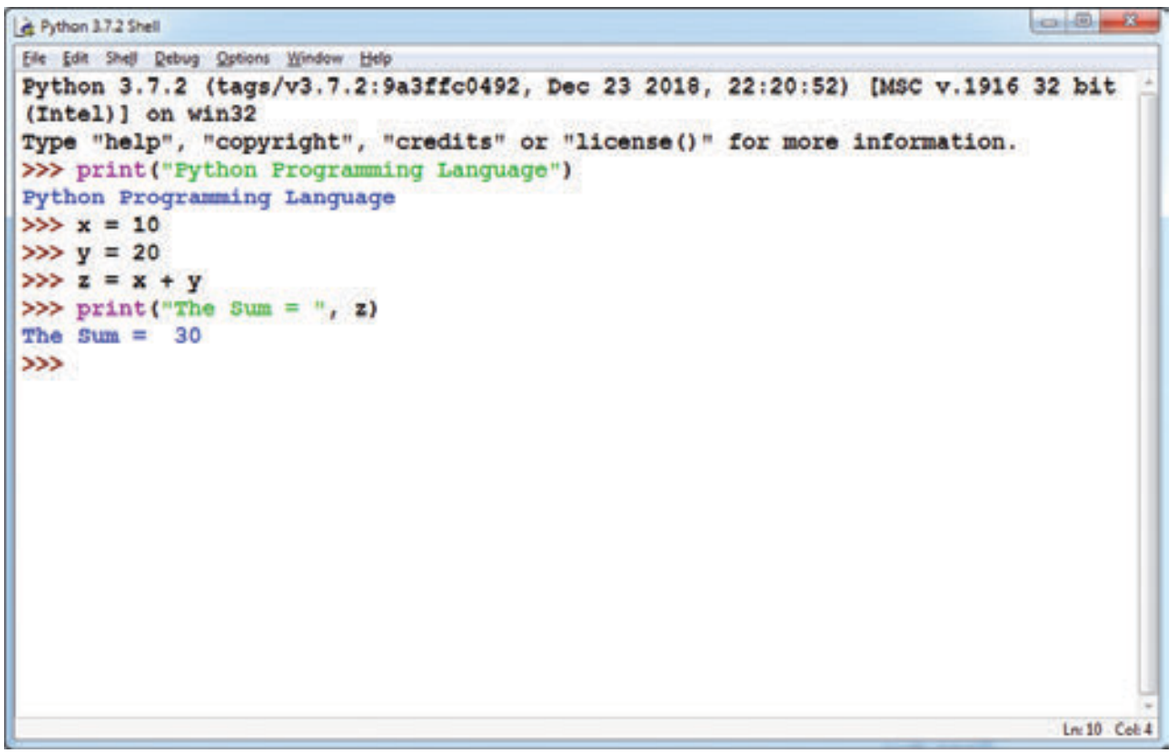
12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:41 PM

مثال1:

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

مثال2:

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



Python Interactive Window

5.3.2 اسکرپٹ وضع پروگرامنگ (Script mode programming)

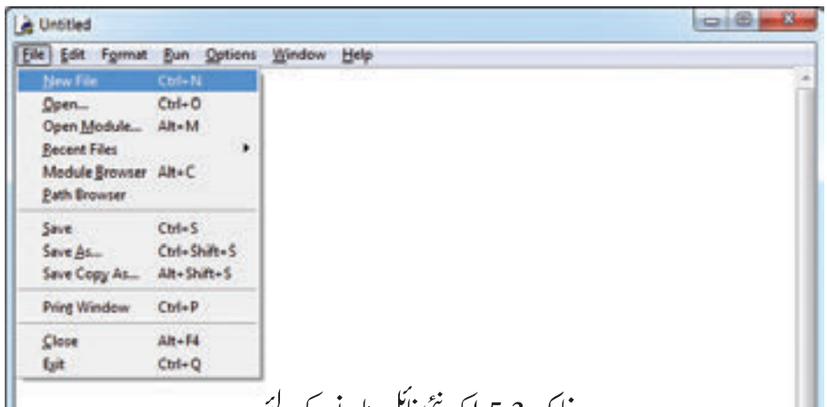
بنیادی طور پر، اسکرپٹ ایک ٹیکسٹ فائل ہے جس میں پائیتھن کے بیانات شامل ہیں۔ پائیتھن اسکرپٹ دوبارہ استعمال کوڈ ہیں۔ ایک بار اسکرپٹ بن جائے تو اسے دوبارہ ٹائپ کیے بغیر بار بار عمل میں لایا جاسکتا ہے۔ اسکرپٹس، ایڈیٹ کرنے کے قابل ہیں۔

(i) پائیتھن میں اسکرپٹ بنانا (Creating scripts in Python)

1۔ File → New File کا انتخاب کریں یا پائیتھن شیل ونڈو میں Ctrl + N دبائیں۔

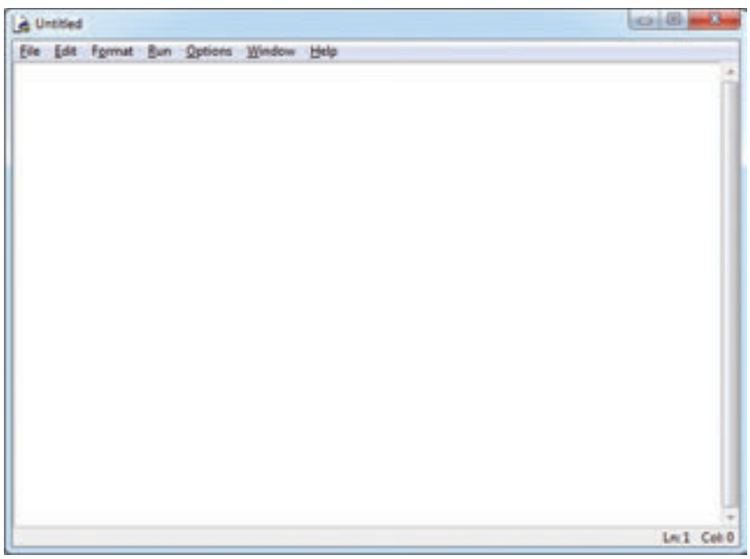
CZSGL

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:41 PM



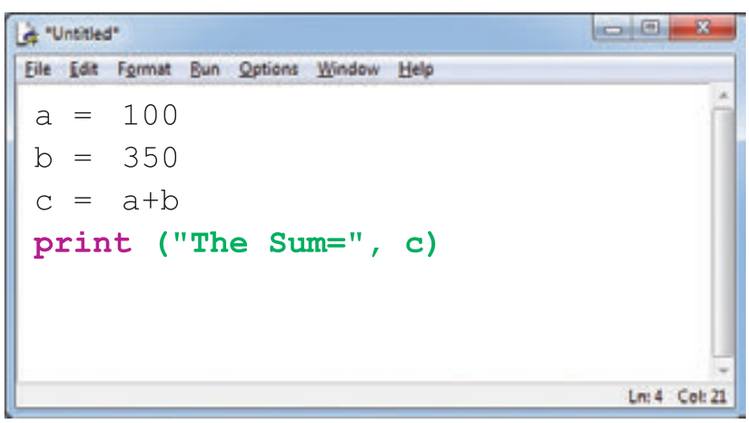
خاکہ 5.3 ایک نئی فائل بنانے کے لئے

2۔ ایک بلاعنوان (untitled) خالی اسکرپٹ ٹیکسٹ ایڈیٹر اسکرین پر ظاہر ہوگا جیسا کہ شکل 5.3(a) میں دکھایا گیا ہے۔



خاکہ 5.3 (a) پیتھان اسکرپٹ ایڈیٹر Untitled, Blank

3۔ اسکرپٹ ایڈیٹر میں درج ذیل کوڈ ٹائپ کریں۔



a =100

b = 350

c = a+b

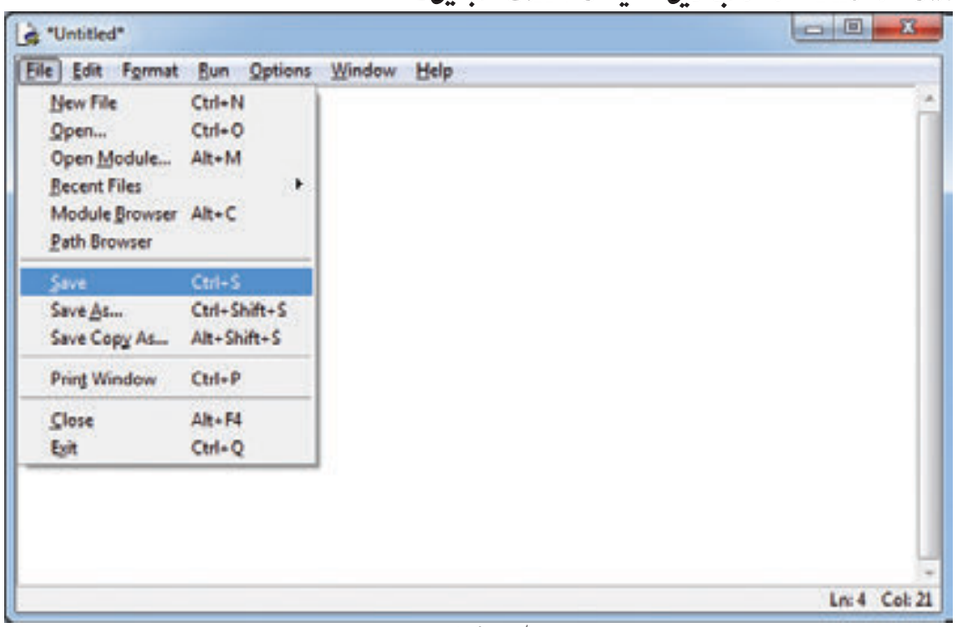
print ("The Sum=", c)

خاکہ 5.4 پیتھان سیمپل کوڈ

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:41 PM

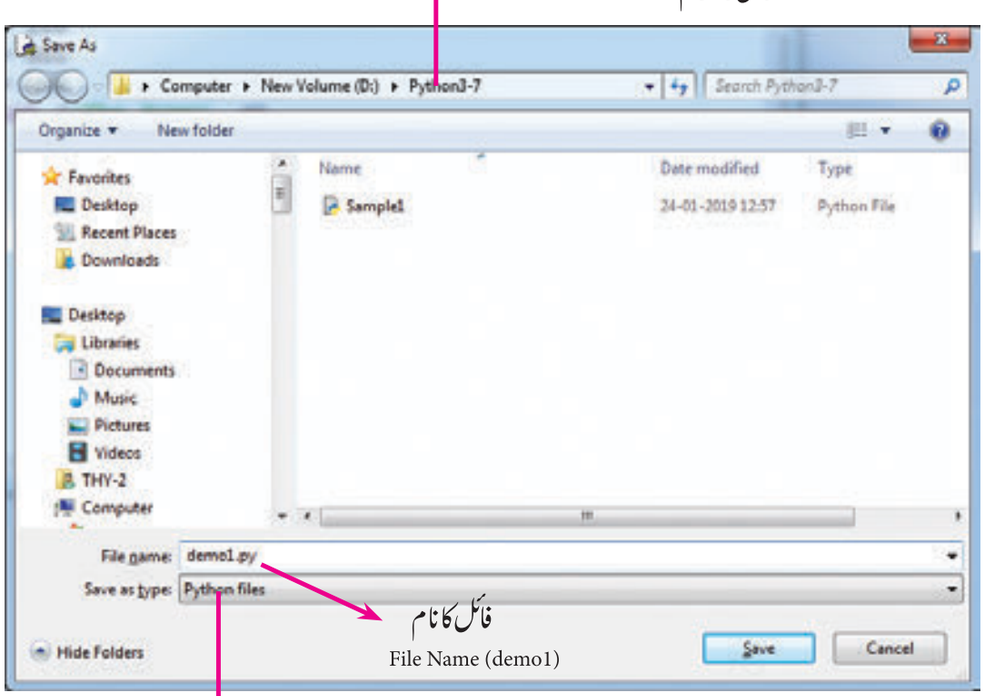
(ii) پائیتھن اسکرپٹ محفوظ (save) کرنا

(1) File → Save کا انتخاب کریں یا Ctrl + S دبائیں۔



خاکہ 5.5 پہلی بار فائل کو save کرنا۔

(2) اب، تصویر 5.6 میں دکھائے مطابق، save as ڈائیلاگ باکس اسکرین پر ظاہر ہوتا ہے



خاکہ 5.6 save as ڈائیلاگ باکس

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:42 PM

(3) save as ڈائیلاگ باکس میں، وہ مقام منتخب کریں جہاں آپ اپنا پائیتھن کوڈ محفوظ کرنا چاہتے ہیں، اور File name کے باکس میں ٹائپ کریں۔ پائیتھن فائلوں کو توسیع .py کے ساتھ ڈیفالٹ محفوظ کیا جاتا ہے۔ اس طرح، پائیتھن اسکرپٹ کو پائیتھن اسکرپٹ ایڈیٹر کا استعمال کرتے ہوئے تخلیق کرتے وقت، فائل کی توسیع کی وضاحت کرنے کی ضرورت نہیں ہے۔

(4) آخر میں، اپنے پائیتھن اسکرپٹ کو محفوظ کرنے کے لئے save کے بٹن پر کلک کریں۔

(iii) پائیتھن اسکرپٹ کو چلانا

(1) Run → Run module کا انتخاب کریں یا F5 دبائیں۔

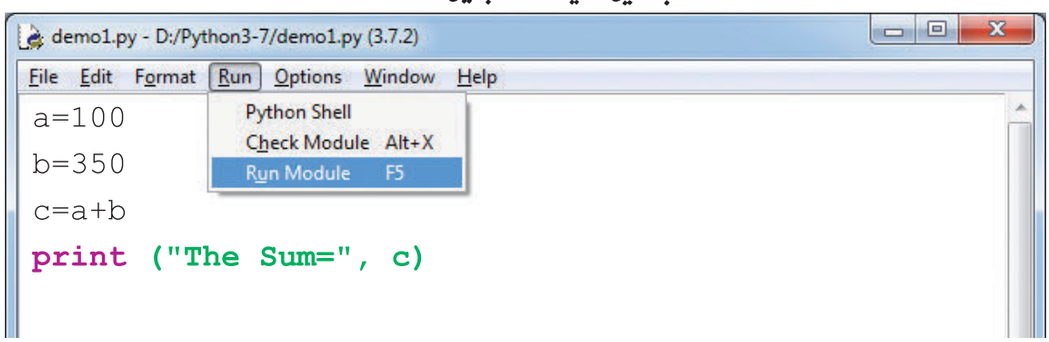
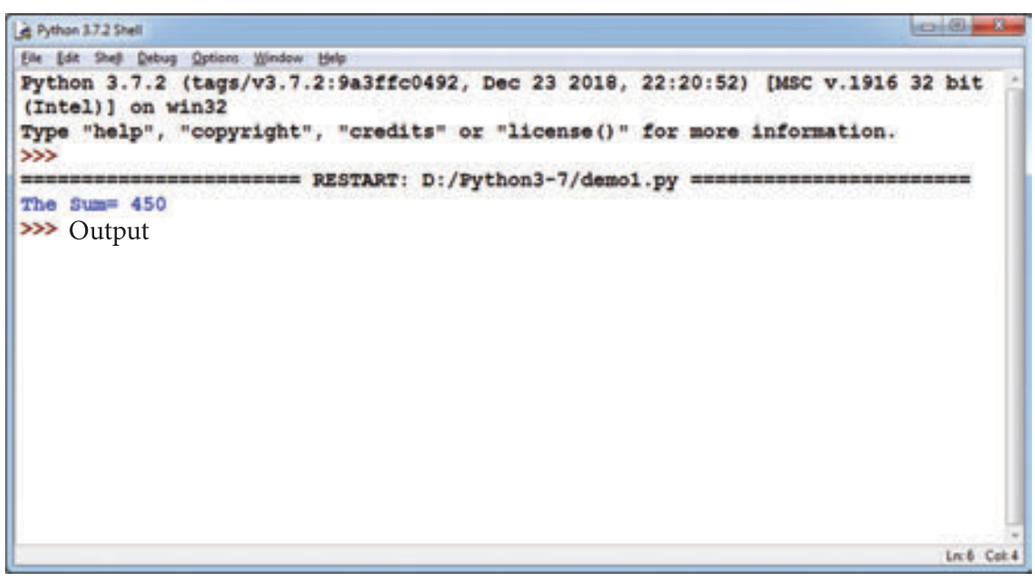


Figure 5.7 – To Execute Python Script

(2) اگر آپ کے کوڈ میں کوئی خرابی ہے تو، اسے IDLE ونڈو میں سرخ رنگ میں دکھایا جائے گا، اور پائیتھن غلطی error کی قسم کو بیان کرتے ہیں۔ غلطیوں کو درست کرنے کے لئے، اسکرپٹ ایڈیٹر پر واپس جائیں، اصلاح کریں، فائل کو Ctrl + S یا فائل کا استعمال کرکے save کریں۔ اسے دوبارہ محفوظ کریں اور اسے چلائیں۔

(3) تمام خرابی سے پاک کوڈ (error free code) کے لئے، آؤٹ پٹ پائیتھن کی IDLE ونڈو میں ظاہر ہوگی۔ خاکہ 5.8



Python Script Output Window خاکہ 5.8 پیتھان اسکرپٹ اوٹ پٹ ونڈو

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:42 PM

5.4 ان پٹ اور آؤٹ پٹ فنکشن Input and Output Functions

ایک پروگرام مطلوبہ کام کو پورا کرنے کے لئے صارف کے ساتھ بات چیت کرنے کی ضرورت ہے۔ یہ ان پٹ آؤٹ پٹ افعال کا استعمال کرتے ہوئے حاصل کیا جاسکتا ہے۔ input() فنکشن صارف کے چلتے وقت اعداد و شمار میں داخل ہونے میں مدد کرتا ہے اور عمل کے بعد اسکرین پر پروگرام کے نتائج کو ظاہر کرنے کے لئے آؤٹ پٹ فنکشن print() استعمال ہوتا ہے۔

5.4.1 پرنٹ() فنکشن (The Print() function)

پائیتھن میں، print() فنکشن اسکرین پر نتائج ظاہر کرنے کے لئے استعمال ہوتا ہے۔ print() کے لیے مندرجہ ذیل syntax ہے۔

مثال

```
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” ……)
```

مثال

```
>>> 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
```

print() مانیٹر پر طباعت کرنے سے پہلے اس کے اظہار کی تشخیص کرتا ہے۔ print() ایک پورا بیان دکھاتا ہے جو print() کے اندر مخصوص ہے۔ کوما Comma(,) ایک سے زیادہ آئٹم پرنٹ کرنے کے لئے print() میں جداکار (separator) کے طور پر استعمال ہوتا ہے۔

5.4.2 ان پٹ () فنکشن (Input () function)

پائیتھن میں، input() فنکشن کو رَن ٹائم کے وقت ڈیٹا کو ان پٹ کے طور پر قبول کرنے کے لئے استعمال کیا جاتا ہے۔

input() فنکشن کے لئے syntax یہ ہے۔

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


12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:42 PM

جہاں،نحو میں فوری سٹرنگ صارف کے لئے بیان یا پیغام ہوتا ہے، یہ جاننے کے لئے کہ کیا ان پٹ دیا جاسکتا ہے۔
اگر فوری طور پر string استعمال کیا جاتا ہے تو، یہ مانیٹر پر ظاہر ہوتا ہے۔ صارف ان پٹ آلہ سے متوقع ڈیٹا فراہم کرسکتا ہے۔ input() کی بورڈ سے جو ٹائپ ہوتا ہے اسے لے جاتا ہے اور درج کردہ ڈیٹا کو دیئے جانے والے متغیر میں اسٹور کرتا ہے۔ اگر input() میں فوری اشارے نہیں دیئے جاتے ہیں تو اسکرین پر کوئی پیغام ظاہر نہیں ہوتا ہے۔ لہذا، صارف کو معلوم نہیں ہوگا کہ ان پٹ کے طور پر کیا ٹائپ کرنا ہے۔

مثال 1: فوری سٹرنگ کے ساتھ input()

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

مثال 2 : اشارے کے بغیر input()

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

نوٹ کریں کہ مثال-2 میں، input() میں کوئی فوری سٹرنگ نہیں ہے، اس طرح صارف کو معلوم نہیں ہوگا کہ ان پٹ کے طور پر کیا ٹائپ کرنا ہے۔ اگر صارف مذکورہ بالا مثال میں دیئے گئے غیر متعلقہ اعداد و شمار کو داخل کرتا ہے، تو آؤٹ پٹ غیر متوقع ہوگا۔ لہذا، اپنے پروگرام کو زیادہ متعامل بنانے کے لئے، input() کے ساتھ فوری تار فراہم کریں۔

input() تمام ڈیٹا کو اسٹرنگ یا حروف کے طور پر قبول کرتا ہے لیکن عدد کے طور پر نہیں۔ اگر ایک عددی قیمت داخل کی جاتی ہے تو، input کی قیمتوں کو واضح طور پر اعداد کی قسم میں تبدیل کیا جانا چاہئے۔ int() فنکشن کا استعمال اسٹرنگ ڈیٹا کو واضح طور پر سالم اعداد کو ڈیٹا میں تبدیل کرنے کے لئے کیا جاتا ہے۔ ہم اس طرح کے مزید افعال کے بارے میں بعد کے ابواب میں سیکھیں گے۔

مثال 3:

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

Output: آؤٹ پٹ

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


12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:42 PM

مثال 4 : مندرجہ بالا پروگرام کے لئے متبادل طریقہ

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

Output: اوٹ پٹ

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

5.5 پائیتھن میں تبصرہ (Comments in Python)

پائیتھن میں، تبصرے ہیش علامت (#) سے شروع ہوتے ہیں۔ خطوط جو # سے شروع ہوتی ہیں ان کو تبصرے سمجھا جاتا ہے اور پائیتھن کے ترجمان نے اسے نظر انداز کیا ہے۔ تبصرے ایک لائن ہو سکتے ہیں یا ایک سے زیادہ لائنیں نہیں ہو سکتی ہیں۔ ملٹی لائن تبصرے ذیل میں دیئے گئے مطابق # (ٹرپل کوٹس) (triple quotes) کے ایک سیٹ کے ساتھ منسلک کیے جانے چاہئیں۔

```
#  جس میں ایک سے زیادہ لائن شامل ہیں۔
#  یہ ایک اکہری خط والا تبصرہ ہے۔
#  یہ ایک ملٹی لائن والا تبصرہ ہے۔
جس میں ایک سے زیادہ خط ہوتے ہیں # ۔
```

5.6 انڈینٹیشن (Indentation)

پائیتھن پروگراموں کے بلاکس کی وضاحت کے لئے وہائٹ اسپیس جیسے spaces اور tabs کا استعمال کرتے ہیں جبکہ دیگر زبانیں جیسے C، C++، java منحنی خطوط والے قوسین { } کو کوڈ کے بلاک، فنکشن یا لوپ کے باڈی اور سلیکشن کمانڈ کی نشاندہی کرتے ہیں۔ انڈینٹیشن میں خالی جگہوں (spaces and tabs) کی تعداد طے نہیں ہے، لیکن بلاک کے اندر تمام بیانات کو ایک ہی مقدار میں خالی جگہوں کے ساتھ پوشیدہ کیا جانا چاہئے۔

5.7 ٹوکن (Tokens)

پائیتھن ہر منطقی لکیر (logical line) کو ٹوکن کے نام سے جانا جاتا ہے۔ ابتدائی لغوی اجزاء (elementary lexical components) کی ترتیب میں توڑ دیتا ہے۔ عام ٹوکن کے اقسام یہ ہیں۔

1) شناخت گر (Identifiers)
2) اہم الفاظ (Keywords)
3) آپریٹرز (Operators)
4) حد بندی کرنے والے (Delimiters)
5) (Literals)

ٹوکن، شناخت دہندگان یا مطلوبہ الفاظ کے مابین وائٹ اسپیس علیحدگی ضروری ہے۔

5.7.1 شناخت گر (Identifiers)

شناخت گر ایک نام ہے جو متغیر، فنکشن، کلاس، ماڈیول یا شے کی شناخت کے لئے استعمال ہوتا ہے۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:43 PM

- شناخت کنندہ کو حرف تہجی (A...Z یا a...z) یا انڈر سکور (_) سے شروع کرنا ہوگا۔
- شناخت کرنے والوں میں ہندسے شامل ہو سکتے ہیں۔ (0 ... 9)
- پائیتھن شناخت کرنے والے حروف جلی اور حروف خفی حساس ہوتے ہیں، یعنی بڑے اور چھوٹے حروف الگ الگ ہوتے ہیں۔
- شناخت کرنے والوں کو ایک پائیتھن کے اہم الفاظ (python Keyword) نہیں ہونا چاہئے۔
- پائیتھن شناخت کنندگان میں اوقافی کردار جیسے ٪ ، $ ، @ وغیرہ کی اجازت نہیں دیتا ہے۔

درست شناخت گر کی مثال (Example of valid identifiers)

sum , total_marks, regno, num1

غلط شناخت کاروں کی مثال (Example of invalid identifiers)

12Name, names, total-mark, continue

5.7.2 کلیدی الفاظ (اہم الفاظ) (Keywords)

کلیدی الفاظ پروگرام کے ڈھانچے کو پہچاننے کے لئے پائیتھن مترجم (interpreter) کے ذریعہ استعمال کردہ خصوصی الفاظ ہیں۔ چونکہ یہ الفاظ مترجم کے لئے مخصوص معنی رکھتے ہیں، لہذا یہ کسی اور مقصد کے لئے استعمال نہیں ہو سکتے ہیں۔

جدول 5.1 پائیتھن کے اہم الفاظ

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	
break	except	In	raise	

5.7.3 آپریٹرز (operators)

کمپیوٹر پروگرامنگ زبانوں میں آپریٹرز خصوصی علامت ہیں جو کمپیوٹیشن، مشروط مماثلت وغیرہ کی نمائندگی کرتے ہیں۔ استعمال شدہ آپریٹر کی قیمت کو اوپریںڈز (Operands) کہتے ہیں۔ آپریٹرز کو ریاضی (Arithmetic)، متعلقہ (relational)، منطقی (logical)، تفویض (assignment) وغیرہ کے طور پر درجہ بندی کیا جاتا ہے، جب آپریٹر کے ساتھ استعمال ہونے پر قیمت اور متغیرات کو اوپریںڈز (operands) کے نام سے جانا جاتا ہے۔

(i) ریاضی کے آپریٹرز (Arithmetic operators)

ایک ریاضی کا آپریٹر ایک حسابی آپریٹر ہے، جو دو operands کرتا ہے اور ان پر حساب کتاب کرتا ہے۔ وہ سادہ ریاضی کے لئے استعمال ہوتے ہیں۔ زیادہ تر کمپیوٹرز زبانوں میں ایسے آپریٹرز کا ایک مجموعہ ہوتا ہے، جسے مختلف اقسام کے ترتیب وار حساب کتاب کرنے کیلئے مساوات کے اندر استعمال کیا جاتا ہے۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:43 PM

پائیتھن مندرجہ ذیل ریاضی آپریٹرز کی حمایت کرتے ہیں۔

آپریٹر - آپریشن **Operator - Operation**	مثالیں **Examples**	نتیجہ **Result**
Assume a=100 and b=10. Evaluate the following expressions		
+ (Addition)	>>> a + b	110
- (Subtraction)	>>>a – b	90
* (Multiplication)	>>> a*b	1000
/ (Divisioin)	>>> a / b	10.0
% (Modulus)	>>> a % 30	10
** (Exponent)	>>> a ** 2	10000
// (Floor Division)	>>> a//30 (Integer Division)	3

Program 5.1 To test Arithmetic Operators: پروگرام 5.1 حسابی اعمال کی جانچ کے لئے

```
#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: اوٹ پٹ

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

(ii) متعلقہ یا تقابلی آپریٹرز (Relational or comparative operators)

ایک متعلقہ آپریٹر (relational operator) کو موازناتی آپریٹر (Comparative operator) بھی کہا جاتا ہے جو دو operands کے مابین تعلقات کی جانچ کرتا ہے۔ اگر تعلق true ہے تو، یہ true کا نتیجہ دیتا ہے۔ بصورت دیگر یہ False نتیجہ دیتا ہے۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:43 PM

پائیتھن مندرجہ ذیل متعلقہ آپریٹرز کی حمایت کرتا ہے۔

Operator - Operation آپریٹر- آپریشن	Examples مثالیں	Result نتیجہ
Assume the value of a=100 and b=35. Evaluate the following expressions.		
== (is Equal)	>>> a==b	False
> (Greater than)	>>> a > b	True
< (Less than)	>>> a < b	False
>= (Greater than or Equal to)	>>> a >= b	True
<= (Less than or Equal to)	>>> a <= b	False
!= (Not equal to)	>>> a != b	True

Coding 5.2 To test Relational Operators: کوڈنگ 5.2 متعلقہ آپریٹرز کی جانچ کے لئے

```
#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
```

Output: اوٹ پٹ

```
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
```

(iii) منطقی آپریٹرز (Logical operators)

پائیتھن میں، منطقی آپریٹرز، دیئے گئے متعلقہ اظہار پر منطقی کارروائیوں کے لئے استعمال ہوتے ہیں۔ تین منطقی آپریٹر ہیں۔ وہ and، or اور not ہیں۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:44 PM

Operator آپریٹر	Example مثال	Result نتیجہ
Assume a = 97 and b = 35, Evaluate the following Logical expressions		
or	>>> a>b or a==b	True
and	>>> a>b and a==b	False
not	>>> not a>b	False i.e. Not True

پروگرام 5.3 منطقی آپریٹر کی جانچ کے لئے (To test Logical operators)

Example – Code مثال - کوڈ

```
#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
```

Example - Result مثال - نتیجہ

```
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) تفویض آپریٹرز (Assignment operators)

پائیتھن میں = متغیر میں قیمتوں کو تفویض کرنے کے لئے ایک سادہ تفویضی آپریٹر ہے۔ a کے لئے 5 کی قیمت اور b کے لئے 10 کی قیمت تفویض کرنے کے لئے a = 5 اور b = 10 کرتے ہیں۔ ان دو تفویضات کے بیانات کو a,b=5,10 کی طرح دیا جاتا ہے، جو متغیرات a اور b کے دائیں جانب 5 اور 10 کو بالترتیب تفویض کرتا ہے۔ پیتھان میں کئی مرکب آپریٹر ہیں، جیسے

+= ، -= ، *= ، /= ، %= ، **= اور //= بھی دستیاب ہیں۔

Operator آپریٹر	Description وضاحت	Example مثال
Assume x=10		
=	Assigns right side operands to left variable	>>> x=10 >>> b="Computer"
+=	Added and assign back the result to left operand i.e. x=30	>>> x+=20 # x=x+20
-=	Subtracted and assign back the result to left operand i.e. x=25	>>> x-=5 # x=x-5
*=	Multiplied and assign back the result to left operand i.e. x=125	>>> x*=5 # x=x*5
/=	Divided and assign back the result to left operand i.e. x=62.5	>>> x/=2 # x=x/2

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:44 PM

%=	Taken modulus(Remainder) using two operands and assign the result to left operand i.e. x=2.5	>>> x%=3 # x=x%3
**=	Performed exponential (power) calculation on operators and assign value to the left operand i.e. x=6.25	>>> x**=2 # x=x**2
//=	Performed floor division on operators and assign value to the left operand i.e. x=2.0	>>> x//=3

Program 5.4 To test Assignment Operators: پروگرام 5.4 تفویضی آپریٹر کی جانچ

Program Coding پروگرام کوڈنگ

```
#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
```

Output اوٹ پٹ

```
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
```

(v) مشروط آپریٹر (Conditional operator)

ٹرنری آپریٹر کو مشروط آپریٹر کے نام سے بھی جانا جاتا ہے، جو کسی حالت کے درست یا غلط ہونے کی بنا پر جانچ پڑتال کرتا ہے۔ یہ صرف ایک لائن میں کسی بھی حالت کی جانچ کرنے کی اجازت دیتا ہے، تاکہ کثیر لائن (multiline) کی جگہ if-else کوڈ کو مربوط کرتا ہے۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:44 PM

مشروط آپریٹر کے لئے syntax یہ ہے۔

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

مثال :

```
min= 50 if 49<50 else 70 # min = 50
min= 50 if 49>50 else 70 # min = 70
```

پروگرام 5.5 مشروط آپریٹر (ٹرنری) کی جانچ (To test conditional (ternary) operator)

```
# 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
```

Output: آؤٹ پٹ

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

5.7.4 حد بندی کرنے والے (Delimiters)

پائیتھن علامتوں اور علامت کے امتزاج کو اظہار خیالات، فہرستوں، لغات اور ڈوروں میں ڈلیمیٹر کے طور پر استعمال کرتے ہیں۔ مندرجہ ذیل حد بندی کرنے والے ہیں۔

(	)	[	]	{	}
,	:	.	‘	=	;
+=	-=	*=	/=	//=	%=
&=	|=	^=	>>=	<<=	**=

5.7.5 لٹرلس (Literals)

لٹرلس ایک خام ڈیٹا ہے جو متغیر یا مستقل کو دیا جاتا ہے۔ پائیتھن میں، مختلف قسم کے لٹرلس موجود ہیں۔

1۔ عددی
2۔ سٹرنگ
3۔ بولین

(i) عددی لٹرلس (Numeric literals)

عددی لٹرلس ہندسوں پر مشتمل ہوتے ہیں اور نا قابل تغیر (بدلاؤ) ہوتے ہیں۔ عددی لٹرلس 3 مختلف عددی اقسام کے سالم اعداد، فلوٹ اور کامپلکس (پیچیدہ) سے تعلق رکھتے ہیں۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:44 PM

Program 5.6 : To demonstrate Numeric literals پروگرام 5.6 عددی لٹرلس کی وضاحت کرنا

```
# 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
```

Output: اوٹ پٹ

```
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 9 x = 1.0
```

(ii) سٹرنگ لٹرلس (String Literals)

پائیتھن میں سٹرنگ لٹریل حروف کی ترتیب ہے، جو ' quotes سے گھیرے ہوئے ہیں۔ پائیتھن ایک سٹرنگ کے لئے سنگل' ، ڈبل" اور ٹرپل ''' quotes کی حمایت کرتا ہے۔ ایک حرفی لٹرل ایک اکہرا حرف ہے جو اکہرے یا دوہرے quote سے گھرا ہوا ہے، جس کے چاروں طرف واحد یا ڈبل قیمتیں ہیں۔ ٹرپل کوٹ ''' ''' والی قدر کثیر خطی سٹرنگ لٹرل دینے کے لئے استعمال ہوتی ہے۔

Program 5.7 To test String Literals پروگرام 5.7 سٹرنگ لٹرل کی جانچ کے لئے

```
# 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
```

Output: اوٹ پٹ

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


12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:44 PM

(iii) بولین لٹریلز (Boolean literals)

ایک بولین لٹرل میں دو قیمتیں ہوسکتی ہیں۔ : True یا False

Program 5.8 To test Boolean Literals: پروگرام 5.8 بولین لٹرل کی جانچ کے لئے

```
# 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
```

Output:

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

(iv) Escape کا سلسلہ (Escape sequence)

پائیتھن کی اسٹرنگس میں back slash "\" خاص حرف(Character) ہے، جسے escape کردار بھی کہا جاتا ہے۔ یہ مخصوص خالی جگہوں کی نمائندگی کرنے میں استعمال ہوتا ہے: "\t" ایک ٹیب ہے، "\n" ایک نئی لائن ہے، اور "\r" خط کے آخر کو واپس کرنا (Carriage return) ہے۔ مثال کے طور پر پیغام "It's raining" پرنٹ کرنے کے لئے، پائیتھن کا کمانڈ اس طرح سے ہے۔

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

پائیتھن مندرجہ ذیل escape characters کی حمایت کرتا ہے۔

Escape sequence character	Description وضاحت	Example مثال	Output اوٹ پٹ
\\	Backslash	>>> print("\\test")	\test
\’	Single-quote	>>> print("Doesn\'t")	Doesn't
\”	Double-quote	>>> print("\"Python\"")	"Python"
\n	New line	print("Python","\n","Lang..")	Python Lang..
\t	Tab	print("Python","\t","Lang..")	Python Lang..

5.8 پائیتھن ڈیٹا کی اقسام (Python Data types)

پائیتھن میں ڈیٹا کی تمام قیمتیں آبجیکٹ یا شئے ہیں اور ہر شے یا قیمت کی ایک قسم ہوتی ہے۔ پائیتھن میں بلٹ ان یا بنیادی ڈیٹا کی اقسام ہیں۔ جیسے number، String، Boolean، tuples، lists اور dictionaries۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:45 PM

5.8. عددی ڈیٹا کی قسم (Number data type)

پائیتھن میں بلٹ ان عددی اشیاء (built-in number objects)، سالم اعداد، فلوٹنگ پوائنٹ نمبر (Floating point number) اور اور کامپلکس نمبروں (complex numbers) کی حمایت کرتا ہے۔

سالم ڈیٹا یا تو عشری ہندسہ، آکٹل یا ہکساڈیسمل (Octal or hexadecimal) ہوسکتے ہیں۔ آکٹل سالم اعداد ہندسہ 0 (صفر) کے ساتھ 'o' آکٹل ہندسہ کے لئے اور 0X (صفر اور X یا x) کو ہکساڈیسمل ہندسہ کے لئے استعمال کرتے ہیں۔ L کو طویل سالمی عدد کے لئے استعمال کرتے ہیں۔

مثال :

```
102, 4567, 567          # Decimal integers
O102, o876, O432        # Octal integers
OX102, oX876, OX432     # Hexadecimal integers
34L, 523L               # Long decimal integers
```

Floating point data سے مراد عشری ہندسے جس میں اعشاریہ بھی ہوں، ان اعداد کی نمائندگی کرتا ہے۔ قوت نمائی ڈیٹا میں عشری ہندسوں کا حصہ، عشری نقطہ، قوت نما کا حصہ اور اس کے ساتھ ایک یا زیادہ ہندسے ہوتے ہیں۔

Example :

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

دو floating point values سے کامپلکس اعداد بنے ہوئے ہیں، ایک حقیقی حصہ (real part) اور دوسرا مجازی حصہ (imaginart part)۔

5.8.2 بولین ڈیٹا کی قسم (Boolean data type)

بولین کے ڈیٹا میں دو قیمتیں ہوسکتی ہیں: True یا False

مثال :

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

5.8.3 اسٹرنگ ڈیٹا کی قسم (String data type)

اسٹرنگ ڈیٹا کو اکہرے quote ' یا ڈبل quote " یا ٹرپل quote ''' سے منسلک کیا جاسکتا ہے۔

مثال :

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


12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:45 PM

یادرکھنے کے نکات: Points to remember:

- پائیتھن ایک عمومی مقصد کی پروگرامنگ زبان ہے جو گائڈووان روسم Guido Van Rossum نے تشکیل دیا ہے۔
- پائیتھن شیل دو طریقوں سے استعمال کیا جاسکتا ہے، جیسے انٹرایکٹو موڈ اور اسکرپٹ موڈ۔
- پائیتھن پروگرام بلاکس کی وضاحت کے لئے وائٹ اسپیس (خالی جگہوں اور ٹیب) کا استعمال کرتے ہیں۔
- ٹوکن، شناخت دہندگان یا مطلوبہ الفاظ کے مابین وائٹ اسپیس کی علیحدگی ضروری ہے۔
- مطلوبہ کام کو پورا کرنے کے لئے کسی پروگرام کو اختتامی صارف سے بات چیت کرنے کی ضرورت ہوتی ہے، یہ ان پٹ، آؤٹ پٹ سہولت کا استعمال کرتے ہوئے کیا جاتا ہے۔
- پائیتھن ہر منطقی خط کو ٹوکن کے نام سے جانتا ہے۔ ابتدائی لغوی اجزاء (lexical components) کی ترتیب میں توڑ دیتا ہے۔
- اہم الفاظ (key words) خاص الفاظ ہیں جو پروگرام کی ساخت کو پہچاننے کے لئے پائیتھن مترجم کے ذریعہ استعمال ہوتے ہیں۔

محاسبہ Evaluation

حصہ - 1

صحیح جواب کا انتخاب کریں۔ (1 نمبر)

1۔ پائیتھن کس نے تیار کیا؟

A) Ritchie B) Guido Van Rossum
C) Bill Gates D) Sundar Pitchai

2۔ پائیتھن کا prompt یہ اشارہ کرتا ہے کہ مترجم (interpreter) ہدایت قبول کرنے کے لئے تیار ہے۔

A) >>> B) <<< C) # D) <<

3۔ نیا پائیتھن پروگرام بنانے کے لئے مندرجہ ذیل میں سے کون سا شارٹ کٹ استعمال ہوتا ہے؟

A) Ctrl + C B) Ctrl + F
C) Ctrl + B D) Ctrl + N

4۔ پائیتھن پروگرام میں کمنٹس (comments) دینے کے لئے درج ذیل میں سے کون سی علامت (Character) استعمال کی جاتی ہے؟

A) # B) & C) @ D) $

5۔ یہ علامت ایک لائن پر ایک سے زیادہ آئٹم پرنٹ کرنے کے لئے استعمال ہوتی ہے۔

A) سیمی کولن (;) B) ڈولر ($) C) کوما (،) D) کولن (:)

6۔ مندرجہ ذیل میں سے کون سا ٹوکن نہیں ہے؟

A) ترجمان (interpreters) B) شناخت گر (Identifiers)
C) اہم الفاظ (Key words) D) آپریٹرز (Operators)

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:45 PM

7۔ مندرجہ ذیل میں سے کونسا پائیتھن میں کلیدی لفظ (اہم لفظ) (key word) نہیں ہے؟

A) break B) while C) continue D) operators

8۔ کس آپریٹر کو موازناتی آپریٹر (comparative operator) بھی کہا جاتا ہے؟

A) ریاضی (Arithmetic) B) متعلقہ (relational)

C) منطقی (logical) D) تفویض (Assignment)

9۔ مندرجہ ذیل میں سے کون سا منطقی آپریٹر (logical operator) نہیں ہے؟

A) and B) or C) not D) Assignment

10۔ کون سے آپریٹر کو مشروط آپریٹر (conditional operator) کہا جاتا ہے؟

A) ٹرنری Ternary B) متعلقہ Relational

C) منطقی Logical D) تفویضی Assignment

حصہ - II

مندرجہ ذیل سوالات کے جوابات دیں: (2 نمبرات)

1۔ پائیتھن پروگرام کو جانچنے کے لئے کون سے مختلف طریقے (modes) استعمال کیے جاسکتے ہیں؟

2۔ ٹوکن (Tokens) پر مختصر نوٹ لکھیں۔

3۔ وہ مختلف آپریٹرز کیا ہیں، جن کو پائیتھن میں استعمال کیا جاسکتا ہے؟

4۔ لٹرل (literal) کیا ہے؟ لٹرلس کے قسموں کی وضاحت کریں۔

5۔ قوت نما کے ڈیٹا سے متعلق مختصر نوٹ لکھیں؟

حصہ - III

مندرجہ ذیل سوالات کے جوابات دیں: (3 نمبرات)

1۔ مثالوں کے ساتھ ریاضی کے آپریٹر (Arithmetic operator) پر مختصر نوٹ لکھیں۔

2۔ تفویضی آپریٹرز (Assignment operators) کیا، جن کو پائیتھن میں استعمال کیا جاتا ہے ؟

3۔ مثال کے ساتھ ٹرنری آپریٹر (ternary operator) کی وضاحت کریں۔

4۔ Escape سلسلے پر مثالوں کے ساتھ مختصر نوٹ لکھیں۔

5۔ سٹرنگ لٹرل (String literal) کیا ہیں؟ وضاحت کریں۔

حصہ - IV

مندرجہ ذیل سوالات کے جوابات دیں: (5 نمبرات)

1۔ اسکرپٹ موڈ پروگرامنگ (script mode programming) کے طریقہ کار کو تفصیل سے بیان کریں۔

2۔ مثالوں کے ساتھ input() اور print() فنکشنز کو بیان کریں۔

3۔ پائیتھن میں ٹوکن (Tokens) سے متعلق تفصیل سے گفتگو کریں۔

12 CS UNIT 5 CDR Set
Sunday, April 16, 2023 9:35:46 PM

CHAPTER 6 باب 6

CONTROL STRUCTURES کنٹرول ساختیں

اکائی II



سیکھنے کے مقاصد

اس سبق کا مطالعہ کرنے کے بعد، طلبا اس قابل ہوسکیں گے:

- پائیتھن زبان میں کنٹرول کے مختلف بہاؤ کے بارے میں معلومات حاصل کرنا۔
- syntax کے ذریعے یہ سیکھیں گے کہ پروگرام کے بہاؤ کی کارکردگی کو بڑھانے میں کس طرح مشروط تعمیر کا استعمال ہوتی ہے۔
- پروگرام کے حصے کو مخصوص تعداد یا وقت تک یا یہاں تک کہ مطلوبہ حالت کو پہنچ جائے، اس کو دوہرانے کے لئے کوڈ تیار کرنا۔

6.1 تعارف Introduction

پروگراموں میں بیانات کا مجموعہ شامل ہوسکتا ہے۔ یہ بیانات قابل عمل طبقات ہیں جو نتیجہ برآمد کرتے ہیں۔ عام طور پر، بیانات کو ایک دوسرے کے بعد مسلسل عمل درآمد کیا جاتا ہے۔ یعنی بیانات کو ایک کے بعد دیگر عمل درآمد کیا جاتا ہے۔ ہماری حقیقی زندگی کے پروگرامنگ میں ایسے حالات پیدا ہوسکتے ہیں جہاں ہمیں کسی طبقہ یا بیانات کے سیٹ کو چھوڑنے کی ضرورت ہوتی ہے اور کسی دوسرے طبقہ کو کسی حالت کی جانچ کی بنیاد پر انجام دینے کی ضرورت ہوتی ہے۔ اسے متبادل (alternative) یا شاخیں (branching) کہا جاتا ہے۔ نیز، ہمیں بیانات کا ایک سیٹ متعدد بار دوہرا کر عمل میں لانے کی ضرورت پڑسکتی ہے، جسے تکرار (iteration) یا لوپنگ (looping) کہتے ہیں۔ اس باب میں ہم پائیتھن میں مختلف کنٹرول ساختیں، ان کی ترکیب پر توجہ مرکوز کریں گے اور ان کو استعمال کرتے ہوئے پروگراموں کو تیار کرنے کا طریقہ سیکھیں گے۔

6.2 کنٹرول ساختیں (Control structures)

پروگرام کا ایک بیان جو پروگرام کے ایک حصے سے دوسرے حصے میں کنٹرول کو jump کرتا ہے، اسے کنٹرول ڈھانچہ (Control structure) یا کنٹرول بیان (control statement) کہا جاتا ہے۔ جیسا کہ آپ نے پہلے ہی C++ میں سیکھا ہے، یہ کنٹرول بیانات عمل کی حالت کے لحاظ سے عمل یا پروگرام کے کنٹرول بہاؤ کو تبدیل کرنے کے لئے استعمال کیے جانے والے مرکب بیانات ہیں۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:36 PM

کنٹرول کی تین اہم ساختیں ہیں۔



3.2.1 سلسلہ وار بیان (Sequential statement)

ایک سلسلہ وار بیان، بیانات کے تسلسل پر مشتمل ہوتا ہے، جسے ایک کے بعد ایک عمل درآمد کیا جاتا ہے۔ آپ کا نام، پتہ اور فون نمبر پرنٹ کرنے کے لئے ایک کوڈ ترتیب وار بیان کی مثال ہے۔

مثال 6.1

```
# Program to print your name and address - example for sequential statement
print ("Hello! This is Shyam")
print ("43, Second Lane, North Car Street, TN")
```

Output

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

6.2.2 متبادل یا شاخوں کا بیان (Alternative or Branching statement)

روزمرہ کی زندگی میں ہمیں مختلف فیصلے کرنے اور اپنے مقصد کے حصول کے لئے ایک متبادل راستہ کے انتخاب کی ضرورت ہے۔ جب ہماری حسب معمول سڑک میں کوئی رکاوٹ آجاتی ہے تو ہم اپنی منزل تک پہنچنے کے لئے ایک متبادل راستہ اختیار کرتے ہیں۔ اس قسم کی فیصلہ سازی وہ ہے، جو ہم متبادل یا برانچنگ اسٹیٹمنٹ کے ذریعہ سیکھنا چاہتے ہیں۔(یہ جانچنا کہ دیا گیا عدد مثبت ہے یا منفی، طاق ہے یا جفت، یہ سب متبادل یا برانچنگ اسٹیٹمنٹ کا استعمال کرتے ہوئے کیا جاسکتا ہے۔

پائیتھن مندرجہ ذیل اقسام کے متبادل یا برانچنگ والے بیانات فراہم کرتا ہے۔

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

(i) simple if بیان (Simple if statement)

فیصلہ سازی کے تمام بیانات میں سب سے سادہ ترین بیان simple if ہے۔ یہ متعلقہ (relational) یا منطقی اظہار کی صورت میں ہونی چاہئے۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:37 PM

Syntax:

if <condition>:

statements-block1

مذکورہ syntax میں if شرط صحیح ہے تو-بلاک 1 پر عمل درآمد کیا جائے گا۔

مثال 6.2

```
# Program to check the age and print whether eligible for voting
x=int (input("Enter your age :"))
if x>=18:
        print ("You are eligible for voting")
```

Output 1:

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

Output 2:

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

جیسا کہ آپ دیکھ سکتے ہیں دوسرے عمل درآمد(execution) میں کوئی آؤٹ پٹ نہیں چھاپا گیا ہے،صرف پائیتھن کاprompt ہی دکھائے گا کیونکہ جب یہ حالت ناکام (fail) ہوجاتی ہے تو پروگرام متبادل عمل کی جانچ نہیں کرتا ہے۔

(ii) if ..else بیان

if..else بیان، true block کے ساتھ ساتھ false block کو چیک کرنے کے لیے کنٹرول فراہم کرتا ہے۔ذیل میں 'if..else' بیان کا syntax ہے۔

Syntax:

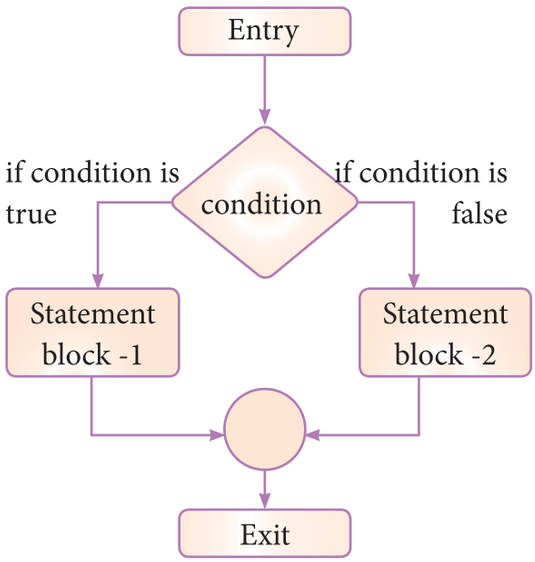
if <condition>:

statements-block 1

else:

statements-block 2

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:37 PM



خاکہ 6.1 if..else بیان کو عمل درآمد کرنا

چنانچہ if..else بیان، دو امکانات فراہم کرتا ہے اور حالت یہ طے کرتی ہے کہ کون سا BLOCK عمل میں لایا جائے۔

مثال 6.3 : # یہ جانچ کرنے کے لئے پروگرام کہ قبول کیا گیا عدد طاق ہے یا جفت
Example 6.3: #Program to check if the accepted number odd or even

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

Output 1:

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

Output 2:

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

اوپر کے پروگرام کو دوبارہ لکھنے کا ایک متبادل طریقہ پائیتھن میں بھی دستیاب ہے۔ مکمل if..else اس طرح بھی لکھ سکتے ہیں:

Syntax:

variable = variable1 if condition else variable 2

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:37 PM

نوٹ

اگراس کی وضاحت کی گئی ہے، if کی جانچ،اگر یہ درست ہےتو، متغیر1 کی قیمت اسائنمنٹ (تفویض) کے بائیں جانب متغیر میں محفوظ کی جاتی ہے، بصورت دیگر متغیر 2 کوقدر کی حیثیت سے لیا جاتا ہے۔

مثال 6.4: # یہ جانچ کرنے کے لئے پروگرام کہ قبول شدہ عدد طاق ہے یا جفت (if...else کے متبادل طریقہ کواستعمال کرتے ہوئے)

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

Output 1:

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

Output 2:

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

(iii) Nested if..elif...else بیان :

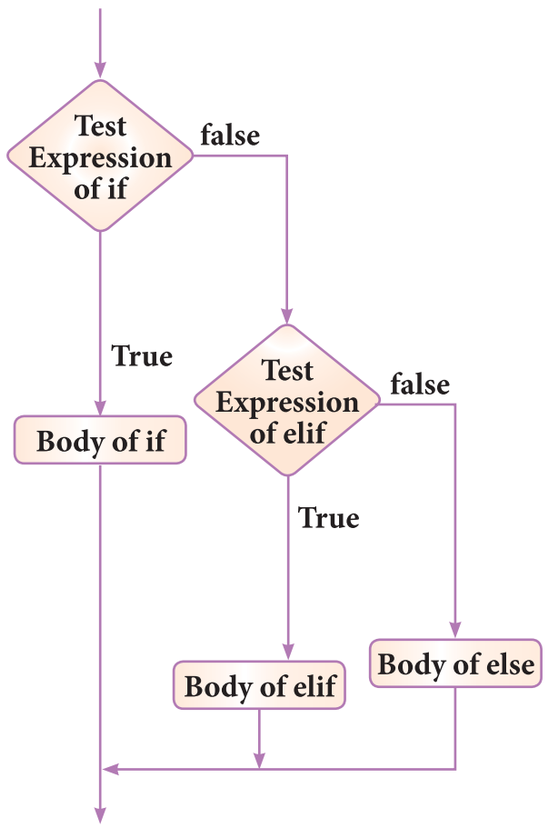
جب ہمیں if بیانوں کی ایک زنجیر بنانے کی ضرورت پڑے گی تو 'else' کی بجائے clause 'elif' استعمال کی جائے گی۔

Syntax:

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

اگر مذکورہ بالا if..elif..else کے syntax میں، حالت-1 سے جانچ کیا جاتا ہے، اگر یہ true ہوتو statement-block 1 عمل درآمد ہوگا، بصورت دیگر کنٹرول حالت 2 کی جانچ کرتا ہے، اگر یہ بیانات true ہیں-block2 پر عمل درآمد کیا جاتا ہے اور یہاں تک کہ اگر یہ بیان بھی fail ہوجاتا ہےتو statement-block n ، جس کو else حصہ میں بتایا گیا ہے، اس پر عمل درآمد کیا جاتا ہے۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:37 PM



خاکہ 6.2 if..elif..else بیانات کوعمل درآمد کرنا

'elif' clause کے ساتھ if..else-if..else بیانات شامل ہوکر ایک if..elif...else بنتا ہے، جس کا مخفف 'else if' لکھا جاتا ہے۔ ایک 'if' بیان میں 'elif' clause کی کوئی حد نہیں ہے، جسے استعمال کیا جاسکتا ہے، مگر آخر میں ایک 'else' کو رکھنا چاہئے۔

نوٹ

if..elif..else بیان، nested if بیان کے مشابہ ہے، جس کو تم نے C++ میں سیکھا تھا۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:37 PM

مثال 6.5 : # nested if statement کے استعمال کی وضاحت کے لئے پروگرام

Example 6.5: #Program to illustrate the use of nested if statement

Average	***Grade***
>=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")
```

Output 1:

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

Output 2 :

```
Enter mark in first subject : 67
```

نوٹ :

ہماری مثال میں موجود if -statement کے کوڈ کے دونوں بلاکوں کو چار خالی جگہوں سے indent کیا گیا ہے، جس کو پیتھان میں ایک مثالی مقدار مانی جاتی ہے۔ دیگر اکثر پروگرامنگ کی زبانوں میں، indentation کو صرف کوڈ کو اچھا دکھائی دینے کے لئے استعمال ہوتا ہے۔ مگر پیتھان میں اس سے یہ ظاہر کیا جاتا ہے کہ یہ بیان کس کوڈ کے بلاک سے تعلق رکھتا ہے۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:37 PM

مثال 6.5a : # ایک بیان میں 'in' اور 'not in' کے استعمال کی وضاحت کے لئے پروگرام
Example 6.5a: #Program to illustrate the use of 'in' and 'not in' in if statement

```
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')
```

Output 1:

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

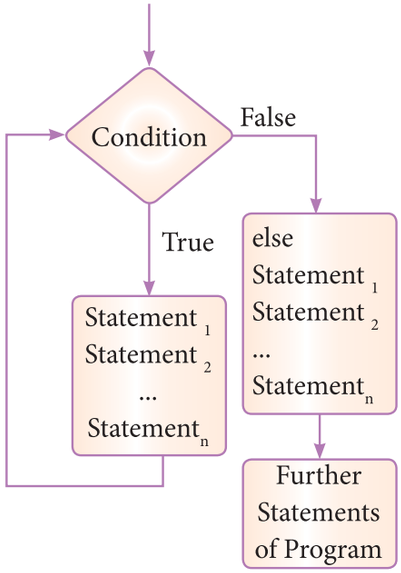
Output 2:

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

6.2.3 دوہرانے والے (تکرار) یا لوپنگ تعمیرات (Iteration or Looing constructs)

تکرار یا لوپ کا استعمال ایسی صورت میں ہوتا ہے جب صارف کو کئی بار کوڈ بلاک پر عمل کرنے کی ضرورت ہوتی ہے یا جب تک کہ شرط پوری نہیں ہوجاتی۔ایک loop بیان، ایک بیان یا بیانات کے گروپ کو کئی بار انجام دینے کی اجازت دیتا ہے۔





خاکہ 6.3 لوپنگ کی تعمیر کس طرح کی جاتی ہے، اس کی وضاحت کے لئے خاکہ

پائیتھن دو قسم کے لوپنگ تعمیرات فراہم کرتا ہے:

- **while** لوپ while loop
- **for** لوپ for loop

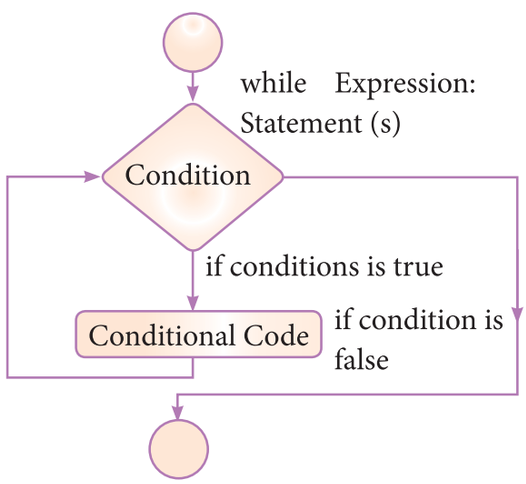
12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:38 PM

while (i) لوپ

پائیتھن میں while loop کے لئے syntax مندرجہ ذیل ہے۔

Syntax:

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



خاکہ 6.4 while loop پر عمل درآمد کرنا

while loop میں کسی بھی بولین بیان کے لئے جس کا نتیجہ True یا False ہو۔ while کا else حصہ، اس کا اختیاری حصہ ہے۔ statements block 1 کو عمل درآمد کیا جاتا ہے یہاں تک کہ شرط True ہو۔ اگر else کے حصہ کو لکھا جائے تو، اس کو اُس وقت تک عمل در آمد کیا جاتا ہے یہاں تک کہ شرط false ہو۔ یاد رکھیں کہ while loop، لوپ کی entry check کی قسم میں شامل ہے، یعنی یہ اس وقت تک ایک بار بھی عمل درآمد نہیں کیا جاسکتا، جب تک کہ شروع میں شرط False نکلے۔

مثال 6.6 while loop کے استعمال کی وضاحت کے لئے پروگرام- 10 تا 15 تک کے تمام اعداد کو پرنٹ کرنے کے لئے۔

```
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
```

Output:

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


12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:38 PM

نوٹ

کہ کنٹرول متغیر i ہو، جس کی ابتدا 10 سے کی جاتی ہے، حالت کی جانچ i <=15 کی جاتی ہے، اگر i صحیح قیمت پرنٹ ہوجاتی ہے، تو پھر کنٹرول متغیر i کو i=i+1 کی طرح اپ ڈیٹ کردیا جاتا ہے (اس کو shorthand assignment operator کو استعمال کرتے ہوئے اس طرح بھی لکھا جاسکتا ہے۔ i+=1)۔ جب i کی قیمت 16 ہوجاتی ہے تو شرط کی جانچ False اور یہ لوپ کو ختم کردیتا ہے۔

نوٹ

print کا اختتام ہوسکتا ہے، sep کو بطور parameter لیتے ہوئے، جب ہم کسی escape sequence کو لیتے ہیں، جیسے tab کے لئے '/t' ، نئے خط کے لئے '/n' اور اسی طرح یہ جاری رہتا ہے۔ پیرامیٹر sep کو بعض مخصوص علامتوں کے ساتھ جیسے کاما , سیمی کولن ; وغیرہ کو بطور جداکار کے استعمال کیا جاتا ہے۔ (یاد رکھیں کہ جس نظریہ کو تم نے پچھلے باب میں فارمیٹنگ میں پرنٹ کے لئے () ، استعمال کیا جاتا ہے)۔

while loop میں else کے حصہ کو استعمال کرنے کے لئے درج ذیل کی ایک مثال یہ ہے۔

مثال 6.7 : else کے حصہ کے ساتھ while loop کے استعمال کی وضاحت کے لئے پروگرام۔

```
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)
```

Output: 1

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

for loop (ii)

عام طور پر for loop کو definite loop کہا جاتا ہے، کیوں کہ پروگرامر یہ جانتا ہے کہ کہ ٹھیک کتنی مرتبہ یہ لوپ عمل میں لایا گیا ہے۔

Syntax:

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

ترتیب اشیاء کو متواتر دہرانے کے لئے پئی تھان میں .... for بیان کو looping statement کی طرح استعمال کیا جاتا ہے۔ یعنی اشیاء متواتر ظاہر ہوتے ہیں۔ یہاں تک کہ ایک اسٹرنگ میں تواتر ترتیب وار یا غیر ترتیب وار قیمت کا مجموعہ ہوتا ہے۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:38 PM

control variable کے مشابہ ہے۔ عام طور پر python میں ابتدا، انتہا اور بڑھتی ہوئی قیمت کے لئے for لوپ، range() فنکشن کو استعمال کرتا ہے۔ start سے stop-1 تک range() فہرست کی قیمتوں کو تیار کرتا ہے۔

range() کے لئے syntax اس طرح ہے۔

Example 6.8 (a) مثال

```
for x in "Hello World":
    print(x, end=' ')
```

Output:

```
H e l l o  W o r l d
```

Example 6.8 (b) مثال

```
for x in (1,2,3,4,5):
    print("Hello World")
```

Output:

```
Hello World
Hello World
Hello World
Hello World
Hello World
```

قابو والا متغیر دوہرانے والے ہر ایک تواتر کی ہر شئے کا معائنہ کرتا ہے، یہاں تک کہ وہ تواتر کے آخری شئے تک پہنچ جاتا ہے۔

اوپر کی مثال 6.8 (a) میں ہم نے "Hello World" نام کی ایک اسٹرنگ، تواتر کی طرح بنائی۔ ابتدا میں x کی قیمت کو اسٹرنگ کے پہلے عنصر کی طرح مانا گیا۔ (یعنی 'H')، لہٰذا لوپ کے اندر پرنٹ کا بیان مکمل ہو گیا۔ اس کے بعد قابو والے متغیر x کو اسٹرنگ کے دوسرے عنصر کی طرف تجدید کی جاتی ہے اور دوبارہ پرنٹ بیان کو مکمل کیا جاتا ہے۔ اس طرح اسٹرنگ کے آخری عنصر تک لوپ کا معائنہ کیا جاتا ہے۔

اسی طرح مثال 6.8(b) "Hello world" کی اسٹرنگ کو پانچ مرتبہ پرنٹ کرتا ہے یہاں تک کہ قابو والا متغیر x، دئے گئے تواتر کے آخری عنصر تک پہنچ جاتا ہے۔

تواتر میں قیمت کو دستی طور پر (manually) چلائے جانے کے بجائے ہم range() استعمال کر سکتے ہیں۔

range() ایک خود کار اندر سے بنا ہوا فنکشن ہے، جو دو عددی وقفوں کے درمیان کی قیمتوں کے سلسلہ کو تیار کرتا ہے۔

range() کا syntax درج ذیل ہے۔

```
range (start,stop,[step])
```

جس میں

Start : ابتدائی قیمت کو بتاتا ہے۔

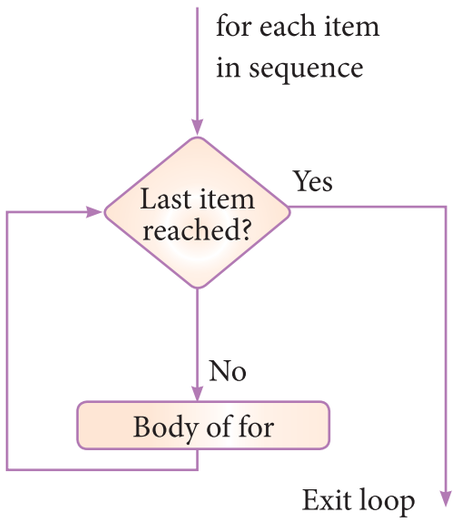
stop : انتہائی قیمت کو بتاتا ہے۔

step : بڑھتی ہوئی قیمت کو بتاتا ہے۔ یہ اختیاری حصہ ہے۔

Example 6.8: Examples for range() مثال 6.8 (c) : range() کے لئے مثالیں

```
range (1,30,1)   will start the range of values from 1 and end at 29
range (2,30,2)   will start the range of values from 2 and end at 28
range (30,3,-3)  - will start the range of values from 30 and end at 6
range (20)       will consider this value 20 as the end value(or upper limit) and starts the
                 range count from 0 to 19 (remember always range() will work till stop -1
                 value only)
```


12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:39 PM



خاکہ 6.5 for لوپ کے لئے عمل درآمد کرنا

خاکہ 6.9 : # اکہری ہندسہ والے جفت اعداد کو پرنٹ کرنے کے لئے for لوپ کی وضاحت کا پروگرام

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

Output:

```
2 4 6 8
```

for لوپ کے else حصہ کو استعمال کرنے کی وضاحت اس طرح سے ہے؛

مثال 6.10 : # else حصہ والے کے ساتھ اکہری ہندسہ والے جفت اعداد کو پرنٹ کرنے کے لئے for لوپ کی وضاحت کا پروگرام

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

Output:

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

نوٹ

پائیتھن میں، لوپ اور دوسرے کنٹرول بیانات میں indentation بہت ہی اہم ہے۔ indentation سے صرف بلاکس اور سب بلاکس تخلیق ہوتے ہیں جیسے، جس طرح C++ وغیرہ جیسی زبانوں میں { } کے ایک سیٹ کے اندر بلاکس بنائے جاتے ہیں۔

یہاں ایک اور پروگرام ہے جو 1 سے 100 تک کے اعداد کا حاصل جمع کے range() کو استعمال کرنے کی وضاحت کرتا ہے۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:39 PM

مثال 6.11 : # 1 سے 100 تک کے اعداد کا حاصل جمع محسوب کرنے کے لئے پروگرام۔

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

Output:

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

اوپر کے کوڈ میں n کی ابتدا 100 سے کی گئی ہے۔ حاصل جمع کی ابتدا 0 سے کی گئی ہے۔ for لوپ 1 سے عمل درآمد ہوتا ہے، ہر ایک تکرار کے لئے حاصل جمع کی قیمت counter variable کے ساتھ شامل ہوتی ہے اور حاصل جمع میں محفوظ ہو جاتی ہے۔ غور کیجئے کہ for لوپ 1 سے اوپری حد تک مکرر (iterate) ہوتا ہے۔ یعنی n کی قیمت 100 ہے، لہٰذا یہ لوپ قیمتوں میں تکرار (دوہرانا) صرف 1 سے 99 تک ہوتی ہے۔ یعنی اسی وجہ سے ہم بالائی حد n+1 کو مقرر کیا گیا ہے۔

نوٹ

range() کی قیمتوں کو string، list، dictionary، وغیرہ سے بھی لے سکتے ہیں، جن کو بعد کے ابواب میں دیکھیں گے۔

range() کے لئے string کے استعمال کی وضاحت کے لئے مثال

مثال 6.12 : range() کے لئے string کے استعمال کی وضاحت کے لئے پروگرام

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

Output

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

(iii) اندرونی لوپ کی ساخت (Nested loop structure)

ایک لوپ کے اندر ایک اور لوپ رکھی ہو تو اسے nested loop structure کہا جاتا ہے۔ ایک while کے اندر ایک اور while؛ for کے اندر ایک اور for، for کے اندر while اور while کے اندر for، اس طرح سے nested loops بنائے گئے ہیں۔

ایک لوپ کو درج ذیل نمونے میں پرنٹ کرنے کے لئے وضاحت درج ذیل مثال میں دی گئی ہے۔

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


12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:39 PM

مثال 6.13 : for کے اندر while لوپ کو nested loop کی طرح وضاحت کرنے کے لئے پروگرام

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

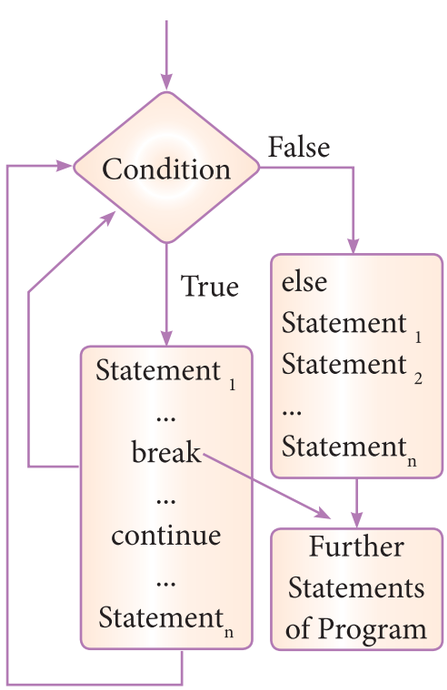
Output:

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

6.2.4 Python میں jump statement

Python میں jump statement، پروگرام کے ایک حصے سے دوسرے حصے میں غیر مشروط کنٹرول منتقل کرنے کے لئے استعمال کیا جاتا ہے۔ پائیتھن میں jump statement حاصل کرنے کے لئے تین کلیدی الفاظ ہیں۔ break، continue، pass

مندرجہ ذیل فلوچارٹ میں break اور continue کے استعمال کی مثال دی گئی ہے۔



تصویر 6.6 ایک لوپ کی ساخت میں break، continue بیانات کی ساخت

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:39 PM

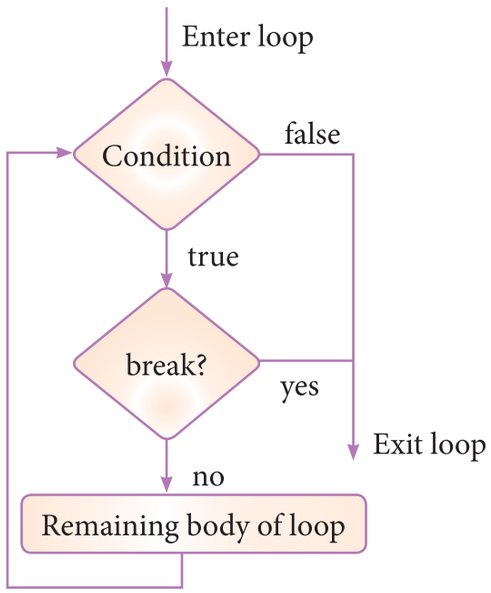
(i) break statement

break statement اس میں موجود لوپ کو ختم کرتا ہے۔ پروگرام کا کنٹرول لوپ کی باڈی کے فوراً بعد statement کی طرف جاتا ہے۔ while یا for لوپ، اس وقت تک تکرار (iterate) کرتا ہے، جب تک اس کی شرط false نہ آجائے، مگر کوئی بھی break statement کے ساتھ لوپ کے کنٹرول کا تبادلہ کرسکتا ہے (ختم کرسکتا ہے)۔ جب break statement پر عمل درآمد کیا جاتا ہے تو پروگرام کے کنٹرول کا بہاؤ لوپ سے باہر آجاتا ہے اور قطعے (segment) کے کوڈ پر عمل درآمد کرنا شروع کردیتا ہے۔

اگر break statement، نسٹڈ لوپ (ایک لوپ کے اندر ایک اور لوپ) ہو تو، سب سے اندرونی لوپ میں break ختم ہوجائے گا۔

Syntax:

break



خاکہ 6.7 break statement کا عمل

for لوپ اور while لوپ کے break statement کا عمل درج ذیل میں بتایا گیا ہے۔

```
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
```


12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:39 PM

مثال 6.14 : لوپ کے اندر break statement کو استعمال کرنے کی وضاحت کے لئے پروگرام

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

Output:

```
Jump Stat
```

اوپر کا پروگرام دئے گئے "Jump statement" میں دوہراتے ہوئے (تکرار کرتے ہوئے) (Iteration) اسٹرنگ کی طرح ہوگا۔ دئے گئے ہر string sequence کا ہر حرف لوپ بلاک کے باہر منتقل ہوجائے گا یا ختم ہوجائے گا۔ جیسا کہ output میں بتایا گیا ہے، اس کو اس وقت تک نمائش کرتے رہے گا یہاں تک کہ حرف 'e' جانچ کیا جائے گا، اس کے بعد لوپ ختم ہوجائے گا۔

ہر کسی کو ایک اہم بات نوٹ کرنی چاہئے کہ ''if ' لوپ اگر break کے حصہ سے چھوٹ جائے تو else کا حصہ عمل درآمد نہیں ہوگا''۔ اس کی وضاحت کے لئے آیئے پہلے کے پروگرام کو else کے حصے کے ساتھ بڑھائیں اور دیکھیں گے کہ اس کا output کیا ہوگا۔

مثال 6.15 : for لوپ کے اندر break statement کے استعمال کے وضاحت کے لئے پروگرام

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

Output:

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

غور کیجئے کہ break statement نے لوپ کے else حصے کو نظر انداز (skipped) کر دیا ہے اور لوپ بلاک کے بعد کنٹرول اگلی لائن کی طرف تبدیل ہوگئی ہے۔

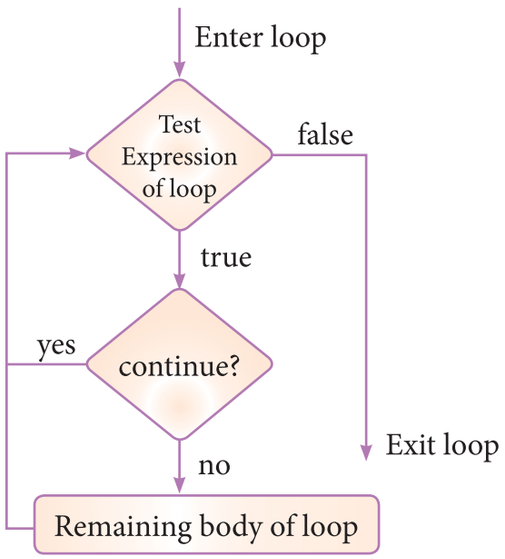
continue statement (ii)

break statement کے برعکس continue statement کو لوپ کے باقی حصہ کو skip کرنے اور اس کے بعد کے تکرار (iteration) کو شروع کرنے کے لئے استعمال ہوتا ہے۔

Syntax:

continue

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:40 PM



خاکہ 6.8 continue statement کا عمل

continue statement کو for اور while لوپ کے عمل کو درج ذیل میں بتایا گیا ہے۔

```
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
```

خاکہ 6.16 continue statement کو for لوپ کے اندر استعمال کرنے کی وضاحت کے لئے پروگرام

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

Output:

```
Jump Statement
End of the program
```


12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:40 PM

جس طرح کا پروگرام ہم نے break statement کے لئے لکھا تھا، یہ بھی اسی طرح کا پروگرام ہے،فرق صرف اتنا ہے کہ ہمیں break کی جگہ continue لکھنا ہے۔ output میں آپ دیکھو گے کہ حرف e کے سوا دیگر تمام حروف print ہو جائیں گے۔

(iii) pass statement

python پروگرامنگ میں pass statement، ایک null statement ہے۔ pass statement کو جب ایک interpreter کے ذریعہ چلایا جاتا ہے تو یہ مکمل طور پر نظر انداز ہو جاتا ہے۔ جب pass کو چلایا جاتا ہے تو کچھ بھی نہیں ہوگا، اس کا نتیجہ no operation پر ختم ہوگا۔

جب آپ کو کسی بلاک کے اندر عمل درآمد کرنا ہو اور جب آپ کسی statement یا command کو اس کے اندر نہیں لانا چاہتے ہیں تو if یا within لوپ کی تعمیر کے اندر pass statement استعمال کیا جا سکتا ہے۔

Syntax:

pass

مثال 6.17 pass statement کے استعمال کی وضاحت کے لئے پروگرام

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

Output:

```
Enter any number :3
```

غیر صفر کی قیمت قبول کی جاتی ہے۔
جب مندرجہ بالا کوڈ پر عمل درآمد ہوتا ہے اگر input value، 0 (صفر) ہے تو پھر کوئی عمل نہیں کیا جائے گا، دیگر تمام ان پٹ ویلیوز کے لئے آؤٹ پٹ مندرجہ ذیل ہوگا۔

نوٹ

pass statement عام طور پر placeholder کے طور پر استعمال ہوتا ہے۔ جب ہمارے پاس لوپ یا فنکشن ہوتا ہے جسے مستقبل میں لاگو کیا جانا ہے اور اب نہیں، تو ہم اس طرح کے function یا loop جس کے جسمانی قطعے (body segments) کے اندر تیار نہیں کر سکتے، کیوں کہ interpreter ایک error ظاہر کرے گا۔ اس سے بچنے کے لئے ہم pass statement کو استعمال کر کے ایک ایسا جسم (body) تیار کر سکتے ہیں، جو کچھ نہیں کرتا ہے۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:40 PM

مثال6.18 : ایک لوپ کے pass statement کے استعمال کی وضاحت کے لئے پروگرام۔

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

Output:

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

یاد رکھنے کے لئے نکات: Points to remember:

- پروگراموں میں بیانات کا مجموعہ شامل ہوسکتا ہے۔ یہ بیانات قابل عمل طبقات ہیں جو نتیجہ برآمد کرتے ہیں۔ عام طور پر، بیانات کو ایک دوسرے کے بعد مسلسل عمل درآمد کیا جاتا ہے۔
- پروگرام کا ایک بیان جو پروگرام کے ایک حصے سے دوسرے حصے میں کنٹرول کو jump کرتا ہے، اسے کنٹرول ڈھانچہ (Control structure) یا کنٹرول بیان (control statement) کہا جاتا ہے۔
- کنٹرول کے بہاؤ کی تین اقسام ہیں۔
  - تسلسل (sequencing)
  - شاخ یا متبادل (branching or alternative)
  - تکرار (Iteration)
- python میں، برانچنگ مختلف قسم کے if ڈھانچے کا استعمال کرتے ہوئے کی جاتی ہے۔
- python میں پروگرامنگ میں indentation اہم کردار ادا کرتا ہے، یہ انڈینٹیشن ہیں، جو بیانات کی گروہ بندی کرتے ہیں، اس لئے { } استعمال کرنے کی ضرورت نہیں ہے۔
- پائیتھن ترجمان (Interpreter) تمام تر errors کی نشان دہی کرے گا۔
- ایک رن ٹائم پر ان پٹ کو قبول کرنے کے لئے، پائیتھن کے پچھلے ورژنز raw_input() کی حمایت کرتے تھے، جب کہ تازہ ترین ورژن input() کی حمایت کرتے ہیں۔
- output کو استعمال کرنے والے کے اختیار میں رکھنے کے لئے escape تسلسل کو استعمال کرتے ہوئے print() کی حمایت کرتا ہے۔
- ایک لوپ میں قیمتوں کی حد کو فراہم کرنے کے لئے range() استعمال ہوتا ہے۔
- python میں break، continue، pass وغیرہ jump statement کی طرح کام کرتے ہیں۔
- pass statement، ایک null statement ہے۔ یہ عام طور پر ایک placeholder کی طرح استعمال ہوتا ہے۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:40 PM

Hands on Experience عملی مشق

1۔ ایک پروگرام لکھیں تا کہ یہ معلوم کیا جا سکے کہ دیا ہوا حرف، حرف علت(vowel) ہے یا نہیں۔

2۔ if..else..elif بیان کا استعمال کرتے ہوئے تین نمبروں میں سب سے چھوٹے عدد کی جانچ کریں۔

3۔ ایک پروگرام کے ذریعہ جانچئے کہ دیا گیا عدد مثبت، منفی یا صفر ہے۔

4۔ فبونیکی سیریز (n رقموں تک) .....0 1 1 2 3 4 5 ظاہر کرنے کے لئے ایک پروگرام لکھیں۔

5۔ n تک قدرتی نمبروں کی تعداد (حاصل جمع) ظاہر کرنے کے لئے ایک پروگرام لکھیں۔

6۔ یہ معلوم کرنے کے لئے ایک پروگرام لکھیں کہ دیا گیا عدد پالینڈوم ہے یا نہیں۔

7۔ مندرجہ ذیل نمونہ کو پرنٹ کرنے کے لئے ایک پروگرام لکھیں۔

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

8۔ یہ معلوم کرنے کے لئے کہ سال لیپ سال ہے یا نہیں، جانچنے کے لئے ایک پروگرام لکھیں۔

F5IBP

Evaluation محاسبہ

صحیح جواب کا انتخاب کریں۔ (1 مارک)

حصہ - I

1۔ پائیتھن میں کتنی اہم کنٹرول ساختیں موجود ہیں ؟

A) 3 B) 4 C) 5 D) 6

2۔ elif کا مخفف یہ سمجھا جاتا ہے۔

A) nested if B) if..else C) else if D) if..elif

3۔ ان میں سے کون پائیتھن پروگرامنگ میں اہم کردار ادا کرتا ہے؟

A) statements B) control C) structure D) indentation

4۔ کون سا بیان عام طور پر place holder کے طور پر استعمال ہوتا ہے؟

A) continue B) break C) pass D) goto

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:40 PM

5۔ if statement میں شرط، اس شکل میں ہونی چاہئے۔

A) ریاضی یا متعلقہ اظہار (Arithmetic or Relational expression)

B) ریاضی یا منطقی اظہار (Arithmetic or Logic expression)

C) متعلقہ یا منطقی اظہار (Relational or Logical expression)

D) ریاضی (Arithmetic)

6۔ سب سے زیادہ آسان لوپ کون سا ہے؟

A) do..while B) while C) for D) if..elif

7۔ درج ذیل ٹکڑے (snippet) کا output کیا ہے ؟

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

A) 12 B) 123

C) 1234 D) 124

8۔ درج ذیل ٹکڑے (snippet) کا output کیا ہے ؟

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

A) False B) True

C) 0 D) no output

9۔ ان میں سے کونسا jump statement نہیں ہے ؟

A) for B) pass C) continue D) break

10۔ blank میں کونسے اشارے (punctuation) کو استعمال کرنا چاہئے ؟

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

A) ; B) :

C) :: D) !

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:41 PM

حصہ -II

مندرجہ ذیل سوالات کے جوابات دیجئے۔ (2 مارکس)

1۔ پائیتھن میں کنٹرول ساختوں (control structures) کی فہرست بنائیں۔

2۔ break statement پر نوٹ لکھیں۔

3۔ if..else statement کے لئے syntax لکھیں۔

4۔ control structures کی وضاحت کریں.

5۔ لوپ میں range() پر ایک نوٹ لکھیں۔

حصہ -III

مندرجہ ذیل سوالات کے جوابات دیجئے۔ (3 مارکس)

1۔ درج ذیل کو ظاہر (display) کرنے کے لئے ایک پروگرام لکھیں۔

A

A B

A B C

A B C D

A B C D E

2۔ if..else کی ساخت پر ایک نوٹ لکھئے۔

3۔ if..else..elif بیان کو استعمال کرتے ہوئے 3 سب سے زیادہ بڑے اعداد کے لئے ایک پروگرام لکھئے۔

4۔ ایک while loop کے لئے syntax لکھئے۔

5۔ break اور continue statement کے درمیان فرق کیجئے۔

حصہ - IV

مندرجہ ذیل سوالات کے جوابات لکھئے (5 مارکس)

1۔ for loop سے متعلق تفصیل سے نوٹ لکھیں۔

2۔ ایک مناسب مثال کے ساتھ if..else..elif بیان پر ایک تفصیلی نوٹ لکھیں۔

3۔ تمام 3 ہندسی طاق اعداد کو ظاہر کرنے کے لئے ایک ایک پروگرام لکھیں۔

4۔ ایک دیئے گئے عدد کے ضرب کی جدول کو ظاہر کرنے کے لئے ایک پروگرام لکھیں۔

12 CS UNIT 6 CDR Set
Monday, April 17, 2023 3:49:41 PM

PYTHON FUNCTIONS پائیتھن فنکشنس

Learning Objectives سیکھنے کے مقاصد

اس باب کا مطالعہ کرنے کے بعد طلباء:

- فنکشن کے تصور اور ان کی اقسام کو سمجھیں گے۔
- built in functions اور user defined کے درمیان فرق جان سکیں گے۔
- یہ جان لیں گے کہ فنکشن کو کیسے استعمال (call) کیا جائے۔
- function کے دلائل کو سمجھ سکیں گے۔
- نامعلوم فنکشن کو جان سکیں گے۔
- ریاضی اور بعض اسٹرنگ کے فنکشنس کو جان سکیں گے۔

7.1 Introduction تعارف

افعال کو کوڈ کے بلاکس کا نام دیا گیا ہے، جو مخصوص کام کرنے کے لئے بنائے گئے ہیں۔ جب آپ کسی خاص کام کو انجام دینا چاہتے ہیں جس کی وضاحت آپ نے کسی فنکشن میں کی ہے تو، آپ اس کے ذمہ دار فنکشن کا نام دیتے ہیں۔ اگر آپ کو اپنے پورے پروگرام میں متعدد بار یہ کام انجام دینے کی ضرورت ہے تو، آپ کو بار بار ایک ہی کام کے لئے تمام کوڈ ٹائپ کرنے کی ضرورت نہیں ہے۔ آپ صرف اس کام کو سنبھالنے کے لئے مختص فنکشن کو کہتے ہیں، اور وہ کال (عمل) پائیتھن کو فنکشن کے اندر کوڈ چلانے کے لئے کہتی ہے۔ آپ کو معلوم ہوگا کہ فنکشنس کا استعمال آپ کے پروگراموں کو لکھنے، پڑھنے، جانچنے اور غلطیوں کو ٹھیک کرنے میں آسان تر بناتا ہے۔

فنکشنس کے اہم فوائد یہ ہیں :

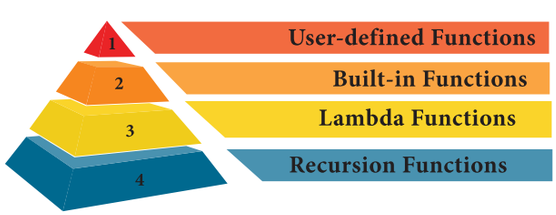
- یہ تکرار سے گریز کرتا ہے اور کوڈ کو دوبارہ استعمال کرنے کا اعلیٰ درجہ دیتا ہے۔
- یہ آپ کے اپلی کیشن کے لئے بہتر موڈولریٹی (modularity) فراہم کرتا ہے۔

نوٹ

فنکشنس کچھ اور نہیں، بلکہ متعلقہ بیانات کے ایک گروپ ہیں، جو مخصوص کام انجام دیتے ہیں۔

7.1.1 فنکشنس (افعال) کی اقسام

بنیادی طور پر، ہم افعال کو درج ذیل اقسام میں تقسیم کر سکتے ہیں۔



خاکہ 7.1 python کے فنکشنس کی قسمیں

فنکشنس Functions	وضاحت Description
user defined فنکشن	صارف کی طرف سے وضاحت کردہ فنکشنس
built-in فنکشنس	پیتھان کے اندر built in فنکشنس
lambda فنکشنس	بے نام فنکشن ہیں جو anonymous فنکشن ہیں۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:18 PM

مکرر (recursion) فنکشن	فنکشنس جو اپنے آپ کو مکرر (recursive) کہلاتے ہیں۔

جدول 7.1 پیتھان کے فنکشن اور اس کی وضاحت

7.2 فنکشنس کی وضاحت (Defining functions)

کچھ functionality کو بنانے اور استعمال کرنے کے لئے افعال کی وضاحت ہونی چاہئے۔ بہت سے بلٹ ان فنکشنز ہیں جو پائیتھن زبان کے ساتھ آتے ہیں (مثال کے طور پر، print() function)، لیکن آپ اپنے فنکشن کی بھی وضاحت کر سکتے ہیں۔

- افعال کی وضاحت کرتے وقت متعدد چیزیں ہوتی ہیں جن کو نوٹ کرنے کی ضرورت ہوتی ہے۔

7.2.1 User defined فنکشن کے لئے syntax

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

نوٹ

اوپر کے syntax میں، وہ متن جو [ ] کے اندر دیا گیا ہے، اختیاری ہے۔

بلاک

ایک بلاک کوڈ کی ایک یا ایک سے زیادہ لائنوں کو ایک ساتھ گروپ کیا جاتا ہے تاکہ ان پر عمل درآمد کے دوران بیانات کی ایک بڑی ترتیب کے طور پر سلوک کیا جائے۔ پائیتھن میں، بلاک میں بیانات انڈینٹیشن کے ساتھ لکھے جاتے ہیں۔ عام طور پر، ایک بلاک شروع ہوتا ہے جب ایک لائن (چار spaces سے) انڈینٹ ہوجاتی ہے اور بلاک کے تمام بیانات ایک جیسے انڈینٹ کی سطح پر ہونے چاہئیں۔

Nested Block: Nested بلاک

ایک بلاک کے اندر ایک اور بلاک nested block کہلاتا ہے۔ جب پہلے بلاک کا بیان اکہرے ٹیب کے space سے انڈینٹ کیا جاتا ہے تو دوسرے بلاک کا بیان دو ٹیب کے space سے انڈینٹ کیا جاتا ہے۔

فنکشن کی وضاحت کے لئے یہ ایک مثال ہے۔

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

- فنکشن بلاک کی شروعات کلیدی لفظ "def" سے ہوتی ہے، جس کے بعد فنکشن کا نام اور قوسین ( ) ہوتا ہے۔
- جب آپ کسی فنکشن کی وضاحت کرتے ہیں تو کوئی بھی ان پٹ پیرامیٹرز یا دلائل ان قوسین کے اندر رکھے جانے چاہئیں۔
- کوڈ بلاک ہمیشہ بڑی کولون (:) کے بعد آتا ہے۔
- بیان "return [expression]" ایک فنکشن کے طور پر خارج ہوتا ہے، جو اختیاری طور پر کال کرنے والے کے سامنے اظہار رائے واپس کرتا ہے۔ ایک "return" جس میں کوئی دلیل نہیں ہے، اس میں return نہیں ہے۔

نوٹ

پائیتھن کے کلیدی الفاظ کو فنکشن نام کے طور پر استعمال نہیں کیا جانا چاہئے۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:18 PM

آیئے اب عمل میں آنے والے فنکشن کو چیک کریں، تا کہ آپ دیکھ سکتے ہیں کہ وہ کسی پروگرام میں کیسے کام کرتے ہیں۔ دیئے ہوئے string کو ظاہر کرنے کے لئے ایک سادہ فنکشن کی ایک مثال یہ ہے۔

مثال : 7.3.1

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

7.2.2 User defined فنکشن کے فوائد (Advantages of User-defined functions)

1۔ فنکشنس ہمیں پروگرام کو ماڈیولز (modules) میں تقسیم کرنے میں مدد کرتے ہیں۔ اس سے کوڈ کا نظم ونسق آسان ہو جاتا ہے۔

2۔ یہ کوڈ دوبارہ استعمال پر عمل درآمد کرتا ہے۔ ہر بار جب آپ کو بیانات کی ترتیب پر عمل درآمد کرنے کی ضرورت ہوتی ہے تو، آپ کو فنکشن کو کال کرنے کی ضرورت ہوتی ہے۔

3۔ functions، ہمیں آسانی سے فعالیت کو تبدیل کرنے کی اجازت دیتا ہے، اور مختلف پروگرامز مختلف افعال پر کام کر سکتے ہیں۔

7.3 ایک function کو کال کرنا (عمل میں لانا) (Calling a function)

مثال 7.2-1 سے hello()function کو کال کرنے کے لئے، آپ یہ کوڈ استعمال کرتے ہیں:

جب آپ function "hello()" کو عمل میں لاتے ہیں، تو پروگرام مندرجہ ذیل سٹرنگ کو آؤٹ پٹ کے طور پر ظاہر کرتا ہے:

Output

hello – Python

متبادل کے طور پر hello() فنکشن کو کال کرنے کے لئے پروگرام درج ذیل string کی طرح output دیتا ہے۔

مثال : 7.3.2

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

اگر return میں کوئی دلیل نہیں ہے تو، آؤٹ پٹ کے آخری بیان کے بطور "None" دکھایا جائے گا۔

مذکورہ بالا فنکشن درج ذیل کا output دے گا۔

Output:

hello – Python

None

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:19 PM

7.4 فنکشن میں پیرامیٹرس کو pass کرنا (Passing parameters in functions)

پیرامیٹرز یا دلائل کو فنکشنس تک پہنچایا(pass) جاسکتا ہے۔

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

آیئے افعال کی وضاحت کرتے ہوئے پیرامیٹرز کا استعمال دیکھیں۔آپ جوتوسین میں پیرامیٹر رکھتے ہیں وہ فنکشن کے ذریعہ ہی استعمال ہوگا۔ آپ فنکشن میں ہرطرح کا ڈیٹا منتقل کرسکتے ہیں۔یہ ایک مثال پروگرام ہے جوایک فنکشن کی وضاحت کرتا ہے جوفنکشن میں پیرامیٹرز کو pass کرنے میں مدد کرتا ہے۔

مثال: 7.4

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

مندرجہ بالا کوڈ چوڑائی اور اونچائی کی قیمتوں کو w اور h میٹرز کوتفویض کرتا ہے۔یہ پیرامیٹرز فنکشن "area" (رقبہ) کی تشکیل میں استعمال ہوتے ہیں۔جب آپ مندرجہ بالا فنکشن کو کال کرتے ہیں تو، یہ چوڑائی اوراونچائی کی مصنوعات کو output کے طور پر واپس کرتا ہے۔

3 اور 5 کی قیمت بالترتیب چوڑائی w اور بلندی h کو منتقل کردی جاتی ہے، فنکشن 15 کو output کے طور پر واپس دے گا۔

کیا تم جانتے ہو

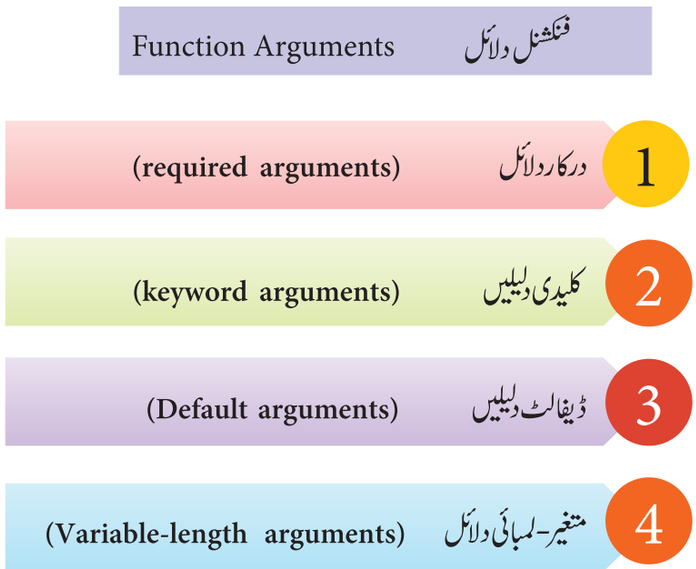
ہم اکثر اصطلاحات پیرامیٹرز اور دلائل کو ایک دوسرے کے ساتھ بدل بدل کر استعمال کرتے ہیں۔تاہم، ان میں تھوڑا سا فرق ہے۔ پیرامیٹرز وہ تغیرات ہیں جو فنکشن کی وضاحت میں استعمال ہوتے ہیں جبکہ دلائل وہ قیمتیں ہیں، جن کو ہم فنکشن پیرامیٹرز میں منتقل کرتے ہیں۔

7.5 فنکشنل دلائل (Function arguments)

دلائل کسی فنکشن کو کال کرنے (عمل میں لانے) کے لئے استعمال ہوتے ہیں اور بنیادی طور پر 4 قسم کے فنکشن ہوتے ہیں جو کوئی استعمال کرسکتا ہے

Required arguments، key arguments، default arguments اور variable length arguments۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:19 PM



7.5.1 درکار دلائل (Required arguments)

Required arguments وہ دلائل ہیں جو صحیح مقامی ترتیب کے ساتھ کسی فنکشن میں بھیجے جاتے ہیں۔
یہاں، فنکشن کال میں دلائل کی تعداد فنکشن کی وضاحت کے عین مطابق ہونی چاہئے۔ درکار output حاصل کرنے کیلئے syntax errors کو روکنے کے لئے آپ کو کم از کم ایک پیرامیٹر کی ضرورت ہے۔

مثال: 7.5.1

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

جب مذکورہ کوڈ پر عمل درآمد کیا جاتا ہے تو، یہ درج ذیل error بتاتا ہے۔

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

مذکورہ کوڈ میں printstring() کی بجائے اگر ہم printstring("Welcome") استعمال کرتے ہیں تو آؤٹ پٹ اس طرح ہوتا ہے۔

Output:

Example - Required arguments
Welcome

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:19 PM

7.5.2 کلیدی الفاظ کے دلائل (keyword arguments)

پیرامیٹرز کے پیرامیٹر کے ناموں سے ان کے پہچاننے کے بعد keyword argument فنکشنس میں شامل ہوں گے۔ keyword argument کی قیمت پیرامیٹر کے نام کے ساتھ جڑتی(matched) ہے اور اسی طرح، کوئی بھی دلائل کو نامناسب ترتیب میں رکھ سکتا ہے (ترتیب میں نہیں)۔

7.5.2 (a) مثال:

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

جب مندرجہ بالا کوڈ عمل درآمد ہوتا ہے، تو یہ مندرجہ ذیل output تیار کرتا ہے:

Output:

Example-1 Keyword arguments
Name :Gshan

7.5.2 (b) مثال:

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

جب مندرجہ بالا کوڈ عمل درآمد ہوتا ہے، تو یہ مندرجہ ذیل نتیجہ دیتا ہے:

TypeError: printdata() got an unexpected keyword argument 'name1'

7.5.2 (c) مثال:

```
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")
```


12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:19 PM

جب مندرجہ بالا کوڈ عمل درآمد ہوتا ہے، تو یہ مندرجہ ذیل نتیجہ دیتا ہے:

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

نوٹ :

مندرجۂ بالا پروگرام میں پیرامیٹرز کی ترتیب کو تبدیل کیا گیا ہے۔

7.5.3 پہلے سے طے شدہ دلائل (Default arguments)

پائیتھن میں ڈیفالٹ دلیل ایک ایسی دلیل ہے جو ڈیفالٹ ویلیو لیتی ہے اگر فنکشن کال میں کوئی ویلیو نہیں دی جاتی ہے۔ مندرجہ ذیل مثال، پہلے سے طے شدہ دلائل کا استعمال کرتی ہے، جو جب کوئی دلیل منظور نہیں ہوتی ہے تو پہلے سے طے شدہ تنخواہ پرنٹ کرتی ہے۔

مثال: 7.5.3

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

جب مندرجہ بالا کوڈ عمل درآمد ہوتا ہے، تو یہ مندرجہ ذیل نتیجہ دیتا ہے:

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

جب مندرجہ بالا کوڈ کو print info("Ram",2000) کی طرح تبدیل کیا جاتا ہے، تو یہ مندرجہ ذیل آؤٹ پٹ تیار کرتا ہے:

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

مذکورہ کوڈ میں، 2000 کی قیمت دلیل کی تنخواہ کی طرح منتقل کر دی گئی ہے، پہلے ہی تنخواہ کے لئے مختص شدہ پہلے سے طے شدہ قیمت نظر انداز کر دی جاتی ہے۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:20 PM

7.5.4 متغیر لمبائی کے دلائل (Variable-length arguments)

کچھ واقعات میں آپ کو پہلے سے بیان کیے گئے بیانات سے کہیں زیادہ دلائل پاس کرنے کی ضرورت پڑسکتی ہے۔اس کی وضاحت کرنے کیلئے فنکشن میں واپس جانا ایک تکلیف دہ عمل ہوسکتا ہے۔اس کی بجائے متغیر لمبائی کے دلائل استعمال کیے جاسکتے ہیں۔یہ function کی وضاحت میں نہیں بتائے گئے ہیں اور اس طرح کے دلائل کی وضاحت کے لئے asterisk(*) کا استعمال کیا جاتا ہے۔

آیئے دیکھتے ہیں کہ جب sum() function میں 3 سے زیادہ دلائل پاس کرتے ہیں تو کیا ہوتا ہے۔

7.5.4 مثال:

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

جب مندرجہ بالا کوڈ عمل درآمد ہوتا ہے، تو یہ مندرجہ ذیل نتیجہ دیتا ہے:

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

7.5.4.1 متغیر لمبائی کے دلائل کے لئے syntax (Syntax - Variable-Length Arguments)

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

7.5.4.1 مثال :

```
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)
```

Output:

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



اپنا محاسبہ کریں Evaluate Yourself

مذکورہ پروگرام میں فنکشن کا نام printnos تمام جگہوں پر پرنٹ نام کے طور پر تبدیل کریں۔جہاں بھی اسے استعمال کیا جائے اور مناسب ڈیٹا سابقہ دیں۔ مثال printnames ('mala', 'kala','bala') کو printnos(10,20,30) کی طرح پرنٹ کریں۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:20 PM

متغیر لمبائی دلائل میں ہم دو طریقوں کا استعمال کرتے ہوئے دلائل پاس کر سکتے ہیں۔

1۔ غیر کلیدی الفاظ کے متغیر دلائل (Non keyword variable arguments)

2۔ کلیدی لفظ کے متغیر دلائل (Keyword variable arguments)

غیر کلیدی الفاظ کے متغیر دلائل کو tuples کہا جاتا ہے۔ آپ بعد کے ابواب میں ٹیپلوں کے بارے میں مزید معلومات حاصل کریں گے۔ دیئے گئے پروگرام میں غیر مطلوبہ الفاظ کی متغیر دلیل کی مثال ہے۔

نوٹ

کلیدی الفاظ کے متغیر دلائل اس کتاب کے دائرہ کار سے باہر ہیں۔

کیا تم جانتے ہو

پائیتھن کی print() function بذات خود اس طرح کے فنکشن کی ایک مثال ہے جو متغیر لمبائی کے دلائل کی تائید کرتی ہے۔

7.6 بے نام فنکشنس (Anonymous functions) — 7.6 Anonymous Functions

بے نام فنکشن کیا ہے؟

پائیتھن میں، بے نام فنکشن ایک ایسا فنکشن ہوتا ہے جس کی تعریف بغیر کسی نام کے کی جاتی ہے۔ اگر چہ عام افعال def کی ورڈ کا استعمال کرتے ہوئے بیان کیے جاتے ہیں، پائیتھن میں بے نام افعال lambda کی ورڈ کا استعمال کرتے ہوئے بیان کیے جاتے ہیں۔ لہذا، بے نام افعال کو lambda افعال بھی کہا جاتا ہے۔

lambda یا anonymous فنکشن کا استعمال کیا ہے ؟

- lambda فنکشن زیادہ تر چھوٹے اور ایک بار کے بے نام فنکشن بنانے کے لئے استعمال ہوتا ہے۔
- lambda فنکشن بنیادی طور پر افعال جیسے filter() ، map() اور reduce()۔

نوٹ

filter() ، map() اور reduce() فنکشن وغیرہ اس کتاب کے دائرہ کار سے باہر ہیں۔

کیا تم جانتے ہو

lambda فنکشن بہت سارے دلائل لے سکتا ہے اور اسے اظہار کی شکل میں ایک قیمت لوٹانی چاہئے۔

lambda فنکشن صرف عالمی متغیر تک رسائی حاصل کر سکتا ہے۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:20 PM

7.6.1 بے نام افعال کے لئے syntax (Syntax of Anonymous Functions)

بے نام افعال کے لئے ترکیب مندرجہ ذیل ہے۔

```
lambda [argument(s)] :expression
```

مثال: 7.6.1

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

Output:

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

مندرجہ بالا lambda فنکشن جو دلیل arg1 کے ساتھ دلیل arg2 کو جوڑتا ہے اور متغیر رقم میں نتیجہ کو اسٹور کرتا ہے۔ نتیجہ print() کا استعمال کرتے ہوئے ظاہر ہوتا ہے۔

7.7 return کا بیان (The return statement)

- return statement آپ کے فنکشن سے باہر نکل جاتا ہے اور اس کے کال کرنے والے کو ایک قیمت لوٹاتا ہے۔ عام طور پر افعال کا نقطہ ان پٹ لینا اور کچھ لوٹانا ہے۔
- return statement اس وقت استعمال کیا جاتا ہے جب کوئی فنکشن اپنے caller کو کوئی قیمت return کرنے کے لئے تیار ہو۔ لہذا، صرف ایک ہی return statement، رن ٹائم پر عمل میں لایا جاتا ہے، حالانکہ اس فنکشن میں ایک سے زیادہ return statement ہوتے ہیں۔
- فنکشن کی تعریف میں کسی بھی تعداد میں return statement کی اجازت ہے لیکن رن ٹائم کے وقت ان میں سے صرف ایک پر عمل درآمد ہوتا ہے

7.7.1 return کے لئے Syntax (Syntax of return)

```
return [expression list ]
```

اس بیان میں اظہار رائے ہوسکتا ہے جس کی جانچ پڑتال کی جاتی ہے اور قیمت واپس کردی جاتی ہے۔ اگر بیان میں کوئی اظہار رائے موجود نہیں ہے یا واپسی کا بیان خود کسی فنکشن کے اندر موجود نہیں ہے تو پھر فنکشن کوئی بھی اعتراض نہیں لوٹائے گا۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:21 PM

مثال : 7.7.1

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

Output 1:

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

Output 2:

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

7.8 متغیرات کا دائرہ کار (Scope of variables)

متغیر کی گنجائش سے مراد پروگرام کا وہ حصہ ہوتا ہے، جہاں یہ قابل رسائی ہوتا ہے، یعنی وہ علاقہ جہاں آپ اسے refer(use) کر سکتے ہیں۔ ہم کہہ سکتے ہیں کہ دائرہ کار میں متغیرات اور ان کی قیمتوں کا موجودہ مجموعہ ہے۔ ہم دو طرح کے دائرہ کار کا مطالعہ کریں گے- local scope اور global scope۔

7.8.1 مقامی اسکوپ (local scope)

فنکشن کے جسم کے اندر یا مقامی دائرہ کار میں اعلان کردہ متغیر کو مقامی متغیر کے نام سے جانا جاتا ہے۔

مقامی متغیر کے قانون (اصول) (Rules of local variable)

- مقامی اسکوپ کے ساتھ ایک متغیر تک صرف اس فنکشن/ بلاک کے اندر ہی رسائی حاصل کی جا سکتی ہے جس میں اس میں تخلیق کیا گیا ہو۔
- جب فنکشن/ بلاک کے اندر متغیر تیار کیا جاتا ہے تو، متغیر اس کے لئے مقامی ہو جاتا ہے۔
- ایک مقامی متغیر صرف اس صورت میں موجود ہے جب کام انجام دے رہا ہو۔
- باضابطہ دلائل (formal arguments) بھی ایک فنکشن کے لئے بھی مقامی ہیں۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:21 PM

مثال:7.8.1(a) مقامی متغیر بنائیں۔ Example : Create a Local Variable

```
def loc():
        y=0 # local scope
        print(y)
loc()
```

Output:

```
0
```

مثال: 7.8.1(b) دائرہ کار سے باہر مقامی متغیر تک رسائی Example : Accessing local variable outside the scope

```
def loc():
        y = "local"
loc()
print(y)
```

جب ہم مذکورہ کوڈ کو چلاتے ہیں تو، آؤٹ پٹ میں درج ذیل error دکھائی دیتا ہے۔

مذکورہ بالا error اس لئے ہوتی ہے، کیونکہ ہم عالمی سطح پر مقامی متغیر 'y' تک رسائی حاصل کرنے کی کوشش کر رہے ہیں۔

```
NameError: name 'y' is not defined
```

7.8.2 عالمی دائرہ کار (Global scope)

پروگرام میں کہیں بھی عالمی دائرہ کار کے ساتھ ایک متغیر استعمال کیا جاسکتا ہے۔ یہ کسی بھی فنکشن/ بلاک کے دائرہ کار سے باہر متغیر کی وضاحت کرکے پیدا کیا جاسکتا ہے۔

عالمی کلیدی الفاظ کے لئے اصول (Rules of global keyword)

پائیتھن میں عالمی کلیدی الفاظ کے بنیادی اصول یہ ہیں:

- جب ہم کسی فنکشن سے باہر کسی متغیر کی وضاحت کرتے ہیں تو، یہ بطور ڈیفالٹ عالمی ہوتا ہے۔ آپ کو عالمی کلیدی لفظ استعمال کرنے کی ضرورت نہیں ہے۔
- ہم کسی فنکشن میں عالمی متغیر کو پڑھنے اور لکھنے کے لئے عالمی کلیدی الفاظ کا استعمال کرتے ہیں۔
- کسی فنکشن سے باہر عالمی مطلوبہ الفاظ کے استعمال کا کوئی اثر نہیں ہوتا ہے۔

عالمی کلیدی الفاظ کا استعمال (use of global keyword)

مثال: 7.8.2(a) ایک فنکشن کے اندر سے عالمی متغیر تک رسائی Example : Accessing global Variable From Inside a Function

```
c = 1          # global variable
def add():
        print(c)
        add()
```

Output:

```
1
```


12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:21 PM

مثال:7.8.2(b) ایک فنکشن کے اندر سے عالمی متغیر تک رسائی

```
c = 1            # global variable
def add():
        print(c)
        add()
```

Output:

```
1
```

نوٹ

عالمی کلیدی لفظ استعمال کیے بغیر ہم فنکشن کے اندر عالمی متغیر کو تبدیل نہیں کر سکتے۔ لیکن ہم صرف عالمی متغیر تک رسائی حاصل کر سکتے ہیں۔

مثال: 7.8.2© ایک فنکشن کے اندر سے عالمی متغیر میں تبدیلی

```
c = 1            # global variable
def add():
        c = c + 2 # increment c b  y 2
        print(c)
add()
```

Output:

```
Unbound Local Error: local variable 'c' referenced before assignment
```

مثال: عالمی مطلوبہ الفاظ کا استعمال کرتے ہوئے ایک فنکشن کے اندر سے عالمی متغیر کو تبدیل کرنا

```
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)
```

Output:

```
Inside add() function  x value is : 5
In main x value is : 5
```

مذکورہ پروگرام میں، x کو عالمی متغیر کے طور پر بیان کیا گیا ہے۔ **add() function** کے اندر، x کے لئے **global keyword** استعمال کیا جاتا ہے اور ہم متغیر x کو 5 تک بڑھا دیتے ہیں۔ اب ہم عالمی سطح پر x کی تبدیلی کو فنکشن کے باہر دیکھ سکتے ہیں، یعنی x کی قیمت 5 ہے۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:21 PM

7.8.3 عالمی اور مقامی متغیرات (Global and local variables)

یہاں، ہم ایک ہی کوڈ میں عالمی متغیرات اور مقامی متغیرات کو کس طرح استعمال کرتے ہیں، اس کا طریقہ دکھائیں گے۔

مثال: 7.8.3(a) عالمی اور مقامی متغیرات کو ایک ہی کوڈ میں استعمال کرنا

```
x=8                    # x is a global variable
def loc():
        global x
        y = "local"
        x = x * 2
        print(x)
        print(y)
loc()
```

Output:

```
16
local
```

مندرجہ بالا پروگرام میں، loc() function میں ہم x کو عالمی اور y کو مقامی متغیر کے طور پر اعلان کرتے ہیں۔

loc() function میں کال کرنے کے بعد، x کی قیمت 16 ہو جاتی ہے، کیونکہ ہم نے x = x * 2 استعمال کیا ہے۔ اس کے بعد، ہم مقامی متغیر y کی قیمت پرنٹ کرتے ہیں۔

مثال : 7.8.3(b) ایک ہی نام والے عالمی متغیر اور مقامی متغیر

```
x = 5
def loc():
        x = 10
        print ("local x:", x)
loc()
print ("global x:", x)
```

Output:

```
local x: 10
global x: 5
```

مذکورہ کوڈ میں، ہم عالمی متغیر اور مقامی متغیر دونوں کے لئے ایک ہی نام x استعمال کرتے ہیں۔ جب ہم ایک ہی متغیر کو پرنٹ کرتے ہیں تو ہمیں مختلف نتیجہ ملتا ہے، کیونکہ متغیر دونوں اسکوپ میں اعلان کیا جاتا ہے، یعنی function loc() کے اندر موجود مقامی دائرہ کار اور loc() کے باہر عالمی دائرہ کار۔

آؤٹ پٹ : local x: 10 کو متغیر کا مقامی اسکوپ کہا جاتا ہے۔

آؤٹ پٹ : global x: 5 کو متغیر کا عالمی دائرہ کار کہا جاتا ہے۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:21 PM

7.9 لائبریریوں کا استعمال کرتے ہوئے فنکشنس

7.9.1 بلٹ ان اور ریاضی کے افعال

Function فنکشن	وضاحت	Syntax	Example مثال
abs ( )	ایک عدد کی مطلق قیمت کو return کہتی ہے۔ دلیل ہوسکتا ہے ایک سالم عدد ہو یا floating point number	abs (x)	x=20 y=-23.2 print('x = ', abs(x)) print('y = ', abs(y)) **Output:** x = 20 y = 23.2
ord ( )	دئے گئے Unicode کے ASCII قیمت کو return کرتی ہے۔ یہ فنکشن chr() کا معکوس (اُلٹا) ہے۔	ord (c)	c= 'a' d= 'A' print ('c = ',ord (c)) print ('A = ',ord (d)) **Output:** c = 97 A = 65
chr ( )	دی گئی ASCII کی قیمت کے لئے Unicode return کو character کرتی ہے۔ یہ فنکشن ord() کا معکوس (اُلٹا) ہے۔	chr (i)	c=65 d=43 print (chr (c)) prin t(chr (d)) **Output:** A +
bin ( )	دئے گئے سالم عدد کے لئے binary string جس میں سابقہ ob لگا ہو، اس کو return کرتی ہے۔ نوٹ: اس فنکشن کے علاوہ format() کو بھی استعمال کیا جاسکتا ہے۔	bin (i)	x=15 y=101 print ('15 in binary : ',bin (x)) print ('101 in binary : ',bin (y)) **Output:** 15 in binary : 0b1111 101 in binary : 0b1100101

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:22 PM

type ( )	دئے گئے اکہرے شئے کے لئے شئے کی قسم کو return کرتا ہے۔ نوٹ: یہ فنکشن اکہرے شئے کے پیرامیٹر میں استعمال ہوتا ہے۔	type (object)	x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s)) **Output:** <class 'float'> <class 'str'> <class 'bool'>
id ( )	id()، شئے کی identity(شناخت) کو return کرتا ہے، یعنی شئے کے پتہ کو حافظہ سے return کرتا ہے۔ نوٹ: آپ کے سسٹم میں x اور y کے پتہ میں فرق ہوسکتا ہے۔	id (object)	x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y)) **Output:** address of x is : 1357486752 address of y is : 13480736
min ( )	ایک list میں کم سے کم قیمت کو return کرتی ہے۔	min (list)	MyList = [21,76,98,23] print ('Minimum of MyList :', min(MyList)) **Output:** Minimum of MyList : 21
max ( )	ایک list میں زیادہ سے زیادہ قیمت کو return کرتی ہے۔	max (list)	MyList = [21,76,98,23] print ('Maximum of MyList :', max(MyList)) **Output:** Maximum of MyList : 98
sum ( )	ایک list میں موجود قیمتوں کے حاصل جمع کو return کرتی ہے۔	sum (list)	MyList = [21,76,98,23] print ('Sum of MyList :', sum(MyList)) **Output:** Sum of MyList : 218

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:22 PM

format ( )	دیئے گئے format کے output کی بنیاد پر return کرتی ہے۔ 1۔ثنائی فارمٹ۔ اعداد کو قاعدہ 2 کے نظام میں output دیتا ہے۔ 2۔ آکٹل فارمٹ ۔ اعداد کو قاعدہ 8 کے نظام میں output دیتا ہے۔ 3۔ مخصوص نقطہ والی ترقیم۔عدد کو ایک مخصوص نقطہ پر ظاہر کرتا ہے۔ اس کا ڈیفالٹ درست عدد 6 ہے۔	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')) **Output:** x value in binary : 1110 y value in octal : 31 y value in Fixed-point no : 25.000000
round ( )	قریبی سالم عدد کو اس کے return میں input کرتا ہے۔ 1۔ پہلا دلائل (عدد) کو مخصوص قیمت پر نشان لگانے کے لئے استعمال کیا گیا ہے۔	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))

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:22 PM

	2۔ دوسرے دلائل (n ہندسے) کو نشان کرنے کے بعد درکار عشری ہندسے کی تعداد کو ظاہر کرتا ہے۔		**Output: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)) **Output:2** 18.0 17.9 17.89
pow ( )	ab محسوب کرنے کے لئے return دیتا ہے۔ یعنی (a**b)۔ a کی قوت b	pow (a,b)	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) print (pow (a+b,3)) **Output:** 25 125.0 343

ریاضی کے فنکشنس (Mathematical Table)

نوٹ

کسی پروگرام میں ریاضی کے تمام فنکشنس استعمال کرنے سے پہلے **import math** کے ماڈیول کی وضاحت کریں۔

Function فنکشن	وضاحت	Syntax	Example مثال
floor ( )	بڑے سے بڑا سالم عدد جو x سے کم یا مساوی ہے، اس کو return کرتا ہے۔	math.floor (x)	x=26.7 y=-26.7 z=-23.2 print (math.floor (x)) print (math.floor (y)) print (math.floor (z)) **Output:** 26 -27 -24

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:22 PM

ceil ( )	چھوٹے سے چھوٹا سالم عدد جو x سے زیادہ یا مساوی ہے، اس کو **return** کرتا ہے۔	math.ceil (x)	x= 26.7 y= -26.7 z= -23.2 print (math.ceil (x)) print (math.ceil (y)) print (math.ceil (z)) **Output:** 27 -26 -23
sqrt ( )	x کے جذرالمربع کو **return** کرتا ہے۔ غور کیجئے : x، صفر سے بڑا ہونا چاہئے۔	sqrt (x)	a= 30 b= 49 c= 25.5 print (math.sqrt (a)) print (math.sqrt (b)) print (math.sqrt (c)) **Output:** 5.477225575051661 7.0 5.049752469181039

7.9.2 افعال میں ترکیب (Composition in functions)

افعال میں ترکیب کیا ہے ؟ (What is composition in functions ?)

کسی فنکشن کے ذریعہ واپس کی گئی قیمت کسی دوسرے فنکشن کے لئے بطور دلیل **nested** کے طور پر استعمال کی جاسکتی ہے۔ اسے ترکیب کہتے ہیں۔ مثال کے طور پر، اگر ہم صارف سے اعداد کے طور پر ایک عددی قیمت یا اظہار لینا چاہتے ہیں تو، ہم فنکشن **input()** کا استعمال کرتے ہوئے صارف سے **input string** لیتے ہیں اور اس کی قیمت کا اندازہ کرنے کے لئے **eval()** فنکشن لگاتے ہیں، مثال کے طور پر

مثال :

```
# 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 پائیتھن کے مکرر فنکشنس (Python recursive functions)

جب کوئی فنکشن خود کو بلاتا ہے تو اسے تکرار **(recursion)** کے نام سے جانا جاتا ہے۔ تکرار لوپ کی طرح کام کرتی ہے لیکن بعض اوقات لوپ کے مقابلے میں تکرار کو استعمال کرنے میں زیادہ معنی خیز ہوتا ہے۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:22 PM

آپ کسی بھی لوپ کو تبدیل کر سکتے ہیں۔ تکرار ایک بار بار چلنے والا فنکشن خود بلاتا (call) ہے۔ ذرا تصور کریں کہ اگر کسی حالت سے روکا نہیں گیا تو عمل لامحدود مدت تک دوبارہ ہو جائے گا! اس طرح کے فنکشن کو لامحدود مکرر کے نام سے جانا جاتا ہے۔ کسی بھی مکرر فنکشن میں جس حالت کا اطلاق ہوتا ہے، اسے بنیادی حالت کے طور پر جانا جاتا ہے۔ ہر مکرر فنکشن میں ایک بنیادی حالت لازمی ہے، ورنہ یہ لامحدود لوپ کی طرح کام کرتا رہے گا۔

مکرر فنکشن کس طرح کام کرتا ہے، اس کا جائزہ لینا۔

1۔ تکرار کرنے والی فنکشن کو کچھ بیرونی کوڈ کے ذریعہ کہا جاتا ہے۔

2۔ اگر بنیادی شرط پوری ہو جائے تو پروگرام معنی خیز output دیتا ہے اور خارج ہو جاتا ہے۔

3۔ بصورت دیگر، فنکشن کچھ ضروری پروسیسنگ کرتا ہے اور پھر اپنے آپ کو تکرار جاری رکھنے کے لئے بلاتا (call) ہے۔

factorial محسوب کرنے کے لئے مکرر فنکشن کی یہ ایک مثال ہے۔

مثال : 7.10

```
def fact(n):
        if n == 0:
                return
        else:
                return n * fact (n-1)
        print (fact (0))
        print (fact (5))
Output:
        1
        120
```

کیا تم جانتے ہو؟

جب زیادہ سے زیادہ تکرار، موازنہ سے تجاوز کر جاتی ہے تو print(fact(2000)) ایک runtime error دکھائے گا۔ یہ اس وقت ہوگا جب پیتھان ڈیفالٹ کے طور پر، 1000 کالوں کے بعد مکرر فنکشنوں کو بلانا (عمل میں لانا) بند کر دے گا۔ اس حد کو تبدیل کرنے کے لئے sys.setrecursionlimit (limit_value) اجازت دیتا ہے۔

مثال :

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
        if n == 0:
        return
 else:
        return n * fact(n-1)
print(fact (2000))
```


12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:22 PM

Points to remember: نکات برائے یادداشت :

- فنکشنس کوڈ بلاکس کا نام دیا گیا ہے جو ایک خاص کام کرنے کے لئے ڈیزائن کیا گیا ہے۔
- فنکشنس کی اقسام user defined، built-in، lambda اور recursion ہیں۔
- فنکشنس بلاکس کی شروعات کلیدی حرف "def" سے ہوتی ہے، جس کے بعد فنکشن کا نام اور قوسین () شامل ہوتے ہیں۔
- بغیر دلیل (arguments) کے ساتھ "return" کی طرح ہے۔ پائیتھن میں return statement اختیاری ہے۔
- پائیتھن میں، بلاک میں بیانات کا آغاز انڈینٹیشن سے ہونا چاہئے۔
- بلاک کے اندر موجود بلاک کو نیسٹ بلاک کہا جاتا ہے۔
- دلائل (arguments) کسی فنکشن کو کال کرنے کے لئے استعمال ہوتے ہیں اور بنیادی طور پر 4 قسم کے فنکشنس ہوتے ہیں، جو ہر کوئی استعمال کرسکتا ہے required arguments، keyword arguments، default arguments، اور variable-length arguments
- required arguments وہ دلائل ہیں، جو صحیح مقامی ترتیب کے مطابق کسی فنکشن کو بھیجے جاتے ہیں۔
- پیرامیٹرز کو پیرامیٹر کے ناموں سے ان کے پہچاننے کے بعد keyword arguments میں شامل ہوں گے۔
- ایک پائیتھن فنکشن کی وضاحت میں پیرامیٹرز کے لئے پہلے سے طے شدہ اقدار دینے کی اجازت دیتا ہے۔ ہم اسے default arguments کہتے ہیں۔
- متغیر لمبائی دلائل فنکشن کی وضاحت میں بیان نہیں کیے جاتے ہیں اور اس طرح کے دلائل (arguments) کی وضاحت کے لئے ایک asterisk (*) کا استعمال کیا جاتا ہے۔
- بے نام فنکشن (Anonomous function) ایک ایسا فنکشن ہے، جس کی تعریف بغیر کسی نام کے کی جاتی ہے۔
- متغیر کے اسکوپ سے مراد پروگرام کا وہ حصہ ہوتا ہے، جہاں یہ قابل رسائی ہوتا ہے، یعنی وہ علاقہ جہاں آپ اسے حوالہ (استعمال) کر سکتے ہیں۔
- کسی فنکشن کے ذریعہ واپس کی گئی قیمت کسی دوسرے فنکشن کے لئے بطور دلیل انداز میں استعمال کی جاسکتی ہے۔ اسے ترکیب (composition) کہتے ہیں۔
- ایک فنکشن جو خود کو بلاتا (call) ہے، اسے تکرار (recursion) کے نام سے جانا جاتا ہے۔ تکرار لوپ کی طرح کام کرتی ہے، لیکن بعض اوقات لوپ کے بجائے تکرار کو استعمال کرنے میں زیادہ معنی خیز ہوتا ہے۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:23 PM

Hands on Experience عملی مشق



1۔ اوپر کے پروگرام میں درج ذیل کوڈ کو آزمائیں۔

Slno	code	Result
1	printinfo(“3500”)	
2	printinfo(“3500””Sri”)	
3	printinfo(name=”balu”)	
4	printinfo(“Jose”,1234)	
5	printinfo(“ ”,salary=1234)	

2۔ درج ذیل افعال کا محاسبہ کریں اور آؤٹ پٹ لکھیں۔

Slno	Function	Output
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)	

3۔ درج ذیل افعال کا محاسبہ کریں اور آؤٹ پٹ لکھیں۔

Slno	function	Output
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)	

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:23 PM

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)	



Evaluation محاسبہ

حصہ I

صحیح جواب کا انتخاب کریں: (1 نمبر)

1۔ کوڈ کے ایک named blocks ایک خاص کام کرنے کے لئے تیار کیے گئے ہیں، ان کو یہ کہا جاتا ہے۔

(a) لوپ (b) برانچنگ (c) فنکشن (d) بلاک

2۔ ایک فنکشن جو خود کو بلاتا (Call) ہے، اس کو یہ کہا جاتا ہے۔

(a) بلٹ ان (builtin) (b) تکرار (recursion)

(c) لیمبڈا (lambda) (d) واپسی (return)

3۔ کون سا فنکشن بے نام (anonymous) اور un-named فنکشن کہا جاتا ہے؟

(a) لیمبڈا (lambda) (b) تکرار (recursion)

(c) فنکشن (function) (d) وضاحت (define)

4۔ فنکشن بلاک شروع کرنے کے لئے مندرجہ ذیل میں سے کون سے کلیدی لفظ (keyword) استعمال ہوتا ہے؟

(a) define (b) for (c) finally (d) def

5۔ فنکشن بلاک سے باہر نکلنے کے لئے مندرجہ ذیل میں سے کون سے کلیدی لفظ (keyword) استعمال کیا جاتا ہے؟

(a) define (b) return (c) finally (d) def

6۔ کسی فنکشن کی وضاحت کرتے ہوئے مندرجہ ذیل علامت میں سے کون سا استعمال ہوتا ہے۔

(a) ; (semicolon) (b) . (dot)

(c) : (colon) (d) $ (dollar)

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:23 PM

7۔ کن دلائل (arguments) میں کسی فنکشن میں صحیح مقامی آرڈر (correct positional order) منظور کیا جاتا ہے ؟

(a) Required (b) Keyword

(c) Default (d) length-Variable

8۔ مندرجہ ذیل بیان کو پڑھیں اور صحیح بیانات کا انتخاب کریں۔

(I) پائیتھن میں، آپ کو فنکشن کی وضاحت کرتے وقت ڈیٹا کی مخصوص قسموں کا تذکرہ کرنے کی ضرورت نہیں ہے۔

(II) پائیتھن کے مطلوبہ الفاظ کو فنکشن نام کے طور پر استعمال کیا جاسکتا ہے۔

(a) I صحیح ہے اور II غلط ہے۔ (b) دونوں صحیح ہیں

(c) I غلط ہے اور II صحیح ہے۔ (d) دونوں غلط ہیں

9۔ دیئے گئے بیان کو کامیابی کے ساتھ چلانے کے لئے صحیح انتخاب کریں۔

if _____ : print (x, "is a leap year")

(a) x%2=0 (b) x%4==0

(c) x/4=0 (d) x%4=0

10۔ مندرجہ ذیل میں سے کون سا کلیدی لفظ (keyword)، فنکشن testpython(): کی وضاحت کے لئے استعمال ہوتا ہے؟

(a) define (b) pass (c) def (d) while

حصہ II

مندرجہ ذیل سوالات کے جوابات دیں: (2 نمبرات)

1۔ فنکشن (function) کیا ہے؟

2۔ مختلف قسم کے فنکشن کے بارے میں لکھیں۔

3۔ فنکشن کے اہم فوائد کیا ہیں؟

4۔ متغیر کے دائرہ کار (scope of variable) سے کیا مراد ہے؟ اس کی اقسام کا ذکر کریں۔

5۔ عالمی دائرہ کار (global scope) کی وضاحت کریں۔

6۔ مکرر فنکشن (recursive function) میں بنیادی حالت (base function) کیا ہے؟

7۔ مکرر فنکشن کی حد کس طرح مقرر کی جاتی ہے؟ ایک مثال دیں۔

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:23 PM

حصہ III

مندرجہ ذیل سوالات کے جوابات دیں: (3 مارکس)

1۔ مقامی متغیر (local variable) کے اصول لکھیں۔
2۔ پیتھان میں عالمی کلیدی الفاظ (global keyword) کے بنیادی اصول لکھیں۔
3۔ جب ہم فنکشن کے اندر عالمی متغیر (global variable) کو ترمیم کرتے ہیں تو کیا ہوتا ہے ؟
4۔ ceil() اور floor() میں فرق کیجئے۔
5۔ یہ دیکھنے کے لئے دیا ہوا سال لیپ سال ہے یا نہیں، اس کے لئے پیتھان میں ایک پروگرام لکھئے۔
6۔ فنکشنس میں ترکیب (composition) کیا ہے ؟
7۔ مکرر فنکشن (recursive function) کس طرح کام کرتا ہے ؟
8۔ کسی فنکشن کی وضاحت کرتے وقت کن نکات کو نوٹ کرنا چاہئے ؟

حصہ - IV

مندرجہ ذیل سوالات کے جوابات دیں: (5 مارکس)

1۔ ایک مثال کے ساتھ فنکشن کی مختلف اقسام کی وضاحت کریں۔
2۔ ایک مثال کے ساتھ متغیرات کے دائرہ کار (scope of variable) کی وضاحت کریں۔
3۔ درج ذیل بلٹ ان فنکشنس کی وضاحت کریں۔

(a) id()
(b) chr()
(c) round()
(d) type()
(e) pow()

4۔ دو اعداد کا L.C.M تلاش معلوم کرنے کے لئے پائیتھن کا کوڈ لکھیں۔
5۔ ایک مثال کے ساتھ مکرر فنکشن کی وضاحت کریں۔

کتب حوالہ 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*

12 CS UNIT 7 CDR Set
Monday, April 17, 2023 4:07:24 PM

اکائی II

باب 8

اسٹرنگز اور اسٹرنگ کو برتنا STRINGS AND STRING MANIPULATION

سیکھنے کے مقاصد

اس باب کی تکمیل کے بعد، طالب علم اس قابل ہو جائے گا

- متن پر کارروائی کرنے کا طریقہ جان لیں گے۔
- پائیتھن میں مختلف اسٹرنگ فنکشنس کو سمجھ لیں گے۔
- اسٹرنگ کو فارمیٹ کرنے کا طریقہ جان لیں گے۔
- اسٹرنگ سلائزنگ کے بارے میں جان لیں گے۔
- حقیقی دنیا میں اسٹرنگز اپلی کیشن کے بارے میں جان لیں گے۔

8.1 تعارف

اسٹرنگ پائیتھن میں ڈیٹا کی قسم ہے، جو حروف کی صف (array) کو ہینڈل کرنے کے لئے استعمال ہوتی ہے۔

اسٹرنگ unicode characters کی ایک ترتیب ہے جو حروف، اعداد، یا کسی مخصوص علامت کا ایک مجموعہ ہوسکتا ہے جس میں سنگل، ڈبل یا اس سے بھی ٹرپل ("") quote سے گھرے ہوئے ہوتے ہیں۔

مثال:

‘Welcome to learning Python’

“Welcome to learning Python”

‘‘‘ “Welcome to learning Python” ’’’

پائیتھن میں، strings غیر منقولہ (immutable) ہیں۔ اس کا مطلب ہے، ایک بار جب آپ سٹرنگ کی وضاحت کریں گے تو اس کو عمل درآمد کے دوران تبدیل نہیں کیا جاسکتا ہے۔

8.2 اسٹرنگ کی تشکیل Creating Strings

جیسا کہ ہم پہلے ہی سیکھ چکے ہیں، سنگل یا ڈبل یا ٹرپل واوین quote("") کا استعمال کرکے پائیتھن میں ایک string تیار کی جاسکتی ہے۔ اکہرے واوین ' quote کا اسٹرنگ ایک اور اکہرے واوین quote کی علامت نہیں لے سکتی۔ کیوں کہ مرتب (compiler) کو یہ نہیں معلوم کہ اسٹرنگ کو کہاں سے شروع اور ختم کرنا ہے۔ اس مسئلے کو حل کرنے کے لئے آپ کو دوہرے واوین " quotes کا استعمال کرنا ہوگا۔ اسٹرنگز جس میں دوہرے واوین quote ہوتے ہیں، ان کی وضاحت تہرے واوین """ (Triple quotes) کے اندر ہونی چاہئے۔ triple quotes کے ساتھ اسٹرنگ کی وضاحت سے کثیر خطی اسٹرنگس (multiline strings) بنانے میں مدد ملتی ہے۔

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:29 PM

مثال

```
#A string defined within single quotes
>>> print (‘Greater Chennai Corporation’)
        Greater Chennai Corporation

#single quoted string defined within single quotes
>>> print ('Greater Chennai Corporation's student')
        SyntaxError: invalid syntax

#A string defined within double quotes
>>>print (“Computer Science”)
        Computer Science

#double quoted string defined within double quotes
>>> print ('" "Computer Science" "')
        "Computer Science"

#single and double quoted multiline string defined within triple quotes
>>> 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 ایک اسٹرنگ میں علامتوں تک رسائی حاصل کرنا (Accessing characters in a string)

ایک بار جب آپ اسٹرنگ کی وضاحت کرتے ہیں تو، پائیتھن اس کے ہر علامت (character) کے لئے ایک اشاری قیمت (index value) مختص کرتا ہے۔ان اشاری قیمتوں کو بہ الفاظ دیگر subscript (نیچے چھوٹے حروف میں) کہا جاتا ہے جو اسٹرنگ تک رسائی حاصل کرنے اور جوڑتوڑ کرنے کے لئے استعمال ہوتے ہیں۔ subscript مثبت یا منفی سالمی اعداد ہو سکتے ہیں۔

مثبت سبسکرپٹ 0 کو پہلی علامت اور n-1 کو آخری حرف کے لئے تفویض کیا گیا ہے، جہاں n اسٹرنگ میں حروف کی تعداد ہے۔ الٹی ترتیب میں آخری حرف سے پہلے حرف کو تفویض کردہ منفی اشاریہ -1 سے شروع ہوتا ہے۔

مثال

اسٹرنگ String	S	C	H	O	O	L
مثبت سب اسکرپٹ Positive subscript	0	1	2	3	4	5
منفی سب اسکرپٹ Negative subscript	-6	-5	-4	-3	-2	-1

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:29 PM

مثال 1 :

دئے گئے اسٹرنگ میں ہر علامت کو اپنے مثبت سب اسکرپٹ کے ساتھ رسائی حاصل کرنے کے لئے پروگرام

```
str1 = input ("Enter a string: ")
index=0
for i in str1:
        print ("Subscript[",index,"] : ", i)
        index + = 1
```

Output

```
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
```

مثال 2 :

دئے گئے اسٹرنگ میں ہر علامت کو اپنے منفی سب اسکرپٹ کے ساتھ رسائی حاصل کرنے کے لئے پروگرام

```
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 اسٹرنگس میں ترمیم اور حذف کرنا (Modifying and deleting strings)

جیسا کہ آپ پہلے ہی سیکھ چکے ہیں، پائیتھن میں string غیر منقول (immutable) ہیں۔اس کا مطلب ہے، ایک بار جب آپ سٹرنگ کی وضاحت کر دیتے ہیں تو اس میں ترمیم یا حذف کرنے کی اجازت نہیں ہے۔اگر آپ اسٹرنگ میں ترمیم کرنا چاہتے ہیں تو، موجودہ سٹرنگ متغیر کو ایک نئی سٹرنگ قیمت تفویض کی جانی چاہئے۔

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:30 PM

مثال

```
>>> 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
```

مذکورہ مثال میں، اسٹرنگ متغیر str1 کو بیان 1 میں "How are you" کے ساتھ تفویض کیا گیا ہے۔ اگلے بیان میں، ہم اسٹرنگ کے پہلے حرف کو 'A' کے حرف سے اپ ڈیٹ کرنے کی کوشش کرتے ہیں۔ لیکن پائیتھن اپ ڈیٹ کی اجازت نہیں دے گا اور یہ ٹائپ error ظاہر کرتا ہے۔

اس مسئلے پر قابو پانے کے لئے، آپ موجودہ سٹرنگ کی قیمت کے لئے ایک نئی سٹرنگ ویلیو کی وضاحت کر سکتے ہیں۔ پائیتھن موجودہ سٹرنگ پر نئی سٹرنگ کو مکمل طور پر اوور رائٹ (overwrite) کرتا ہے۔

مثال

```
>>> str1="How are you"
>>> print (str1)
        How are you
>>> str1="How about you"
>>> print (str1)
        How about you
```

عام طور پر پائیتھن اپنے اسٹرنگس میں کسی قسم کی ترمیم کی حمایت نہیں کرتا ہے۔ لیکن، یہ کسی فنکشن میں کسی خاص علامت کے تمام واقعات کو تبدیل کرنے کے لئے ایک فنکشن replace() فراہم کرتا ہے۔

replace function کا عام فارمیٹ:

replace("char1", "char2")

replace، فنکشن char1 سے char 2 کے تمام واقعات کو بدل دیتا ہے۔

مثال:

```
>>> str1="How are you"
>>> print (str1)
        How are you
>>> print (str1.replace("o", "e"))
        Hew are yeu
```

ترمیم ہی کی طرح، پائیتھن ایک خاص علامت کو اسٹرنگ میں حذف (delete) کرنے کی اجازت نہیں دیتا۔
جبکہ آپ del کمانڈ کا استعمال کر کے پوری سٹرنگ متغیر کو ختم کر سکتے ہیں۔

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:30 PM

مثال 3 : اسٹرنگ میں ایک خاص علامت کو حذف کرنے کے لئے کوڈ لائن۔

```
>>> str1="How are you"
>>> del str1[2]
Traceback (most recent call last):
        File "<pyshell#7>", line 1, in <module>
            str1[2]
        TypeError: 'str' object doesn't support item deletion
```

مثال 4 : کوڈ لائن جو ایک اسٹرنگ متغیر کو حذف کرنے کے لئے کوڈ لائن۔

```
>>> 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>
            (str1)
        NameError: name 'str1' is not defined
```

8.5 اسٹرنگ آپریٹرز (String operators)

پائیتھن اسٹرنگ کے اعمال (operators) کے لئے ذیل کے آپریٹرز فراہم کرتا ہے۔ یہ آپریٹرز اسٹرنگ کے جوڑتوڑ (برتنے) میں مفید ہیں۔

(i) باہم ملانا (Concatenation) (+)

دو یا دو سے زیادہ strings کو باہم ملانے کو Concatenation کہا جاتا ہے۔ پیتھان میں پلس (+) آپریٹر strings کو ملانے کیلئے استعمال کیا جاتا ہے۔

مثال

```
>>> "welcome" + "Python"
'welcomePython'
```

(ii) منسلک کرنا (جوڑنا) (Append) (+ =)

کسی موجودہ اسٹرنگ کے اختتام پر مزید اسٹرنگ شامل کرنا append کہلاتا ہے۔ آپریٹر + = کو موجودہ سٹرنگ کے ساتھ ایک نئی سٹرنگ شامل کرنے کے لئے استعمال کیا جاتا ہے۔

مثال

```
>>> str1="Welcome to "
```


12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:30 PM

```
>>> str1+="Learn Python"
>>> print (str1)
```

Welcome to Learn Python

(iii) دہرانا (repeating) (*)

ضرب کا آپریٹر (*) متعدد بار میں اسٹرنگ کو ظاہر کرنے کے لئے استعمال ہوتا ہے۔

مثال

```
>>> str1="Welcome "
>>> print (str1*4)
```

Welcome Welcome Welcome Welcome

(iv) سٹرنگ کو کاٹنا (سلائسنگ) (String slicing)

سلائس ایک اصل سٹرنگ کی ایک ذیلی اسٹرنگ (substring) ہے۔ آپریٹر [ ] اور اشاریہ یا سب اسکرپٹ کی قیمتوں کو استعمال کر کے ذیلی اسٹرنگ کو اصل سٹرنگ سے لیا جا سکتا ہے۔ لہٰذا [ ] کو سلائسنگ آپریٹر کے نام سے بھی جانا جاتا ہے۔ سلائس آپریٹر کا استعمال کرتے ہوئے، آپ کو ایک اصل اسٹرنگ سے ایک یا ایک سے زیادہ ذیلی اسٹرنگس کاٹ سکتے ہیں۔

سلائس آپریشن کی عمومی شکل: (General format of slice operations)

str[start:end]

جس میں start ابتدائی اشاریہ ہے اور end اسٹرنگ میں کسی علامت کا انتہائی اشاریہ ہے۔ پائیتھن آخری درج کردہ اصل اشاریہ کی قیمت سے ایک قیمت کم لیتا ہے۔ مثال کے طور پر، اگر آپ کسی سٹرنگ کے پہلے 4 علامتوں کو سلائس کرنا چاہتے ہیں تو آپ کو 0 سے 5 تک بیان کرنا ہوگا۔ کیونکہ، پائیتھن صرف آخری قیمت کو n-1 کی طرح لیتا ہے۔

مثال I : ایک اسٹرنگ سے صرف ایک علامت کو سلائس کرنا

```
>>> str1="THIRUKKURAL"
>>> print (str1[0])
```

T

مثال II : ایک ذیلی اسٹرنگ سے اشارہ 0 سے 4 کو سلائس کرنا

```
>>> print (str1[0:5])
```

THIRU

مثال III : ایک ذیلی اسٹرنگ سے اشارہ 0 سے 4 کو سلائس کرنا، مگر ابتدائی اشاریہ کی وضاحت کئے بغیر۔

```
>>> print (str1[:5])
```

THIRU

مثال IV : ایک ذیلی اسٹرنگ سے اشارہ 0 سے 4 کو سلائس کرنا، مگر انتہائی اشاریہ کی وضاحت کئے بغیر۔

```
>>> print (str1[6:])
```

KURAL

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:30 PM

مثال V : ایک لوپ سے ایک ذیلی اسٹرنگ (substring) کو سلائس کرنے کے لئے پروگرام

```
str1="COMPUTER"
index=0
for i in str1:
        print (str1[:index+1])
        index+=1
```

Output

```
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) سٹرنگ کو سلائس کرتے وقت چلی جانے والی چال (stride) (Stride when slicing string)

سلائسنگ کے عمل کے دوران، آپ تیسری دلیل کو چال(stride) کی طرح مختص کر سکتے ہیں، جس میں اس سے پہلے حرف کو string سے بازیافت کرنے کے بعد آگے بڑھنے کے لئے حروف کی تعداد سے مراد ہوتا ہے۔ سلائڈ کی ڈیفالٹ ویلیو 1 ہے۔

مثال :

```
>>> 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
```

نوٹ: یاد رکھیں، پائیتھن آخری قیمت کو n-1 کے طور پر لیتا ہے۔

آپ منفی قیمت کو بطور stride (تیسری دلیل) بھی استعمال کر سکتے ہیں۔ اگر آپ کوئی منفی قیمت بتاتے ہیں تو، یہ معکوس ترتیب میں پرنٹ کرتا ہے۔

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:31 PM

مثال :

```
>>> str1 = "Welcome to learn Python"
>>> print(str1[::-2])
nhy re teolW
```

8.6 اسٹرنگ فارمیٹنگ آپریٹرز (String formatting operators)

اسٹرنگ فارمیٹنگ آپریٹر پائیتھن کی دلچسپ خصوصیات میں سے ایک ہے۔فارمیٹنگ آپریٹر % کو strings کی تعمیر کے لئے استعمال کرتا ہے، اسٹرنگس کے حصوں کی جگہ متغیر میں موجود ڈیٹا سے بدل دیا جاتا ہے۔

Syntax:

("String to be display with %val1 and %val2" %(val1, val2))

مثال

```
name = "Rajarajan"
mark = 98
print ("Name: %s and Marks: %d" %(name,mark))
```

Output

```
Name: Rajarajan and Marks: 98
```

8.7 علامتوں کو فارمیٹ کرنا (Formatting characters)

فارمیٹ کے لئے استعمال کی جانے والی علامتیں Format characters	USAGE استعمال
%c	علامتیں (نشانات)
%d (or) %i	علامت والی عشری سالم اعداد
%s	اسٹرنگ
%u	بغیر علامتوں والے عشری سالم اعداد
%o	آکٹل سالم اعداد (قاعدہ 8 والے)
%x or %X	چھ عشری سالم اعداد (خفی حروف x، a-f کی اور جلے حروف X، A-F کی نشان دہی کرتا ہے
%e or %E	قوت نمائی ترقیم
%f	تیرنے والے نقطہ دار اعداد (floating point numbers)
%g or %G	تیرنے والے نقطہ دار یا قوت نما والے چھوٹے اعداد

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:31 PM

پائیتھن میں escape کا سلسلہ (Escape sequence in python)

escape کا سلسلہ بیک سلیش (back slash) سے شروع ہوتا ہے اور اس کی مختلف ترجمانی (interpret) کی جاسکتی ہے۔ جب آپ کو string کی نمائندگی کرنے کے لئے واحد quote ' استعمال ہوتا ہے تو، String کے اندر موجود تمام اکہرے quote ' کو escape کرنا ہوگا۔ ایسا ہی معاملہ دوہرے quotes " کا بھی ہے۔

مثال

```
# String within triple quotes to display a string with single quote
>>> print ('''They said, "What's there?"''')
        They said, "What's there?"

# String within single quotes to display a string with single quote using escape sequence
>>> print ('They said, "What\'s there?"')
        They said, "What's there?"

# String within double quotes to display a string with single quote using escape sequence
>>> print ("They said, \"What's there?\"")
        He said, "What's there?"
```

پائیتھن سے تعاون کردہ escape sequency:

Escape Sequence	وضاحت DESCRIPTION
\newline	Backslash and newline ignored
\\	Backslash
\'	Single quote
\"	Double quote
\a	ASCII Bell
\b	ASCII Backspace
\f	ASCII Form feed
\n	ASCII Linefeed
\r	ASCII Carriage Return
\t	ASCII Horizontal Tab
\v	ASCII Vertical Tab
\ooo	Character with octal value ooo
\xHH	Character with hexadecimal value HH

8.8 فارمیٹ()فنکشن (The format( ) function)

اسٹرنگس کو فارمیٹ کرنے کے لئے format( ) function استعمال کیا جاتا ہے، جو نہایت ہی آسان اور طاقتور فنکشن ہے۔ علامت { } کو replacement fields یا placeholders کی طرح استعمال کیا جاتا تھا، اب ان کی جگہ format( ) فنکشن نے لے لی ہے۔

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:31 PM

مثال

```
num1=int (input("Number 1: "))
num2=int (input("Number 2: "))
print ("The sum of { } and { } is { }".format(num1, num2,(num1+num2)))
```

Out Put

```
Number 1: 34
Number 2: 54
The sum of 34 and 54 is 88
```

8.9 بلٹ ان اسٹرنگ فنکشنس (Built-in String functions)

اسٹرنگ کو برتنے کے لئے پیتھان درج ذیل built-in فنکشنس کی تائید کرتا ہے۔

Syntax	Description وضاحت	Example مثال

len(str)	اسٹرنگ کی طول (علامتوں کی تعداد) کو return کرتا ہے۔	>>> A="Corporation"
		>>> print(len(A))
		11

capitalize( )	اسٹرنگ کے پہلے حرف کو حرف جلی (Capitalize) کرنے کے لئے استعمال ہوتا ہے	>>> city="chennai"
		>>> print(city.capitalize())
		Chennai

center(width, fillchar)	اصلی اسٹرنگ جو کالم کی چوڑائی کو بیچ میں کر کے fillchar سے بھرنے تا کہ اس میں کوئی بھی علامت نہ رہے، اس سے اسٹرنگ کو return کرنے کے لئے استعمال ہوتا ہے۔	>>> str1="Welcome"
		>>> print(str1.center(15,'*'))
		********Welcome*******

find(sub[, start[, end]])	دئے گئے اسٹرنگ میں ذیلی اسٹرنگ کے پہلے بار واقع ہونے کو تلاش کرنے کے لئے استعمال ہوتا ہے۔ یہ اس اشاریہ کو return کرتا ہے، جہاں سے substring شروع ہوتی ہے۔ اگر string میں کوئی بھی substring نہ ہو تو یہ -1 کو return کرتا ہے۔	>>>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

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:31 PM

Syntax	وضاحت	Example مثال
isalnum( )	اگر string میں صرف حروف اور اعداد ہوں تو یہ True کو return کرتا ہے۔ اگر string میں مخصوص علامات جیسے _ ، @ ، # ، * ، وغیرہ۔	>>>str1='Save Earth' >>>str1.isalnum() ***False*** The function returns False as space is an alphanumeric character. >>>'Save1Earth'.isalnum() ***True***
isalpha( )	اگر string میں صرف حروف ہوں تو یہ True کو return کرتا ہے۔ ورنہ False کو return کرتا ہے۔	>>>'Click123'.isalpha() ***False*** >>>'python'.isalpha( ) ***True***
isdigit( )	اگر string میں صرف اعداد ہوں تو یہ True کو return کرتا ہے۔ ورنہ False کو return کرتا ہے۔	>>> str1='Save Earth' >>>print(str1.isdigit( )) ***False***
lower( )	string کے تمام حروف کی پوری نقل کرتا ہے جب string میں صرف حروف خفی (lower case) ہوں۔	>>>str1='SAVE EARTH' >>>print(str1.lower()) ***save earth***
islower( )	اگر string کے تمام حروف میں صرف حروف خفی (lower case) ہوں تو یہ True کو return کرتا ہے۔	>>> str1='welcome' >>>print (str1.islower( )) ***True***
isupper( )	اگر string کے تمام حروف میں صرف حروف جلی (upper case) ہوں تو یہ True کو return کرتا ہے۔	>>> str1='welcome' >>>print (str1.isupper( )) ***False***
upper( )	string کے تمام حروف کی پوری نقل کرتا ہے جب string میں صرف حروف جلی (upper case) ہوں۔	>>> str1='welcome' >>>print (str.upper( )) ***WELCOME***

title( )	title case میں ایک string کو return کرتا ہے۔	>>> str1='education department'
		>>> print(str1.title())
		Education Department

swapcase( )	اپنے مخالف حرف (خفی یا جلی) یا اس کا الٹ ہو تو اس کو تبدیل کر لیتا ہے۔	>>> str1="tAmiL NaDu"
		>>> print(str1.swapcase())
		TaMIl nAdU

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:31 PM

Syntax	Description وضاحت	Example مثال

count(str, beg, end)	دی گئی حد میں آنے والے ذیلی اسٹرنگ کی تعداد کو return کرتا ہے۔ یاد رکھیں کہ ذیلی اسٹرنگ ایک علامت بھی ہوسکتی ہے۔ حد (range) کی شروع اور آخر میں دلائل اختیاری ہیں۔ اگر یہ نہ دیا گیا ہو تو، پیتھان اس کو مکمل string میں تلاش کرتا ہے۔ تلاش ایک case sensitive ہے۔	>>> str1="Raja Raja Chozhan"
		>>> print(str1.count('Raja'))
		2
		>>> print(str1.count('r'))
		0
		>>> prin***t(str1.count('R'))***
		2
		>>> print(str1.count('a'))
		5
		>>> print(str1.count('a',0,5))
		2
		>>> print(str1.count('a',11))
		1

ord(char)	ایک علامت (character) کے ASCII کوڈ کو return کرتا ہے۔	>>> ch = 'A'
		>>> print(ord(ch))
		65
		>>> print(***ord('B'))***
		66

chr(ASII)	ایک علامت (character) جس کے ASCII کی وہ نمائندگی کرتا ہے، اس کو Return کرتا ہے۔	>>> ch=97
		>>> print(chr(ch))
		a
		>>> print(chr(87))
		W

8.10 ممبرشپ آپریٹرز (Membership operators)

اس بات کا تعین کرنے کے لئے کہ ایک اسٹرنگ، دوسری اسٹرنگ میں موجود ہے یا نہیں، اس کے لئے 'in' اور 'not in' آپریٹرز کو تار کے ساتھ استعمال کیا جاسکتا ہے۔ لہذا، ان آپریٹرز کو ممبرشپ آپریٹرز (membership operators) کہا جاتا ہے۔

مثال:

```
str1=input ("Enter a string: ")
str2="chennai"
if str2 in str1:
    print ("Found")
else:
    print ("Not Found")
```

Output : 1

```
Enter a string: Chennai G HSS, Saidapet
Found
```

Output : 2

```
Enter a string: Govt G HSS, Ashok Nagar
Not Found
```


12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:32 PM

مثال 8.11 اسٹرنگز کو استعمال کرنے والے پروگرام (Programs using Strings)

مثال 8.11.1 : پروگرام یہ جانچنے کے لئے کہ دی گئی اسٹرنگ **palindrome** (اُلٹا لفظ بھی بالکل وہی ہو، مثال : ساس، لعل وغیرہ) ہے یا نہیں۔

```
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")
```

Output : 1

```
Enter a string: malayalam
The given string = malayalam
The Reversed string = malayalam
Hence, the given string is Palindrome
```

Output : 2

```
Enter a string: welcome
The given string = welcome
The Reversed string = emoclew
Hence, the given string is not a palindrome
```

مثال 8.11.2: مندرجہ ذیل نمونہ کو ظاہر کرنے کے لئے پروگرام

```
*
* *
* * *
* * * *
* * * * *
str1=' * '
i=1
while i<=5:
    print (str1*i)
    i+=1
```

Output

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


12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:32 PM

مثال 8.11.3: دی گئی string میں حروف علت (vowels) اور حروف صحیح (consonants) کی تعداد ظاہر کرنے کے لئے پروگرام

```
str1=input ("Enter a string: ")
str2="aAeEiIoOuU"
v,c=0,0
for i in str1:
        if i in str2:
                v+=1
        else:
                c+=1
print ("The given string contains { } vowels and { } consonants".format(v,c))
```

Output

```
Enter a string: Tamilnadu School Education
The given string contains 11 vowels and 15 consonants
```

مثال 8.11.4 : ایک Abecedarian series (عناصر جو حروف تہجی کی ترتیب کے لحاظ سے ظاہر ہوتے ہیں) تیار کرنے کے لئے پروگرام

```
str1="ABCDEFGH"
str2="ate"
for i in str1:
        print ((i+str2),end='\t')
```

Output

```
Aate    Bate    Cate    Date    Eate    Fate    Gate    Hate
```

مثال 8.11.5 الفاظ جن میں سے حروف علت (vowels) خارج کرنا ہو، اس کو user سے ایک string قبول کرنے کے لئے پروگرام

```
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)
```

Output

```
Enter a String: Mathematical fundations of Computer Science
The string without vowels:  Mthmtcl fndtns f Cmptr Scnc
```


12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:32 PM

مثال 8.11.6 : ایک اسٹرنگ میں موجود characters کی تعداد کا شمار کرنے کے لئے پروگرام

```
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))
```

Out Put

```
Enter a String: Software Engineering
Enter a character to be searched: e
The given character e is occurs 3 times in the given string
```

یاد رکھنے کے لئے نکات Points to remember:

- سٹرنگ پائیتھن میں ڈیٹا کی قسم ہے۔
- اسٹرنگز ناقابل تبدیل ہیں(immutable)،اس کا مطلب ہے کہ ایک بار جب آپ سٹرنگ کی وضاحت کرتے ہیں، عمل درآمد کے دوران تبدیل نہیں کیا جاسکتا ہے۔
- triple quotes """ کے اندر وضاحت کے دوران کثیر خطی اسٹرنگس بنانے کی اجازت ملتی ہے۔
- اسٹرنگ میں، پائیتھن اس کے ہر کردار کے اشاریہ کے عدد کو مختص کرتا ہے، جسے سب سکرپٹ (subscript) کہا جاتا ہے۔
- سب سکرپٹ مثبت یا منفی سالم عدد ہوسکتا ہے۔
- سلائس ایک اہم اسٹرنگ کی ایک ذیلی اسٹرنگ ہے۔
- سلائسنگ آپریشن میں stride تیسری دلیل (third argument) ہے۔
- escape کی ترتیب backslash \ سے شروع ہوتی ہے اور اس کی مختلف طریقے سے ترجمانی (interpret) کی جاسکتی ہے۔
- اسٹرنگس کو فارمیٹ کرنے کے لئے format( ) function استعمال کیا جاتا ہے، جو نہایت ہی آسان اور طاقتور فنکشن ہے۔
- اس بات کا تعین کرنے کے لئے کہ ایک اسٹرنگ، دوسری اسٹرنگ میں موجود ہے یا نہیں، اس کے لئے 'in' اور 'not in' آپریٹرز کو تار کے ساتھ استعمال کیا جاسکتا ہے۔



عملی مشق (Hands on experience)

1۔ اسٹرنگ کی لمبائی معلوم کرنے کے لئے ایک پائیتھن کا پروگرام لکھیں۔

2۔ دی گئی اسٹرنگ میں ہر حرف کے واقع ہونے کی تعداد گننے کے لئے ایک پروگرام لکھیں۔

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:32 PM

3۔ ایک اسٹرنگ میں میں تمام لائنوں میں ایک ماقبل متن (prefix text) کو شامل کرنے کے لئے ایک پروگرام لکھیں۔

4۔ مخصوص چوڑائی کے دائیں طرف سالم اعداد کے ساتھ '*' کو پرنٹ کرنے کے لئے ایک پروگرام لکھیں۔

5۔ دیئے گئے اسٹرنگ کی مرئی تصویر (آئینہ کا عکس) (mirror image) بنانے کے لئے ایک پروگرام لکھیں۔
مثال کے طور پر، "wel" = "lew" ۔

6۔ اسٹرنگ میں دئے گئے تمام علامتوں (character) کو خارج کرنے کے لئے ایک پروگرام لکھیں۔

7۔ = + آپریٹر کو استعمال کرتے ہوئے ایک string سے دوسری string کو جوڑنے کے لئے ایک پروگرام لکھیں۔

8۔ دو strings کو ادل بدل کرنے کے لئے ایک پروگرام لکھیں۔

9۔ ()replace استعمال کرتے ہوئے ایک اسٹرنگ کی جگہ دوسری اسٹرنگ کو replace کرنے کے لئے ایک پروگرام لکھیں۔

10۔ دی گئی string میں علامتوں، الفاظ اور لائنوں کی تعداد شمار کرنے کے لئے ایک پروگرام لکھیں۔

Evaluation محاسبہ

حصہ - I

صحیح جواب کا انتخاب کریں۔ (1 مارک)

1۔ ذیل میں سے کون سا output مندرجہ ذیل پائیتھن کوڈ ہے ؟

```
str1="TamilNadu"
print(str1[::-1])
```

(a) Tamilnadu (b) Tmlau
(c) udanlimaT d) udaNlimaT

2۔ درج ذیل کوڈ کا output کیا ہوگا ؟

```
str1 = "Chennai Schools"
str1[7] = "-"
```

(a) Chennai-Schools (b) Chenna-School
(c) Type error (D) Chennai

3۔ مندرجہ ذیل میں سے کون سا آپریٹر باہم ملانے (Concatenation) کے لئے استعمال ہوتا ہے ؟
(a) + (b) & (c) * d) =

4۔ ٹرپل کوٹس میں ڈور کی وضاحت سے تخلیق کی اجازت ملتی ہے۔
(a) سنگل لائن سٹرنگز (single line strings) (c) ملٹی لائن سٹرنگز (multiline strings)
(c) ڈبل لائن اسٹرنگ (double line strings) (d) ایک سے زیادہ اسٹرنگز (multiple strings)

5۔ پائیتھن میں اسٹرنگز:
(a) تبدیل (Changeable) (b) تغیر پزیر (mutable)
(c) ناقابل تبدیلی (immutable) (d) لچکدار (flexible)

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:33 PM

6۔ درج ذیل میں سے سلائسنگ آپریٹر (Slicing operator) کون سا ہے؟

(a) { } (b) [ ] (c) < > (d) ( )

7۔ اسٹرائڈ stride کیا ہے ؟

(a) سلائیڈ آپریشن کی اشاریہ قیمت (index value) (b) سلائس آپریشن کی پہلی دلیل (argument)

(c) سلائس آپریشن کی دوسری دلیل(argument) (d) سلائس آپریشن کی تیسری دلیل(argument)

8۔ مندرجہ ذیل میں سے کون سا فارمیٹنگ کریکٹر (formatting character)، اوپری کیس (حروف جلی) میں قوت نمائی ترقیم پرنٹ کرنے کے لئے استعمال ہوتا ہے؟

(a) %e (b) %E (c) %g (d) %n

9۔ مندرجہ ذیل میں سے کون سا پلیس ہولڈرز (place holders) یا متبادل فیلڈ کے طور پر استعمال ہوتا ہے، جو format( ) function کے ساتھ تبدیل ہو جاتا ہے ؟

(a) { } (b) < > (c) + + (d) ^ ^

10۔ کسی اسٹرنگ کا سب اسکرپٹ یہ ہو سکتا ہے۔

(a) مثبت (b) منفی (c) (a) اور (b) دونوں (d) (a) یا (b)

حصہ-II

مندرجہ ذیل سوالات کے جوابات دیں۔ (2 نمبرات)

1۔ اسٹرنگ کیا ہے؟

2۔ کیا آپ پائیتھن کی string میں ترمیم (modify) کر سکتے ہیں ؟

3۔ آپ پائیتھن میں string کو کیسے حذف (delete) کریں گے ؟

4۔ مندرجہ ذیل پائیتھن کوڈ کی پیداوار کیا ہوگی ؟

```
str1 = "School"
print(str1*3)
```

5۔ سلائسنگ (slicing) کیا ہے؟

حصہ-III

درج ذیل سوالات کے جوابات دیں۔ (3 نمبرات)

1۔ دئے گئے نمونے کو ظاہر کرنے کے لئے پائیتھن کا ایک پروگرام لکھیں۔

```
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
```


12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:33 PM

2۔ مناسب مثال کے ساتھ پیروی کے بارے میں ایک مختصر لکھیں۔

capitalize( ) (a) swapcase( ) (b)

3۔ دیئے گئے پائیتھن پروگرام کا output کیا ہوگا ؟

```
str1 = "welcome"
str2 = "to school"
str3=str1[:2]+str2[len(str2)-2:]
print(str3)
```

4۔ format( ) کا استعمال کیا ہے؟ ایک مثال دیں۔

5۔ پائیتھن میں count( ) فنکشن کے بارے میں ایک نوٹ لکھیں۔

حصہ - IV

مندرجہ ذیل سوالات کے جوابات دیں۔ (5 نمبرات)

1۔ مناسب مثال کے ساتھ پائیتھن میں سٹرنگ آپریٹرز کے بارے میں وضاحت کریں۔

کتب حوالہ : 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.*

12 CS UNIT 8 CDR Set
Monday, April 17, 2023 4:55:33 PM

اکائی III

باب 9

فہرستیں، ٹوپلس، سیٹ اور لغت

LISTS, TUPLES, SETS AND DICTIONARY



سیکھنے کے مقاصد

اس باب کا مطالعہ کرنے کے بعد طلباء، اس کے قابل ہو جائیں گے۔

- پائیتھن میں ڈیٹا کی مختلف اقسام کے بنیادی تصورات جیسے لسٹ، ٹوپلس ،سیٹ اور لغت کو سمجھیں۔
- طرح طرح کے افعال کا استعمال کرتے ہوئے فہرست، ٹوپلس ، سیٹ اور لغت کے ساتھ کام کریں۔
- فہرست، ٹوپلس ،سیٹ اور لغات کا استعمال کر کے پائیتھن کے پروگرام لکھنا۔
- فہرست، ٹوپلس اور لغت کے مابین تعلقات کو سمجھیں۔

9.1 Lists کا تعارف (Introduction to List)

پائیتھن پروگرامنگ زبان میں data types کا مجموعہ ہوتا ہے، جس میں لسٹ، ٹیوپلز، سیٹ اور ڈکشنری شامل ہیں۔ پائیتھن میں ایک فہرست کو "sequence data type" کے طور پر جانا جاتا ہے۔ یہ اسکوائر بریکٹ [ ] کے ساتھ منسلک اقدار کا ایک ترتیب وار مجموعہ ہے۔ ایک فہرست کی ہر قیمت کو عنصر element کہا جاتا ہے۔ یہ کسی بھی قسم کی ہو سکتی ہے جیسے اعداد، علامتیں، اسٹرنگس اور یہاں تک کہ nested lists بھی ہوتے ہیں۔ عناصر کو تبدیل یا ادل بدل کیا جا سکتا ہے جس کا مطلب ہے کہ عناصر کو تبدیل، شامل کیا یا ہٹایا جا سکتا ہے۔ ہر عنصر فہرست میں کسی نہ کسی پوزیشن پر ٹکا ہوا ہے۔ کسی عنصر کی پوزیشن صفر سے شروع ہونے والی تعداد کے ساتھ ترتیب دی جاتی ہے جو کسی خاص عنصر کو ڈھونڈنے اور اسے حاصل کرنے کے لئے استعمال ہوتا ہے۔ چنانچہ List، صف (arrays) کی طرح ہیں، جو آپ نے گیارھویں جماعت میں سیکھا تھا۔

9.1.1 پائیتھن میں ایک فہرست بنانا (Create a List in Python)

پائیتھن میں، مربع بریکٹ [ ] کا استعمال کر کے ایک فہرست آسانی سے بنائی جاتی ہے۔ فہرست کے عناصر کو مربع بریکٹ [ ] میں واضح کرنا چاہئے۔ درج ذیل syntax فہرست کی تشکیل کی وضاحت کرتا ہے۔

Syntax:

Variable = [element-1, element-2, element-3 …… element-n]

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:01 PM

مثال

```
Marks = [10, 23, 41, 75]
Fruits = ["Apple", "Orange", "Mango", "Banana"]
MyList = [ ]
```

اوپر کی مثال میں، فہرست کے نشانوں میں چار سالمی اعداد عنصر کے طور پر ہیں۔ دوسری فہرست میں پھلوں کے چار string عنصر ہیں۔ تیسری ایک خالی فہرست (empty list) ہے۔ ایک فہرست کے عناصر کو ایک ہی قسم (homogenous) کے data کی ضرورت نہیں ہے۔ درج ذیل فہرست میں متعدد قسم کے عناصر (multiple type elements) شامل ہیں۔

Mylist = ["Welcome", 3.14, 10, [2, 4, 6]]

اوپر کی مثال میں Mylist میں عنصر کی طرح ایک اور list ہوتی ہے۔ اس طرح کی list کی قسم کو Nested list کہتے ہیں۔

ایک list جس کے اندر ایک اور list عنصر کی طرح شامل ہوتی ہے، اس کو nested list کہتے ہیں۔

9.1.2 فہرست کے عناصر تک رسائی (Accessing List elements)

پائیتھن فہرست کے ہر عنصر کے لئے ایک خودکار اشاریہ کی قیمت صفر سے شروع کرتا ہے۔ فہرست میں کسی عنصر تک رسائی کے لئے اشاریہ کی قیمت (Index value) کا استعمال کیا جاسکتا ہے۔ پائیتھن میں انڈیکس ویلیو ایک سالمی عدد ہے، جو مثبت یا منفی ہوسکتا ہے۔

مثال

Marks = [10, 23, 41, 75]

Marks	10	23	41	75
Index (Positive)	0	1	2	3
Index(Negative)	-4	-3	-2	-1

فہرست کے آغاز سے اشاریہ کی مثبت قیمت کا شمار ہوتا ہے اور منفی قیمت کا مطلب فہرست کے آخر سے پیچھے (یعنی اُلٹی ترتیب میں) شمار کرنا ہوتا ہے۔

کسی فہرست سے کسی عنصر تک رسائی کے لئے، فہرست کا نام لکھیں، اس کے بعد مربع بریکٹ [ ] میں بند عنصر کا اشاریہ بنائیں۔

Syntax:

List_Variable = [E1, E2, E3 ..... En]

print (List_Variable[index of a element])

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:01 PM

مثال : واحد عنصر تک رسائی

```
>>> Marks = [10, 23, 41, 75]
>>> print (Marks[0])
10
```

مندرجہ بالا مثال میں، پرنٹ کمانڈ 10 کو آؤٹ پٹ کے طور پر پرنٹ کرتا ہے، کیونکہ 10 کا انڈیکس صفر ہے۔

مثال: اُلٹی ترتیب میں عناصر تک رسائی حاصل کرنا

```
>>> Marks = [10, 23, 41, 75]
>>> print (Marks[-1])
  75
```

نوٹ

الٹی ترتیب میں کسی عنصر تک رسائی کے لئے منفی انڈیکس کا استعمال کیا جا سکتا ہے۔

(i) فہرست کے تمام عناصر تک رسائی حاصل کرنا (Accessing all elements of a list)

فہرست سے تمام عناصر تک رسائی حاصل کرنے کے لئے لوپس کا استعمال کیا جاتا ہے۔ لوپ کی ابتدائی قیمت صفر ہونی چاہئے۔ صفر 0 ایک فہرست کی ابتدائی اشاریہ قیمت ہے۔

Example

```
Marks = [10, 23, 41, 75]
i = 0
while i < 4:
        print (Marks[i])
        i = i + 1
```

Output

```
10
23
41
75
```

مذکورہ مثال میں، نشانوں کی فہرست میں چار سالمی اعداد (عناصر) شامل ہیں، یعنی، 10، 23، 41، 75۔ ہر عنصر کی ایک اشاریہ قیمت 0 سے ہوتی ہے۔ عناصر کی اشاریہ کی قیمت بالترتیب 0، 1، 2، 3 ہے۔ یہاں، جبکہ لوپ تمام عناصر کو پڑھنے کے لئے استعمال ہوتا ہے۔ لوپ کی ابتدائی قیمت صفر ہے، اور ٹیسٹ کی حالت i <4 ہے، جب تک کہ ٹیسٹ کی حالت true ہے، لوپ اس پر عمل کرتا ہے اور اس سے متعلق آؤٹ پٹ پرنٹ کرتا ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:01 PM

پہلی تکرار کے دوران، i کی قیمت 0 ہے، جہاں حالت true ہے۔اب، مندرجہ ذیل اسٹیٹمنٹ print(Marks[i]) پر عمل درآمد ہوتا ہے اور Marks[0] عنصر یعنی 10 کی قیمت کو پرنٹ کرتا ہے۔

اگلا بیان i = i + 1 کی قیمت کو 0 سے 1 تک بڑھا دیتا ہے۔اب، کنٹرول کی روانی ٹیسٹ کی حالت کو جانچنے کے لئے while statement میں منتقل ہو جاتی ہے۔اس عمل کو Marks لسٹ کے باقی عناصر کو پرنٹ کرنے تک کے لئے دہرایا جاتا ہے، یہاں تک کہ while loop کے جانچ کی حالت False نہ ہو جائے۔

مندرجہ ذیل جدول، لوپ پر عمل درآمد اور پرنٹ کی جانے والی قیمت کو ظاہر کرتی ہے۔

Iteration	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) اُلٹی اشاریہ سازی (Reverse Indexing)

پائیتھن فہرست عناصر (list elements) کیلئے الٹ یا منفی اشاریہ سازی کا اہل بناتا ہے۔اس طرح، پائیتھن مخالف ترتیب میں انڈیکس کی فہرست بناتا ہے۔ پائیتھن لسٹ میں آخری عنصر کی انڈیکس ویلیو 1- اور اس سے پہلے والے عنصر کے ل 2- سیٹ کرتا ہے۔اسے reverse indexing کہا جاتا ہے۔

مثال

```
Marks = [10, 23, 41, 75]
i = -1
while i >= -4:
        print (Marks[i])
        i = i + -1
```

Output

```
75
41
23
10
```

درج ذیل جدول مذکورہ پائیتھن کوڈنگ کے عمل کو ظاہر کرتا ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:02 PM

Iteration	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 فہرست کی لمبائی (List length)

پائیتھن میں len( ) function کسی فہرست کی لمبائی (یعنی، فہرست میں عناصر کی تعداد) تلاش کرنے کے لئے استعمال ہوتا ہے۔ عام طور پر، len( ) function ایک لوپ میں اوپری حد مقرر کرنے کے لئے فہرست کے تمام عناصر کو پڑھنے کے لئے استعمال ہوتا ہے۔ اگر ایک فہرست میں عنصر کی حیثیت سے ایک اور فہرست شامل ہوتی ہے تو، len( ) اندرونی لسٹ کو ایک عنصر کے طور پر واپس (return) کرتا ہے۔

مثال اکہرے عنصر تک رسائی حاصل کرنا

```
>>> MySubject = ["Tamil", "English", "Comp. Science", "Maths"]
>>> len(MySubject)
    4
```

مثال لوپ کا استعمال کر کے ایک فہرست میں عناصر کو ظاہر کرنے کا پروگرام

```
MySubject = ["Tamil", "English", "Comp. Science", "Maths"]
i = 0
while i < len(MySubject):
        print (MySubject[i])
        i = i + 1
```

Output

```
Tamil
English
Comp. Science
Maths
```

9.1.4 for loop استعمال کرنے والے عناصر تک رسائی حاصل کرنا (Accessing elements using for loop)

پائیتھن میں، ایک ایک کر کے لسٹ میں موجود تمام عناصر تک رسائی حاصل کرنے کے لئے *for* loop کا استعمال کیا جاتا ہے۔ یہ بالکل اسی طرح کی ہے جیسے دیگر پروگرامنگ زبان جیسے C++ میں کلیدی الفاظ (keyword) جیسے for کا استعمال کیا جاتا ہے۔

Syntax:

```
for index_var in list:
    print (index_var)
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:02 PM

یہاں، index_var، فہرست میں ہر عنصر کی اشاریہ کی قیمت کی نمائندگی کرتا ہے۔ پائیتھن "for" بیان کو انگریزی میں اس طرح پڑھتا ہے۔"

"For (every) element in (the list of) list and print (the name of the) list items"

مثال:

```
Marks=[23, 45, 67, 78, 98]
for x in Marks:
        print( x )
```

Output

```
23
45
67
78
98
```

مندرجہ بالا مثال میں، مارکس کی فہرست میں 5 عناصر ہیں۔ ہر عنصر کو 0 سے 4 تک اشاریہ دیا جاتا ہے۔ پائیتھن **for loop** اور **print** کو انگریزی میں اس بیانیہ طرح پڑھتا ہے۔

"For (every) element (represented as x) in (the list of) Marks and print (the values of the) elements".

9.1.5 فہرست کے عناصر کو تبدیل کرنا (Changing list elements)

پائیتھن میں، فہرستیں تبدیل (mutable) کی جاسکتی ہیں، جس کا مطلب ہے کہ ان کو تبدیل کیا جاسکتا ہے۔ ایک لسٹ عنصر یا عناصر کی حد کو آسان اسائنمنٹ آپریٹر (=) استعمال کر کے تبدیل کیا جاسکتا ہے۔

Syntax:

List_Variable [index of an element] = Value to be changed
List_Variable [index from : index to] = Values to changed

جس میں ***index from***، انڈیکس کی حد کا ابتدائی اشاریہ ہے۔ ***index to*** اشاریہ میں بالائی حد ہوتی ہے، جسے حد میں شامل نہیں کیا جاتا ہے۔ مثال کے طور پر، اگر آپ **range[0:5]** طے کرتے ہیں تو اس کا مطلب، پیتھان عناصر کے اشاریہ کی طرح صرف **0** سے **4** لیتا ہے۔ لہٰذا، اگر آپ عناصر کی حد 1 سے 4 تک اپ ڈیٹ کرنا چاہتے ہیں تو، اسے **[1:5]** کے طور پر بیان کیا جانا چاہئے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:02 PM

مثال 9.1 آخری قیمت کو update/change کرنے کے پیتھان کا پروگرام

```
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)
```

Output:

```
MyList elements before update...
2
4
5
8
10
MyList elements after updation...
2
4
6
8
10
```

مثال 9.2 قیمتوں کی حد کو update/change کرنے کے پیتھان کا پروگرام

```
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)
```

Output

```
List Odd numbers...
1
3
5
7
9
List Even numbers...
2
4
6
8
10
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:02 PM

9.1.6 ایک فہرست میں مزید عناصر کا اضافہ کرنا
(Adding more elements in a list)

پائیتھن میں موجودہ list میں ایک اکہرے عنصر کو شامل کرنے کے لئے append( ) فنکشن اور ایک سے زیادہ عناصر کو کو ایک ہی عنصر کو شامل کرنے کے لئے extend( ) فنکشن کا استعمال کیا جاتا ہے۔

Syntax:

List.append (element to be added)
List.extend ([elements to be added])

extend( ) فنکشن میں، کثیر عناصر کو مربع بریکٹ [ ] کے اندر فنکشن کے دلائل (argument) کی طرح بیان کیا جانا چاہئے۔

مثال

```
>>> Mylist=[34, 45, 48]
>>> Mylist.append(90)
>>> print(Mylist)
        [34, 45, 48, 90]
```

مندرجہ بالا مثال میں، Mylist تین عناصر کے ساتھ بنائی گئی ہے۔ بیان >>>Mylist.append(90) کے ذریعے، ایک اضافی قیمت 90 شامل کی جاتی ہے، مندرجہ ذیل پرنٹ بیان فہرست MyList میں موجود تمام عناصر کو ظاہر کرتا ہے۔

مثال

```
>>> Mylist.extend([71, 32, 29])
>>> print(Mylist)
        [34, 45, 48, 90, 71, 32, 29]
```

مذکورہ کوڈ میں، extend( ) فنکشن، ایک سے زیادہ عناصر کو شامل کرنے کے لئے استعمال کیا جاتا ہے، پرنٹ بیان، اضافی عناصر کو شامل کرنے کے بعد فہرست کے تمام عناصر کو ظاہر کرتا ہے۔

9.1.7 ایک فہرست میں عناصر کو داخل کرنا
(Inserting elements in a list)

جیسا کہ آپ پہلے ہی سیکھ چکے ہیں، پیتھان میں append( ) فنکشن کو فہرست میں مزید عناصر شامل کرنے کے لئے استعمال کیا جاتا ہے۔ لیکن، اس میں ایک فہرست کے آخر میں عناصر شامل ہیں۔ اگر آپ اپنی مطلوبہ پوزیشن پر کسی عنصر کو شامل کرنا چاہتے ہیں تو، آپ insert( ) فنکشن استعمال کر سکتے ہیں۔ insert( ) فنکشن کا استعمال ایک لسٹ کی کسی بھی پوزیشن پر عنصر شامل کرنے کے لئے کیا جاتا ہے۔

Syntax:

List.insert (position index, element)

مثال

```
>>> 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']
```

مندرجہ بالا مثال میں، insert( ) فنکشن، ایک نئے عنصر 'Ramakrishnan' کو انڈیکس ویلیو 3 میں،، یعنی چوتھی پوزیشن پر داخل کرتا ہے۔ ایک خاص جگہ پر موجودہ عناصر کے درمیان ایک نیا عنصر داخل کرتے وقت، موجودہ عنصر، دائیں طرف ایک پوزیشن کو منتقل ہو جاتا ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:03 PM

9.1.8 ایک فہرست سے عناصر کو حذف کرنا (Deleting elements from a list)

لسٹ سے عنصر کو حذف کرنے کے دو طریقے ہیں، جیسے۔ del اسٹیٹمنٹ اور remove( ) فنکشن۔ del statement کا استعمال معلوم عناصر کو حذف کرنے کے لئے استعمال کیا جاتا ہے، جبکہ remove( ) function کا استعمال، اس فہرست کے عناصر کو حذف کرنے کے لئے ہوتا ہے اگر اس کا انڈیکس نامعلوم ہو۔ del statement پوری فہرست کو حذف کرنے کے لئے بھی استعمال کیا جاسکتا ہے۔

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

مثال

```
>>> MySubjects = ['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> print (MySubjects)
        ['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> del MySubjects[1]
>>> print (MySubjects)
        ['Tamil', 'Telugu', 'Maths']
```

مندرجہ بالا مثال میں، فہرست MySubjects چار عناصر کے ساتھ بنائی گئی ہے۔ print statement فہرست کے تمام عناصر کو ظاہر کرتا ہے۔ >>> del MySubjects[1] بیان میں، ایک عنصر کو حذف کرنا ہے، جس کی اشاریہ کی قیمت 1 ہے اور مندرجہ ذیل پرنٹ، فہرست کے بقیہ عناصر کو ظاہر کرتا ہے۔

مثال

```
>>> del MySubjects[1:3]
>>> print(MySubjects)
        ['Tamil']
```

مندرجہ بالا کوڈز میں، >>> del MySubjects[1:3] دوسرے اور تیسرے عناصر کو فہرست سے حذف (delete) کر دیتی ہیں۔ اشاریہ کی بالائی حد [ ] کے اندر بیان کی گئی ہے، اسے پائیتھن 1- کی طرح لے گا۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:03 PM

مثال

```
>>> del MySubjects
>>> print(MySubjects)
Traceback (most recent call last):
File "<pyshell#9>", line 1, in <module>
print(MySubjects)
NameError: name 'MySubjects' is not de<ned
```

یہاں، **>>>del MySubjects** مکمل طور پر **MySubjects** کو فہرست سے حذف کر دیتا ہے۔ جب آپ عناصر کو پرنٹ کرنے کی کوشش کرتے ہیں تو، پائیتھن ایک **error** ظاہر کرتا ہے کیونکہ فہرست کی وضاحت نہیں کی گئی ہے۔ جس کا مطلب ہے، **MySubjects** کی فہرست مکمل طور پر حذف کر دی گئی ہے۔

جیسا کہ پہلے ہی بیان ہو چکا ہے کہ اگر انڈیکس کی قیمت معلوم نہیں ہو تو، **remove( )** فنکشن ایک یا ایک سے زیادہ عناصر کو حذف کرنے کے لئے بھی استعمال کیا جا سکتا ہے۔ **remove( )** فنکشن کے علاوہ، **pop( )** فنکشن بھی دیئے گئے انڈیکس ویلیو کا استعمال کرتے ہوئے کسی عنصر کو ڈیلیٹ کرنے کے لئے استعمال کیا جا سکتا ہے۔ اگر فہرست نہیں دی جاتی ہے تو **pop( )** فنکشن کسی فہرست کے آخری عنصر کو حذف اور واپس کرتا ہے۔ **clear( )** فنکشن کا استعمال فہرست میں موجود تمام عناصر کو حذف کرنے کے لئے کیا جاتا ہے، یہ صرف عناصر کو حذف کرتا ہے اور فہرست کو برقرار رکھتا ہے۔ یاد رکھیں، **del** بیان پوری فہرست کو حذف کر دیتا ہے۔

Syntax:

List.remove(element) # to delete a particular element
List.pop(index of an element)
List.clear( )

مثال

```
>>> 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']
```

مندرجہ بالا مثال میں، **MyList** کو تین سالم اعداد اور تین اسٹرنگ عناصر کے ساتھ تشکیل دیا گیا ہے، مندرجہ ذیل پرنٹ بیان فہرست میں موجود تمام عناصر کو ظاہر کرتا ہے۔ **>>>MyList.remove(89)** بیان میں، عنصر **89** کو فہرست سے خارج کرتا ہے اور پرنٹ بیان باقی عناصر کو ظاہر کرتا ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:03 PM

مثال

```
>>> MyList.pop(1)
      34
>>> print(MyList)
      [12, 'Kannan', 'Gowrisankar', 'Lenin']
```

مذکورہ کوڈ میں، pop( ) فنکشن کسی خاص عنصر کو اپنی انڈیکس ویلیو کا استعمال کرتے ہوئے ڈیلیٹ کرنے کے لئے استعمال ہوتا ہے، جیسے ہی عنصر کو ڈیلیٹ کیا جاتا ہے، pop( ) فنکشن عنصر کو دکھاتا ہے جو حذف ہو جاتا ہے۔ pop( ) فنکشن کا استعمال، فہرست سے صرف ایک عنصر کو حذف کرنے کے لئے کیا جاتا ہے۔ یاد رکھیں، del بیان ایک سے زیادہ عناصر کو حذف کرتا ہے۔

مثال

```
>>> MyList.clear( )
>>> print(MyList)
      [  ]
```

مذکورہ کوڈ میں، clear( ) فنکشن صرف عناصر کو ہٹاتا ہے اور فہرست کو برقرار رکھتا ہے۔ جب آپ اس فہرست کو پرنٹ کرنے کی کوشش کرتے ہیں جو پہلے ہی صاف (clear) ہو چکی ہے تو، خالی [ ] کسی بھی عناصر کے بغیر ظاہر ہوتا ہے، جس کا مطلب ہے کہ فہرست خالی ہے۔

9.1.9 List اور range ( ) فنکشن (List and range ( ) function)

range( ) ایک ایسا فنکشن ہے جو پائیتھن میں قیمتوں کی ایک سیریز تیار کرنے کے لئے استعمال ہوتا ہے۔ range( ) فنکشن کا استعمال کرتے ہوئے، آپ قیمتوں کے سلسلوں (series) کے ساتھ فہرست تشکیل دے سکتے ہیں۔ range( ) فنکشن میں تین دلائل (arguments) ہیں

Syntax of range ( ) function:

range (start value, end value, step value)

جس میں

- start value - سلسلہ کی ابتدائی قیمت ۔ زیرو ڈیفالٹ ابتدائی قیمت ہے۔
- end value - سلسلہ کی اوپری حد ۔ پائیتھن انتہائی قیمت کی بالائی حد 1- کے طور پر لیتے ہیں۔
- step value - یہ ایک اختیاری دلیل ہے، جو مختلف قیمتوں کے وقفے کو پیدا کرنے کے لئے استعمال کی جاتی ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:04 PM

مثال : 10 تک صحیح اعداد کو پیدا کرنا

```
for x in range (1, 11):
        print(x)
```

Output

```
1
2
3
4
5
6
7
8
9
10
```

مثال : پہلے 10 جفت اعداد کو پیدا کرنا

```
for x in range (2, 11, 2):
      print(x)
```

Output

```
2
4
6
8
10
```

(i) قیمتوں کے سلسلہ کی ایک فہرست بنانا (Creating a list with series of values)

range( ) فنکشن کا استعمال کرتے ہوئے، آپ قیمتوں کے سلسلہ کے ساتھ ایک فہرست تشکیل دے سکتے ہیں۔ range( ) فنکشن کے نتیجے کو لسٹ میں تبدیل کرنے کے لئے، ہمیں list( ) نامی ایک اور فنکشن کی ضرورت ہے۔

list( ) فنکشن، range( ) کو فہرست کے result میں لاتا ہے۔

Syntax:

List_Varibale = list (range ( ))

نوٹ

list( ) فنکشن، پائیتھن میں فہرست (list) بنانے کے لئے بھی استعمال ہوتا ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:04 PM

مثال

```
>>> Even_List = list(range(2,11,2))
>>> print(Even_List)
[2, 4, 6, 8, 10]
```

مذکورہ کوڈ میں، list( ) فنکشن، range( ) کو Even_List elements کی طرح لیتی ہے۔ اس طرح، Even_List elements کی فہرست میں پہلے پانچ جفت اعداد کے عنصر ہوتے ہیں۔

اسی طرح، ہم range( ) فنکشن کا استعمال کرتے ہوئے قیمتوں کی کسی بھی سیریز کو تشکیل دے سکتے ہیں۔ مندرجہ ذیل مثال میں وضاحت کی گئی ہے کہ پہلے 10 طبیعی اعداد کا مربع کیسے بنایا جاتا ہے۔

مثال: پہلے 10 طبیعی اعداد کا مربع معلوم کرنا

```
squares = [ ]
for x in range(1,11):
    s = x ** 2
    squares.append(s)
print (squares)
```

مذکورہ پروگرام میں، "squares" نام کی ایک خالی فہرست تیار کی گئی ہے۔ اس کے بعد، for loop، 1 سے 10 تک کے طبیعی اعداد تیار کرتا ہے۔ اس کے لئے range( ) فنکشن استعمال ہوتا ہے۔ لوپ کے اندر، x کی موجودہ قیمت کو 2 کی قوت تک بڑھا دیا جاتا ہے اور متغیرات میں محفوظ کر دیا جاتا ہے۔ مربع کی ہر نئی قیمت کو "squares" میں شامل کر لیا جاتا ہے۔ آخر میں، پروگرام، آؤٹ پٹ کے طور پر درج ذیل قیمتوں کو ظاہر کرتا ہے۔

Output

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

List comprehensions 9.1.10

list comprehension عنصر کے سلسلہ پیدا کرنے کا ایک آسان طریقہ ہے جو کسی خاص حالت کو پورا کرتا ہے۔

Syntax:

List = [expression for variable in range]

مثال: list comprehension کے تصور کا استعمال کرتے ہوئے پہلے 10 طبیعی اعداد کا مربع معلوم کرنا

```
>>> squares = [ x ** 2 for x in range(1,11) ]
>>> print (squares)
```

Output:

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:04 PM

مندرجہ بالا مثال میں، اظہار x**2 ہر بار دوہرانے (iterate) کے بعد اس کا محاسبہ (evaluated) کیا جاتا ہے۔ قیمتوں کی سیریز تیار کرنے کا یہ ایک شارٹ کٹ طریقہ ہے۔

9.1.11 بعض اہم list functions (Other important list funcion)

Function	Description وضاحت	Syntax	Example مثال
copy ( )	List کی ایک کاپی کو return کرتا ہے۔	List.copy( )	MyList=[12, 12, 36] x = MyList.copy() print(x) **Output:** [12, 12, 36]
count ( )	List میں موجود مشابہ عناصر کے اعداد کو return کرتا ہے	List.count(value)	MyList=[36 ,12 ,12] x = MyList.count(12) print(x) **Output:** 2
index ( )	پہلے مکرر (recurring) عناصر کے اشاریہ کے قیمتوں کو return کرتا ہے۔	List.index(element)	MyList=[36 ,12 ,12] x = MyList.index(12) print(x) **Output:** 0
reverse ( )	List میں موجود عناصر کی ترتیب کو الٹ دیتا ہے۔	List.reverse( )	MyList=[36 ,23 ,12] MyList.reverse() print(MyList) **Output:** [12 ,23 ,36]
sort ( )	List میں عناصر کو ترتیب دیتا ہے۔	List.sort(reverse=True|False, key=myFunc)	
Both arguments are optional • If reverse is set as True, list sorting is in descending order. • Ascending is default. • Key=myFunc; "myFunc" - the name of the user defined function that specifies the sorting criteria. **Note:** sort( ) will affect the original list.		MyList=['Thilothamma', 'Tharani', 'Anitha', 'SaiSree', 'Lavanya'] MyList.sort( ) print(MyList) MyList.sort(reverse=True) print(MyList) **Output:** ['Anitha', 'Lavanya', 'SaiSree', 'Tharani', 'Thilothamma'] ['Thilothamma', 'Tharani', 'SaiSree', 'Lavanya', 'Anitha']	

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:04 PM

max( )	List میں موجود زیادہ سے زیادہ قیمت کو return کرتا ہے۔	max(list)	MyList=[21,76,98,23] print(max(MyList)) **Output:** 98
min( )	List میں موجود کم سے کم قیمت کو return کرتا ہے۔	min(list)	MyList=[21,76,98,23] print(min(MyList)) **Output:** 21
sum( )	List میں موجود قیمتوں کے حاصل جمع کو return کرتا ہے۔	sum(list)	MyList=[21,76,98,23] print(sum(MyList)) **Output:** 218

9.1.12 list کا استعمال کرتے ہوئے پروگرام (Programs using List)

پروگرام 1 : ایک ایسا پروگرام لکھئے جو 1 سے 20 تک کے اعداد کی ایک فہرست تیار کرے جو 4 سے تقسیم پذیر ہو۔

```
divBy4=[ ]
for i in range(21):
        if (i%4==0):
                divBy4.append(i)
print(divBy4)
```

Output

[0, 4, 8, 12, 16, 20]

پروگرام 2 : BRICS کے ممبر ممالک کی فہرست کی وضاحت کے لئے ایک پروگرام لکھیں۔ جانچئے کہ ایک ملک BRICS کا ممبر ملک ہے یا نہیں۔

```
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")
```

Output

Enter the name of the country: India
India is the member of BRICS

Output

Enter the name of the country: Japan
Japan is not a member of BRICS

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:04 PM

پروگرام 3 : چھ مضامین کے مارکس کو پڑھنے اور ہر مضمون میں حاصل کردہ مارکس کو پرنٹ کرنے اور کل مارکس ظاہر کرنے کے لئے پائیتھن کا پروگرام

```
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(j1+,subjects[j],marks[j]))
print("Total Marks = ", sum(marks))
```

Output

```
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
```

پروگرام 4 : ایک فہرست میں سے 5 اشیاء کی قیمتوں کو پڑھ کر اور پھر تمام قیمتوں کے حاصل جمع کو دکھانے اور اوسط معلوم کرنے کے لئے پائیتھن کا پروگرام

```
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))
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:04 PM

Output:

```
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
```

پروگرام5 : سالانہ 1 لاکھ سے زیادہ کمانے والے ملازمین کی تعداد گننے کے لئے پائیتھن کا پروگرام۔
ملازمین کی n تعداد کی ماہانہ تنخواہ دی گئی ہے۔

```
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))
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:05 PM

Output:

```
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
```

پروگرام 6 : 1 سے 10 تک کی حد میں نمبروں کی فہرست بنانے کے لئے ایک پروگرام لکھیں۔
پھر تمام جفت اعداد کو فہرست سے حذف کریں اور حتمی فہرست پرنٹ کریں۔

```
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)
```

Output

```
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]
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:05 PM

پروگرام 7 : فبونیکی سیریز میں پیدا کرنے کے لئے ایک پروگرام لکھیں اور اسے فہرست list میں اسٹور کریں۔ پھر تمام قیمتوں کا حاصل جمع تلاش کریں۔

```
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)
```

Output

```
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 ٹیوپلس (Tuples)

ٹیوپلس کا تعارف

ٹیوپلس متعدد قیمتوں پر مشتمل ہوتے ہیں جن کو کوما (,) سے الگ کیا جاتا ہے اور قوسین ( ) کے اندر بند ہوتا تھا۔ list ہی کی طرح Tupules ہے۔ List ہی کی طرح tuple ہے، ۔ List میں قیمتوں کو تبدیل کیا جاسکتا ہے، لیکن ٹیوپل (Tuple) میں نہیں۔

کیا تم جانتے ہو!

اصطلاح Tuple لاطینی لفظ سے ماخوذ ہے اور یہ تعداد کے تسلسل کو حاصل کرنے کی نمائندگی کرتی ہے۔

single(1), double(2), triple(3), quadruple(4), quintuple(5), sextuple(6), septuple(7), octuple(8), ..., n-tuple, ...,

9.2.1 فہرست پر ٹوپلس کے فوائد (Advantages of Tuples over list)

1۔ ایک list کے عناصر تبدیل (mutable) کئے جاسکتے ہیں، جب کہ Tuple میں پائے جانے والے عناصر تبدیل نہیں کئے جاسکتے (immutable)، tuples اور list میں یہی اہم فرق ہے۔

2۔ ایک List کے عناصر مربع بریکٹ [ ] میں بند ہوتے ہیں۔ لیکن، ٹیوپل کے عناصر قوسین ( ) کے ذریعہ بند ہوتے ہیں۔

3۔ فہرست کی بہ نسبت tuples تیز دہراتا (iterate) ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:05 PM

9.2.2 Tuples تیارکرنا (Creating Tuples)

ٹیوپلس بنانا لسٹ کے مشابہ ہے۔ایک فہرست میں،عناصر کومربع بریکٹ [ ] کے اندر بیان کیا جاتا ہے،جبکہ ٹیوپل میں،وہ قوسین( ) کے ذریعہ منسلک ہوسکتے ہیں۔ٹیوپل کے عناصر کی وضاحت، قوسین کے بغیر بھی کی جاسکتی ہے۔چاہے عناصر کوقوسین کے اندر بیان کیا جائے یا قوسین کے بغیر، اس سے اس کے فنکشن میں کوئی فرق نہیں پڑتا۔

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

مثال

```
>>> 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) Tuple( ) فنکشن کا استعمال کرتے ہوئے ٹیوپل تیارکرنا (Creating tuples using tuple( ) function)

ایک list سے tuples تیارکرنے کے لئے tuple( ) فنکشن کا استعمال کیا جاتا ہے۔ جب آپ فہرست سے ایک tuple تشکیل دیتے ہیں تو، عناصر کومربع بریکٹ [ ] کے اندر بند کیا جانا چاہئے۔

Syntax:

Tuple_Name = tuple([list elements])

Example

```
>>> MyTup3 = tuple( [23, 45, 90] )
>>> print(MyTup3)
        (23, 45, 90)
>>> type (MyTup3)
        <class 'tuple'>
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:05 PM

نوٹ

پائتھان کے شئے کے data type کو جاننے کے لئے Type( ) فنکشن استعمال کیا جاتا ہے۔

(ii) اکہرے عنصر کو تشکیل دینا (Creating single element tuple)

ایک اکہرے عنصر کے ساتھ ٹیوپل بنانے کے دوران، عنصر کے آخر میں کوما , شامل کریں۔ کوما , کی غیر موجودگی میں، پائیتھن عنصر کو ایک عام ڈیٹا ٹائپ سمجھ لے گا، ٹیوپل نہیں۔ اکہرے عنصر کے ساتھ ٹیوپل تیار کرنا "Singleton" ٹیوپل کہلاتا ہے۔

مثال

```
>>> MyTup4 = (10)
>>> type(MyTup4)
        <class 'int'>
>>> MyTup5 = (10,)
>>> type(MyTup5)
        <class 'tuples'>
```

9.2.3 ایک ٹیوپل میں قیمتوں تک رسائی (Accessing values in a Tuple)

List کی طرح، ٹیوپل کے ہر عنصر کا ایک انڈیکس نمبر صفر 0 سے شروع ہوتا ہے۔ انڈیکس نمبر کے استعمال سے ٹیوپل کے عناصر تک آسانی سے رسائی حاصل کی جاسکتی ہے۔

مثال

```
>>> 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)
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:05 PM

9.2.4 Tuple کو اپ ڈیٹ اور ڈیلیٹ کرنا (Update and Delete Tuple)

جیسا کہ آپ جانتے ہیں کہ ایک ٹیوپل غیر منقولہ ہے، لہذا ٹیوپل میں موجود عناصر کو تبدیل نہیں کیا جا سکتا۔ ٹیوپل میں قیمتوں کو ردوبدل کرنے کے بجائے، دو ٹیوپلس کو جوڑنا یا مکمل ٹیوپل کو حذف کرنا ممکن ہے۔

مثال

```
# Program to join two tuples    # دو ٹوپلس کو جوڑنے کے لئے پروگرام
Tup1 = (2,4,6,8,10)
Tup2 = (1,3,5,7,9)
Tup3 = Tup1 + Tup2
print(Tup3)
```

Output

```
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

پورے ٹیوپل کو حذف (delete) کرنے کے لئے، **del** کمانڈ استعمال کی جا سکتی ہے۔

Syntax:

del tuple_name

مثال

```
Tup1 = (2,4,6,8,10)
print("The elements of Tup1 is ", Tup1)
del Tup1
print (Tup1)
```

Output:

```
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
```

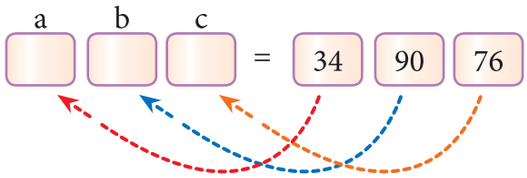
نوٹ کریں، مندرجہ بالا کوڈ میں **print statement** عناصر کو پرنٹ کرتا ہے۔ اس کے بعد، **del** بیان پورے **tuple** کو حذف کر دیتا ہے۔ جب آپ حذف شدہ **tuple** کو پرنٹ کرنے کی کوشش کرتے ہیں تو، پائیتھن **error** ظاہر کرتا ہے۔

9.2.5 ٹیوپل تفویض (Tuple Assignment)

tuple assignment، پائیتھن میں ایک طاقتور خصوصیت ہے۔ اسائنمنٹ آپریٹر کے بائیں جانب ایک متغیر کو تفویض کے دائیں جانب کی قیمتوں کو تفویض کرنے کی اجازت دیتا ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:06 PM

ہر قیمت کو اس کے متعلق متغیر کی طرف تفویض کیا گیا ہے۔



مثال

```
>>> (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
```

نوٹ کریں، جب آپ کسی ٹیوپل کی قیمت کو تفویض کرتے ہیں تو، اس بات کو یقینی بنائیں کہ اسائنمنٹ آپریٹر کے دونوں اطراف کے قیمتوں کی تعداد یکساں ہو، ورنہ پائیتھن ایک **error** بتائے گا۔

9.2.6 ٹیوپلس کے متعدد (کثیر) قیمتوں کو return کرنا (Returning multiple values in Tuples)

فنکشن ایک وقت میں صرف ایک ہی قیمت **return** کرسکتا ہے، لیکن پائیتھن ایک فنکشن سے ایک سے زیادہ قیمت کو **return** کرتے ہیں۔ پائیتھن کئی قیمتوں کو گروپ کرتا ہے اور ان کو ایک ساتھ **return** کرتا ہے۔

مثال : ایک فہرست میں زیادہ سے زیادہ اور کم سے کم قیمتوں کو return کرنے کے لئے پروگرام

```
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)
```

Output:

```
Maximum value =  99
Minimum value =  1
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:06 PM

9.2.7 نیسٹڈ ٹیوپلس (Nested tuples)

پائیتھن میں، ایک ٹیوپل کی وضاحت دوسرے ٹیوپل کے اندر کی جاسکتی ہے۔ نیسٹڈ ٹیپل کہا جاتا ہے۔ nested ٹیوپل میں، ہر ٹیوپل کو عنصر سمجھا جاتا ہے۔ nested ٹیوپل پر مشتمل تمام عناصر تک رسائی حاصل کرنے کے لئے for لوپ کارآمد ہوگا۔

مثال :

```
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)
```

Output:

```
('Vinodini', 'XII-F', 98.7)
('Soundarya', 'XII-H', 97.5)
('Tharani', 'XII-F', 95.3)
('Saisri', 'XII-G', 93.8)
```

نوٹ:

List میں استعمال ہونے والے تمام functions کا اطلاق Tuples کے لئے بھی ہوسکتا ہے۔

9.2.8 ٹیوپلس استعمال کرنے والے پروگرام (Programs using Tuples)

پروگرام 1 : ٹیوپل کی تفویض کا استعمال کرتے ہوئے دو قیمتوں کو تبدیل کرنے کے لئے ایک پروگرام لکھیں۔

```
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)
```

Output:

```
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
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:06 PM

پروگرام 2 : کسی ایسے فنکشن کا استعمال کرتے ہوئے ایک پروگرام لکھیں جو اس دائرے کا رقبہ اور اس کے احاطہ کو return کرے، جس کا نصف قطر ایک دلیل (argument) کی طرح گزرے۔ ٹیوپل تفویض کو استعمال کرتے ہوئے دو قیمتیں معلوم کریں۔

```
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)
```

Output:

```
Enter the Radius: 5
Area of the circle =  78.5
Circumference of the circle =  31.400000000000002
```

پروگرام 3 : ایک پروگرام لکھیں جس میں مثبت اور منفی اعداد کی ایک list ہو۔ ایک نیا ٹیوپل بنائیں۔ جس میں list سے صرف مثبت اعداد ہوں۔

```
Numbers = (5, -8, 6, 8, -4, 3, 1)
Positive = ( )
for i in Numbers:
    if i > 0:
        Positive += (i, )
print("Positive Numbers: ", Positive)
```

Output:

```
Positive Numbers:  (5, 6, 8, 3, 1)
```

9.3 سیٹ (مجموعے) (Sets)

تعارف

پائیتھن میں، data type کی ایک اور قسم کا مجموعہ set کہلاتا ہے۔ ایک سیٹ نقل (duplicate) کے بغیر ایک تغیر پذیر (mutable) اور بے ترتیب عناصر کا مجموعہ ہے۔ اس کا مطلب ہے کہ ایک سیٹ کے اندر موجود عناصر کو دہرایا نہیں جا سکتا۔ اس خصوصیت کو ممبر شپ کی جانچ اور نقل (duplicate) عناصر کو ختم کرنے کا عمل شامل ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:06 PM

9.3.1 سیٹ (مجموعہ) بنانا (Creating a set)

ایک سیٹ کے اندر تمام عناصر کو comma کے ساتھ الگ کر کے { } کے اندر تیار کیا گیا ہے۔ set( ) فنکشن کو پائیتھن میں سیٹ بنانے کے لئے بھی استعمال کیا جاتا ہے۔



Syntax:

Set_Variable = {E1, E2, E3 …….. En}

مثال

```
>>> 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}
```

مندرجہ بالا مثالوں میں، سیٹ S1 مختلف اقسام کے عناصر کے ساتھ تخلیق کیا گیا ہے، جس میں نقلی قیمتیں (duplicate values) نہیں ہیں۔ جبکہ سیٹ S2 کو نقلی قیمتوں کے ساتھ تشکیل دیا گیا ہے، لیکن پائیتھن اس نقل کے درمیان صرف ایک عنصر کو قبول کرتا ہے۔ جس کا مطلب ہے کہ پائیتھن نے نقلی قیمتوں کو ہٹا دیا، کیونکہ پائیتھن کے سیٹ میں نقلی (duplicate) عناصر نہیں ہو سکتے۔

نوٹ

جب آپ ایک سیٹ سے عناصر کو پرنٹ کرتے ہیں تو، پائیتھن مختلف ترتیب میں قیمتوں کو ظاہر کرتا ہے۔

9.3.2 list یا tupule کو استعمال کرتے ہوئے ایک سیٹ بنانا (Creating Set using List or Tuple)

ایک list یا Tuple کو set( ) فنکشن کا استعمال کر کے سیٹ (مجموعہ) کے طور پر تبدیل کیا جا سکتا ہے۔ یہ بہت آسان طریقہ کار ہے۔ پہلے آپ کو ایک list یا Tuple بنانا ہو، اس کے بعد اس کے متغیر کو set( ) فنکشن میں دلیل (argument) کے طور پر رکھیں۔

مثال

```
MyList=[2,4,6,8,10]
MySet=set(MyList)
print(MySet)
```

Output:

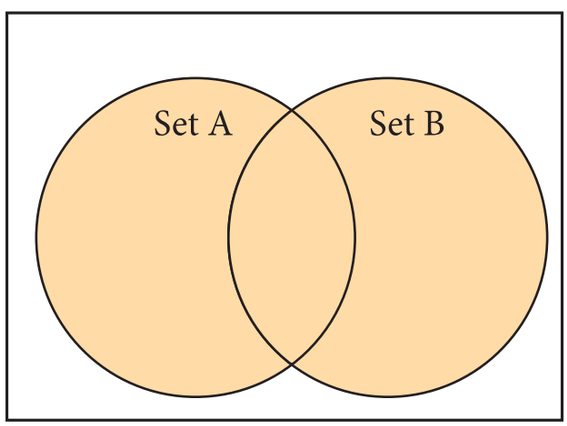
```
{2, 4, 6, 8, 10}
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:06 PM

9.3.3 سیٹ پر اعمال (Set operations)

جیسا کہ آپ نے ریاضی میں سیکھا ہے کہ پیتھان بھی مجموعہ پر اعمال جیسے اتحاد(union)، تقاطع (Intersection)، فرق (difference) اور تشاکلی فرق (symmetric difference) کی بھی حمایت کرتا ہے۔

(i) یونین (اتحاد) (Union) :اس میں دو یا زیادہ سیٹوں کے تمام عناصر شامل ہیں۔



پائیتھن میں، آپریٹر | دو سیٹوں کو جوڑنے کے لئے استعمال ہوتا ہے۔ پائیتھن میں union( ) فنکشن، دو سیٹوں کو جوڑنے کے لئے بھی استعمال کیا جاتا ہے۔

مثال : union operator کے ذریعہ دو مجموعوں کا اتحاد (union) کے لئے پروگرام

```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)
```

Output:

```
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

مثال : union function کے ذریعہ دو مجموعوں کا اتحاد (union) کے لئے پروگرام

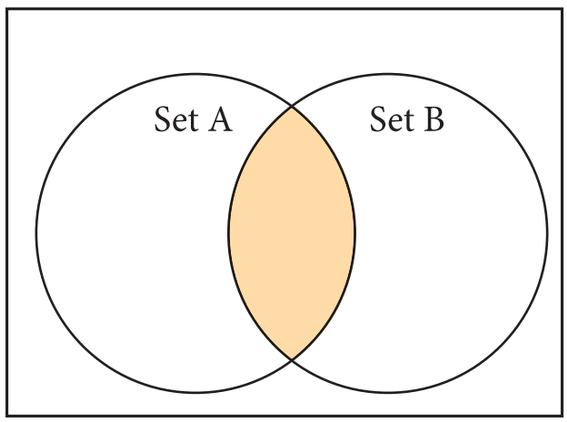
```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
set_U=set_A.union(set_B)
print(set_U)
```

Output:

```
{'D', 2, 4, 6, 8, 'B', 'C', 'A'}
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:07 PM

(ii) تقاطع (Intersection) : اس میں دوسیٹوں کے مشترک (common) عناصر شامل ہیں۔



پیتھان میں دومجموعوں کے تقاطع کے لئے آپریٹر & کواستعمال کیا جاتا ہے۔پیتھان میں intersection( ) فنکشن کواستعمال کرکے دومجموعوں کا تقاطع کیا جاتا ہے۔

مثال: intersection operator کواستعمال کرکے دومجموعوں کے تقاطع کا پروگرام

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```

Output:

```
{'A', 'D'}
```

مثال: intersection function کواستعمال کرکے دومجموعوں کے تقاطع کا پروگرام

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.intersection(set_B))
```

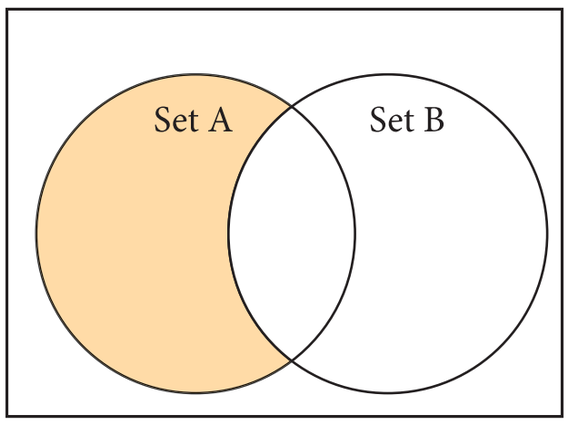
Output:

```
{'A', 'D'}
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:07 PM

(iii) فرق یا تفریق (Difference)

اس میں وہ تمام عناصر شامل ہیں جو پہلے سیٹ میں ہیں(فرض کیجئے set A) ، لیکن دوسرے سیٹ میں نہیں(فرض کیجئے set B)۔



تفریق کی علامت (-) آپریٹر کو پائیتھن میں سیٹ آپریشن میں فرق کرنے کے لئے استعمال کیا جاتا ہے۔( ) difference فنکشن کو difference operation کرنے کے لئے استعمال کیا جاتا ہے۔

مثال: minus operator کو استعمال کر کے دو مجموعوں کی تفریق کا پروگرام

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
```

Output:

```
{2, 4}
```

مثال: minus function کو استعمال کر کے دو مجموعوں کی تفریق کا پروگرام

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.difference(set_B))
```

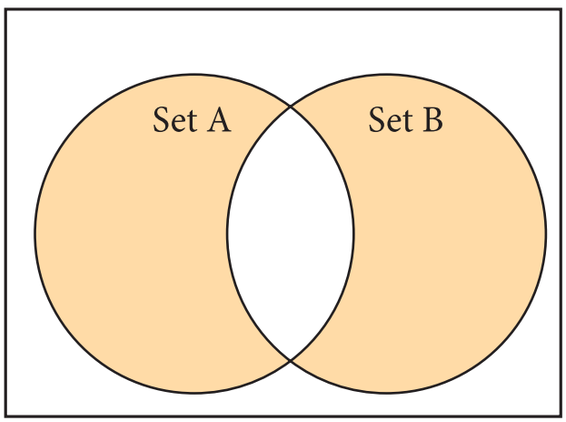
Output:

```
{2, 4}
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:07 PM

(iv) تشاکلی فرق (symmetric difference)

اس میں وہ تمام عناصر شامل ہیں جو دونوں سیٹ میں موجود ہیں (فرض کیجئے سیٹ A اور B) لیکن وہ عناصر نہیں جو دو سیٹوں میں مشترک ہیں۔



پیتھان میں Caret (^) آپریٹر کو تشاکلی فرق معلوم کے لئے استعمال کیا جاتا ہے۔ symmetric_difference ( ) فنکشن کو اسی آپریشن کے لئے استعمال کیا جاتا ہے۔

مثال: Caret operator استعمال کرتے ہوئے دو سیٹوں کے تشاکلی فرق معلوم کرنے کے لئے پروگرام

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A ^ set_B)
```

Output:

```
{2, 4, 'B', 'C'}
```

مثال: symmetric difference function استعمال کرتے ہوئے دو سیٹوں کے تشاکلی فرق معلوم کرنے کے لئے پروگرام

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.symmetric_difference(set_B))
```

Output:

```
{2, 4, 'B', 'C'}
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:07 PM

9.3.4 سیٹ کا استعمال کرتے ہوئے پروگرام (Programs using Sets)

پروگرام 1 : پروگرام جو عدد اولیٰ کا ایک سیٹ اور دوسرے سیٹ میں جفت اعداد کا ایک سیٹ تیار کرتا ہے۔ اتحاد(union)، تقاطع (Intersection)، فرق (difference) اور تشاکلی فرق (symmetric difference) کے اعمال کو واضح کرنا۔

مثال

```
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:
                ifi%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))
```

Output:

```
Even Numbers:  {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
Prime Numbers:  {2, 3, 4, 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 ڈکشنری (Dictionaries)

تعارف

پائیتھن میں، dictionary ، عناصر کا ایک مخلوط مجموعہ ہے۔ دیگر ڈیٹا کے مجموعہ کی قسم جیسے list یا Tuple، کے بجائے dictionary کی قسم اپنے عناصر کے ساتھ ایک کلید (key) کو محفوظ کرتی ہے۔ پائیتھن میں ڈکشنری کی keys کو جدا کرنے کے لئے colon (:) استعمال کیا جاتا ہے، جبکہ comma کو عناصر کے لئے جداکار (separator) کے طور پر استعمال کیا جاتا ہے۔ اہم قیمت کی جوڑیوں (key value pairs) کو { } کے ساتھ بند ہوتے ہیں۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:07 PM

Syntax of defining a dictionary:

```
Dictionary_Name = {    Key_1: Value_1,
                       Key_2:Value_2,
                       ……..
                       Key_n:Value_n
                       }
```

Dictionary میں موجود key ، حروف جلی اور حروف خفی حساس ہوتی ہے، اور یہ کسی بھی صحیح (Valid) پائیتھن کی قسم کی ہوسکتی ہے۔

9.4.1 ایک ڈکشنری تیار کرنا (Creating a Dictionary)

```
# Empty dictionary
    Dict1 = { }

# Dictionary with Key
    Dict_Stud = { RollNo: 1234, Name:Murali, Class:XII, Marks:451}
```

9.4.2 ڈکشنری کی تفہیم (Dictionary Comprehensions)

پائیتھن میں، ڈکشنری پیدا کرنے کا ایک اور طریقہ comprehension ہے۔ ذیل میں ایسی ڈکشنری تخلیق کرنے کا syntax ہے۔

Syntax

```
Dict = { expression for variable in sequence [if condition] }
```

if شرط اختیاری ہے اور if کی وضاحت کی گئی ہے تو یہ بھی ڈکشنری کی تخلیق کا ایک طریقہ ہے۔ سلسلہ میں صرف اس کی قیمتوں کا محاسبہ، اس اظہار کے ذریعے کیا جاتا ہے جو شرط کو پورا کرتا ہے۔

مثال

```
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 کسی ڈکشنری سے عناصر تک رسائی، اضافہ، ترمیم اور حذف کرنا۔ (Accessing, Adding, Modifying and Deleting elements from a Dictionary)

List اور Tuples ہی کی طرح، ایک ڈکشنری سے تمام عناصر تک رسائی ہوتی ہے۔ سادہ print فنکشن کا استعمال کر کے تمام عناصر تک رسائی حاصل کی جاتی ہے۔ اگر آپ کسی خاص عنصر تک رسائی حاصل کرنا چاہتے ہیں تو، key کے ساتھ مربع بریکٹ [ ] بھی استعمال کیا جاسکتا ہے۔

100 95 75 25 5 0

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:07 PM

مثال: ایک ڈکشنری میں محفوظ کردہ تمام قیمتوں تک رسائی کا پروگرام

```
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'])
```

Output:

```
{'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
```

نوٹ کریں کہ، پہلا پر **print** بیان لغت کی تمام قیمتوں کو پرنٹ کرتا ہے۔ دوسرے بیانات صرف مخصوص قیمتوں کو ہی چھاپ رہے ہیں جو مربع بریکٹ [ ] کے اندر دی گئی ہیں۔

کسی موجودہ ڈکشنری میں، آپ **key** کے ساتھ قیمت کو تفویض کر کے مزید قیمتیں شامل کر سکتے ہیں۔ درج ذیل **syntax** کو ڈکشنری میں مزید عناصر شامل کرنے اور سمجھنے کے لئے استعمال کیا جاتا ہے۔

dictionary_name [key] = value/element

مثال : ڈکشنری میں ایک نئی قیمت شامل کرنے کا پروگرام

```
MyDict = {      'Reg_No': '1221',
                'Name' : 'Tamilselvi',
                'Schod' : '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'])
```

لغت میں کسی قیمت میں ترمیم کرنا عناصر کو شامل کرنے کے مترادف ہے۔ جب آپ کسی **key** کو کوئی قیمت تفویض کرتے ہیں تو، اس سے پرانی قیمت کو آسانی سے ختم کر دیا جائے گا۔

پائیتھن ڈکشنری میں، کسی خاص عنصر کو حذف کرنے کے لئے **del** کی ورڈ استعمال ہوتا ہے۔ **clear( )** فنکشن کسی لغت میں موجود تمام عناصر کو حذف کرنے کے لئے استعمال ہوتا ہے۔ ڈکشنری کو ختم کرنے کے لئے، آپ ڈکشنری کے نام کے ساتھ **del** مطلوبہ الفاظ استعمال کر سکتے ہیں۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:08 PM

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
```

مثال: ڈکشنری سے عناصر کو حذف کرنے کا پروگرام اور آخر میں ڈکشنری کو حذف کر دیں۔

```
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
```

Output:

```
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 list اور dictionary کے درمیان فرق (Difference between List and Dictionary)

(1) List عناصر کا ایک ترتیب وار مجموعہ ہے۔لیکن، dictionary ایک data کی ساخت ہے، جو ایک عنصر (key) کو دوسرے (value) کے ساتھ جوڑنے کے لئے استعمال ہوتا ہے۔

(2) انڈیکس کی قیمتوں کو کسی خاص عنصر تک رسائی کے لئے استعمال کیا جا سکتا ہے۔لیکن، ڈکشنری میں کلید (key)، اشاریہ کی نمائندگی کرتی ہے۔ یاد رکھیں، کلید اسٹرنگ کا ایک عدد ہو سکتا ہے۔

(3) List کو کسی قیمت کو دیکھنے کے لئے استعمال کیا جاتا ہے جبکہ ایک ڈکشنری کو ایک قیمت لینے اور دوسری قیمت تلاش کرنے کے لئے استعمال کی جاتی ہے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:08 PM

Points to remember: یادرکھنے کے لئے نکات

- پائیتھن پروگرامنگ زبان میں data types کا مجموعہ ہوتا ہے، جس میں لسٹ، ٹیوپلز، سیٹ اور ڈکشنری شامل ہیں۔
- ایک list کو "sequence data type" کے نام سے جانا جاتا ہے۔ ایک list کی ہر قیمت کو عنصر کہا جاتا ہے۔
- list کے عناصر کو مربع بریکٹ [ ] کے درمیان میں بیان کرنا چاہئے۔
- ہر عنصر کی ایک انوکھی قیمت ہوتی ہے جسے انڈیکس نمبر کہتے ہیں اور یہ صفر سے شروع ہوتی ہے۔
- پائیتھن اشاریہ کی حیثیت سے مثبت اور منفی قیمتوں کی اجازت دیتا ہے۔
- ایک list سے تمام عناصر تک رسائی حاصل کرنے میں لوپس کا استعمال کیا جاتا ہے۔
- for لوپ ایک ایک کر کے تمام عناصر تک رسائی حاصل کرنے کے لئے ایک موزوں لوپ ہے۔
- list میں مزید عناصر کو شامل کرنے کے لئے append( )، extend( ) اور insert( ) فنکشنس استعمال کیے جاتے ہیں۔
- del، remove( )، اور pop( ) کا استعمال list سے عناصر کو حذف کرنے کے لئے کیا جاتا ہے۔
- range( ) فنکشن، قیمتوں کی ایک سیریز کو تیار کرنے کے لئے استعمال ہوتا ہے۔
- tuples متعدد قیمتوں پر مشتمل ہوتے ہیں جو comma کے ذریعہ الگ ہو جاتے ہیں اور قوسین ( ) کے اندر بند ہوتے ہیں۔
- list کی بہ نسبت tuples میں دوہرانے (iterating) کا اثر بہت تیز ہے۔
- tuple( ) فنکشن کا استعمال list سے ٹیپلس بنانے کے لئے بھی کیا جاتا ہے۔
- اکہرے عنصر کے ساتھ tuple تیار کرنے کو "singleton" ٹیوپل کہا جاتا ہے۔
- ایک سیٹ (مجموعہ) نقل (duplicate) کے بغیر ایک تغیر پذیر (mutable) اور بے ترتیب عناصر کا مجموعہ ہے۔
- ایک سیٹ کے اندر تمام عناصر کو comma کے ساتھ الگ کر کے { } کے اندر تیار کیا گیا ہے۔
- ایک ڈکشنری عناصر کا ایک مخلوط مجموعہ ہے۔



Hands on Experience عملی مشق

1۔ list سے ڈپلیکیٹ کو ہٹانے کے لئے ایک پروگرام لکھیں۔

2۔ ایک ٹیپل میں زیادہ سے زیادہ قیمت کو print کرنے کے لئے ایک پروگرام لکھیں۔

3۔ while لوپ کا استعمال کرتے ہوئے، ایک tuples کے تمام اعداد کا حاصل جمع معلوم کرنے کے لئے ایک پروگرام لکھیں۔

4۔ ایک list میں موجود تمام جفت اعداد کا حاصل جمع معلوم کرنے کے لئے ایک پروگرام لکھیں۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:08 PM

5۔ لوپ کا استعمال کرتے ہوئے ایک list کو الٹنے کے لئے ایک پروگرام لکھیں۔

6۔ ایک list میں مخصوص مقام پر ایک قیمت کو داخل کرنے کے لئے ایک پروگرام لکھیں۔

7۔ اعداد 1 سے 50 تک کی ایک list تیار کریں جو یا تو 3 سے تقسیم پذیر ہوں یا 6 سے تقسیم پذیر ہوں، اس کے لئے ایک پروگرام لکھیں۔

8۔ اعداد 1 سے 20 تک کی حد میں ایک list تیار کریں۔ اس کے بعد اس list سے تمام اعداد کو حذف کریں جو 3 سے تقسیم پذیر ہوں۔

9۔ list میں ظاہر ہونے والی ایک قیمت جو متعدد بار واقع ہوتی ہے، اس کو شمار کرنے کے لئے ایک پروگرام لکھیں۔
ایسا کرنے کے لئے لوپ کا استعمال کریں۔

10۔ ایک پروگرام لکھیں جو ڈکشنری میں زیادہ سے زیادہ اور کم سے کم قیمت print کرے۔

TS1FQ

Evaluation محاسبہ

حصہ - I

صحیح جواب کا انتخاب کریں۔ (1 مارک)

1۔ data type کے مجموعہ کے سلسلے میں غیر موزوں کا انتخاب کریں۔

(a) List (b) Tuple (c) Dictionary (d) Loop

2۔ فرض کیجئے List 1= [2,4,6,8,10] ہو تو print(List1[-2]) کا نتیجہ اس طرح ہوگا۔

(a) 10 (b) 8 (c) 4 (d) 6

3۔ ایک list میں موجود عناصر کی تعداد کو شمار کرنے کے لئے درج ذیل میں سے کون سا فنکشن استعمال ہوتا ہے؟

(a) count() (b) find() (c)len() (d) index()

4۔ اگر List= [10,20,30,40,50] ہو تو List[2]=35 کا نتیجہ اس طرح ہوگا۔

(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۔ اگر List= [17,23,41,10] ہو تو List.append(32) کا نتیجہ اس طرح ہوگا۔

(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۔ موجودہ List میں ایک سے زیادہ عنصر شامل کرنے کے لئے مندرجہ ذیل میں سے کون سا پائیتھن کا فنکشن استعمال کیا جانا چاہئے؟

(a) append() (b) append_more() (c)extend() (d) more()

7۔ مندرجہ ذیل پائیتھن کوڈ کا نتیجہ کیا ہوگا؟

```
S=[x**2 for x in range(5)]
```


12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:09 PM

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]

8۔ پائیتھن میں type( ) فنکشن کا استعمال کیا ہے؟

(a) ایک Tuple بنانے کے لئے

(b) ایک tuple میں کسی عنصر کے قسم کو جاننے کے لئے

(c) پائیتھن آبجیکٹ کے data type کو جاننے کے لئے

(d) ایک list بنانے کے لئے

9۔ مندرجہ ذیل میں سے کون سا بیان درست نہیں ہے؟

(a) ایک list تبدیل پذیر (mutable) ہے۔

(b) ایک tuple غیر تبدیل پذیر (immutable) ہے۔

(c) append( ) فنکشن کسی عنصر کو شامل کرنے کے لئے استعمال ہوتا ہے۔

(d) ایک فہرست میں عناصر کو شامل کرنے کے لئے tuple میں extend( ) فنکشن استعمال کیا جاتا ہے۔

10۔ فرض کیجئے setA={3,6,9}، SetB={1,3,9} ہے۔ درج ذیل کمانڈ کے ٹکڑے (snippet) کا نتیجہ کیا ہوگا ؟

print(setA|setB)

(a) {3,6,9,1,3,9} (b) {3,9} (c) {1} (d) {1,3,6,9}

11۔ مندرجہ ذیل سیٹ آپریشن میں سے کس میں دونوں مجموعوں کے تمام عناصر شامل ہوں گے، مگر اُس میں نہیں جو دونوں مجموعوں میں مشترک ہوں۔

(a) Symmetric difference (b) Difference

(c) Intersection (d) Union

12۔ پائیتھن کی keys، میں ڈکشنری اس طرح ظاہر کی جاتی ہے۔

حصہ-II

مندرجہ ذیل سوالات کے جوابات دیں۔ (2 مارکس)

1۔ پائیتھن میں List کیا ہے؟

2۔ آپ الٹ ترتیب میں List کے عناصر تک کیسے رسائی حاصل کریں گے؟

3۔ مندرجہ ذیل پائیتھن کوڈ میں x کی قیمت کیا ہوگی؟

```
List1=[2,4,6[1,3,5]]
x=len(list1)
```

4۔ List میں del اور remove() فنکشن کے درمیان فرق کیجئے۔

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:09 PM

5۔ n عناصر کی تعداد کے ساتھ ایک tuple تیار کرنے کے لئے syntax لکھیں۔

6۔ پائیتھن میں مجموعہ (set) کیا ہے؟

حصہ - III

درج ذیل سوالات کے جوابات دیں (3 مارکس)

1۔ list کی بہ نسبت tuples کے فوائد کیا ہیں؟

2۔ sort( ) سے متعلق ایک مختصر نوٹ لکھیں۔

3۔ درج ذیل کوڈ کا نتیجہ کیا ہوگا؟

```
list = [2**x for x in range(5)]

print(list)
```

4۔ ایک مثال کے ساتھ ڈکشنری میں del اور clear( ) کے درمیان فرق کی وضاحت کریں۔

5۔ پائیتھن کے ذریعہ تائید کردہ مجموعہ کے اعمال (set operation) کی ایک فہرست بنائیے۔

6۔ list اور dictionary کے درمیان کیا فرق ہے؟ What are the difference between List and Dictionary?

حصہ - IV

مندرجہ ذیل سوالات کے جوابات دیں (5 مارکس)

1۔ ایک list میں عنصر داخل کرنے کے مختلف طریقے کیا ہیں؟ مناسب مثال کے ساتھ واضح کریں۔

2۔ range( ) کا مقصد کیا ہے؟ ایک مثال کے ساتھ واضح کریں۔

3۔ nested tuple کیا ہے؟ ایک مثال کے ساتھ واضح کریں۔

4۔ مناسب مثال کے ساتھ پائیتھن کے ذریعہ تائید کردہ مختلف مجموعہ کے اعمال (Set operationis) کی وضاحت کریں۔

References حوالہ جات

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.

12 CS UNIT 9 CDR Set
Monday, April 17, 2023 5:01:09 PM

اکائی III

باب 10

PYTHON CLASSES AND OBJECTS

پائیتھن، کلاسس اور آبجکٹس (اشیاء)



سیکھنے کے مقاصد:

اس باب کی تکمیل کے بعد، طالب علم اس قابل ہو جائے گا۔

- آبجیکٹ اوریئنٹڈ پروگرامنگ کے بنیادی تصورات جیسے: کلاسس، آبجیکٹ، کنسٹرکٹر اور ڈسٹرکٹر وغیرہ کو سمجھ سکے گا۔
- پائیتھن میں کلاسس اور آبجکٹس تیار کرنے کا علم حاصل کرے گا۔
- تعمیر کاروں (کنسٹرکٹر) کے ساتھ کلاسس بنائے گا۔
- کلاسوں کا استعمال کر کے پائیتھن میں پیچیدہ پروگرام لکھ سکے گا۔

10.1 تعارف Introduction

پائیتھن ایک آبجیکٹ اوریئنٹڈ پروگرامنگ زبان ہے۔ آبجیکٹ اوریئنٹڈ پروگرامنگ کی کلیدی خصوصیات کلاسس (Classes) اور آبجیکٹ (objects) ہیں۔ classes اور objects کے نظریاتی تصورات C++ سے بہت ملتے جلتے ہیں۔ لیکن، C++ کے مقابلے میں پائیتھن میں classes اور objects کی تیاری اور نفاذ بہت آسان ہے۔

کلاسس، پائیتھن کے عمارت کا مرکزی بلاک ہے۔ آبجیکٹ، ڈیٹا اور فنکشن کا مجموعہ ہے جو ان ڈیٹا پر عمل کرتا ہے۔ object کے لئے class ایک نمونہ (template) ہے۔ آبجیکٹ اوریئنٹڈ پروگرامنگ کے تصور کے مطابق، objects یا class یا کلاس متغیر (class variable) کے طور پر بھی کہا جاتا ہے۔ پائیتھن میں، ہر چیز ایک شے (object) ہے۔ مثال کے طور پر، تمام سالمی متغیرات جو ہم اپنے پروگرام میں استعمال کرتے ہیں وہ کلاس int. کا ایک object ہے۔ اسی طرح تمام سٹرنگ متغیرات بھی کلاس سٹرنگ کے objects ہیں۔

10.2 کلاسوں کی وضاحت کرنا (Defining classes)

پائیتھن میں، کلاس کی تعریف class keyword کے ذریعہ کی جاتی ہے۔ ہر طبقے کا ایک انوکھا نام ہوتا ہے جس کے بعد colon (:) آتا ہے۔

Syntax:

```
class class_name:
    statement_1
    statement_2
    ……
    ……
    statement_n
```


12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:38:58 PM

جس میں، ایک کلاس کی وضاحت میں موجود بیان(Statement)، ایک variable declaration، decision control، لوپ، یہاں تک کہ ایک function definition بھی ہوسکتا ہے۔ کلاس کے اندر واضح کردہ متغیرات کو "class variable" کہا جاتا ہے اور functions کو 'methods" کہا جاتا ہے۔ class variables اور methods دونوں کو ملا کر کلاس کے ممبرس کہا جاتا ہے۔ class variable کی رسائی objects یا class کے instance کے ذریعہ کی جاتی ہے۔ پائیتھن پروگرام میں کہیں بھی ایک کلاس کی وضاحت کی جاسکتی ہے۔

مثال : ایک کلاس کی وضاحت کے لئے پروگرام

```
class Sample:
    x, y = 10, 20          # class variables
```

مذکورہ کوڈ میں، کلاس کا نام sample ہے اور اس میں دو متغیرات x اور y ہیں جن کی ابتدائی قیمت بالترتیب 10 اور 20 ہے۔ کلاس کے اندر بیان کردہ قیمتوں تک رسائی کے لئے، آپ کو object یا کلاس کے instance کی ضرورت ہوگی۔

10.3 آبجیکٹس کی تیاری (creating objects)



ایک بار کلاس تیار ہونے کے بعد، آپ کو اس کلاس کا کوئی object یا instance بنانا چاہئے۔ آبجیکٹ بنانے کے عمل کو "class instantation" کہا جاتا ہے۔

Syntax:

Object_name = class_name( )

نوٹ کریں کہ class instantation، فنکشن کی ترقیم (function notation) کو استعمال کرتا ہے۔ یعنی class_name( ) with

10.4 کلاس ممبروں تک رسائی (Accessing Class members)

کسی بھی کلاس ممبر ، یعنی class variable یا طریقہ (function) تک رسائی حاصل کرنے کے لئے object کے ساتھ dot(.) operator استعمال کیا جاتا ہے۔

Syntax:

Object_name . class_member

مثال : ایک کلاس کی وضاحت کرنے اور اس کے member variables تک رسائی کے لئے پروگرام۔

```
class Sample:
    #class variables
    x, y = 10, 20
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
```


12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:38:59 PM

اوپر کے کوڈ میں کلاس کا نام sample ہے۔ کلاس کے اندر، ہم نے x اور y متغیر کو بالترتیب 10 اور 20 ابتدائی قیمت کے ساتھ تفویض کیا ہے۔ ان دونوں متغیرات کو کلاس متغیر یا کلاس کے ممبر متغیر کہا جاتا ہے۔ class instantation کے عمل میں، ہم نے کلاس کے ممبروں تک رسائی کے لئے ایک آبجیکٹ S تیار کیا ہے۔ پہلے دو پرنٹ بیانات (print statements) آسانی سے کلاس متغیر x اور y کی قیمت پرنٹ کرتے ہیں اور آخری پرنٹ بیان (last print statement) میں دو قیمتیں شامل کرتے ہیں اور result پرنٹ کرتے ہیں۔

10.5 کلاس کے طریقے (Class methods)

پائیتھن کلاس فنکشن یا طریقہ بہت معمولی فرق کے ساتھ عام فنکشن سے ملتا جلتا ہے کہ، کلاس کے طریقہ کار میں خود کو سب سے پہلے دلیل (argument) کو رکھنا چاہئے، جس کا نام self ہوتا ہے۔ جب ہم طریقہ کو کال کریں (چلائیں گے) تو اس دلیل کے لئے کوئی قیمت پاس کرنے کی ضرورت نہیں ہے۔ پائیتھن اپنی قیمت خود بخود فراہم کرتا ہے۔ یہاں تک کہ اگر کوئی طریقہ دلائل نہیں لیتا ہے، تو اس کی وضاحت خود کو پہلی دلیل جسے self کہا جاتا ہے، اس سے کی جانی چاہئے۔ اگر ایک طریقہ صرف ایک دلیل کو قبول کرنے کے لئے بیان کیا گیا ہے تو یہ اسے دو دلائل کے طور پر لے جائے گا یعنی self اور defined argument ۔

جب آپ کلاس کے اندر کلاس متغیر کا اعلان کرتے ہیں تو، class name اور dot operator کے ذریعہ طریقوں کو پہلے سے لگایا جانا چاہئے۔

نوٹ

کلاس کے اندر واضح کردہ بیانات کو مناسب طریقے سے انڈنٹ (indent) کرنا چاہئے۔

مثال 10.2: کلاس کا استعمال کرتے ہوئے کل اور اوسط مارکس معلوم کرنے کا پروگرام

```
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()
```

مذکورہ پروگرام میں، کلاس کی وضاحت کے بعد، ایک object، S تیار کی جاتی ہے۔ بیان S.process( ) فنکشن کو مطلوبہ آؤٹ پٹ حاصل کرنے کے لئے کہتا ہے۔

12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:38:59 PM

غور کریں کہ،ہم نے بالترتیب 45،91،71 کی قیمتوں کے ساتھ تین متغیروں کو مارک1، مارک2 اور مارک3 قرار دیا ہے۔ہم نے self argument کے ساتھ ایک طریقہ کی وضاحت کی ہے،جس کا نام process ہے۔ جس کا مطلب ہے کہ،ہم اس طریقہ کار میں کوئی قیمت نہیں گزاریں گے۔ پہلے، process method ، کلاس متغیر کی قیمتوں کو جوڑتا ہے، متغیر حاصل جمع میں نتیجہ کو محفوظ کرتا ہے، اوسط معلوم کرتا ہے اور نتیجہ کو ظاہر کرتا ہے۔

اس طرح مندرجہ بالا کوڈ مندرجہ ذیل آؤٹ پٹ کو دکھائے گا۔

Output

Total Marks = 207

Average Marks = 69.0

مثال : کلاس کو استعمال کر کے دیا گیا عدد طاق ہے یا جفت معلوم کرنے کے لئے پروگرام

```
class Odd_Even:
        even = 0                    #class varaiable
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.

Output 1

```
Enter a value: 4
4  is Even number
```

Output 2

```
Enter a value: 5
5  is Odd number
```

10.6 پائیتھن میں تعمیر کار(constructor)اور تباہ کار(destructor) (Constructor and destructor in Python)

constructor ایک خاص فنکشن ہوتا ہے، جو class کی کوئی object تیار ہونے پر خود بخود چلنے لگتا ہے۔ پائیتھن میں، ایک خاص فنکشن ہے جس کا نام "init" ہے جو بطور constructor کام کرتا ہے۔اس کا آغاز double underscore کے ساتھ ہونا چاہئے۔ یہ فنکشن ایک عام function کے طور پر کام کرے گا۔ لیکن فرق صرف اتنا ہے کہ جب یہ object تیار ہو جائے تو یہ خود بخود عمل میں آجاتا ہے۔اس فنکشن کی دلیل کے ساتھ یا بغیر تعریف کی جاسکتی ہے۔ یہ طریقہ کلاس متغیرات کو شروع کرنے کے لئے استعمال کیا جاتا ہے۔

12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:38:59 PM

General format of __init__ method (Constructor function)

```
def __init__(self, [args ……..]):
        <statements>
```

Example : Program to illustrate Constructor

```
class Sample:
        def __init__(self, num):
                print("Constructor of class Sample...")
                self.num=num
                print("The value is :", num)
S=Sample(10)
```

مذکورہ بالا کلاس "sample" میں صرف ایک کنسٹرکٹر ہے جس میں ایک دلیل (argument) ہے، اس کا نام num. ہے۔ جب کنسٹرکٹر کو چلایا جاتا ہے تو، پہلے پرنٹ اسٹیٹمنٹ، "Constructor of class sample....", پرنٹ کرتا ہے، پھر، کنسٹرکٹر کے passing value کو self.num تفویض کیا گیا ہے اور بالآخر دئے گئے سٹرنگ کے ساتھ یہ قیمت کو print کر دیتا ہے۔

مذکورہ بالا کنسٹرکٹر خود بخود عمل درآمد کرتا ہے، جب کوئی S object اصلی پیرامیٹر 10 کے ساتھ تشکیل دیا جاتا ہے۔ لہٰذا python ، درج ذیل output دے گا۔

Constructor of class Sample...

The value is : 10

class variable کو constructor کے اندر واضح کیا گیا ہے، اس سے کلاس میں تیار شدہ objects کی تعداد، کا شمار constructor کرتا ہے۔

مثال : تیار شدہ objects شمار کرنے کے لئے class variable کے وضاحت کا پروگرام

```
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)
```


12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:00 PM

مذکورہ پروگرام میں، کلاس متغیر num کلاس نمونے کی تینوں objects کے ساتھ مشترک ہے۔ اس کی ابتداء صفر 0 کردی جاتی ہے اور ہر بار جب کوئی object تیار ہوجاتا ہے تو num کی تعداد میں 1 کا اضافہ ہوتا ہے۔ چونکہ، تمام objects کو متغیر ساجھے داری (share) کرتا ہے، 1 کے اضافہ کے ساتھ num میں کی جانے والی تبدیلی، دیگر اشیاء میں بھی ظاہر ہوتی ہے۔ اس طرح مندرجہ بالا پروگرام ذیل کا output دیتا ہے۔

Output

```
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
```

نوٹ : کلاس متغیر (Class variable)، C++ میں static type کے مشابہ ہے۔

جب ایک object، اس کے دائرہ کار (scope) سے خارج ہوجاتی ہے تو Destructor ایک مخصوص طریقہ ہے، جو خود سے عمل در آمد ہوجاتا ہے۔ جب کوئی شے دائرہ کار سے خارج ہوجاتی ہے تو destructor خود بخود عمل میں لایا جاتا ہے۔ یہ constructor کے بالکل برعکس ہے۔ پائیتھن میں، ( )_del_ طریقہ destructor کے طور پر استعمال ہوتا ہے۔

مثال : ( )__del__ کے طریقہ کار کے بارے میں واضح کرنے کا پروگرام

```
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)
```

10.7 عوامی اور نجی ڈیٹا کے ممبران (Public and Private data members)

متغیرات جو کلاس کے اندر بیان ہوتی ہیں، وہ بطور ڈیفالٹ عوامی (public) ہوتی ہیں۔ ان متغیرات کو dot operator کے استعمال سے پروگرام سے رسائی جاسکتا ہے۔

ایک متغیر جو double underscore کے سابقہ میں ہوتا ہے، فطری طور پر نجی (private) بن جاتا ہے۔ ان متغیرات کو صرف کلاس کے اندر ہی حاصل کیا جاسکتا ہے۔

12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:00 PM

مثال : نجی اور عوامی تغیرات کی وضاحت کے لئے پروگرام۔

```
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)
```

مذکورہ پروگرام میں، n1 اور n2 دو class variables قرار دیئے گئے ہیں۔ متغیر n1 ایک عوامی متغیر ہے اور n2 نجی متغیر ہے۔ member display( ) کا طریقہ ان دونوں متغیروں کو منتقل کی گئی قیمتوں کو ظاہر کرنے کے لئے بیان کیا گیا ہے۔

کلاس کے اندر بیان کردہ print statement کامیابی کے ساتھ n1 اور n2 کی قیمتوں کو ظاہر کریں گے، حالانکہ کلاس متغیر n2 نجی ہے۔ کیونکہ، اس معاملے میں، n2 کلاس کے اندر بیان کردہ طریقہ کے ذریعہ کہا جاتا ہے۔ لیکن، جب ہم کلاس سے باہر n2 کی قیمت تک رسائی حاصل کرنے کی کوشش کرتے ہیں تو پیتھان error ظاہر کرتا ہے۔ کیونکہ، کلاس کے باہر سے نجی متغیر (private variable) تک رسائی حاصل نہیں کی جاسکتی ہے۔

Output

```
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'
```


12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:00 PM

10.8 classess اور objects کی وضاحت کے لئے پروگرام کا نمونہ

پروگرام 1 : ایک دائرہ کا رقبہ اور احاطہ معلوم کرنے کے لئے ایک پروگرام لکھئے۔

```
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())
```

Output:

```
Enter Radius: 5
The Area = 78.5
The Circumference = 31.400000000000002
```

پروگرام 2: دی گئی اسٹرنگ میں موجود جلی حروف (upper case)، خفی حروف (lower case)، حروف علت Vowels)، صحیح حروف (consonants) اور خالی جگہوں (spaces) کی تعداد کو ایک اسٹرنگ قبول کر کے print کرنے کے لئے ایک پروگرام لکھئے۔

```
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
```


12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:00 PM

```
                    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()
```

Output:

```
Enter a String:Welcome To Learn Computer Science
The given string contains...
4 Uppercase letters
25 Lowercase letters
12 Vowels
13 Consonants
4 Spaces
```

Points to remember یادرکھنے کے لئے نکات

- پائیتھن ایک آبجیکٹ اوریینٹڈ پروگرامنگ زبان ہے۔
- classes اور objects آبجیکٹ اوریینٹڈ پروگرامنگ کی کلیدی خصوصیات ہیں۔
- پائیتھن میں،کلیدی الفاظ key word کلاس استعمال کرکے ایک کلاس کی وضاحت کی جاتی ہے۔
- کلاس کے اندر بیان کردہ متغیرات کو"class variable" کہا جاتا ہےاورفنکشن کو "Methods" کہا جاتا ہے۔
- آبجیکٹ بنانے کے عمل کو' "class instantation کہا جاتا ہے۔
- تعمیرکار (Constructor) ایک خاص فنکشن ہوتا ہے جوکلاس کا کوئی object تیار ہونے پرخود بخود سرانجام دیتا ہے۔
- پائیتھن میں،ایک خاص فنکشن ہوتا ہے جسے"init" کہتے ہیں بطور کنسٹرکٹر استعمال ہوتا ہے۔
- ڈسٹریکٹر یہ بھی ہے کہ جب کوئی شے دائرہ کار سے خارج ہوجائے تو خود بخود عمل درآمد ہوجاتا ہے۔
- پائیتھن میں، ( )__del__ طریقہ کوڈسٹرکٹر کے طور پر استعمال کیا جاتا ہے۔
- double underscore کے ساتھ آگے لگائے جانے والا متغیر فطری طور پر نجی بن جاتا ہے۔

12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:01 PM

Hands on Experience عملی مشق

1۔ class استعمال کرتے ہوئے، طلباء کے نام اور نشانات کو فہرست میں جمع کریں اور کل نمبرات کو print کرنے کے لئے ایک پروگرام لکھیں

2۔ مثلث کے تین اضلاع کو قبول کرنے اور اس کے رقبے کو پرنٹ کرنے کے لئے class کا استعمال کرتے ہوئے ایک پروگرام لکھیں۔

3۔ دو فاصلوں کو پڑھنے، ڈیپلے کرنے، جمع کرنے اور تفریق کرنے کیلئے مینو سے چلنے والے پروگرام (menu driver program) لکھیں۔

6MVJH

Evaluation محاسبہ

حصہ - I

صحیح جواب منتخب کریں۔ (1 مارک)

1۔ مندرجہ ذیل میں سے کون سی خصوصیات آبجیکٹ اورینٹڈ پروگرامنگ زبان کے ہیں؟

(a) constructor اور classes (b) constructor اور objects

(c) classes اور objects (d) constructor اور destructor

2۔ ایک class کے اندر وضاحت کردہ functions :

(a) functions (b) Module

(c) Methods (d) section

3۔ کلاس کے ممبر، کس آپریٹر کے ذریعے ان تک رسائی حاصل کرتے ہیں ؟

(a) & (b) .

(c) # (d) %

4۔ جب کوئی object تیار ہو جائے تو مندرجہ ذیل میں سے کون سا طریقہ خود بخود چلنے لگتا ہے ؟

(a) __object__( ) (b) __del__( )

(c) __func__( ) (d) __init__( )

5۔ ایک نجی کلاس متغیر کے آگے یہ ہوتا ہے۔

(a) __ (b) &&

(c) ## (d) **

6۔ مندرجہ ذیل میں سے کون سا طریقہ ڈسٹریکٹر (destructor) کے طور پر استعمال ہوتا ہے؟

(a) __init__( ) (b) __dest__( )

(c) __rem__( ) (d) __del__( )

100 95 75 25 5 0

12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:01 PM

7۔ مندرجہ ذیل میں سے کون سا کلاس اعلامیہ درست ہے؟

(a) کلاس کلاس_نام (b) کلاس کلاس_نام <>

(c) کلاس کلاس_نام: (d) کلاس کلاس_نام][

8۔ مندرجہ ذیل میں سے کون سا پروگرام مندرجہ ذیل ہے؟ کلاس کا طالب علم:

```
class Student:
    def __init__(self, name):
        self.name=name
S=Student("Tamil")
```

(a) Error (b) Tamil

(c) name (d) self

9۔ نجی کلاس متغیر مندرجہ ذیل میں سے کون سا ہے؟

(a) __num (b) ##num

(c) $$num (d) &&num

10۔ کسی شے کو بنانے کے عمل کو اس طرح کہا جاتا ہے:

(a) تعمیر کنندہ (b) ڈسٹرکٹر

(c) ابتدائی کریں (d) انسٹیٹیشن

حصہ - II

مندرجہ ذیل سوالات کے جوابات لکھئے۔ **(2 مارکس)**

1۔ کلاس کیا ہے ؟

2۔ **instantiation** کیا ہے ؟

3۔ مندرجہ ذیل پروگرام کا **output** کیا ہے؟

```
class Sample:
    __num=10
    def disp(self):
        print(self.__num)
S=Sample()
S.disp()
print(S.__num)
```

4۔ آپ پائیتھن میں کنسٹرکٹر (**constructor**) کیسے بنائیں گے؟

5۔ ڈسٹرکٹر (**destructor**) کا مقصد کیا ہے؟

12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:01 PM

حصہ - III

درج ذیل سوالات کے جوابات لکھئے۔ (3 مارکس)

1۔ کلاس ممبرس (class members) کیا ہیں؟ آپ اسے کیسے واضح کریں گے؟

2۔ ایک دونجی کلاس متغیر (private class variables) والی کلاس لکھئیا ور ایک طریقہ استعمال کر کے حاصل جمع (Sum) کو print کیجئے۔

3۔ دیئے گئے آؤٹ پٹ کو حاصل کرنے کے لئے مندرجہ ذیل پروگرام میں error تلاش کیجئے؟

```
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()
```

Output

```
Fruit 1 = Apple,   Fruit 2 = Mango
```

4۔ مندرجہ ذیل پروگرام کا output کیا ہے؟

```
class Greeting:
def __init__(self, name):
    self.__name = name
def display(self):
    print("Good Morning ", self.__name)
obj=Greeting('Bindu Madhavan')
obj.display()
```

5۔ پائیتھن میں تعمیر کار (constructor) اور ڈسٹرکٹر (destructor) کی تعریف کیسے کی جاتی ہے ؟

حصہ - IV

مندرجہ ذیل سوالات کے جوابات لکھئے۔ (5 مارکس)

1۔ اسٹیشنری آئٹمز کو شامل کرنے یا حذف کرنے کے لئے مینو سے چلنے والا پروگرام (menu driven program) لکھئے۔

اشیاء اور ان کے کاروباری شناخت (brand) کو محفوظ کرنے کے لئے آپ کو dictionary کا استعمال کرنا چاہئے۔

References حوالہ جات

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.*

12 CS UNIT 10 CDR Set
Monday, April 17, 2023 5:39:01 PM

ڈیٹا بیس کے تصورات DATABASE CONCEPTS

سیکھنے کے مقاصد

اس باب کی تکمیل پر، طالب علم یہ جان چکا ہوگا۔

- ایک ڈیٹا بیس اور تعلقی ڈیٹا بیس کا تصور۔
- ڈیٹا بیس کے مختلف اجزاء۔
- ڈیٹا بیس ماڈل کی قسمیں۔
- تعلقات کی اقسام۔
- تعلقی الجبرا کے تصورات۔

تعارف Introduction

ڈیٹا بیس اعداد و شمار کا ایک منظّم ذخیرہ ہوتا ہے، جو عام طور پر کمپیوٹر سسٹم سے الیکٹرانک طور پر اسٹور اور حاصل کیا جاتا ہے۔ اصطلاح "database" بھی کسی بھی DBMS، ڈیٹا بیس سسٹم یا ڈیٹا بیس سے وابستہ اپلی کیشن کے لئے استعمال ہوتا ہے۔ ان کے مابین قریبی تعلقات کی وجہ سے، اکثر ڈیٹا بیس اور DBMS دونوں کو استعمال کرنے کے لئے اصطلاح "database" عام طور پر استعمال کی جاتی ہے۔ اسکول میں کلاس کا رجسٹر ایک ایسا ڈیٹا بیس ہوتا ہے جہاں ناموں کو حرف تہجی سے ترتیب دیا جاتا ہے۔ جب سے لوگوں نے چیزوں کو ریکارڈ کرنا شروع کیا ہے، ڈیٹا بیس، ہمارے آس پاس جمع ہو رہا ہے۔ یہاں ہم الیکٹرانک data کی طرف توجہ مرکوز کرتے ہیں۔

11.1 ڈیٹا (Data)

ڈیٹا ایک خام حقائق ہیں، جو کمپیوٹر میں محفوظ ہیں۔ ڈیٹا میں کوئی بھی علامت، متن، لفظ یا ایک نمبر شامل ہوسکتا ہے۔

مثال : 600006، DPI کیمپس، SCERT، چنئی، کالج روڈ۔

11.2 معلومات (اطلاعات) (Information)

معلومات کو ڈیٹا فارمیٹ (formatted data) کہا جاتا ہے، جو قابل استعمال طریقے سے استعمال کرنے کی اجازت دیتا ہے۔

مثال :

SCERT
College Road
DPI Campus
Chennai 600006

SYC51

جیسا کہ آپ دیکھ سکتے ہیں کہ مندرجہ بالا مثال سے، data الفاظ اور اعداد کا ایک سیٹ کے طور پر ظاہر ہوتے ہیں۔ تاہم، جب اعداد و شمار پر کارروائی، منظّم اور فارمیٹ کیا جاتا ہے تو، اس سے SCERT کے ادارے سے رابطہ کے پتے کے بارے میں معنی خیز معلومات مل جاتی ہے۔

11.3 ڈیٹا بیس (Database)

ڈیٹا بیس ایک طرح سے متعلق اعداد و شمار کا ذخیرہ ہے جس سے اعداد و شمار تک آسانی سے access، manage اور updated کیا جاسکتا ہے۔ ڈیٹا بیس سافٹ ویئر یا ہارڈ ویئر پر مبنی ہوسکتا ہے، جس میں ڈیٹا کو محفوظ کرنے کا ایک ہی مقصد ہوتا ہے۔

11.4 ڈیٹا بیس مینجمنٹ سسٹم (DBMS)

DBMS ایک ایسا سافٹ ویئر ہے جو ہمیں ڈیٹا بیس بنانے، اس کی وضاحت کرنے اور اسے برتنے کرنے کی سہولت دیتا ہے، جس سے صارفین آسانی سے ڈیٹا کو اسٹور، پراسس اور analyse کرسکتے ہیں۔ DBMS ہمیں ایک انٹرفیس یا ایک ٹول فراہم کرتا ہے، جس میں ڈیٹا بیس بنانے، اعداد و شمار کو ذخیرہ کرنے اور ڈیٹا کو اپ ڈیٹ کرنے کے لئے مختلف کاروائیاں انجام دی جاتی ہیں۔

ڈیٹا بیس یہ متعدد صارفین (multiple users) کی صورت میں بھی ڈیٹا مستقل مزاجی (data consistency) کو برقرار رکھتا ہے۔

DBMS سافٹ ویئر کی مثالیں یہ ہیں: Foxpro, dbase

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:17 PM

11.4.1 نسبتی ڈیٹابیس مینجمنٹ سسٹم (Relational Database Management System)(RDBMS)

ڈیٹابیس مینجمنٹ سسٹم سے اعلیٰ ورژن نسبتی ڈیٹابیس مینجمنٹ سسٹم ہے۔ یہ ڈیٹا کو زیادہ کارکردہ طریقے سے عمل میں لاتا ہے۔
یہ ڈیٹا کو ٹیبلٹ کی شکل میں

11.4.2 نسبتی ڈیٹابیس مینجمنٹ سسٹم کی خصوصیات (Characteristics of Relational Database Management System)

1۔ ڈیٹا کو محفوظ کرتا اور عمل میں لاتا ہے۔	RDBMS ایک ڈیٹابیس میں ڈیٹا کو (محفوظ، ترمیم اور منسوخ) کرنے کی سہولت رکھتا ہے۔
2۔ فضول پن میں کمی (Reduced Redundancy)	جدید دنیا میں ہارڈ ڈرائیوز (hard drives) بہت سستی ہیں، لیکن اس سے قبل جب ہارڈ ڈرائیوز بہت مہنگی ہوتی تھیں۔ ڈیٹابیس میں ڈیٹا کی غیر ضروری تکرار (دوہرانے) کو محفوظ کرنا ایک بہت بڑا مسئلہ تھا لیکن RDBMS عام عمل کی پیروی کرتا ہے جو اعداد و شمار کو اس طرح تقسیم کرتا ہے کہ کم سے کم بار دوہرایا جائے۔
3۔ ڈیٹا میں مستقل مزاجی (data consistency)	براہ راست data کو مستقل طور پر اپ ڈیٹ کیا جاتا ہے اور نئی معلومات کو شامل کیا جاتا ہے، data کی مستقل مزاجی کو برقرار رکھنا ایک چیلنج بن سکتا ہے۔ لیکن RDBMS اسے خود ہی سنبھالتا ہے۔
4۔ ایک سے زیادہ صارف اور ہمہ وقت تک رسائی کی حمایت کرتے ہیں۔	RDBMS ایک ہی وقت میں متعدد صارفین کو اس پر کام کرنے کی اجازت دیتا ہے (ڈیٹا کو اپ ڈیٹ کرنے، داخل کرنے، حذف کرنے) اور پھر بھی data میں مستقل مزاجی (data consistency) برقرار رکھنے کا انتظام کرتے ہیں۔
5۔ سوال کرنے کی زبان	RDBMS صارفین کو ایک آسان سوال کرنے کی زبان فراہم کرتا ہے، جس کا استعمال کرتے ہوئے ڈیٹابیس میں آسانی سے ڈیٹا لاکر، داخل کرکے، حذف کرکے اور اپ ڈیٹ بھی کیا جاسکتا ہے۔
6۔ سیکیورٹی (تحفظ)	غیر اختیاری رسائی حاصل کرنیوالوں سے RDBMS ڈیٹا کی نگرانی اور حفاظت کا بھی خیال رکھتا ہے۔ ایک عام RDBMS میں، ہم مختلف رسائی کی اجازت والے صارف اکاؤنٹ بنا سکتے ہیں۔ جسے ہم صارف تک رسائی کو محدود کرکے آسانی سے اپنے ڈیٹا کو محفوظ کرسکتے ہیں۔
7۔ DBMS لین دین کی حمایت کرتا ہے۔	یہ ہمیں حقیقت میں ڈیٹا کی سالمیت کو بہتر طریقے سے سنبھالنے اور ان کا انتظام کرنے کی سہولت دیتا ہے۔ ورلڈ اپیلی کیشنز جہاں ملٹی تھریڈنگ بڑے پیمانے پر استعمال ہوتی ہے۔

11.4.2 RDBMS کے فوائد (Advantages of RDBMS)

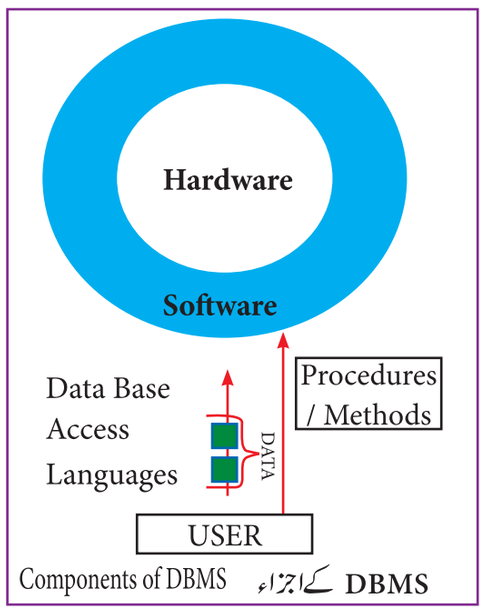
- اپلی کیشن کے پروگرام کو الگ کرنا
- کم سے کم ڈیٹا کی نقل محفوظ کرنا یا فضول ڈیٹا رکھنا
- سوال کی زبان (query language) کے استعمال سے data کی آسانی سے بازیافت
- ترقی کے لئے کم وقت اور دیکھ بھال (نگرانی)

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:17 PM

11.4.4 DBMS کے اجزاء
(Components of DBMS)

DBMS کو پانچ بڑے اجزاء میں تقسیم کیا جاسکتا ہے۔

1۔ ہارڈ ویئر
2۔ سافٹ ویئر
3۔ ڈیٹا
4۔ عمل/طریقے (Procedures/methods)
5۔ ڈیٹا بیس تک رسائی کی زبانیں



خاکہ 11.1

1۔ ہارڈ ویئر (Hardware):

کمپیوٹر، ہارڈ ڈسک، Data لئے I/O چینلز، اور Data کو محفوظ کرنے کے لئے کوئی دیگر طبیعی جز (Physical component)

2۔ سافٹ ویئر (Software):

یہ اہم جزو ایک پروگرام ہے جو ہر چیز کو کنٹرول کرتا ہے۔ DBMS سافٹ ویئر ڈیٹا بیس تک رسائی کی زبان کو سمجھنے کی صلاحیت رکھتا ہے اور اس پر عملدرآمد کے لئے ڈیٹا بیس کمانڈوں کی ترجمانی کرتا ہے۔

3۔ ڈیٹا:

یہ وہ وسیلہ (ذریعہ) ہے جس کے لئے DBMS تیار کیا گیا ہے۔ DBMS ڈیٹا کو اسٹور اور استعمال کرنے کے لئے تیار کیا گیا ہے۔

4۔ طریقہ کار/طریقے (Procedures/methods):

وہ ڈیٹا بیس مینجمنٹ سسٹم کو استعمال کرنے کے لئے عام ہدایات ہیں جیسے DBMS کی تنصیب، بیک اپ لینے کے لئے ڈیٹا بیس کا انتظام کرنا، رپورٹ تیار کرنا وغیرہ۔

5۔ ڈیٹا بیس تک رسائی کی زبانیں:
(Database access languages)

وہ زبانیں ہیں جو کسی بھی ڈیٹا بیس میں محفوظ ڈیٹا تک رسائی، داخل کرنے، اپ ڈیٹ کرنے اور اسے حذف کرنے کے لئے کمانڈ لکھنے استعمال ہوتی ہیں۔

مشہور DBMS کی مثالیں: Dbase, FoxPro

11.5 ڈیٹا بیس کی ساخت (Database structure)

ایک جدول، متعلقہ ڈیٹا کا پورا مجموعہ ہے، جسے فائل یا table کہا جاتا ہے، جہاں ڈیٹا کو قطار اور کالم کے طور پر منظم کیا جاتا ہے۔

ٹیبل کی ہر صف ریکارڈ کی نمائندگی کرتی ہے، جو ہر ڈیٹا بیس اندراج کے لئے ڈیٹا کا ایک سیٹ ہوتا ہے۔

جدول کا ہر کالم ایک فیلڈ کی نمائندگی کرتا ہے، جو ڈیٹا کے ہر ٹکڑے یا آئٹم کو ریکارڈ کے درمیان مخصوص زمرے یا اعداد و شمار کی قسموں میں گروپ کرتا ہے۔ مثال کے طور پر

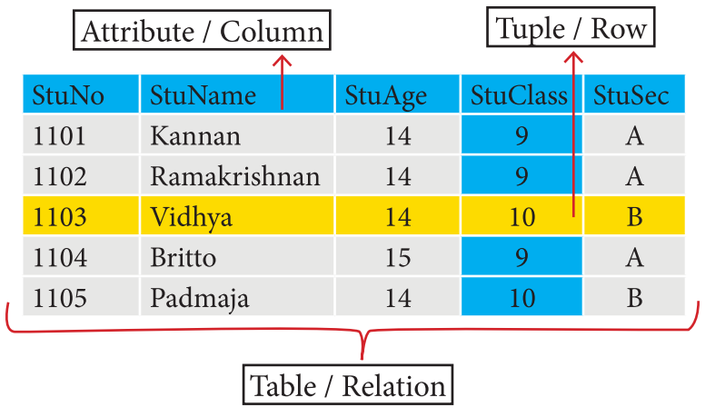
Stu.no., StuName, StuAge, StuClass, StuSec.

ایک جدول (table) کو ایک RELATION کے نام سے جانا جاتا ہے۔

ایک صف (row) کو TUPLE کہا جاتا ہے۔

ایک کالم (قطار) کو ATTRIBUTE کے نام سے جانا جاتا ہے۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:18 PM



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 ڈیٹابیس کی ساخت

خاکہ 11.2

11.6 ڈیٹاماڈل (Data model)

- ایک ڈیٹاماڈل بیان کرتا ہے کہ مکمل نفاذ کے بعد کس طرح سافٹ ویر سے ڈیٹا کی نمائندگی کی جاسکتی ہے اور اسے حاصل کیا جاسکتا ہے۔
- یہ پیچیدہ حقیقی دنیا کے ڈیٹا اکٹھا کرنے والے ماحول کا ایک سادہ حاصل کرنے کا طریقہ ہے۔
- ڈیٹاماڈل کا بنیادی مقصد یہ بتانا ہے کہ ترقی مکمل ہونے کے بعد حتمی نظام یا سوفٹ ویر کیسا نظر آئے گا۔

11.6.1 ڈیٹاماڈل کی قسمیں (Types of Data Model)

ڈیٹاماڈل کی مختلف قسمیں درج ذیل ہیں.

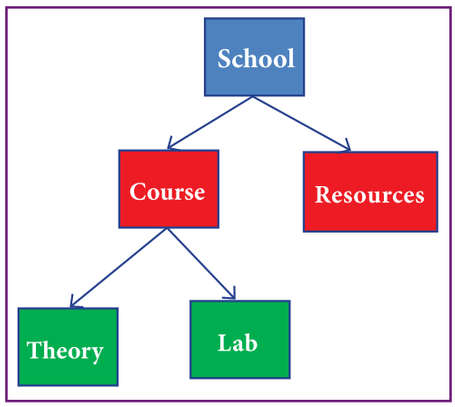
- درجہ بندی کا ماڈل
- تعلقی ماڈل
- نیٹ ورک ڈیٹابیس ماڈل
- ہستی کے تعلق کا ماڈل
- آبجیکٹ ماڈل

1۔ درجہ بندی کا ماڈل (Hierarchical Model)

درجہ بندی کا ماڈل IBM نے انفارمیشن مینجمنٹ سسٹم کے طور پر تیار کیا تھا۔

درجہ بندی کے ماڈل میں، ڈیٹا کو سادہ درخت کے ساخت کی طرح دکھایا جاتا ہے۔ یہ ماڈل ایک سے کئی تعلقات کی نمائندگی کرتا ہے، یعنی والدین اور بچے کے تعلقات۔ ایک بچے کے صرف ایک والدین ہوسکتے ہیں لیکن ایک والدین کے بہت سے بچے ہوسکتے ہیں۔ یہ ماڈل بنیادی طور پر IBM کے مین فریم کمپیوٹرز (main frame computers) میں استعمال ہوتا ہے۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:18 PM



خاکہ 11.3 درجہ بندی کا ماڈل

2۔ تعلقی ماڈل (Relational Model)

متعلقہ ڈیٹا بیس ماڈل کو پہلی بار E.F. Codd نے 1970 میں تجویز کیا تھا۔ آج کل، یہ دنیا بھر میں ڈیٹا بیس کی ایپلی کیشنز کے لئے استعمال ہونے والا سب سے وسیع ڈیٹا ماڈل ہے۔

تعلقی ماڈل میں ڈیٹا کی بنیادی ساخت جدول (تعلقات) ہیں۔ کسی خاص قسم سے متعلق تمام معلومات اسی ٹیبل کی صف میں محفوظ کیا جاتا ہے۔ لہذا جدول کو تعلقی ماڈل میں تعلقات کے طور پر بھی جانا جاتا ہے۔ ایک تعلقی کلید (relation key) ، ایک وصف (attribute) ہے، جو کسی خاص Tuple (تعلق میں صف) (جدول 1) کی شناخت کرتی ہے۔

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

خاکہ 11.4 تعلقی ماڈل **Relational Model**

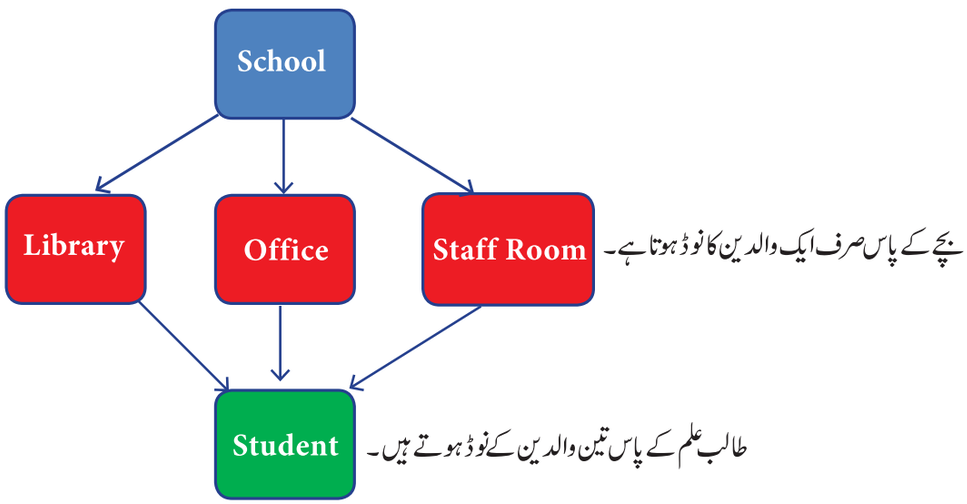
3۔ نیٹ ورک ماڈل (Network Model)

نیٹ ورک ڈیٹا بیس کا ماڈل درجہ بندی کے ڈیٹا ماڈل کی ایک توسیعی شکل ہے۔ درجہ بندی اور نیٹ ورک کے ڈیٹا ماڈل کے درمیان فرق یہ ہے:

- درجہ بندی کے ماڈل میں، بچوں کے ریکارڈ میں صرف ایک والدین کا نوڈ ہوتا ہے۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:18 PM

- نیٹ ورک ماڈل میں، بچے کے پاس بہت سے والدین نوڈس ہو سکتے ہیں۔ یہ بہت سے تعلقات میں ڈیٹا کی نمائندگی کرتا ہے۔
- ڈیٹا تک رسائی حاصل کرنے کے لئے یہ ماڈل آسان اور تیز تر ہے۔



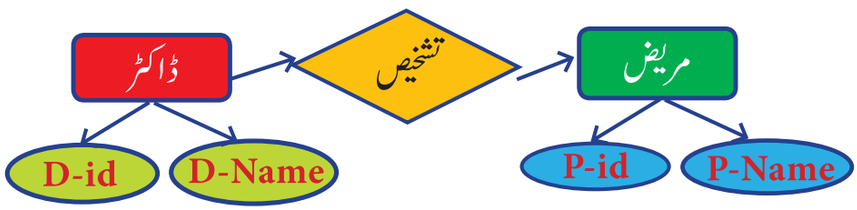
Network Model خاکہ 11.5 نٹ ورک ماڈل

اسکول والدین کی نمائندگی کرتا ہے۔
لائبریری، دفتر اور عملہ کا کمرہ ، بچے سے اسکول کا تعلق ہے۔ (پیرنٹ نوڈ)۔
طالب علم، لائبریری، دفتر اور عملہ کا کمرہ (ایک سے کئی تعلقات) کا تعلق بچے سے ہے۔

4۔ ہستی کے تعلقات کا ماڈل (ER ماڈل) (Entity Relationship Model)

اس ڈیٹا بیس ماڈل میں، شئے کو وجود اور اس کی خصوصیات کو ہستیوں میں تقسیم کر کے تعلق قائم کیا جاتا ہے۔
اس کو Chen نے 1976 میں تیار کیا تھا۔ یہ ماڈل ڈیٹا بیس کے لئے نظریاتی ڈیزائن تیار کرنے میں کارآمد ہے۔Data کے منطقی نظارے کو ڈیزائن کرنا بہت آسان اور آسان ہے۔ڈویلپر تعمیر کردہ ER ماڈل دیکھ کر سسٹم کو آسانی سے سمجھ سکتا ہے۔
ایک مستطیل (rectangle) ہستیوں کی نمائندگی کرتا ہے۔جیسے۔ڈاکٹر اور مریض
بیضوی شکل (ellipse) صفات کی نمائندگی کرتے ہیں جیسے۔ D-id، D-name، P-id، D-name۔ صفات (attributes) خصوصیات کو بیان کرتی ہیں اور ہر ہستی ڈیٹا بیس میں محفوظ ڈیٹا کا ایک اہم حصہ بن جاتی ہے۔
کھجوری شکل (Diamond shape) ER خاکوں میں تعلقات کی نمائندگی کرتا ہے۔
جیسے۔ڈاکٹر مریض کی تشخیص کرتا ہے۔

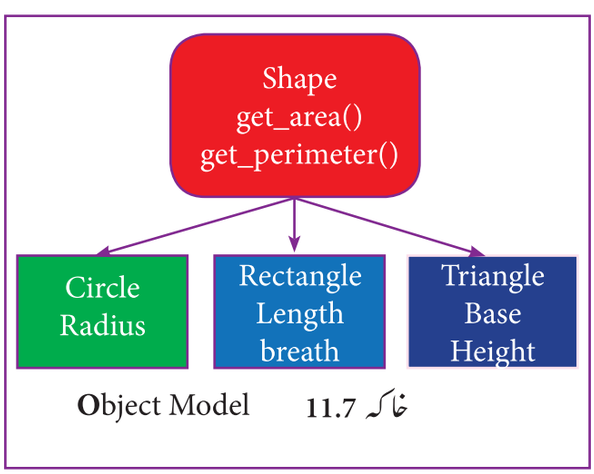
12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:19 PM



ER model خاکہ 11.6

5۔ آبجیکٹ ماڈل (Object Model)

آبجیکٹ ماڈل ڈیٹا کو اشیاء، اوصاف اور طریقوں، کلاسس اور درجہ بندی کی شکل میں محفوظ کرتا ہے۔ یہ ماڈل زیادہ پیچیدہ ایپلی کیشنز، جیسے جغرافیائی انفارمیشن سسٹم (GIS)، سائنسی تجربات، انجینئرنگ ڈیزائن اور تیاری کو سنبھالتا ہے۔ یہ فائل مینجمنٹ سسٹم میں استعمال ہوتا ہے۔ یہ حقیقی دنیا کی اشیاء، اوصاف اور طرز عمل کی نمائندگی کرتا ہے۔ یہ ایک واضح ماڈیولر ساخت فراہم کرتا ہے۔ موجودہ کوڈ کو برقرار رکھنے اور اس میں ترمیم کرنا آسان ہے۔



Object Model خاکہ 11.7

آبجیکٹ ماڈل کی ایک مثال شکل، دائرہ، مستطیل اور مثلث اس ماڈل میں شامل ہیں۔

- دائرہ کا وصف (attribute) اس کا نصف قطر ہے۔
- مستطیل کے اوصاف (attributes) اس کی لمبائی اور چوڑائی ہیں۔
- مثلث کے اوصاف (attributes)، اس کا قاعدہ اور اونچائی ہے۔
- اشیاء دائرہ، مستطیل اور مثلث، شئے کی شکل سے درجہ بندی (inherit) میں ملتے ہیں۔

11.6.2 DBMS کے صارفین کی اقسام۔ (Types of DBMS Users)

ڈیٹابیس اڈمنسٹریٹر (Database Administrator)

ڈیٹابیس ایڈمنسٹریٹر یا DBA وہ ہے جو مکمل ڈیٹابیس مینجمنٹ سسٹم کا انتظام کرتا ہے۔ DBMS کی حفاظت اور نگرانی DBA کرتا ہے، لائسنس کی چابیاں کا انتظام، صارف کے کھاتوں کا انتظام اور رسائی وغیرہ کا خیال رکھتا ہے۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:19 PM

اپلی کیشن پروگرامرز یا سافٹ ویئر ڈویلپرز (Application programmers or Software developers)

یہ صارف گروپ کے DBMS کے پرزے تیار کرنے اور ڈیزائن کرنے میں شامل ہے۔

آخر میں استعمال کرنے والا (آخری صارف) (End user)

تمام جدید اپلی کیشنز، ویب یا موبائل، صارف کا ڈیٹا اسٹور کریں۔ اپلی کیشن کو اس طرح پروگرام کیا جاتا ہے کہ وہ صارف کا ڈیٹا اکٹھا کرتے ہیں اور اپنے سرور پر چلنے والے DBMS سسٹم پر ڈیٹا اسٹور کرتے ہیں۔ آخری صارف وہی ہیں جو ڈیٹا کو اسٹور، بازیافت، اپ ڈیٹ اور حذف کرتے ہیں۔

ڈیٹا بیس ڈیزائنرز (Database designers)

ڈیٹا کی نمائندگی اور ذخیرہ کرنے کی مناسب ساخت کا انتخاب کرنے کے لئے ڈیٹا بیس میں ذخیرہ کرنے والے، ڈیٹا کی شناخت کے ذمہ دار ہیں۔

11.7 DBMS اور RDBMS کے درمیان فرق (Difference between DBMS and RDBMS)

موازنہ کی بنیاد	DBMS	RDBMS
توسیع	ڈیٹا بیس مینجمنٹ سسٹم	متعلقہ ڈیٹا بیس مینجمنٹ سسٹم
ڈیٹا کو محفوظ کرنا	نیویگیشنل ماڈل یعنی منسلک ریکارڈ کے ذریعہ	متعلقہ ماڈل (جدولوں میں)۔ یعنی جدول میں صف اور قطار طور پر ڈیٹا۔
ڈیٹا میں فضول پن (Data redundancy)	نمائش ہوتی ہے	غیر موجود
معمول بنانا (Normalisation)	انجام نہیں دیا گیا	فضول پن کو کم کرنے کے لئے RDBMS نارملائزیشن کا استعمال کرتا ہے۔
ڈیٹا تک رسائی	زیادہ وقت لگتا ہے	DBMS کے مقابلے میں تیز
چابیاں اور اشاریہ جات Keys and indexes	استعمال نہیں کرتا	تعلقات قائم کرنے کے لئے استعمال کیا جاتا ہے کلیدیں RDBMS میں استعمال ہوتی ہیں۔
ٹرانزیکشن مینجمنٹ	کم کارکردگی، Error کے ساتھ اور غیر محفوظ	کارکردگی اور محفوظ
تقسیم شدہ ڈیٹا بیس	RDBMS کے ذریعہ تعاون یافتہ	سہولت مہیا نہیں
مثال	Dbase, FoxPro.	SQL server, Oracle, mysql, MariaDB, SQLite.

ڈیٹا بیس کو معمول پر لانے کی تجویز سب سے پہلے Dr. Edgar F Codd نے RDBMS کے لازمی حصے کے طور پر کی تھی تاکہ ڈیٹا میں فضول پن کی کمی اور ڈیٹا کی سالمیت کو بہتر بنایا جاسکے۔ یہ قواعد E F Codd Rules کے نام سے مشہور ہیں۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:19 PM

11.8 تعلقات کی اقسام (Types of Relationships)

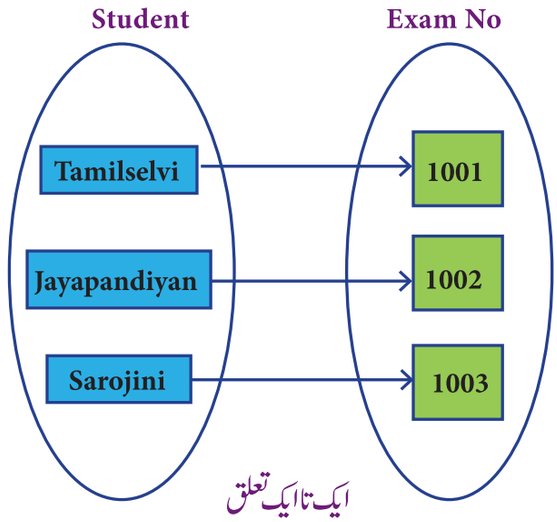
ذیل میں ڈیٹا بیس میں استعمال ہونے والے تعلقات کی اقسام درج ذیل ہیں۔

1۔ ایک سے ایک رشتہ۔
2۔ ایک سے زیادہ تعلقات۔
3۔ ایک سے زیادہ تعلقات۔
4۔ بہت سے تعلقات۔

1۔ایک سے ایک کا تعلق (One-to-one Relationship)

تعلقات میں، ایک ہستی کا تعلق صرف ایک ہی دوسری ہستی سے ہوتا ہے۔ جدول کی ایک قطار دوسرے جدول میں صرف ایک ہی قطار سے منسلک ہے اور اس کا برعکس بھی۔

مثال کے طور پر: ایک طالب علم کے پاس صرف ایک امتحان نمبر ہوسکتا ہے۔



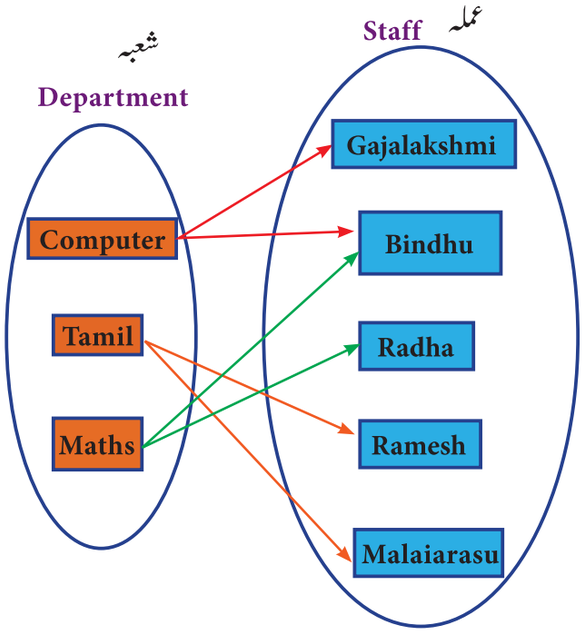
ایک تا ایک تعلق

خاکہ 11.8

ایک سے کئی کا تعلق (One-to-Many Relationship)

ایک سے کئی کے تعلق میں، ایک ہستی کا تعلق بہت ساری دیگر ہستیوں سے کیا جاتا ہے۔

جدول A میں موجود ایک صف کو جدول B میں موجود ایک صف کے ساتھ جوڑا گیا ہے، مگر جدول B کی ایک صف کو جدول A کی صرف ایک صف کے ساتھ جوڑا گیا ہے۔ مثال : ایک شعبے کے کئی ممبر ہو سکتے ہیں۔



ایک سے کئی کی نقشہ سازی (One to Many Mapping)

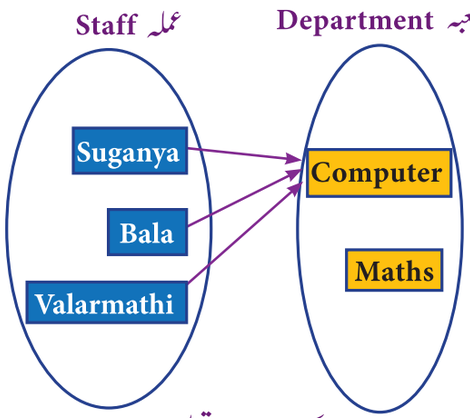
خاکہ 11.9

3۔کئی سے ایک کا تعلق (Many-to-one Relationship)

کئی سے ایک کے تعلق میں کئی ہستیوں کو صرف ایک ہستی کے ساتھ جوڑا جاتا ہے۔

مثال کے طور پر : ایک محکمہ میں کام کرنے والے عملے کے متعدد ممبران۔

عملے کے ممبروں کی جدول میں متعدد صفوں کو شعبہ کی جدول کے صرف ایک صف سے جوڑا گیا ہے۔



کئی سے ایک کا تعلق

خاکہ 11.10

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:19 PM

4۔ کئی سے کئی کا تعلق
(Many-to-many relationship)

کئی سے کئی کا تعلق اس وقت ہوتا ہے، جب کسی ٹیبل میں متعدد ریکارڈس کو کسی دوسرے ٹیبل کے متعدد ریکارڈوں کے ساتھ جوڑا جاتا ہو۔

مثال 1 : صارفین اور مصنوعات

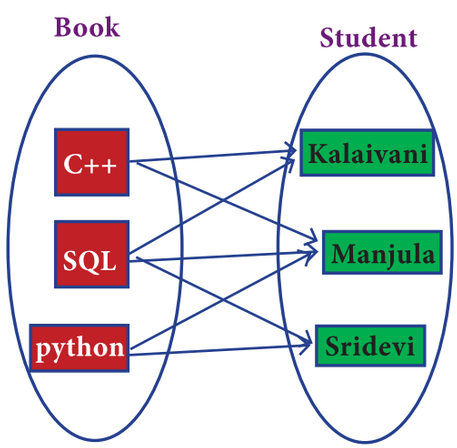
صارفین مختلف مصنوعات خرید سکتے ہیں اور بہت سارے گاہکوں کے ذریعہ مصنوعات خریدی جاسکتی ہیں۔

مثال 2 : طلباء اور نصاب

ایک طالب علم بہت سارے کورسس کے لئے اندراج کرسکتا ہے اور ایک کورس میں بہت سارے طلباء شامل ہوسکتے ہیں۔

مثال 3 : کتابیں اور طالب علم

ایک لائبریری میں بہت سی کتابیں، بہت سارے طلباء کو جاری کی جاتی ہیں۔



کئی سے کئی کا تعلق Many to Many Relationship

خاکہ 11.11

کیا تم جانتے ہو



متعلقہ ماڈل کو Edgar Frank Codd نے ایجاد کیا، جسے (Father of relational database) کہا جاتا ہے۔ Data کے ایک عام نمونہ کے طور پر اور اس کے بعد اس کو Chris Date کے علاوہ Hugh Darwen اور دیگروں نے فروغ دیا۔

11.9 DBMS میں متعلقہ الجبرا

متعلقہ الجبرا کیا ہے ؟

متعلقہ الجبرا کو سب سے پہلے Edgar F Codd نے IBM میں رہتے وقت تیار کیا۔ تعلقی ڈیٹا بیس میں ڈیٹا کو محفوظ کرتے وقت ڈیٹا کی نمونہ سازی کے لئے اور اس سے متعلق سوالات کو واضح کرنے کے لئے استعمال کیا جاتا ہے۔

متعلقہ الجبرا طریقہ وار سوالات کی زبان ہے (procedural query language)۔ SQL کے استعمال سے ڈیٹا بیس کے جدول کے سوالات کو جواب دینے کے لئے یہ استعمال ہوتا ہے۔

متعلقہ الجبرا کے اعمال کو ایک تعلق (جدول) پر بار بار انجام دیا جاتا ہے، تاکہ ایک output حاصل کیا جا سکے۔ ان اعمال کا output ایک نیا تعلق ہے، جو ایک یا ایک سے زیادہ تعلقات کے input سے بنا ہوا ہوتا ہے۔

متعلقہ الجبرا کو مختلف گروپوں میں تقسیم کیا گیا ہے۔

Unary Relational operations

SELECT (Symbol : σ)

PROJET (Symbol : Π)

مجموعہ کے نظریہ سے متعلقہ الجبرا کے اعمال

- UNION (∪)
- INTERSECTION (∩)
- DIFFERENCE (–)
- CARTESIAN PRODUCT (X)

منتخب کریں (علامت: σ) (SELECT)

عام شکل $\sigma_c(R)$ جس میں ایک تعلق R اور ایک شرط C کے ساتھ R کی خصوصیات پر۔

دی گئی شرط کے ساتھ ٹیوپل میں تحتی مجموعہ کا انتخاب کرنے کے لئے SELECT operation استعمال کیا جاتا ہے۔

C کی شرط پوری نہیں کرنے والے تمام ٹیوپل کو SELECT فلٹر کرتا ہے۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:20 PM

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

جدول 11.1

σ_{course} = "Big Data" **(STUDENT)**

Studno	Name	Course	Year
cs1	Kannan	Big Data	II
cs3	Lenin	Big Data	I

(Symbol : Π) PROJECT

input کے تعلق کے تمام اوصاف **(attributes)** کو **projection** ختم کر دیتا ہے۔ **projection** کا طریقہ ایک تعلق کی وضاحت کرتا ہے، جس میں تعلق **(relation)** کا ایک عمودی تحتی مجموعہ ہے۔

Example 1 using Table 11.1

مثال 1 : جدول 11.1 کو استعمال کرتے ہوئے۔

Π_{course} (STUDENT)

Result

Course
Big Data
R language
Python Programming

نوٹ:

نتیجہ میں نقلی صف خارج کر دی گئی ہے۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:20 PM

مثال 2 : (جدول 11.1 کا استعمال کرتے ہوئے)

$\Pi_{\text{studno, course}}$ (STUDENT)

Result

Studno	Course
cs1	Big Data
cs2	R language
cs3	Big Data
cs4	Python Programming

UNION (علامت : ∪)

اس میں وہ تمام ٹیوپلس شامل ہیں، جو ٹیبل A یا B میں موجود ہیں۔ اس سے نقلیں (duplicates) بھی نکال دی جاتی ہیں۔

Set A کا یونین Set B کا اظہار A ∪ B کی طرح کیا جائے گا۔

مثال 3 : مندرجہ ذیل جدولوں پر غور کریں۔

Table A	
Studno	**Name**
cs1	Kannan
cs3	Lenin
cs4	Padmaja

Table B	
Studno	**Name**
cs1	Kannan
cs2	GowriShankarn
cs3	Lenin

Table 11.2

Result

Table A ∪ B	
Studno	**Name**
cs1	Kannan
cs2	GowriShankar
cs3	Lenin
cs4	Padmaja

مجموعہ کا فرق (علامت : –) (SET DIFFERENCE)

A – B کا نتیجہ ایک رشتہ ہے، جس میں وہ تمام ٹیوپلس شامل ہیں جو A میں ہیں، لیکن B میں نہیں ہیں۔

A کے وصف کے نام (attribute name) کو B کے وصف کے نام کے ساتھ جوڑا جاتا ہے۔

مثال 4 (جدول 11.2 کا استعمال کرتے ہوئے)

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:20 PM

Result

Table A – B	
cs4	Padmaja

تقاطع (علامت : ∩) (INTERSECTION) A ∩ B

اس تعلق کی وضاحت کرتا ہے جس میں تمام ٹیپل کا ایک مجموعہ ہوتا ہے جو A اور B دونوں میں ہوتا ہے۔ تاہم، A اور B میں اتحاد ہونا چاہئے۔

مثال 5 (جدول 11.2 کا استعمال کرتے ہوئے)

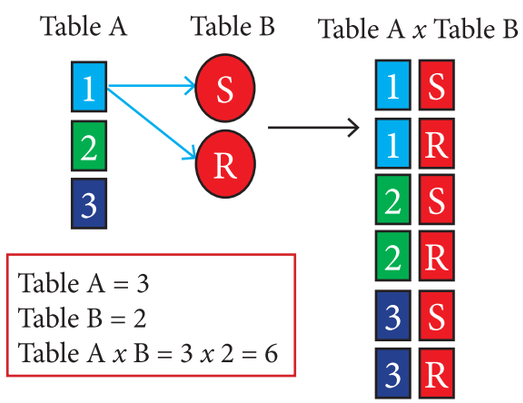
A ∩ B	
cs1	Kannan
cs3	Lenin

حاصل ضرب یا کارتیسی حاصل ضرب (علامت:X)(PRODUCT OR CARTESIAN PRODUCT)

کراس پروڈکٹ (Cross product) دو رشتوں کو جوڑنے کا ایک طریقہ ہے۔ نتیجہ کے تعلق میں جوڑنے والے دونوں تعلقات مشترکہ طور پر شامل ہیں۔

A × B کا مطلب، A مرتبہ B ہے، جہاں تعلق A اور B کے مختلف اوصاف (attributes) ہیں۔

اس طرح کے عمل، دو تعلقات کی قطاروں کو ضم (merge) کرنے میں معاون ہے۔



کارتیسی حاصل ضرب

خاکہ 11.12

Table A		Table B	
studno	**name**	**studno**	**subject**
cs1	Kannan	cs28	Big Data
cs2	Gowri Shankar	cs62	R language
cs4	Padmaja	cs25	python programming

Table 11.3

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:20 PM

کارتیسی حاصل ضرب : جدول A × جدول 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

Points to remember: یادرکھنے کے لئے نکات

- DBMS ایک کمپیوٹر پر مبنی ریکارڈ رکھنے کا نظام ہے۔
- ڈیٹا ایک غیر عمل شدہ ڈیٹا ہے، جس میں کسی بھی علامت، متن، لفظ یا عدد ہوتا ہے، جس کا کوئی معنی نہیں ہوتا۔
- معلومات یا اطلاعات عمل درآمد، منظّم اور فارمیٹ کردہ ڈیٹا ہے۔
- RDBMS کی مثالوں میں mysql ، oracle ، sql server ، ibm db2 شامل ہیں۔
- فضول پن (redundancy) کا مطلب ہے ڈیٹا بیس میں ڈیٹا کی نقل (duplication) ہے۔
- ڈیٹا مستقل مزاجی (consistency) کا مطلب، ڈیٹا بیس کی تمام صورتوں میں ڈیٹا کی قیمت ایک جیسی ہوتی ہے۔
- ڈیٹا کی سالمیت (integrity) غیر اختیاری صارفین کی سیکیورٹی ہے۔
- جدول کو ایک تعلق کہا جاتا ہے۔
- ایک صف (row) کو ٹیوپل (tuple) کہا جاتا ہے۔
- ایک قطار (کالم) کو ایک وصف (attribute) کے طور پر جانا جاتا ہے۔
- ڈیٹا ماڈل کی قسمیں: درجہ بندی Hierarchical، متعلقہ relational، نیٹ ورک network ، ای آر ER اور آبجیکٹ ماڈل Object model ہیں۔
- درجہ بندی کا ماڈل ایک سادہ درخت کی طرح ہے جس کی ساخت، ایک سے ایک کے تعلق کی طرح ہے، جس کو والدین-بچہ کا تعلق کہتے ہیں۔
- متعلقہ ماڈل تعلقات یا جدولوں میں ڈیٹا کی نمائندگی کرتا ہے۔
- نیٹ ورک ماڈل Hierarchical ماڈل کی طرح ہے، لیکن اس سے ایک سے زیادہ والدین کے ریکارڈ کی اجازت ملتی ہے۔
- ER ماڈل، ہستیوں، اوصاف اور تعلقات پر مشتمل ہے۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:21 PM

Points to remember: یادرکھنے کے لئے نکات

- آبجیکٹ ماڈل، ڈیٹا، اشیاء، اوصاف، طریقوں، کلاسس اور inheritance کےطور پرمحفوظ کرتا ہے۔
- حسب معمول (Normalisation) لانے سے، ڈیٹا کا فضول پن کم ہوتا ہے اور ڈیٹا کی سالمیت بہتر ہوتی ہے۔
- مختلف اقسام کے تعلقات ایک سے ایک، ایک سے کئی، کئی سے ایک اور کئی سے کئی تعلقات ہیں۔
- ڈیٹا بیس نارملائزیشن کی تجویز Dr. Edgar F Codd نے کی تھی۔
- متعلقہ الجبرا کو متعلقہ ڈیٹا بیس میں ذخیرہ کرنے والے ڈیٹا کو ماڈلنگ اور اس پر سوالات کی وضاحت کے لئے استعمال کیا جاتا ہے۔

Evaluation محاسبہ

حصہ - A

صحیح جواب کا انتخاب کریں۔ (ایک مارک)

1۔ DBMS کا پھیلاؤ کیا ہے؟

a) ڈیٹا بیس مینجمنٹ سمبل
b) ڈیٹا بیس مینجنگ سسٹم
c) ڈیٹا بیس مینجمنٹ سسٹم
d) ڈیٹا بیسک مینجمنٹ سسٹم

2۔ ایک جدول اس طرح جانی جاتی ہے۔

a) ٹیوپل
b) وصف (attribute)
c) تعلق (relation)
d) ہستی (entity)

3۔ کون سا ڈیٹا بیس ماڈل والدین اور بچہ کے تعلقات کی نمائندگی کرتا ہے؟

a) متعلقہ (relational)
b) نیٹ ورک (Network)
c) درجہ بندی (Hierarchical)
d) آبجیکٹ (Object)

4۔ متعلقہ ڈیٹا بیس ماڈل کو پہلے انہوں نے تجویز کیا۔

a) E F Codd
b) E E Codd
c) E F Cadd
d) E F Codder

5۔ درجہ بندی کا ماڈل (hierarchical model) کس قسم کے تعلقات کی نمائندگی کرتا ہے ؟

a) ایک سے ایک one-to-one
b) ایک سے کئی one-to-many
c) کئی سے ایک many-to-one
d) کئی سے کئی many-to-many

6۔ مندرجہ ذیل سے بابائے تعلقاتی ڈیٹا بیس (father of Relational database) کس کو کہتے ہیں؟

a) Chris Date
b) Hugh Darween
c) Edgar Frank Codd
d) Edgar Frank Cadd

7۔ مندرجہ ذیل میں سے کون ایک RDBMS ہے؟

a) Dbase
b) Foxpro
c) Microsoft Access
d) SQlite

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:21 PM

8۔ SELECT Statement کے لئے کون سی علامت استعمال ہوتی ہے؟

a) σ b) Π c) X d) Ω

9۔ ایک tuple، اس طرح بھی جانا جاتا ہے۔

a) جدول(table) b) صف (row) c) وصف (attribute) d) فیلڈ (field)

10۔ E R ماڈل کس نے تیار کیا؟

a) Chin b) E F Codd c) Chend d) Chand

حصہ - B

مندرجہ ذیل سوالات کا جوابات دیں۔ (2 مارکس)

1۔ ایک ڈیٹا بیس کی کچھ مثالوں کا ذکر کریں۔

2۔ RDBMS کے کچھ مثالوں کی فہرست بنائیے۔

3۔ ڈیٹا کی مستقل مزاجی (data consistency) کیا ہے؟

4۔ درجہ بندی (Hierarchical) اور نیٹ ورک (Net work) کے ڈیٹا ماڈل میں کیا فرق ہے؟

5۔ عام کرنا (Normalization) کیا ہے؟

حصہ - C

درج ذیل سوالات کے جوابات دیجئے۔ (3 مارکس)

1۔ SELECT اور PROJECT کمانڈ میں کیا فرق ہے ؟

2۔ DBA کا کیا کردار ہے ؟

3۔ ایک مثال کے ساتھ کارتیسی حاصل ضرب کی وضاحت کیجئے۔

4۔ ایک مثال کے ساتھ آبجیکٹ ماڈل کی وضاحت کیجئے۔

5۔ مختلف قسم کے DBMS صارفین (DBMS users) پر ایک نوٹ لکھئے۔

حصہ - D

مندرجہ ذیل سوالات کے جوابات دیجئے۔ (5 مارکس)

1۔ مختلف قسم کے ڈیٹا ماڈل کی وضاحت کریں۔

2۔ مختلف اقسام کے تعلقات کے نقشہ سازی (relationship mapping) کو بیان کریں۔

3۔ DBMS اور RDBMS میں فرق کیجئے۔

4۔ متعلقہ الجبرا میں مختلف آپریٹرز کو مناسب مثالوں کے ساتھ وضاحت کریں۔

5۔ RDBMS کے خصوصیات کی وضاحت کریں۔

12 CS UNIT 11 CDR Set
Monday, April 17, 2023 6:21:21 PM

اکائی IV

باب 12

اسٹرکچرڈ کوری لینگویج (Structured Query Language)

OPE46

سیکھنے کے مقاصد

اس سبق کا مطالعہ کرنے کے بعد، طلبا اس قابل ہوسکیں گے:

- SQL کی پروسیسنگ کی مہارت
- SQL کے اجزاء۔
- فیلڈوں اور ریکارڈوں کی وضاحت کرکے ایک جدول تیار کرنا۔
- مختلف برتنے والے اعمال جیسے inserting، updating، deleting، وغیرہ کرنا۔
- مختلف رکاوٹوں (Constraints) کے بارے میں جاننے کے لئے اور اسے جدول میں لاگو کرنا۔
- مختلف شقوں کو لاگو کرکے جدول میں سوالات (queries) پیدا کرنا۔
- کسی موجودہ جدول کی ساخت میں ترمیم کرنا۔
- ریکارڈس، جدول وغیرہ کو حذف کرنے کے کمانڈس، اور کمانڈس کو دوبارہ واپس کرنا۔

12.1 SQL کا تعارف Introduction to SQL

سٹرکچرڈ کوئری لینگوئج (SQL) ڈیٹا بیس تک رسائی حاصل کرنے اور اسے استعمال کرنے کے لئے پروگرامنگ کی ایک معیاری زبان ہے۔ SQL صارف کو ڈیٹا بیس کے مابین اطلاعات تیار کرنے، بازیافت کرنے، تبدیل کرنے اور منتقلی کی اجازت دیتا ہے۔ یہ ایک ایسی زبان ہے جو متعلقہ ڈیٹا بیس مینجمنٹ سسٹم (RDBMS) میں ڈیٹا کے نظم ونسق تک رسائی کے لئے ڈیزائن کی گئی ہے۔

SQL کے بہت سے ورژن ہیں۔ اصل ورژن IBM کے تحقیقی مرکز میں تیار کیا گیا تھا اور اصل میں اسے 1970 کے اوائل میں Sequel کہا جاتا تھا۔ بعد میں زبان کو SQL میں تبدیل کردیا گیا۔ 1986 میں، ANIS (امریکن نیشنل اسٹینڈرڈ انسٹی ٹیوٹ) نے ایک SQL اسٹینڈرڈ شائع کیا، جو 1992 میں ایک بار پھر اپ ڈیٹ ہوا تھا، تازہ ترین SQL، 2008 میں جاری کیا گیا تھا اور اسے SQL 2008 کے نام سے موسوم کیا گیا تھا۔

نوٹ

SQL کا تازہ ترین معیار SQL 2008 ہے، جسے 2008 میں جاری کیا گیا۔

12.2 RDBMS میں SQL کا کردار Role of SQL in RDBMS

RDBMS کا مطلب ہے Relational Database Management System۔ Oracle، MySQL، MS server، SQL، IBM DB2 اور Microsoft Access وغیرہ RDBMS کے packages ہیں۔ SQL ایسی زبان ہے، جسے ڈیٹا بیس میں ڈیٹا تک رسائی کے لئے استعمال کی جاتی ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:39 PM

عام طور پر، ڈیٹا بیس ان جدولوں کا ایک مجموعہ ہے جو ڈیٹا کے سیٹ کو اسٹور کرتا ہے جن کو دوسرے ایپلی کیشنز میں استعمال کرنے کے لئے استفسار (query) کیا جاسکتا ہے۔ ڈیٹا بیس مینجمنٹ سسٹم ڈیٹا بیس پلیٹ فارم کی ترقی، انتظامیہ اور استعمال کی حمایت کرتا ہے۔ RDBMS ایک قسم کا DBMS ہے جس میں ایک صف پر مبنی جدول کی ساخت ہے جو متعلقہ ڈیٹا عناصر کو جوڑتا ہے اور اس میں تیار کرنے، پڑھنے، اپ ڈیٹ اور حذف کرنے سے متعلق افعال شامل ہیں، جس کو اجتماعی طور پر CRUD کہا جاتا ہے۔

RDBMS میں موجود ڈیٹا کو ڈیٹا بیس اشیاء میں رکھا جاتا ہے، جسے ٹیبل کہتے ہیں۔ ایک ٹیبل متعلقہ ڈیٹا اندراجات کا ایک مجموعہ ہے اور اس میں صف اور قطار شامل ہیں۔

جدول کی ایک قطار (کالم) فیلڈ کہلاتی ہے جس کو جدول میں موجود ہر ریکارڈ کے بارے میں مخصوص متعلقہ معلومات کو برقرار رکھنے کے لئے ڈیزائن کیا گیا ہے۔ یہ ایک عمودی ہستی ہے جس میں ایک جدول میں مخصوص فیلڈ سے وابستہ تمام معلومات شامل ہیں۔ طلباء کی جدول میں موجود فیلڈ میں type، AdmnNo، StudName، StudAge، StudClass، Place وغیرہ ہوسکتے ہیں۔

ریکارڈ ایک صف ہے، جو متعلقہ فیلڈوں یا کالموں کا ایک مجموعہ ہے جو ایک جدول میں موجود ہے۔ ایک ریکارڈ کسی جدول میں ایک افقی ہستی ہوتی ہے جو طالب علم کی جدول میں کسی خاص طالب علم کے تفصیلات کی نمائندگی کرتی ہے۔

12.3 SQL کے پراسیسنگ کا ہنر (Processing Skills of SQL)

SQL کی پروسیسنگ کی مختلف مہارتیں یہ ہیں:

1۔ SQL DDL : Data Definition Language (DDL) نہ صرف تعلق کے schemas (ساختوں) کو واضح کرنے کا کمانڈ ہے، بلکہ relations کو delete کرنے، Indexes کو تیار کرنے، اور relation schemas کی ترمیم کے لئے بھی کمانڈ مہیا کرتے ہیں۔

2۔ SQL DML : Data Manipulation Language (DML) میں ڈیٹا بیس میں کمانڈ کو داخل کرنے، حذف کرنے اور ٹیوپل میں ترمیم کرنے کے احکامات شامل ہیں۔

3۔ SQL : Embedded Data Manipulation Language کی ایمبیڈڈ شکل اعلیٰ سطحی پروگرامنگ زبانوں میں استعمال ہوتی ہے۔

4۔ SQL : View definition میں جدولوں کو دیکھنے کی وضاحت کے لئے بھی احکامات شامل ہیں۔

5۔ اختیار (Authorization) : SQL میں تعلقات تک رسائی کے حقوق اور جدولوں کو دیکھنے کے کمانڈز شامل ہیں۔

6۔ سالمیت (Integrity) : SQL حالت کا استعمال کرتے ہوئے سالمیت کی جانچ پڑتال کے لئے فارم فراہم کرتا ہے۔

7۔ ٹرانزیکشن کنٹرول (Transaction control):SQL میں فائل ٹرانزیکشن کے کمانڈز اور ٹرانزیکشن پروسیسنگ پر قابو پانے شامل ہیں۔

کیا تم جانتے ہو

SQL- اسٹرکچرڈ کوویری لینگوئج ایک ایسی زبان ہے جو ڈیٹا بیس تک رسائی کے لئے استعمال کی جاتی ہے، جبکہ SQL ایک ڈیٹا بیس مینجمنٹ سسٹم ہے، جیسے SQL server، Oracle، Informix، Postgres، وغیرہ۔ MySQL ایک RDBMS ہے۔

ضمیمہ-1 میں MySQL نصب کرنے کا حوالہ دیں۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:39 PM

12.4 ڈیٹابیس تیار کرنا (Creating Database)

1۔ ایک ڈیٹابیس بنانے کے لئے prompt: میں درج ذیل کمانڈ ٹائپ کریں۔

```
CREATE DATABASE database_name;
```

مثال کے طور پر جدولوں میں ڈیٹابیش میں ذخیرہ کرنے کے لئے

```
CREATE DATABASE stud;
```

2۔ ڈیٹابیس کے ساتھ کام کرنے کے لئے، درج ذیل کمانڈ کو ٹائپ کریں۔

```
USE DATABASE;
```

مثال کے طور پر تیار کردہ stud database کو استعمال کرنے کے لئے یہ کمانڈ استعمال کریں۔

```
USE stud;
```

کیا تم جانتے ہو

WAMP کا مطلب ہے "Windows، Apache، MySQL اور PHP"۔ ونڈوز سسٹم کے لئے LAMP کا ایک متغیر WAMP ہے، اور اس کو سافٹ ویر کے بنڈل (Apache, MySQL , PHP) کے طور پر انسٹال کیا جاتا ہے۔ اس کو اکثر ویب ڈولپمنٹ اور اندرونی جانچ کے لئے استعمال کیا جاتا ہے، لیکن اس کا استعمال براہ راست live websites کو serve کرنے کے لئے بھی ہوتا ہے۔

12.5 SQL کے اجزاء (Components of SQL)

SQL کمانڈ کو پانچ قسموں میں تقسیم کیا گیا ہے۔

- DML - Data Manipulation Language
- DDL - Data Definition Language
- DCL - Data Control Language
- TCL - Transaction Control Language
- DQL - Data Query Language

12.5.1 ڈیٹا کے برتنے کی زبان (DDL) (Data Definition Language)

ڈیٹا ڈیفینیشن لینگویج (DDL) SQL بیانات پر مشتمل ہوتا ہے جو ڈیٹابیس کی ساخت یا Schema کی وضاحت کے لئے استعمال ہوتا ہے۔ یہ آسانی سے database schema کی وضاحت کے ساتھ کام کرتا ہے اور ڈیٹابیس میں ڈیٹابیس اشیاء کی ساخت کو بنانے اور اس میں ترمیم کرنے کے لئے استعمال ہوتا ہے۔

DDL، ڈیٹابیس سسٹم کے ذریعہ استعمال ہونے والے اسٹوریج کی ساخت اور رسائی کے طریقوں کی وضاحت کرنے کے لئے ایک مجموعہ فراہم کرتا ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:39 PM

DDL مندرجہ ذیل کام کرتا ہے:

1۔ اس کو ڈیٹا ڈویژن جیسے Data item، segment، record اور database file کی قسم کی شناخت کرنا چاہئے۔

2۔ یہ ہر ڈیٹا قسم جیسے، item type، record type، file type اور data base وغیرہ کو ایک انوکھا نام دیتا ہے۔

3۔ اس کو مناسب اعداد و شمار کی قسم کی وضاحت کرنا چاہئے۔

4۔ اس کو ڈیٹا آئٹم کے سائز کی وضاحت کرنی چاہئے۔

5۔ یہ قیمتوں کی حد کی وضاحت کرسکتا ہے جو ڈیٹا آئٹم استعمال کرسکتی ہے۔

6۔ یہ غیر اختیار ڈیٹا داخلے کو روکنے کے لئے اور خفیہ تالے (privacy locks) کی وضاحت کرسکتا ہے۔

SQL کے کمانڈ جو ڈیٹا ڈیفینیشن لینگویج (DDL) کے تحت آتے ہیں، وہ یہ ہیں:

Create	ڈیٹا بیس میں ٹیبل بنانے کے لئے۔
Alter	ڈیٹا بیس کی ساخت کو تبدیل کرنے کے لئے۔
Drop	ڈیٹا بیس سے ٹیبل ڈیلیٹ کرنے کے لئے۔
Truncate	تمام ٹیبل سے تمام ریکارڈز کو ہٹادیں، ان ریکارڈوں کی جگہ جگہ بھی جاری کریں۔

12.5.2 ڈیٹا کو برتنے کی زبان (DML) (Data Manipulation Language)

ڈیٹا کو برتنے کی زبان (DML) (Data Manipulation Language) ایک کمپیوٹر پروگرامنگ زبان ہے، جو ڈیٹا بیس میں ڈیٹا کو شامل کرنے (داخل کرنے)، حذف کرنے (ڈیلیٹ کرنے) اور ترمیم (اپ ڈیٹ) کرنے کے لئے استعمال کی جاتی ہے۔ SQL میں، ڈیٹا کو برتنے کی زبان میں SQL-ڈیٹا کی تبدیلی کے بیانات شامل ہیں، جو ذخیرہ شدہ ڈیٹا میں ترمیم کرتے ہیں لیکن ڈیٹا بیس ٹیبل کی schema نہیں۔

Database schema کی وضاحت اور ڈیٹا بیس کو واضح کرنے کے بعد، DML کے ذریعہ اظہار کردہ طریقہ کار کے ایک سیٹ کا استعمال کرکے ڈیٹا کو جوڑ توڑ کیا جاسکتا ہے۔

ڈیٹا کو برتنے سے ہمارا مطلب یہ ہے :

- ڈیٹا بیس میں نئی معلومات شامل کرنا۔
- ڈیٹا بیس میں ذخیرہ شدہ معلومات کی بازیافت۔
- ڈیٹا بیس سے معلومات کو حذف کرنا۔
- ڈیٹا بیس میں محفوظ ڈیٹا میں ترمیم۔

بنیادی طور پر DML دو قسموں کا ہوتا ہے۔

Procedural DML - کسی صارف سے یہ بیان کرنے کی ضرورت ہوتی ہے کہ کون سے ڈیٹا کی ضرورت ہے اور اسے کیسے حاصل کیا جائے۔

Non-Procedural DML- کسی صارف سے یہ بتانے کی ضرورت ہوتی ہے کہ کس ڈیٹا کو حاصل کرنے کا طریقہ بتائے بغیر اس کی ضرورت ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:40 PM

ڈیٹا کو برتنے والی زبان (DML) کے تحت آنے والے SQL کمانڈز یہ ہیں :

Insert	جدول میں ڈیٹا کو داخل کرتا ہے۔
Update	ٹیبل کے اندر موجود ڈیٹا کو اپ ڈیٹ کرتا ہے۔
Delete	حذف کریں ایک میز سے تمام ریکارڈز کو حذف کرتا ہے، لیکن ان کے ذریعہ خالی شدہ جگہ کو نہیں۔

12.5.3 ڈیٹا کنٹرول کی زبان (DCL) (Data Control Language)

ڈیٹا کنٹرول کی زبان (DCL) (Data Control Language) ایک پروگرامنگ زبان ہے، جو ڈیٹا بیس میں محفوظ ڈیٹا تک رسائی کو کنٹرول کرنے کے لئے استعمال ہوتی ہے۔اس کا استعمال ڈیٹا بیس کے اختیار کو کنٹرول کرنے کے لئے کیا جاتا ہے۔ ڈیٹا بیس کے تمام کام انجام دینے کے لئے مراعات کی ضرورت ہوتی ہے، جیسے سلسلہ بنانا، جدول کو دیکھنا وغیرہ۔

SQL کمانڈز جو ڈیٹا کنٹرول لینگویج کے تحت آتے ہیں، وہ یہ ہیں۔

Grant	ایک یا زیادہ صارفین کو مخصوص کام انجام دینے کی اجازت دیتا ہے۔
Revoke	GRANT بیان کے ذریعہ دی گئی رسائی کی اجازت واپس لے لیں۔

12.5.4 ٹران سیکشنل کنٹرول زبان (TCL) (Transactional Control Language)

ٹرانسیکشنل کمانڈز ڈیٹا بیس میں لین دین کا انتظام کرنے کے لئے استعمال ہوتے ہیں۔ان کا استعمال DML کے بیانات کے ذریعہ ڈیٹا میں کی جانے والی تبدیلیوں کو منظّم کرنے کے لئے کیا جاتا ہے۔

Transfer Control Language کے تحت آنے والے SQL کمانڈ یہ ہیں:

Commit	ڈیٹا بیس میں کسی بھی لین دین کو مستقل طور پر بچاتا ہے۔
Rollback	آخری بیک ریاست میں ڈیٹا بیس کو بحال کریں۔
Save point	پوائنٹ محفوظ کریں عارضی طور پر ٹرانسیکشن کو بچائیں، تاکہ آپ رول بیک ہوسکیں۔

12.5.5 ڈیٹا QUERY کی زبان (Data Query Language)

ڈیٹا کوئری لینگویج میں ڈیٹا بیس سے ڈیٹا کے استفسار (Query) یا بازیافت کرنے کے لئے استعمال ہونے والے کمانڈز شامل ہیں۔

Data Query Language میں ایسی ہی ایک SQL کمانڈ ہے۔

Select	یہ میز سے ریکارڈ دکھاتا ہے۔

12.6 ڈیٹا کی اقسام (Data Types)

ڈیٹا بیس میں موجود ڈیٹا، اس میں محفوظ کردہ قیمت کی بنیاد پر محفوظ کیا جاتا ہے۔ اس کی شناخت ڈیٹا کی ڈیٹا ٹائپ کی حیثیت سے ہوتی ہے یا ہر فیلڈ میں ڈیٹا کی قسم تفویض کرکے۔ دئے گئے فیلڈ میں تمام قیمتیں ایک ہی قسم کی ہونی چاہئے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:40 PM

ANSI SQL معیار صرف متن اور اعداد و شمار کے ڈیٹا کی قسم کو تسلیم کرتا ہے، جبکہ کچھ تجارتی پروگرام دوسرے ڈیٹا ٹائپ جیسے تاریخ اور وقت وغیرہ کا استعمال کرتے ہیں۔ ANSI کے ڈیٹا کی اقسام کو جدول 12.1 میں نیچے درج کیا گیا ہے۔

DataType	Description وضاحت
char (علامتیں)	اسٹرنگ کے چوڑائی کی قیمت متعین ہوتی ہے۔ اس قسم کی قیمتوں کو اکہرے واوین ' کے اندر بند کیا جاتا ہے۔ مثال کے طور پر 's Anu کو 'Anu' 's' لکھا جاتا ہے۔
varchar	اسٹرنگ کے چوڑائی کی قیمت غیر متعین ہوتی ہے۔ یہ char ہی کے مشابہ ہے، مگر اس میں ڈیٹا کے اندراج کی مقدار کافی مختلف ہوتی ہے۔
(اعشاریہ) dec (Decimal)	یہ ایک عشری اعداد کی نمائندگی کرتا ہے، جیسے 15.12، 0.123 وغیرہ۔ یہاں سائز کی دلیل (argument) دو حصوں پر مشتمل ہے: درستگی (precision) اور پیمانہ (scale) ۔ درستگی، یہ اشارہ کرتا ہے کہ عدد میں کتنے ہندسے ہو سکتے ہیں اور پیمانہ، اعشاریہ کے دائیں جانب موجود ہندسوں کی تعداد کو ظاہر کرتا ہے۔ جسامت (5,2) یہ ظاہر کرتی ہے کہ اس میں precision 5 اور scale 2 ہے۔ scale کی قیمت ہمیشہ، precision کی قیمت سے کم ہوگی۔
numeric	یہ اعشاریہ ہی کی طرح ہی ہے سوائے یہ کہ اس میں موجود ہندسوں کی تعداد precision argument سے زیادہ نہیں ہو سکتی۔
int (سالم عدد) (Integer)	یہ ایسے اعداد کو ظاہر کرتا ہے، جس میں اعشاریہ نہیں ہوتے۔ اس لئے یہاں size argument استعمال نہیں ہوتا
smallint	یہ بھی سالم عدد ہی کی طرح ہے، مگر default size، سالمی عدد سے چھوٹا ہوگا۔
float	قاعدہ 10 کے نظام میں، قوت نمائی ترقیم میں یہ ایک چھوٹا floating point کا عدد ہے اور کسی precision کو ظاہر کرنے کے لئے اس کی وضاحت زیادہ سے زیادہ 64 تک کی جاتی ہے۔
real	یہ بھی float ہی کی طرح ہے، سوائے اس کے کہ اس میں size argument کو استعمال نہیں کیا جاتا اور اس کے precision کی وضاحت بھی زیادہ سے زیادہ 64 تک کی جاتی ہے۔
double (دگنا)	real ہی کی طرح سوائے اس کے کہ اس میں precision سے زیادہ ہو سکتی ہے۔

جدول 12.1

12.7 SQL کے کمانڈس اور ان کے افعال (SQL Commands and their Functions)

ڈیٹا کو ذخیرہ کرنے کا واحد طریقہ جدول ہیں، لہذا تمام معلومات کو جدول کی شکل میں ترتیب دینا ہوگا۔ SQL ڈیٹا بیس پر کام کرنے کے لئے پہلے سے طے شدہ کمانڈ فراہم کرتا ہے۔

Keywords (کلیدی الفاظ) SQL میں ان کا ایک خاص معنی ہے۔ وہ ہدایات کے طور پر سمجھے جاتے ہیں۔

Commands (احکامات) وہ صارف کی طرف سے ڈیٹا بیس کو دی جانے والی ہدایات ہیں جو statements بھی کہلاتے ہیں۔

Clauses (شقیں) وہ ایک کلیدی لفظ کے ساتھ شروع ہوتا ہے اور کلیدی لفظ (keyword) اور دلیل (argument) پر مشتمل ہوتے ہیں۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:41 PM

Arguments (دلائل) ایک clause کو مکمل کرنے کے لئے دی گئی قیمتیں arguments کہلاتی ہیں۔

12.7.1 DDL کے کمانڈز

CREATE TABLE command

آپ CREATE TABLE کمانڈ کا استعمال کرکے ایک ٹیبل تشکیل دے سکتے ہیں۔ جب ایک ٹیبل بن جاتا ہے تو، اس کے کالموں (قطاروں) کو نام جاتا ہے، ڈیٹا کی اقسام اور سائز کی وضاحت کی جاتی ہے۔ ہر ٹیبل میں کم از کم ایک کالم ہونا ضروری ہے۔

CREATE TABLE کمانڈ کے لئے Syntax یہ ہے۔

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]
(<column name><data type>[<size>].....
);
```

اب ہم کلاس کے طلباء کے بارے میں کچھ معلومات کو ڈیٹا بیس میں ذخیرہ کرنے کے لئے مذکورہ syntax کا استعمال کریں گے۔ اس کے لئے، آپ کو پہلے ٹیبل بنانے کی ضرورت ہے۔ طلباء کی جدول بنانے کے لئے، آیئے ہم طلباء سے متعلق کچھ معلومات لیتے ہیں جیسے admission number، جسے ہم اسے مختصر شکل میں (admno)، طالب علم کا نام (name)، صنف، عمر، مقام، وغیرہ استعمال کر سکتے ہیں۔ آیئے ایک ٹیبل تیار کریں جس میں فیلڈ کے نام Admno ، Name، Gender، Age اور Place ہوں۔

SQL کمانڈ حسب ذیل ہوگا:

```
CREATE TABLE Student
    (Admno   integer,
     Name  char(20),
     Gender  char(1),
     Age     integer,
     Place char(10),
     );
```

مذکورہ بالا ایک سادہ جدول کی ساخت ہے، جو بغیر کسی پابندی کے ہے۔ آپ Data کی قسم کو محدود کرنے کے لئے رکاوٹیں (constraints) بھی مرتب کر سکتے ہیں، جس کے لئے تم جدول کی fields میں جاسکتے ہو۔ constraints اس طرح کے Data کو محدود کرنے کے لئے استعمال کی جاتی ہیں جو کسی ٹیبل میں جاسکتی ہیں۔ یہ ڈیٹا بیس میں موجود ڈیٹا کی درستگی (accuracy) اور اعتبار (reliability) کو یقینی بناتا ہے۔ رکاوٹیں یا تو کالم کی سطح یا جدول کی سطح پر ہوسکتی ہیں۔

نوٹ

رکاوٹیں (Constraints)، ایک ایسی شرط ہے جو ایک فیلڈ یا فیلڈ کے ایک سیٹ پر عمل پذیر ہوتی ہیں۔

Column constraint: کالم کی رکاوٹ — کالم کی رکاوٹ صرف انفرادی کالم پر لاگو ہوتی ہے۔

Table constraint : ٹیبل کی رکاوٹ — ٹیبل کی رکاوٹ ایک یا زیادہ کالموں کے گروپ پر لاگو ہوتی ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:41 PM

رکاوٹ کے ساتھ بنائے گئے ٹیبل کے نحو کو ذیل میں بتایا گیا ہے۔

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]<column constraint>,
(<column name><data type>[<size>]<column constraint>……
<table constraint>(<column name>,[<column name>….])…..
);
```

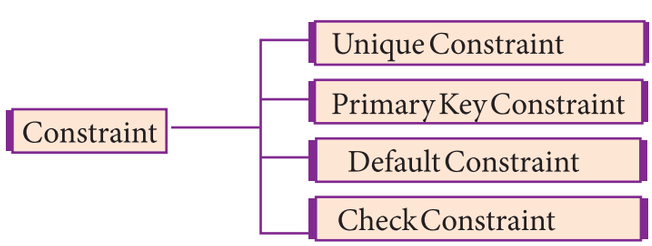
طلباء کی جدول کے لئے مندرجہ ذیل ایک مثال ہے جس میں کالم رکاوٹ "NOT NULL" ہے۔ یہ رکاوٹ ایک فیلڈ کو ہمیشہ کچھ نہ کچھ قیمت رکھنے کے لئے نافذ کرتی ہے۔

```
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),
);
```

مندرجہ بالا کمانڈ ایک ٹیبل "student" تیار کرتا ہے جس میں Admno کے فیلڈ، جو سالم عدد کی قسم کے ہوتے ہیں، اس کی وضاحت NULL NOT کی طرح کی گئی ہے۔ جس کا مطلب ہے کہ ان دونوں فیلڈز میں قیمتیں ہونی چاہئیں۔ صنف، عمر اور مقام کے کھیتوں میں کوئی رکاوٹ نہیں ہے۔

12.7.2 رکاوٹوں کی قسم (Type of Constraints)

رکاوٹیں ڈیٹا بیس کی سالمیت کو یقینی بناتی ہیں، لہذا ڈیٹا بیس کی سالمیت کو رکاوٹوں کے طور پر جانا جاتا ہے۔ مختلف قسم کی رکاوٹیں یہ ہیں:



12.7.2.1 انوکھی پابندی (Unique Constraint)

یہ رکاوٹ کو یہ یقینی بناتی ہے کہ مخصوص کالموں (قطاروں) میں کسی بھی دو صفوں میں ایک جیسی قیمت نہیں ہونی چاہئے۔ مثال کے طور پر طلباء کی جدول کے Admno پر UNIQUE constraint کا اطلاق ہوتا ہے، جس سے یہ یقینی ہو جاتا ہے کہ کوئی بھی دو طلباء کا admission number ایک جیسا نہیں ہونا چاہئے، ایسے موقع پر اس رکاوٹ کو استعمال کیا جا سکتا ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:42 PM

```
CREATE TABLE Student
(
Admno integer NOT NULL UNIQUE, → Unique constraint
Name char (20) NOT NULL,
Gender char (1),
Age integer,
Place char (10),
);
```

انوکھی رکاوٹوں کا اطلاق صرف ان فیلڈز پر ہوسکتا ہے جن کو NOT NULL کے طور پر اعلان کیا گیا ہے۔

جب ایک ہی میدان میں دو رکاوٹیں لگائی جاتی ہیں تو، اسے متعدد رکاوٹوں (multiple constraints) کے نام سے جانا جاتا ہے۔ مندرجہ بالا ایک سے زیادہ رکاوٹوں میں NOT NULL اور UNIQUE کا اطلاق کسی ایک فیلڈ Admno پر ہوتا ہے۔ رکاوٹوں کو الگ کرنے کے لئے space استعمال ہوتا ہے اور field definition کے اختتام پر ایک کوما (،) شامل ہوجاتا ہے۔ ان دو رکاوٹوں کو شامل کر کے فیلڈ Admno کو کچھ قدر لینا چاہئے۔ یعنی اس میں NULL نہیں ہوگا اور اسے نقل (duplicate) نہیں کیا جانا چاہئے۔

12.7.2.2 بنیادی کلیدی رکاوٹ (Primary Key Constraint)

یہ رکاوٹ کسی فیلڈ کو پرائمری کلید قرار دیتی ہے جو ریکارڈ کو الگ الگ شناخت کرنے میں معاون ہے۔ یہ انوکھا رکاوٹ کی طرح ہے سوائے اس کے کہ کسی ٹیبل کے صرف ایک فیلڈ کو بنیادی کلید مقرر کیا جاسکے۔ پرائمری کلید NULL قدروں کی اجازت نہیں دیتی ہے اور اس وجہ سے کسی فیلڈ کو بنیادی کلید کے طور پر NOT NULL رکاوٹ کو اعلان کیا جانا ضروری ہے۔

طالب علم کی جدول میں بنیادی کلیدی رکاوٹ (Primary Key Constraint) ظاہر کرنے کی مثال :

```
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),
);
```

مندرجہ بالا مثال میں Admno فیلڈ کو بنیادی کلید کے طور پر مرتب کیا گیا ہے اور اس وجہ سے ہمیں ریکارڈ کو منفرد انداز میں شناخت کرنے میں مدد ملے گی، اس کو بھی NOT NULL رکھا گیا ہے، لہذا اس فیلڈ کی قیمت خالی نہیں ہوسکتی ہے۔

12.7.2.3 ڈیفالٹ رکاوٹ (DEFAULT Constraint)

DEFAULT رکاوٹ کو فیلڈ کیلئے ڈیفالٹ قیمت تفویض کرنے کے لئے استعمال کیا جاتا ہے۔ جب DEFAULT رکاوٹ والے مخصوص فیلڈ کے لئے کوئی قیمت نہیں دی جاتی ہے تو، خود بخود پہلے سے طے شدہ قیمت فیلڈ کو تفویض کردی جائے گی۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:42 PM

مثال کے طور پر طلباء کی جدول میں DEFAULT رکاوٹ ظاہر کرنا:

```
CREATE TABLE Student
(
Admno integer   NOT NULL  PRIMARY KEY,
Name char(20)NOT NULL,
Gender char(1),
Age integer  DEFAULT = "17", → Default Constraint
Place char(10),
);
```

مندرجہ بالا مثال میں "Age" فیلڈ میں 17 کی ڈیفالٹ قیمت تفویض کی گئی ہے، لہذا جب صارف کے ذریعہ عمر میں کوئی قیمت داخل نہیں کی جاتی ہے تو، یہ خود بخود 17 کو عمر کی تفویض کر دیتا ہے۔

12.7.2.4 رکاوٹ چیک کریں (Check Constraint)

یہ رکاوٹ فیلڈ کے لئے رکھی گئی حد کی قیمت طے کرنے میں معاون ہے۔ جب ہم کسی ایک کالم پر check constraint کی وضاحت کرتے ہیں تو، اس فیلڈ میں صرف محدود قیمت کی اجازت دیتا ہے۔ مثال کے طور پر طالب علموں کی جدول میں check constraint نیچے بتایا گیا ہے۔:

```
CREATE TABLE Student
(
Admno integer   NOT NULL PRIMARY KEY
Name char(20)NOT NULL,
Gender char(1),
Age integer (CHECK<=19), → Check Constraint
Place char(10),
);
```

مذکورہ مثال میں Check Constraint، عمر کے فیلڈ پر متعین ہے، جہاں عمر کی قیمت 19 سے کم یا اس کے برابر ہونی چاہئے۔



نوٹ:

check constraint شرط کے لئے تعلقی اور منطقی آپریٹرز کا استعمال کر سکتی ہے۔

12.7.2.5 ٹیبل رکاوٹ (Table Constraint)

جب table constraint کے فیلڈز کے گروپ پر لاگو ہوتی ہے تو، اسے table constraint کے نام سے جانا جاتا ہے۔ ٹیبل کی رکاوٹ عام طور پر ٹیبل تعریف کے آخر میں دی جاتی ہے۔ آیئے مندرجہ ذیل فیلڈز Admno، Firstname ، Lastname، Gender ، Age، Place: کے ذریعہ کے ساتھ ایک نئی جدول لیں۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:42 PM

```
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
);
```

مندرجہ بالا مثال میں دو فیلڈ ہوتے ہیں۔ Firstname اور Lastname بنیادی کلید کے طور پر بیان کیے گئے ہیں، جو ایک table constraint ہے۔

12.7.3 DML کمانڈز (DML Commands)

جب جدول کی schema یا ساخت تیار ہو جائے تو ٹیبل میں قیمتیں شامل کی جا سکتی ہیں۔

DML کمانڈز جدول میں صفیں داخل کرنے، حذف کرنے اور تازہ کاری کرنے پر مشتمل ہیں۔

12.7.3.1 (i) INSERT کمانڈ (INSERT Command)

INSERT کمانڈ، ڈیٹا بیس میں نیا ڈیٹا شامل کرنے یا ٹیبل میں نئے ریکارڈ شامل کرنے میں مدد کرتی ہے۔ کمانڈ کا استعمال مندرجہ ذیل ہے۔

```
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 (102, 'Adarsh', 'M', 18, 'Delhi');
```

دو جدید ریکارڈوں کو جدول میں شامل کیا جاتا ہے، جیسا کہ نیچے بتایا گیا ہے۔

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi

قیمتوں کی ترتیب، CREATE TABLE کمانڈ کے کالموں کی ترتیب سے ملنا چاہیئے۔ اگر تمام کالموں میں ڈیٹا کو داخل کیا جانا چاہیئے تو کالموں کے نام کو مخصوص کرنا اختیاری ہوگا۔ طالب علم کی جدول میں قیمتوں کو داخل کرنے کے لئے کمانڈ کو درج ذیل طریقہ سے بھی استعمال کیا جا سکتا ہے

```
INSERT INTO Student VALUES ( 102, 'Akshith', 'M','17',' Bangalore');
```

102	Akshith	M	17	Bangalore

مذکورہ کمانڈ طلبہ کی جدول میں ریکارڈ داخل کرتی ہے۔

صرف بعض کالموں میں ڈیٹا کو شامل کرنے کے لئے جس میں کالم کا نام اور اس کا ڈیٹا مخصوص کیا جاتا ہے۔ یہ اس طرح سے کیا جا سکتا ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:43 PM

INSERT INTO Student(Admno, Name, Place) VALUES (103, 'Ayush', 'Delhi');

103	Ayush	M	18	Delhi

اوپر کا کمانڈ درج ذیل ریکارڈ کو شامل کرتا ہے جس کی ڈیفالٹ قیمت جنس کے لئے M ہوتی ہے اور عمر کے لئے 18۔

INSERT INTO Student (Admno, Name, Place) VALUES (104, 'Abinandh', 'Chennai');

104	Abinandh	M	18	Chennai

طالب علم کی جدول میں درج ذیل ڈیٹا ہوتے ہیں۔

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 کمانڈ میں فیلڈ نہیں دئے گئے ہیں اور یہ ڈیفالٹ قیمتیں لے لے گا، اگر ان کو متعین کیا گیا ہو تو، ورنہ اس میں NULL محفوظ کیا جائے گا۔

کیا تم جانتے ہو

INSERT کمانڈ میں وہ فیلڈ جو چھوڑ دی گئی ہے، وہ سب متعین شدہ ڈیفالٹ قیمتیں یا NULL قیمتیں لے لے گا۔

12.7.3.2 DELETE کمانڈ (DELETE COMMAND)

DELETE Command، مستقل طور پر ٹیبل سے ایک یا زیادہ ریکارڈز کو ہٹا دیتا ہے۔ یہ پورے صف کو ہٹا دیتا ہے، صرف صف کے انفرادی فیلڈز کو نہیں۔ چنانچہ فیلڈ دلیل (Field argument) کی ضرورت نہیں ہے۔ DELETE کمانڈ کا استعمال مندرجہ ذیل ہے۔

DELETE FROM table-name WHERE condition;

مثال کے طور پر ریکارڈ کو حذف کرنا، جس کا داخلہ نمبر 104 ہے کمانڈ ذیل میں دیا گیا ہے:

DELETE FROM Student WHERE Admno=104;

104	Abinandh	M	18	Chennai

مندرجہ ذیل ریکارڈ طلباء کی جدول سے حذف ہو گیا ہے۔

جدول کی تمام صفوں کو حذف کرنے کے لئے، کمانڈ کا استعمال اس طرح ہوتا ہے:

DELETE * FROM Student;

جدول اب خالی ہوگی اور DROP کمانڈ کو ختم کیا جا سکتا ہے۔ (سیکشن 12.7.4.3 میں بحث کیا گیا ہے)۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:43 PM

12.7.3.3 UPDATE کمانڈ (UPDATE Command)

UPDATE کمانڈ ڈیٹا بیس میں کچھ یا تمام ڈیٹا ویلیوز کو اپ ڈیٹ کرتی ہے۔ یہ ایک ٹیبل میں ایک یا زیادہ ریکارڈز کو اپ ڈیٹ کرسکتا ہے۔ اپ ڈیٹ کمانڈ صفوں کی وضاحت کرتی ہے کہ WHERE شق اور SET Keyword کا استعمال کرتے ہوئے نیا ڈیٹا استعمال کرتا ہے۔ کمانڈ کا استعمال مندرجہ ذیل ہے۔

UPDATE <table-name> SET column-name = value, column-name = value,…
WHERE condition;

فیلڈز کو اپ ڈیٹ کرنے کے لئے ایک مثال درج ذیل ہے۔

UPDATE Student SET Age = 20 WHERE Place = “Bangalore”;

اوپر کی کمانڈ ان طلباء کی عمر کو 20 بنائے گا، جن کا مقام Bangalore ہے۔ جدول اس طرح Update کی گئی ہے۔

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

متعدد فیلڈوں کو update کرنے کے لئے، متعدد فیلڈ کی تفویض کرنے کی شق SET، ہوگی، جس کو comma سے الگ کیا گیا ہے۔ جدول کو نیچے کی طرح اپ ڈیٹ کیا جائے گا:

UPDATE Student SET Age=18, Place = ‘Chennai’ WHERE Admno = 102;

102	Akshith	M	18	Chennai

درج بالا کمانڈ، ریکارڈ کو درج ذیل طریقہ سے ترمیم کرتا ہے۔

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 بعد اضافی DDL کمانڈس (Some Additional DDL commands)

12.7.4.1 ALTER COMMAND

جدول کی ساخت جیسے کالم کو شامل کرنا، موجودہ کالم کے نام کو تبدیل کرنا، کسی کالم کے ڈیٹا کی قسم کو تبدیل کرنا یا کالم کی جسامت کو تبدیل کرنا، جدول کو حذف کرنا، وغیرہ کے لئے ALTER کمانڈ استعمال کیا جاتا ہے۔ اس کو اس طرح کیا جاتا ہے۔

ALTER TABLE <table-name> ADD <column-name><data type><size>;

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:44 PM

طالب علم کی جدول میں char قسم کے Address میں نیا کالم شامل کرنے کے لئے یہ کمانڈ استعمال کیا جاتا ہے۔

ALTER TABLE Student ADD Address char;

جدول میں موجود کالم کی ترمیم کے لئے ALTER TABLE کمانڈ کو MODIFY شق کے ساتھ اس طرح استعمال کیا جاتا ہے۔

ALTER <table-name> MODIFY<column-name><data type><size>;

ALTER TABLE Student MODIFY Address char (25);

The above command will modify the address column of the Student table to now hold 25 characters.

موجودہ کالم کے نام کو تبدیل کرنے کے لئے ALTER کمانڈ کو اس طرح استعمال کیا جاتا ہے۔

ALTER <table-name> RENAME old-column-name TO new-column-name;

طلباء کی جدول سے کالم CITY کو نکالنے کے لئے یہ کمانڈ استعمال کیا جاتا ہے۔

ALTER TABLE Student RENAME Address TO City;

The **ALTER** command can also be used to remove a column or all columns, for example to remove a particular column, the **DROP COLUMN** is used with the **ALTER TABLE** to remove a particular field, the command can be used as:

ALTER <table-name> DROP COLUMN <column-name>;

To remove the column City from the Student table, the command is used as :

ALTER TABLE Student DROP COLUMN City;

(ii) TRUNCATE command — 12.7.4.2 TRUNCATE کمانڈ

جدول میں موجود تمام صفوں کو حذف کرنے کے لئے TRUNCATE کمانڈ استعمال ہوتا ہے۔ ساخت یوں ہی رہتی ہے اور جدول سے جگہ خالی ہو جاتی ہے۔ TRUNCATE کمانڈ کے لئے syntax یہ ہے۔

TRUNCATE TABLE table-name;

مثال کے طور پر طالب علم کے جدول میں موجود تمام record کو delete کرنے کے لئے اور جدول کو delete کرنے کے لئے SQL بیان اس طرح دیا جاتا ہے۔

TRUNCATE TABLE Student;

student نامی جدول نکل جاتا ہے (خارج ہو جاتا ہے) اور جگہ خالی ہو جاتی ہے۔

DROP TABLE Command 12.7.4.3

DROP TABLE کمانڈ ڈیٹا بیس سے ٹیبل کو ہٹانے کے لئے استعمال کیا جاتا ہے۔ اگر آپ ٹیبل ہٹا دیتے ہیں تو ٹیبل کی ساری صفیں حذف ہو جاتی ہیں اور ٹیبل ڈھانچے کو ڈیٹا بیس سے ہٹا دیا جاتا ہے۔ ایک بار ٹیبل ہٹا دیتے ہیں تو ہم اسے واپس نہیں لا سکتے ہیں، لہذا ڈراپ ٹیبل کمانڈ کا استعمال کرتے وقت محتاط رہیں۔ لیکن جدول کو خارج کرنے کی بھی ایک شرط ہے۔ یہ جدول خالی ہونا چاہئے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:44 PM

DELETE کمانڈ کا استعمال کرتے ہوئے ٹیبل کی تمام صفوں کو ہٹائیں۔ DELETE کمانڈ کی وضاحت پہلے ہی ہو چکی ہے۔

تمام صفوں کو حذف کرنے کے لئے، کمانڈ اس طرح دیا گیا ہے:

DELETE * FROM Student;

ایک بار جب تمام صفیں حذف ہو جائیں گی، تو DROP TABLE کمانڈ کے ذریعہ ٹیبل کو مندرجہ ذیل طریقے سے حذف کیا جا سکتا ہے۔

DROP TABLE table-name;

مثال کے طور پر اسٹوڈنٹ ٹیبل کو حذف کرنا:

DROP TABLE Student;

DELETE, TRUNCATE AND DROP statement:

حذف کریں DELETE	DELETE کمانڈ اس شرط کی بنیاد پر ٹیبل سے صرف صفیں حذف کرتا ہے جہاں شق کی وضاحت کی گئی ہے یا اگر کوئی شرط نہیں بتائی گئی ہے تو وہ ٹیبل سے تمام صفیں حذف کر دیتی ہے۔ لیکن یہ جدول پر مشتمل جگہ کو آزاد نہیں کرتا ہے۔
ٹرنکائٹ TRUNCATE	TRUNCATE کمانڈ کا استعمال تمام صفوں کو حذف کرنے کے لئے کیا جاتا ہے، جدول کی ساخت میں رہتا ہے اور جدول پر مشتمل جگہ کو آزاد کرتا ہے۔
DROP	DROP کمانڈ کا استعمال ڈیٹا بیس سے کسی شے کو ہٹانے (خارج) کے لئے کیا جاتا ہے۔ اگر آپ ٹیبل خارج کر دیتے DROP ہیں تو، ٹیبل کی ساری صفیں حذف ہو جاتی ہیں اور ٹیبل کی ساخت کو ڈیٹا بیس سے ہٹا دیا جاتا ہے۔ ایک بار جدول ہٹا دی ڈراپ جائے تو ہم اسے واپس نہیں لا سکتے ہیں۔

SELECT command – DQL COMMAND 12.7.5

SQL کے ساتھ کام کرتے وقت ایک سب سے اہم کام سوالات (queries) پیدا کرنا اور ڈیٹا کو بازیافت کرنا ہے۔ ڈیٹا بیس ٹیبل سے مطلوبہ نتیجہ حاصل کرنے کے لئے دیا گیا کمانڈ query کہلاتا ہے۔ SELECT کمانڈ ڈیٹا بیس میں موجود ٹیبل سے ڈیٹا کو پوچھ گچھ کرنے یا بازیافت کرنے کے لئے استعمال ہوتا ہے۔ یہ ایک یا ایک سے زیادہ جدولوں سے ریکارڈوں کے سب سیٹ کو بازیافت کرنے کے لئے استعمال ہوتا ہے۔ SELECT کمانڈ کو مختلف شکلوں میں استعمال کیا جا سکتا ہے۔

SELECT کمانڈ کے لئے syntax :

SELECT <column-list>FROM<table-name>;

- Table-name اس ٹیبل کا نام ہے، جہاں سے معلومات کی بازیافت کی جاتی ہے۔
- Column-list میں ایک یا زیادہ کالم شامل ہیں، جہاں سے ڈیٹا بازیافت کیا جاتا ہے۔

مثال کے طور پر اسٹوڈنٹ ٹیبل سے صرف داخلہ نمبر اور طلباء کا نام دیکھنے کے لئے کمانڈ مندرجہ ذیل ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:44 PM

اگر طلباء کی جدول میں درج ذیل ڈیٹا موجود ہے تو:

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 :

مذکورہ بالا کمانڈ مندرجہ ذیل ڈیٹا کو ظاہر کرے گی۔

Admno	Name
100	Ashish
101	Adarsh
102	Akshith
103	Ayush
104	Abinandh
105	Revathi
106	Devika
107	Hema

جدول کے تمام فیلڈوں اور صفوں کو دیکھنے کے لئے SELECT COMMAND کو اس طرح استعمال کیا جاتا ہے۔

SELECT * FROM STUDENT;

DISTINCT keyword 12.7.5.1

SELECT command کے ساتھ DISTINCT keyword استعمال کر کے جدول کی duplicate صفوں کو خارج کیا جاتا ہے۔ اس سے فضول ڈیٹا (redundant data) کو خارج کرنے میں مددگار ہے۔ مثال کے طور پر :

SELECT DISTINCT Place FROM Student;

مندرجہ ذیل اعداد و شمار کو ظاہر کرے گا:

SELECT * FROM Student;

Place
Chennai
Bangalore
Delhi

مندرجہ بالا آؤٹ پٹ میں آپ دیکھ سکتے ہیں، جگہ کے فیلڈ میں کوئی نقل والی صفیں نہیں ہوں گی۔ جب DISTINCT Keyword استعمال کیا جاتا ہے تو، صرف ایک ہی NULL قیمت واپس آجاتی ہے، یہاں تک کہ اگر زیادہ NULL قیمتیں پائی جائیں۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:45 PM

ALL keyword 12.7.5.2

ALL Keyword ڈپلیکیٹ صفیں برقرار رکھتی ہیں۔ یہ ڈپلیکیٹ اندراجات پر غور کیے بغیر ٹیبل کی ہر صف کو ظاہر کرے گا۔

SELECT ALL Place FROM Student:

مذکورہ بالا کمانڈ ڈپلیکیٹ اندراجات پر غور کیے بغیر ٹیبل کی ہر صف سے جگہ کے فیلڈ کی تمام قیمتوں کو ظاہر کرے گا۔

Place
Chennai
Delhi
Bangalore
Delhi
Chennai
Chennai
Bangalore
Chennai

SELECT command میں WHERE کی شق مطلوبہ نتیجہ حاصل کرنے کے معیار کو واضح کرتی ہے۔

WHERE شق کے ساتھ SELECT command کی عمومی شکل ہے:

```
SELECT <column-name>[,<column-name>,....] FROM <table-name>WHERE condition>;
```

مثال کے طور پر طلباء کا داخلہ نمبر اور صرف ان طلباء کا نام ظاہر کرنے کے لئے جن کا تعلق Chennai سے ہے، SELECT کمانڈ کو ذیل میں استعمال کیا جاتا ہے:

```
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

تعلقی آپریٹرز جیسے = , < , <= , > , >=. < > دو شقوں کا موازنہ کرنے کے لئے استعمال کیا جاسکتا ہے، WHERE شق کے ساتھ SELECT کمانڈ استعمال کیا جاتا ہے۔ منطقی آپریٹرز OR، AND، NOT کو WHERE شق میں تلاش کے حالات کو مربوط کرنے

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:45 PM

کے لئے استعمال کیا جاسکتا ہے۔مثال کے طور پر:

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 کمانڈ کو ان طریقوں سے بھی استعمال کیا جاتا ہے۔

SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 OR Place ="Delhi");

SELECT Admno, Name, Place FROM Student WHERE (NOT Place ="Delhi");

BETWEEN and NOT BETWEEN Keywords 12.7.5.3

BETWEEN keyword قیمتوں کی حدود کو واضح کرتا ہے جو ریکارڈ میں شرط کو True بتاتے ہیں۔اس کی حد میں ایک اعلیٰ قیمت اور ایک ادنیٰ قیمت ہوتی ہے جس کے درمیان معیار کو ہونا چاہئے۔

Admno	Name	Gender	Age	Place
101	Adarsh	M	18	Delhi
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore

BETWEEN آپریٹر کا اُلٹا NOT BETWEEN ہے، جس میں شرط کو پوری نہ کرنے والے رکارڈس ظاہر کرتے ہیں۔

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

IN Keyword 12.7.5.4

IN keyword مطلوبہ لفظ اقدار کی ایک فہرست کی وضاحت کرنے کے لئے استعمال ہوتا ہے جو ریکارڈ اقدار کے ساتھ مماثل ہونا چاہئے۔ دوسرے الفاظ میں یہ ایک سے زیادہ قیمت کے ساتھ کالم کا موازنہ کرنے کے لئے استعمال ہوتا ہے۔یہ ایک OR حالت کی طرح ہے۔

مثال کے طور پر :

SELECT Admno, Name, Place FROM Student WHERE Place IN (“Chennai”, “Delhi”);

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:45 PM

Admno	Name	Place
100	Ashish	Chennai
101	Adarsh	Delhi
103	Ayush	Delhi
104	Abinandh	Chennai
105	Revathi	Chennai
107	Hema	Chennai

NOT IN keyword صرف ان رکارڈوں کو ظاہر کرتا ہے، جو list سے میچ نہیں ہوتا۔

مثال کے طور پر :

SELECT Admno, Name, Place FROM Student WHERE Place NOT IN ("Chennai", "Delhi");

will display students only from places other than "Chennai" and "Delhi".

Admno	Name	Age
102	Akshith	Bangalore
106	Devika	Bangalore

NULL قیمت

جدول کی ایک فیلڈ میں NULL قیمت کو تلاش کرنے کے لئے IS WHERE شق میں IS NULL کو استعمال کیا جاتا ہے۔

مثال کے طور پر list میں تمام طلباء جن کی عمروں میں قیمتیں نہیں ہیں، اس کمانڈ کو استعمال کیا جاتا ہے۔

SELECT * FROM Student WHERE Age IS NULL;

نوٹ:

کسی ٹیبل میں غیر NULL قیمتوں کو NOT NULL کا استعمال کرتے ہوئے درج کیا جاسکتا ہے۔

ORDER BY Clause 12.7.5.5

SQL کو ORDER BY شق کو ایک یا دو کالموں میں ڈیٹا کو بڑھتی ہوئی (عروجی ترتیب) یا نزولی ترتیب میں چھانٹنے کے لئے استعمال کیا جاتا ہے۔

1۔ ORDER BY ڈیٹا کو بڑھتی ہوئی ترتیب میں Default کے طور پر ترتیب دیتا ہے۔

2۔ ہم ڈیٹا کو نزولی ترتیب میں ترتیب دینے کے لئے DESC keyword اور بڑھتی ہوئی ترتیب میں ترتیب دینے کے لئے ASC keyword استعمال کر سکتے ہیں۔

ORDER BY شق اس طرح استعمال ہوتی ہے۔

SELECT <column-name>[,<column-name>,....] FROM <table-name>ORDER BY <column1>,<column2>,...ASC| DESC ;

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:46 PM

مثال کے طور پر:

طلباء کو ان کے ناموں کے حروف تہجی انداز میں ظاہر کرنے کے لئے، یہ کمانڈ استعمال ہوتا ہے

SELECT * FROM Student ORDER BY Name;

مندرجہ بالا طلباء کی جدول اس ترتیب پاتی ہے۔

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

نوٹ:

ORDER BY شق اصلی ٹیبل کو متاثر نہیں کرتی ہے۔

12.7.5.6 WHERE شق (WHERE Clause)

ریکارڈس کو فلٹر کرنے کے لئے WHERE شق کا استعمال کیا جاتا ہے۔ یہ صرف ان ریکارڈوں کو نکالنے میں مدد کرتا ہے جو ایک دی گئی شرط کو پورا کرتے ہیں۔ مثال کے طور پر طلباء کی جدول میں، 18 سال اور اس سے اوپر کے طلباء کی فہرست کو حرف تہجی ترتیب میں ظاہر کرنے کے لئے، کمانڈ ذیل میں دی گئی ہے :

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

طلباء کی فہرست کو نزولی ترتیب میں 18 سال اور اس سے اوپر کے طالب علموں کے ناموں کو ظاہر کرنے کے لئے یہ کمانڈ دیا گیا ہے:

SELECT * FROM Student WHERE Age>=18 ORDER BY Name DESC;

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:46 PM

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

نوٹ:

ایک سے زیادہ فیلڈوں میں چھانٹنے (sorting) کا عمل کیا جاسکتا ہے۔

12.7.5.7 GROUP BY شق (GROUP BY Clause)

GROUP BY شق کے ساتھ SELECT کے بیان کو استعمال کر کے ساتھ طلباء کو صفوں یا کالموں پر یکساں اقدار کے ساتھ گروپ بنانے یا جدول کو گروپوں میں تقسیم کرنے کے لئے کیا جاتا ہے۔ مثال کے طور پر کلاس کے مرد طلباء یا طالب علموں کی تعداد جاننے کے لئے، گروپ کے ذریعہ GROUP استعمال کیا جاسکتا ہے۔ یہ زیادہ تر ڈیٹا بیس سے سمری رپورٹس پیش کرنے کے لئے مجموعی افعال کے ساتھ مل کر استعمال ہوتا ہے۔

GROUP BY شق کے لئے syntax یہ ہے۔

SELECT <column-names> FROM <table-name> GROUP BY <column-name>HAVING condition];

student جدول میں اوپر کی کمانڈ کو apply کرنے کے لئے

SELECT Gender FROM Student GROUP BY Gender;

درج ذیل کمانڈ، نیچے دیا ہوا نتیجہ دے گا۔

Gender
M
F

غور کرنے کی بات یہ ہے کہ صرف دو نتائج واپس آئے ہیں۔ اس کی وجہ یہ ہے کہ ہمارے پاس صرف جنس کی دو قسمیں ہیں 'Male' اور 'Female'۔ GROUP BY شق M کے تمام طلباء کو return کرتا ہے اور ان کو ایک صف میں لاتا ہے۔ یہ F طالبات کے ساتھ بھی ایسا ہی کرتا ہے۔

مثال کے طور پر طلباء کی میز میں Male اور Female طلبہ کی تعداد شمار کرنے کے لئے، درج ذیل کمانڈ دیا گیا ہے:

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:46 PM

SELECT Gender, count(*) FROM Student GROUP BY Gender;

Gender	count(*)
M	5
F	3

نوٹ:

COUNT کے ساتھ NULL کی قیمتوں کو شامل کرنے کے لئے * استعمال کیا جاتا ہے۔

GROUP BY مجموعی افعال کو آزادانہ طور پر ان گروہوں کی ایک سیریز پر لاگو کرتا ہے جن کی تعریف مشترکہ طور پر ایک فیلڈ کی قیمت ہوتی ہے۔ مذکورہ بالا SELECT Statement سے مرد اور خواتین طالب علموں کی تعداد شمار کی جاتی ہے۔

HAVING Clause 12.7.5.8

SELECT statement میں GROUP BY شق کے ساتھ HAVING شق گروپ کو شرط لگایا جاتا ہے، اور ان پر aggregate function بھی شامل ہیں۔ مثال کے طور پر Chennai سے تعلق رکھنے والے Male اور Female طالب علموں کی تعداد شمار کرنا۔

SELECT Gender , count(*) FROM Student GROUP BY Gender HAVING Place = 'Chennai';

Gender	count(*)
M	2
F	2

مندرجہ بالا آؤٹ پٹ جدول طلباء سے چینئی میں مرد اور خواتین طلباء کی تعداد ظاہر کرتی ہے۔

TCL Commands 12.7.6

COMMIT Command 12.7.6.1

COMMIT کمانڈ کا استعمال ڈیٹا بیس میں کسی بھی لین دین کو مستقل طور پر محفوظ کرنے کے لئے کیا جاتا ہے۔ جب کسی بھی DML کمانڈ جیسے INSERT، UPDATE، DELETE کمانڈز استعمال کیے جاتے ہیں تو، ان کمانڈوں کے ذریعہ کی گئی تبدیلیاں مستقل نہیں ہوتی ہیں۔ اس کو مستقل طور پر نشان زد کیا گیا ہے۔ جب ہی SQL پرامپٹ سے کمٹ کمانڈ دیا جائے۔ ایک بار جب COMMIT کمانڈ دیا جاتا ہے تو، کی گئی تبدیلیاں واپس نہیں لاسکتی ہیں۔ COMMIT کمانڈ بطور استعمال ہوتا ہے۔

```
COMMIT;
```

ROLLBACK Command 12.7.6.2

ROLLBACK کمانڈ آخری ڈیٹا بیس، کو ڈیٹا بیس کو بحال کرتی ہے۔ یہ SAVEPOINT کمانڈ کے ساتھ کسی خاص savepoint مقام پر جانے کے لئے استعمال ہوتا ہے۔ ROLLBACK کمانڈ کا syntax ہے۔

```
ROLL BACK TO save point name;
```


12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:46 PM

SAVEPOINT command 12.7.6.3

SAVEPOINT کمانڈ عارضی طور پر ٹرانزیکشن کو بچانے کے لئے استعمال کیا جاتا ہے تا کہ جب بھی ضرورت ہو اس مقام پر آپ رول بیک کر سکیں۔ ہماری جدول کی مختلف ریاستوں کو مختلف ناموں کا استعمال کرتے ہوئے کسی بھی وقت محفوظ کیا جا سکتا ہے اور اس حالت میں رول بیک بیک کمانڈ کا استعمال کر کے کیا جا سکتا ہے۔

```
SAVEPOINT savepoint_name;
```

درج ذیل ڈیٹا موجود Student کی جدول میں COMMIT، SAVEPOINT اور ROLLBACK وغیرہ دکھائی گئی مثال۔

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;
```

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

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:47 PM

```
INSERT INTO Student VALUES(106, '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

Points to remember: یادرکھنے کے نکات:

- SQL ایک ایسی زبان ہے جو تعلقی ڈیٹا بیس بنانے اور چلانے میں مدد دیتی ہے۔
- MySQL ایک ڈیٹا بیس مینجمنٹ سسٹم ہے۔
- SQL کے مختلف اجزاء Data Definition Language (DDL)، Data Manipulation Language (DML)، Data Query Language (DQL)، Transactional Control Language (TCL) اور Data Control Language (DCL) ہیں۔
- DDL جدول بنانے اور حذف کرنے کے لیے بیانات فراہم کرتا ہے۔
- DML جدول میں ڈیٹا داخل کرنے، اپ ڈیٹ کرنے اور حذف کرنے کے لیے بیانات مہیا کرتا ہے۔
- CDL ڈیٹا تک رسائی کی اجازت کے احکامات فراہم کرتا ہے۔
- TCL کمانڈ ڈیٹا بیس میں لین دین کو منظم کرنے کے لیے استعمال ہوتا ہے۔
- DQL کمانڈ ڈیٹا بیس میں سوالات (queries) پیدا کرنے میں مدد کرتا ہے۔
- CREATE TABLE کمانڈ ایک نئی جدول بناتی ہے۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:47 PM

Hands on Experience عملی مشق

1۔ طلباء کے جدول میں اس ترتیب میں استفسار query تیار کریں، جس میں فیلڈ کے نام، عمر، جگہ اور admno کے مطابق بنائیں۔

2۔ 18 سال سے زیادہ عمر کے طلباء کے ساتھ منفرد شہر کے ساتھ طلباء کی جدول دکھانے کے لیے ایک استفسار query بنائیں۔

3۔ درج ذیل فیلڈوں میں ملازم نمبر، ملازم کا نام، عہدہ، شمولیت کی تاریخ اور بنیادی تنخواہ کے ساتھ ملازموں کی ایک جدول بنائیں۔

4۔ مذکورہ جدول میں ملازم نمبر کو primary key کے طور پر مقرر کریں اور کسی بھی فیلڈ میں NULL قیمتوں کو چیک کریں۔

5۔ ان تمام ملازمین کی فہرست تیار کریں جو مینیجر ہیں۔

Evaluation محاسبہ

حصہ - I

صحیح جواب کا انتخاب کریں۔ (1 مارکس)

1۔ کون سا کمانڈ ٹیبل کی ساخت بنانے، تعلقات کو حذف کرنے، اور تعلقاتی schema میں ترمیم کرنے کی وضاحت کرتی ہیں۔

a. DDL b. DML

c. DCL d. DQL

2۔ کون سا کمانڈ جدول کی ساخت کو تبدیل کرنے دیتا ہے ؟

a. SELECT b. ORDER BY

c. MODIFY d. ALTER

3۔ ایک جدول کو حذف کرنے کے لئے command یہ ہے۔

A) DROP B) DELETE

C) DELETE ALL D) ALTER TABLE

4۔ اس کے استعمال سے سوالات (استفسار) (Queries) تیار کئے جاسکتے ہیں۔

a. SELECT b. ORDER BY

c. MODIFY d. ALTER

5۔ ڈیٹا بیس میں ڈیٹا کو چھانٹنے (sort) دینے کے لیے استعمال ہونے والی شق۔

a. SORT BY b. ORDER BY

c. GROUP BY d. SELECT

حصہ - II

II۔ مختصر جوابات لکھیں۔ (2 مارکس)

1۔ ایک سوال (query) لکھیں جو ان تمام طلباء کو منتخب کرے جن کی عمر ترتیب وار 18 سے کم ہو۔

2۔ Unique constraint اور primary key constraint میں فرق کریں۔

3۔ Table constraint اور column constraint کے درمیان فرق لکھیں۔

4۔ SQL کا کون سا جز وجدولوں میں قیمتوں کو داخل کرنے دیتا ہے اور کون سا جدول بنانے کی اجازت دیتا ہے؟

5۔ SQL اور MySQL میں کیا فرق ہے؟

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:47 PM

حصہ - III

III۔ وضاحت کریں۔ (3 مارکس)

1۔ ایک رکاوٹ (constraint) کیا ہے؟ بنیادی کلیدی رکاوٹ (primary key constraint) پر مختصر نوٹ لکھیں۔

2۔ ایک نیا فیلڈ شامل کر کے طالب علم کی جدول کی ساخت میں ترمیم کرنے کے لیے ایک SQL بیان لکھیں۔

3۔ کوئی تین DDL کمانڈ لکھیں۔

4۔ ایک مثال کے ساتھ Savepoint کمانڈ کا استعمال لکھیں۔

5۔ DISTINCT keyword استعمال کرتے ہوئے SQL بیان لکھیں۔

حصہ - IV

IV۔ تفصیل سے وضاحت کریں۔ (5 مارکس)

1۔ مختلف اقسام کی رکاوٹیں (Constraints) اور ان کے افعال لکھیں۔

2۔ درج ذیل ملازم کی جدول پر غور کریں۔ qtns.(i) سے (v) تک کے لیے SQL کمانڈ لکھیں۔

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) تمام ملازمین کی تفصیلات تنخواہ کے نزولی ترتیب میں ظاہر کرنا

(ii) ان تمام ملازمین کو ظاہر کرنا جن کا الاؤنس 5000 اور 7000 کے درمیان ہے۔

(iii) ان ملازمین کو خارج کرنا جو میکانک ہیں۔

(iv) ایک نئی صف شامل کرنا۔

(v) ان تمام ملازمین کی تفصیلات ظاہر کرنا جو آپریٹرز ہیں۔

3۔ SQL کے اجزاء کیا ہیں؟ ہر ایک کے کمانڈ لکھیں۔

4۔ طالب علم کی جدول میں درج ذیل SQL بیانات بنائیں۔

(i) GROUP BY شق کا استعمال کرتے ہوئے SELECT بیان۔

(ii) ORDER BY شق کا استعمال کرتے ہوئے SELECT بیان۔

5۔ کسی بھی پانچ فیلڈز والے ملازم کے لیے ایک جدول بنانے کے لیے SQL کا بیان لکھیں اور ملازم جدول کے لیے table constraint بنائیں۔

12 CS UNIT 12 CDR Set
Monday, April 17, 2023 7:57:48 PM

اکائی IV

باب 13

پائیتھن اور CSV فائلیں PYTHON AND CSV FILES

سیکھنے کے مقاصد

اس باب کی تکمیل کے بعد، طالب علم قابل ہو جائے گا۔

- سمجھنا کہ CSV کیا ہے ؟
- پائیتھن کے پروگراموں میں CSV فائلیں درآمد کرنے کے قابل بنانا۔
- پائیتھن کے پروگراموں کو چلانا اور ڈیبگ کرنا۔

13.1 تعارف Introduction

پائیتھن کے پاس ماڈیولز کی ایک وسیع لائبریری ہے، جو اس کی تقسیم کے ساتھ شامل ہے۔ ماڈیول میں سے ایک CSV ماڈیول ہے جو کہ پائیتھن پروگرامر کو CSV (کوما سے الگ کردہ قیمتیں) فائلوں کو پارس (parse) کرنے کی صلاحیت فراہم کرتا ہے۔ CSV فائل ایک انسانی پڑھنے کے قابل ٹیکسٹ فائل ہے، جہاں ہر لائن کے کئی فیلڈز ہوتے ہیں، کوما یا کسی دوسرے delimiter سے الگ ہوتے ہیں۔ آپ ہر لائن کو صف اور ہر فیلڈ کو کالم سمجھ سکتے ہیں۔ CSV ماڈیول، CSV فائلوں کی اکثریت کو پڑھنے اور لکھنے کے قابل ہو جائے گا۔

13.2 CSV اور XLS فائل فارمیٹس کے درمیان فرق

کوما سے الگ شدہ اقدار (CSV) (Comma separated values) اور eXceL Sheets (XLS) فائل فارمیٹس کے درمیان فرق ہے۔

Excel	CSV
ایکسل ایک بائنری فائل ہے جو فائل میں تمام ورک شیٹس کے بارے میں معلومات رکھتی ہے، بشمول مواد اور فارمیٹنگ دونوں۔	CSV فارمیٹ ایک سادہ ٹیکسٹ فارمیٹ ہے جس میں قیمتوں کی ایک سیریز کو ما سے الگ کیا گیا ہے۔
XLS فائلوں کو صرف ان ایپلیکیشنز کے ذریعے پڑھا جا سکتا ہے، جو خاص طور پر ان کے فارمیٹ کو پڑھنے کے لیے لکھی گئی ہیں، اور صرف اسی طرح لکھی جا سکتی ہیں۔	CSV ونڈوز میں کسی بھی ٹیکسٹ ایڈیٹر جیسے نوٹ پیڈ، ایم ایس ایکسل، اوپن آفس، وغیرہ کے ساتھ کھولا جا سکتا ہے۔
ایکسل ایک اسپریڈ شیٹ ہے جو فائلوں کو اپنے ملکیتی فارمیٹ میں محفوظ کرتی ہے۔ xls یا xlsx ۔	CSV ٹیبلولر معلومات کو ایک محدود شکل والی ٹیکسٹ فائل میں .csv توسیع کے ساتھ محفوظ کرنے کا ایک فارمیٹ ہے۔
ایکسل درآمد شدہ ڈیٹا کو زیادہ میموری استعمال کرتا ہے۔	CSV فائلوں کو درآمد کرنا بہت تیز ہو سکتا ہے، اور یہ کم میموری بھی استعمال کرتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:43 PM

کیا تم جانتے ہو

excel میں محفوظ فائلیں ٹیکسٹ ایڈیٹرز کے ذریعہ نہیں کھولی جاسکتی اور نہ ہی ان میں ترمیم کی جاسکتی ہیں۔

13.3 CSV فائل کا مقصد (Purpose of CSV file)

CSV ایک سادہ فائل فارمیٹ ہے، جو ٹیبلر ڈیٹا (جدول شدہ ڈیٹا) کو ذخیرہ کرنے کے لیے استعمال کیا جاتا ہے، جیسے اسپریڈشیٹ یا ڈیٹا بیس۔ چونکہ وہ سادہ متن ہیں، اس لیے سپریڈشیٹ یا کسی اور اسٹوریج ڈیٹا بیس میں درآمد کرنا آسان ہے۔ قطع نظر اس کے کہ آپ جو مخصوص سافٹ ویئر استعمال کر رہے ہیں۔

آپ CSV فائلوں کو اسپریڈشیٹ پروگرام، جیسے مائیکروسافٹ ایکسل یا ٹیکسٹ ایڈیٹر میں یا ڈیٹا بیس کے ذریعے کھول سکتے ہیں، جس سے انہیں پڑھنے میں آسانی ہوتی ہے۔

نوٹ

CSV فائل چارٹ یا گراف محفوظ نہیں کرسکتی۔ یہ ڈیٹا کو اسٹور کرتا ہے، لیکن اس میں فارمیٹنگ، فارمولے میکر، وغیرہ شامل نہیں ہیں۔

کیا تم جانتے ہو

CSV فائل کو فلیٹ فائل بھی کہا جاتا ہے۔ CSV فارمیٹ میں فائلیں ان پروگراموں سے درآمد اور برآمد کی جاسکتی ہیں، جو ڈیٹا کو میزوں میں محفوظ کرتے ہیں، جیسے *Microsoft Excel* یا *OpenOfficeCalc*

13.4 نوٹ پیڈ (یا کوئی ٹیکسٹ ایڈیٹر) کا استعمال کرتے ہوئے CSV فائل بنانا

CSV فائل ایک ٹیکسٹ فائل ہے، لہذا اسے کسی بھی ٹیکسٹ ایڈیٹر کا استعمال کرتے ہوئے بنایا اور ترمیم کیا جا سکتا ہے، لیکن زیادہ تر CSV فائل اس پروگرام میں اسپریڈشیٹ یا ڈیٹا بیس کو ایکسپورٹ کر کے بنائی جاتی ہے جس نے اسے بنایا ہے۔

13.4.1 CSV نارمل فائل بنانا (Creating CSV normal file)

نوٹ پیڈ میں CSV فائل بنانے کے لیے، پہلے ایک نئی فائل کا استعمال کرتے ہوئے کھولیں۔

File →New or ctrl +N.

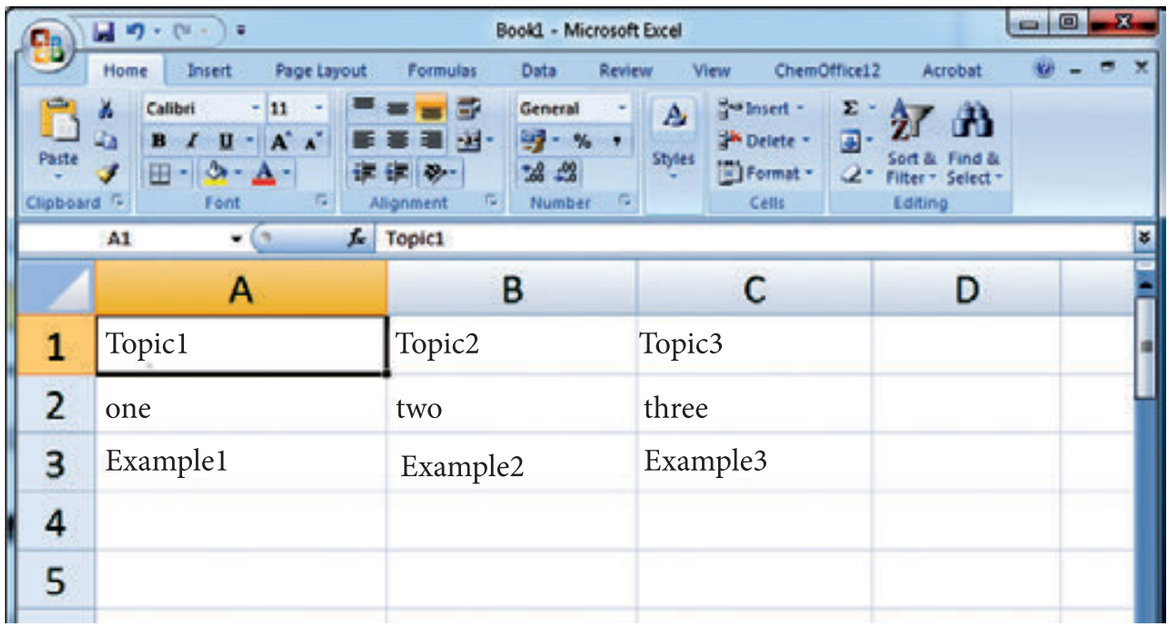
پھر وہ ڈیٹا درج کریں جس میں آپ چاہتے ہیں کہ فائل پر مشتمل ہو، ہر قیمت کو کوما اور ہر صف کو نئی لائن کے ساتھ الگ کریں۔

مثال کے طور پر درج ذیل تفصیلات پر غور کریں۔

```
Topic1,Topic2,Topic3
one,two,three
Example1,Example2,Example3
```

اس مواد کو csv. توسیع کے ساتھ ایک فائل میں محفوظ کریں۔ اس کے بعد آپ اسے مائیکروسافٹ ایکسل یا کسی دوسرے اسپریڈشیٹ پروگرام کے ذریعے کھول سکتے ہیں۔ یہاں ہم نے مائیکروسافٹ ایکسل کا استعمال کرتے ہوئے کھول دیا ہے۔ یہ درج ذیل کی طرح ڈیٹا کا جدول بنائے گا۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:44 PM



خاکہ 13.4.1 CSV File کو جب MS-Excel میں کھولا گیا ہو

مندرجہ بالا CSV فائل میں، آپ دیکھ سکتے ہیں کہ ڈیٹا کے فیلڈوں کو کوما سے الگ کیا گیا ہے۔ اگر ڈیٹا میں خود کوما موجود ہوتا ہو تو کیا ہوتا ہے ؟ اگر آپ کی CSV فائل میں ڈیٹا کے فیلڈز میں کوما (comma) ہوں، تو آپ ان ڈیٹا فیلڈز کو ڈبل کوٹس (واوین) (") میں بند کر کے ان کی حفاظت کر سکتے ہیں ۔ وہ کوما جو آپ کے ڈیٹا کا حصہ ہیں پھر ان کو ماسے الگ رکھے جائیں گے، جو خود فیلڈز کی حد بندی کرتے ہیں۔

13.4.2 CSV فائل بنانا جس میں ڈیٹا کے ساتھ کوما ہو (Creating CSV File that contains comma with data)

مثال کے طور پر، یہ کہتے ہیں کہ ہمارے ایک فیلڈ میں تفصیل میں کوما ہیں۔ اگر ہمارا ڈیٹا نیچے دی گئی مثال کی طرح لگتا ہے:

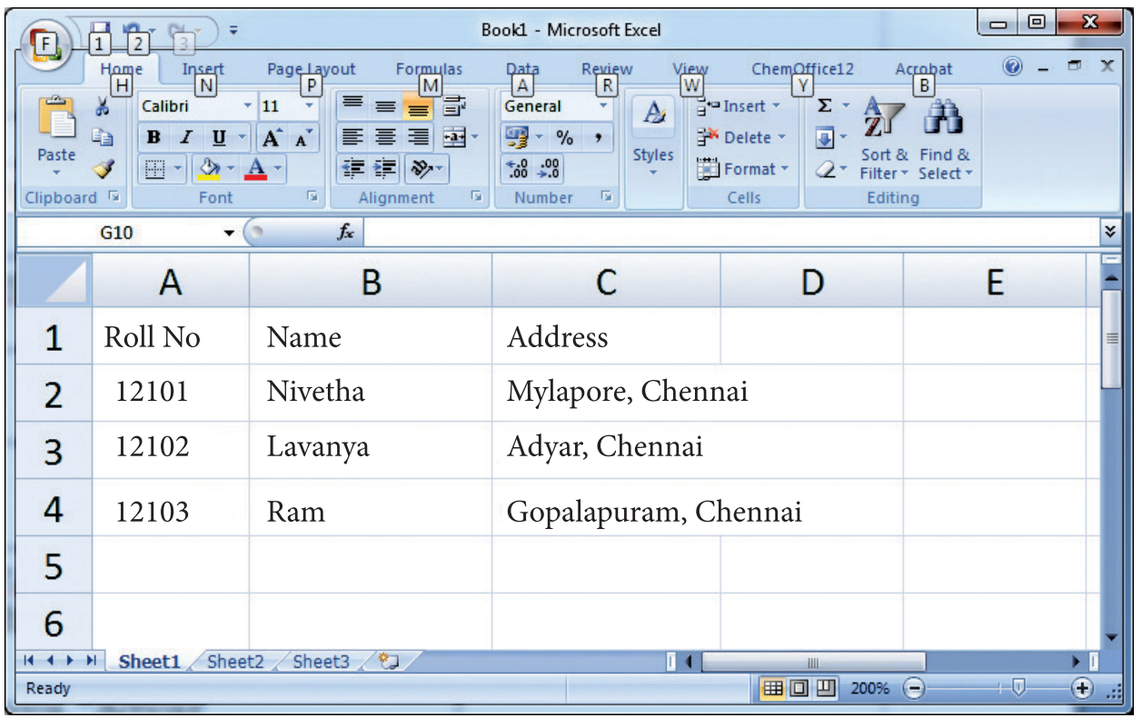
RollNo	Name	Address
12101	Nivetha	Mylapore, Chennai
12102	Lavanya	Adyar, Chennai
12103	Ram	Gopalapuram, Chennai

"Address" کالم میں comma برقرار رکھنے کے لیے، آپ کوٹیشن مارکس (واوین) میں فیلڈز کو بند کر سکتے ہیں۔ مثال کے طور پر:

```
RollNo, Name, Address
12101, Nivetha, “Mylapore, Chennai”
12102, Lavanya, “Adyar, Chennai”
12103, Ram, “Gopalapuram, Chennai”
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:44 PM

جیسا کہ آپ دیکھ سکتے ہیں، صرف وہ فیلڈز جن میں کوما ہوتے ہیں کوٹس میں بند ہوتے ہیں۔ اگر آپ اسے MS Excel میں کھولتے ہیں تو، یہ مندرجہ ذیل کی طرح لگتا ہے۔



خاکہ 13.4.2(a) ایکسل میں کوما کے ساتھ CSV فیلڈ ڈیٹا

نئی لائنوں کے لیے بھی یہی ہے (ایک لائن کی مثال جیسے ایڈرس کا کالم) جس میں آپ کی فیلڈ کے ڈیٹا کا کوئی حصہ ہوتا ہے۔ کوئی بھی فیلڈ جس میں نئی لائن اس کے ڈیٹا کا حصہ ہو اسے ڈبل کوٹس میں بند کرنے کی ضرورت ہے۔

مثال کے طور پر

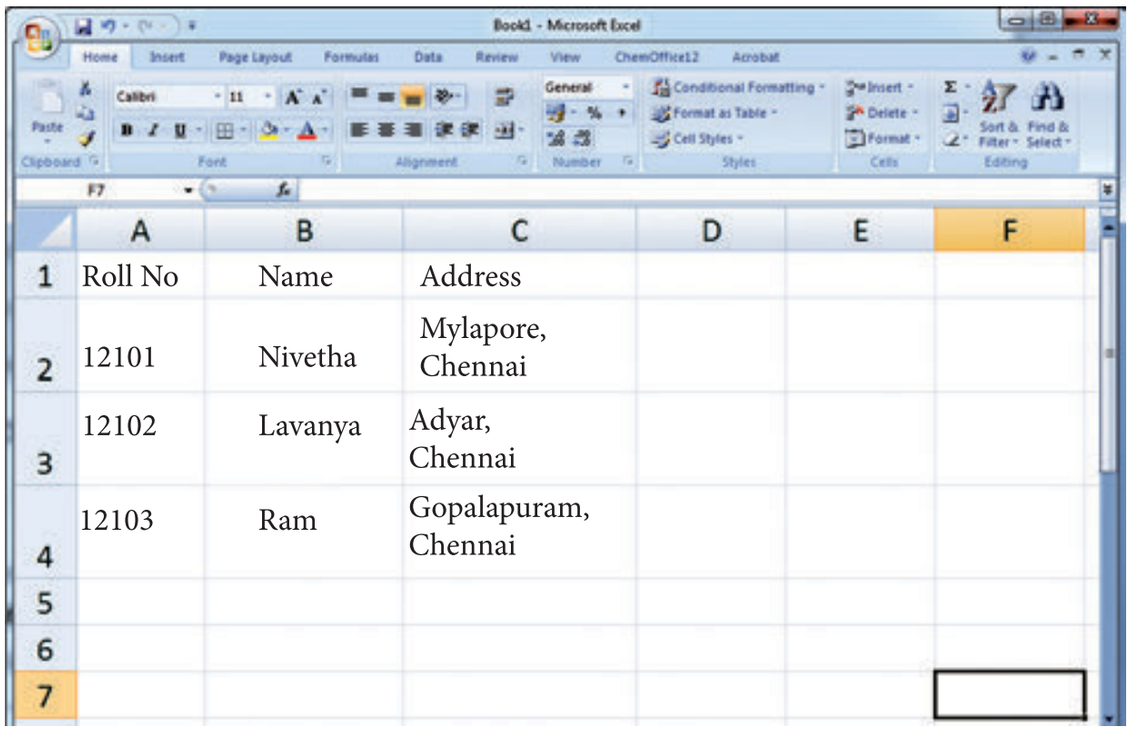
RollNo	Name	Address
12101	Nivetha	Mylapore, Chennai
12102	Lavanya	Adyar, Chennai
12103	Ram	Gopalapuram, Chennai

اسے CSV فائل میں اس طرح لکھا جانا چاہیے۔

```
RollNo, Name, Address
12101, Nivetha, “Mylapore
Chennai”
12102, Lavanya, “Adyar
Chennai”
12103, Ram, “Gopalapuram
Chennai”
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:44 PM

نتیجہ اس طرح نظر آئے گا۔



تصویر 13.4.2 (b) ایکسل میں نئی لائن کے ساتھ CSV فیلڈ ڈیٹا

13.4.3 CSV فائل بنانا، جس میں ڈیٹا کے ساتھ ڈبل کوٹس (واوین) (") ہوں

اگر آپ کے فیلڈوں میں ان کے ڈیٹا کے حصے کے طور پر ڈبل کوٹس (") ہیں، تو اندرونی کوٹیشن مارکس کو دوگنا کرنے کی ضرورت ہے، تاکہ ان کی صحیح تشریح کی جاسکے۔ مثال کے طور پر، درج ذیل ڈیٹا دیا گیا ہے :

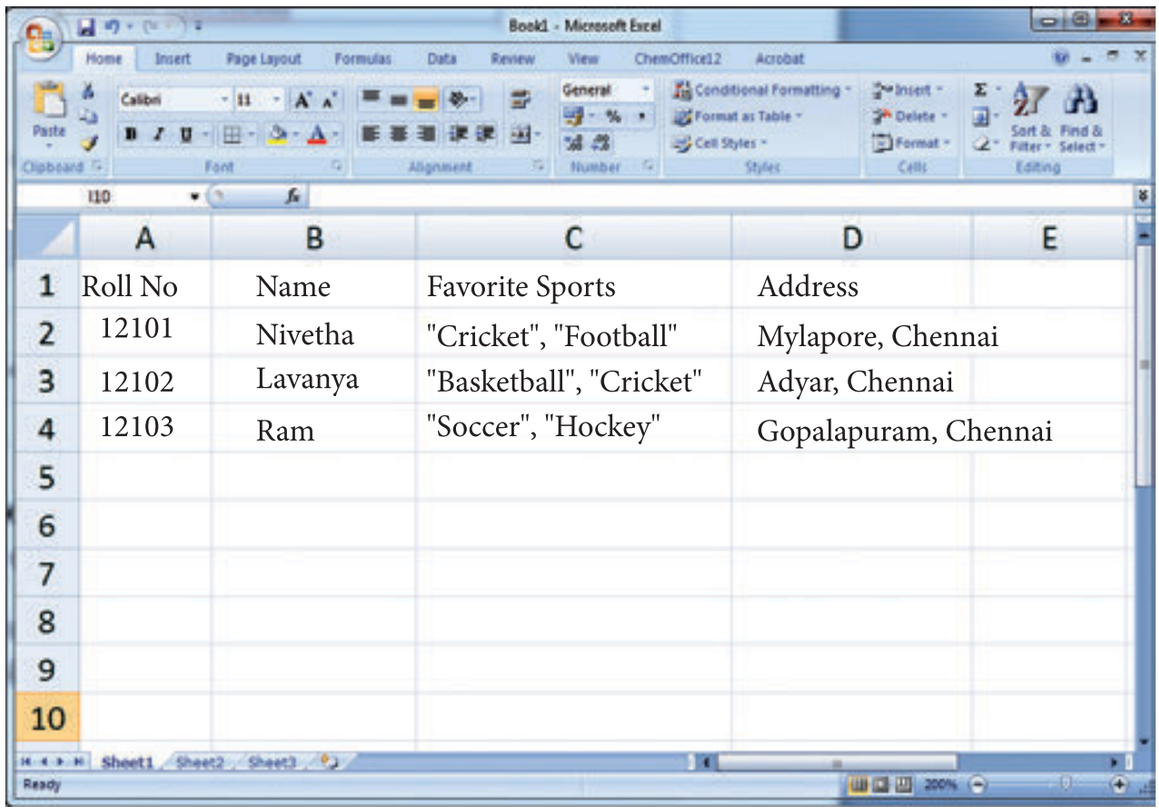
Roll No	Name	Favorite Sports	Address
12101	Nivetha	“Cricket”, “Football”	Mylapore Chennai
12102	Lavanya	“Basketball”, “Cricket”	Adyar Chennai
12103	Ram	“Soccer”, “Hockey”	Gopala puram Chennai

اسے csv فائل میں اس طرح لکھا جانا چاہیے۔

```
RollNo, Name, FavoriteSports, Address
12101, Nivetha,”””” Cricket ””,”” Football ””””, Mylapore chennai
12102, Lavanya,”””” Basketball ””,”” Cricket ””””, Adyar chennai
12103, Ram,”””” Soccer””,”” Hockey””””, Gopalapuram chennai
```

تو آوٹ پٹ اس طرح ہوگا۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:44 PM



	A	B	C	D	E
1	Roll No	Name	Favorite Sports	Address	
2	12101	Nivetha	"Cricket", "Football"	Mylapore, Chennai	
3	12102	Lavanya	"Basketball", "Cricket"	Adyar, Chennai	
4	12103	Ram	"Soccer", "Hockey"	Gopalapuram, Chennai	
5					
6					
7					
8					
9					
10					

خاکہ 13.4.3 ایکسل میں ڈبل کوٹس کے ساتھ CSV فیلڈ ڈیٹا

13.4.4 CSV فائل میں ڈیٹا کو فارمیٹ کرنے کے لیے مندرجہ ذیل قواعد پر عمل کریں۔

1۔ ہر ریکارڈ (ڈیٹا کی صف) کو ایک علیحدہ لائن پر واقع ہونا ہے، جس میں Enter بٹن دبا کر لائن بریک سے حد بندی (delimit) کی جاتی ہے۔
مثال کے طور پر : ↵

xxx,yyy↵

↵ denotes enter Key to be pressed

2۔ فائل میں آخری ریکارڈ ختم ہونے والی لائن ٹوٹ سکتی ہے یا نہیں۔ مثال کے طور پر:

ppp, qqq ↵
yyy, xxx

3۔ ایک اختیاری ہیڈر لائن ہو سکتی ہے جو فائل کی پہلی لائن کے طور پر ظاہر ہو رہی ہے جس میں عام ریکارڈ لائنوں کی شکل ہے۔ ہیڈر میں فائل میں موجود فیلڈز سے متعلقہ نام ہوں گے اور اس میں فیلڈز کی اتنی ہی تعداد ہونی چاہیے جتنی کہ باقی فائل میں ریکارڈ۔ مثال کے طور پر:

field_name1.field_name2,field_name3↵

aaa,bbb,ccc ↵
zzz,yyy,xxx CRLF(**Carriage Return and Line Feed**)

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:45 PM

4۔ ہیڈر اور ہر ریکارڈ کے اندر، ایک یا ایک سے زیادہ فیلڈز ہو سکتے ہیں، کوما سے الگ کر کے۔ خالی جگہوں (spaces) کو ایک فیلڈ کا حصہ سمجھا جاتا ہے اور اسے نظر انداز نہیں کیا جانا چاہیے۔ ریکارڈ میں آخری فیلڈ کے بعد کوما نہیں ہونا چاہیے۔ مثال کے طور پر: **Red , Blue**

5۔ ہر فیلڈ ڈبل کوٹس میں بند ہو سکتا ہے یا نہیں۔ اگر فیلڈز ڈبل کوٹس کے ساتھ بند نہیں ہیں، تو ڈبل کوٹس فیلڈز کے اندر ظاہر نہیں ہو سکتے ہیں۔ مثال کے طور پر:

```
"Red","Blue","Green"↵        # ڈبل کوٹس کے ساتھ فیلڈ ڈیٹا
Black,White,Yellow           # ڈبل کوٹس کے بغیر فیلڈ ڈیٹا
```

6۔ لائن بریکس (CRLF)، ڈبل کوٹس اور کوما پر مشتمل فیلڈز کو ڈبل کوٹس میں بند کیا جانا چاہیے۔ مثال کے طور پر:

```
Red, “,”, Blue CRLF      # کوما خود ایک فیلڈ کی قیمت رکھتی ہے، چنانچہ اسے ڈبل کوٹس سے بند کیا جانا چاہئے۔
Red, Blue , Green
```

7۔ اگر ڈبل کوٹس فیلڈز کو بند کرنے کے لیے استعمال کیے جاتے ہیں، تو فیلڈ کے اندر ظاہر ہونے والے ڈبل کوٹ کو دوسرے ڈبل کوٹ سے پہلے ہونا چاہیے۔ مثال کے طور پر:

```
“Red, ” “Blue”, “Green”,     # ڈبل کوٹس خود ایک فیلڈ کی قیمت رکھتی ہے، چنانچہ اسے ڈبل کوٹس سے بند کیا جانا چاہئے۔
, , White
```

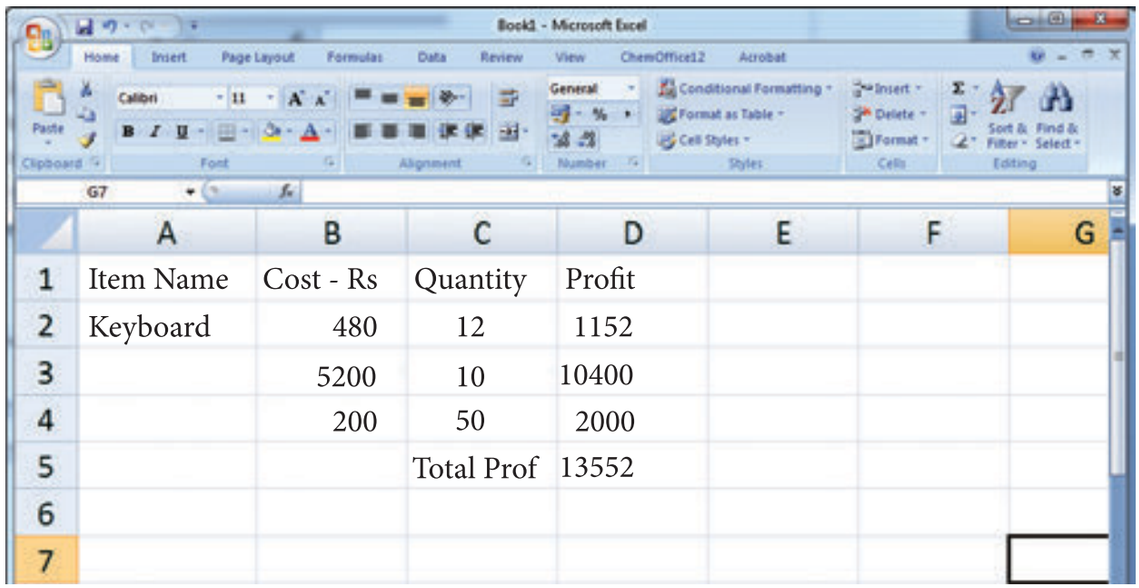
نوٹ:

مذکورہ بالا مثال (White , ,) میں آخری صف دو کوما سے شروع ہوتی ہے، کیونکہ اس صف کے پہلے دو فیلڈز ہماری اسپریڈ شیٹ میں خالی تھے۔ انہیں حذف نہ کریں- دو کوما درکار ہیں تا کہ فیلڈ صف سے صف کے مطابق ہوں۔ ان کو خارج نہیں کیا جا سکتا۔

13.5 مائیکروسافٹ ایکسل کا استعمال کرتے ہوئے ایک CSV فائل بنانا
Create A CSV File Using Microsoft Excel

مائیکروسافٹ ایکسل کا استعمال کرتے ہوئے CSV فائل بنانے کے لیے، ایکسل لانچ کریں اور پھر وہ فائل کھولیں جسے آپ CSV فارمیٹ میں محفوظ کرنا چاہتے ہیں۔ مثال کے طور پر، ذیل میں ہمارے نمونہ ایکسل ورک شیٹ میں موجود ڈیٹا ہے:

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:45 PM

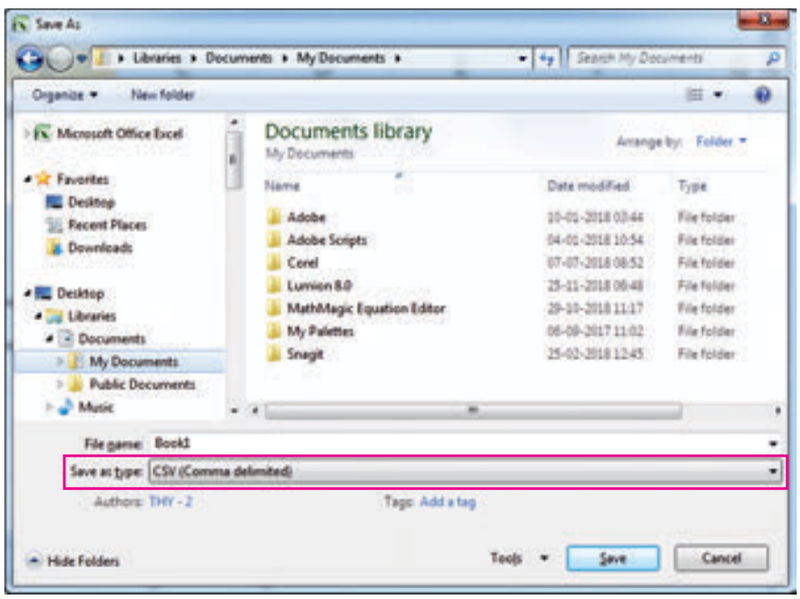


	A	B	C	D	E	F	G
1	Item Name	Cost - Rs	Quantity	Profit			
2	Keyboard	480	12	1152			
3		5200	10	10400			
4		200	50	2000			
5			Total Prof	13552			
6							
7							

خاکہ 13.5 نمونہ ورک شیٹ ڈیٹا (Sample worksheet data)

ورک شیٹ میں ڈیٹا داخل کرنے کے بعد File → Save as آپشن کو منتخب کریں، اور Save as type آپشن کے لیے، CSV (comma delimited) انتخاب کریں یا فائل کا نام توسیع کے ساتھ ٹائپ کریں۔

ایکسل فائل کو بطور CSV محفوظ کرنا۔



خاکہ 13.6 Save as ڈائیلاگ باکس

فائل کو محفوظ کرنے کے بعد، آپ اسے دیکھنے کے لیے یا اسے دستی طور پر ترمیم کرنے کے لیے ٹیکسٹ ایڈیٹر میں کھولنے کے لیے آزاد ہیں۔ اس کے مندرجات درج ذیل سے ملتے جلتے ہوں گے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:45 PM

```
Item Name, Cost-Rs, Quantity, Profit
Keyboard, 480, 12, 1152
Monitor, 5200, 10, 10400
Mouse, 200, 50, 2000
,,Total Profit =,13552
```

13.5.1 مائیکروسافٹ ایکسل کی ایک CSV فائل کھولنے کے لیے

اگر مائیکروسافٹ ایکسل کمپیوٹر پر انسٹال ہو چکا ہے، جب فائل ڈبل کلک کی جائے تو بطور ڈیفالٹ CSV فائلیں ایکسل میں خود بخود کھل جائیں گی۔ اگر آپ CSV فائل کو کھولتے وقت open with پرامپٹ حاصل کر رہے ہیں تو، فائل کھولنے کے لیے دستیاب پروگراموں میں سے مائیکروسافٹ ایکسل کا انتخاب کریں۔

متبادل کے طور پر، آپ مائیکروسافٹ ایکسل کھول سکتے ہیں اور مینو بار میں، File → Open اور CSV فائل منتخب کریں۔
اگر فائل درج نہیں ہے تو، فائل کی قسم کو ٹیکسٹ فائلوں (*.prn ، *.txt ، *.csv) میں تبدیل کرنا یقینی بنائیں۔

> **کیا تم جانتے ہو!**
> اگر کمپیوٹر میں MS Excel اور Open Office Calc دونوں انسٹال ہیں، تو default کے طور پر CSV فائل MS Excel میں کھل جائے گی۔

13.6 پائیتھن کا استعمال کرتے ہوئے ایک CSV فائل پڑھنا اور لکھنا

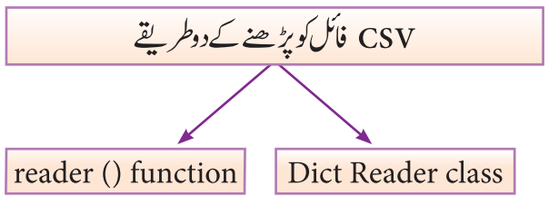
پیتھون CSV نامی ماڈیول فراہم کرتا ہے، اس کا استعمال کرتے ہوئے آپ CSV فائلوں پر کئی آپریشن کر سکتے ہیں۔ CSV لائبریری میں CSV فائلوں سے ڈیٹا کو پڑھنے، لکھنے اور اس پر کارروائی کرنے کے لیے اشیاء اور دیگر کوڈ شامل ہیں۔

> **کیا تم جانتے ہو!**
> CSV فائلوں کو ای کامرس ایپلی کیشنز میں بڑے پیمانے پر استعمال کیا گیا ہے، کیونکہ ان پر عمل کرنا بہت آسان سمجھا جاتا ہے۔

13.6.1 پیتھون کا استعمال کرتے ہوئے ایک CSV فائل پڑھنا

CSV فائل کو پڑھنے کے دو طریقے ہیں۔

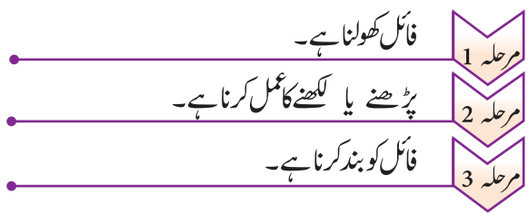
1۔ csv ماڈیول کا ریڈر فنکشن استعمال کریں۔
2۔ DictReader کلاس استعمال کریں۔



خاکہ 13.7 CSV فائل کو پڑھنے کے طریقے

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:46 PM

جب آپ ایک فائل کو پڑھنا یا اس میں لکھنا چاہتے ہیں تو، آپ پہلے اس کو کھولیں۔ جیسے ہی ریڈنگ ختم ہو جاتی ہے، اس کو بند کرنا چاہئے۔ اس لئے کہ ذرائع کو باندھ کر file کو آزاد کرنا ہوتا ہے۔ لہٰذا، پیتھان میں ایک فائل آپریشن درج ذیل ترتیب میں عمل پاتا ہے۔



نوٹ :

فائل کا نام یا مکمل راستہ نام (path name) کی نمائندگی کھلے کمانڈ میں " " یا ' ' کے ساتھ کی جاسکتی ہے۔

پیتھان کے پاس ایک فائل کھولنے کے لیے بلٹ ان فنکشن open() ہے۔ یہ فنکشن ایک فائل آبجیکٹ لوٹاتا ہے، جسے ہینڈل (handle) بھی کہا جاتا ہے، کیونکہ یہ اس کے مطابق فائل کو پڑھنے یا اس میں ترمیم کرنے کے لیے استعمال ہوتا ہے۔

مثال کے طور پر

```
>>> f = open("sample.txt")  # open file in current directory and f is file object
>>> f = open('c:\\pyprg\\ch13sample5.csv')        # specifying full path
```

آپ فائل کھولتے وقت موڈ کی وضاحت کر سکتے ہیں۔ mode میں، آپ وضاحت کر سکتے ہیں کہ آپ 'r' پڑھنا چاہتے ہیں، 'w' لکھنا چاہتے ہیں یا فائل میں 'a' شامل کرنا چاہتے ہیں۔ آپ "text or binary" کی بھی وضاحت کر سکتے ہیں جس میں فائل کھولنی ہے۔

ڈیفالٹ ٹیکسٹ موڈ میں پڑھ رہا ہے۔ اس موڈ میں، فائل سے پڑھتے وقت ڈیٹا string کی شکل میں ہوگا۔

دوسری طرف، بائنری موڈ بائٹس لوٹاتا ہے اور یہ وہ موڈ ہے جو جب غیر ٹیکسٹ فائلوں جیسے امیج یا exe فائلوں سے نمٹتے ہیں، اس وقت استعمال کیا جاتا ہے۔

پیتھون فائل موڈز Python File Modes

موڈ Mode	تفصیل Description
'r'	پڑھنے کے لیے ایک فائل کھولیں۔ (پہلے سے طے شدہ) (default)
'w'	لکھنے کے لیے ایک فائل کھولیں۔ اگر یہ موجود نہیں ہے تو ایک نئی فائل بناتی ہے یا اگر فائل موجود ہے تو اسے کاٹ دیتی ہے۔
'x'	خصوصی تخلیق کے لیے ایک فائل کھولیں۔ اگر فائل پہلے سے موجود ہے تو آپریشن ناکام ہو جاتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:46 PM

'a'	فائل کو کاٹے بغیر اس کے اختتام پر کھولنے کے لیے کھولیں۔ اگر یہ موجود نہیں ہے تو ایک نئی فائل بناتا ہے
't'	ٹیکسٹ موڈ میں کھولیں۔ (پہلے سے طے شدہ) (default)
'b'	بائنری موڈ میں کھولیں۔
'+'	اپ ڈیٹ کرنے کے لیے ایک فائل کھولیں (پڑھنے اور لکھنے)

```
f=open("sample.txt")
#equivalent to 'r' or 'rt'
f = open("sample.txt",'w')          # write in text mode
f = open("image1.bmp",'r+b')        # read and write in binary mode
```

پیتھون کے پاس ایک ردی garbage جمع کرنے والا ہوتا ہے، جو غیر حوالہ شدہ اشیاء (unreferred objects) کو صاف کرتا ہے۔ لیکن، کسی کو فائل کو بند کرنے کے لیے اس پر انحصار نہیں کرنا چاہیے۔

```
f = open("test.txt")      # چونکہ کوئی موڈ متعین نہیں ہے، پہلے سے طے شدہ موڈ rt استعمال کیا جاتا ہے۔
# perform file operations
f.close()
```

مندرجہ بالا طریقہ مکمل طور پر محفوظ نہیں ہے۔ اگر کوئی رعایت اس وقت ہوتی ہے جب آپ فائل کے ساتھ کچھ آپریشن کر رہے ہوں تو کوڈ فائل کو بند کیے بغیر باہر نکل جاتا ہے۔ ایسا کرنے کا بہترین طریقہ "with" بیان کا استعمال ہے۔ یہ یقینی بناتا ہے کہ فائل بند ہے جب with کا بلاک باہر نکل جائے۔ آپ کو واضح طور پر close() طریقہ پر کال کرنے کی ضرورت نہیں ہے۔ یہ اندرونی طور پر کیا جاتا ہے۔

```
with open("test.txt",'r') as f:
# f is file object to perform file operations
```

فائل کو بند کرنے سے وہ وسائل آزاد ہو جائیں گے جو فائل کے ساتھ بندھے ہوئے تھے اور پائیتھن close() طریقہ استعمال کرتے ہوئے کیے جاتے ہیں۔

```
f = open("sample.txt")
# perform file operations
f.close()
```

13.6.1.1 CSV ماڈیول کا ریڈر فنکشن (CSV Module's Reader function)

آپ csv.reader() طریقہ کی مدد سے CSV فائل کے مندرجات کو پڑھ سکتے ہیں۔ ریڈر فنکشن کو ڈیزائن کیا گیا ہے کہ فائل کی ہر لائن لے اور تمام کالموں کی فہرست بنائے۔ پھر، آپ صرف وہ کالم منتخب کریں جس کے لیے آپ متغیر ڈیٹا چاہتے ہیں۔ اس طریقہ کار کو استعمال کرتے ہوئے کوئی شخص مختلف فارمیٹس کی csv فائلوں سے ڈیٹا پڑھ سکتا ہے، جیسے کہ quotes(" ")، pipe(|) اور کاما (,)۔

csv.reader() کے لئے syntax یہ ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:46 PM

```
csv.reader(fileobject,delimiter,fmtparams)
```

جس میں

فائل آبجیکٹ :- فائل کا راستہ (path) اور موڈ گزرتا ہے۔

ڈیلیمیٹر :- ایک اختیاری پیرامیٹر جس میں معیاری بولیاں dilects جیسے | شامل ہیں، ان کو خارج کیا جاسکتا ہے۔

fmtparams :- اختیاری پیرامیٹر جو بولیوں کی ڈیفالٹ قیمتوں کو ختم کرنے میں مدد کرتا ہے quoting، skipinitialspace وغیرہ کو چھوڑ دیا جاسکتا ہے۔

1. CSV file - data with default delimiter comma (,)
2. CSV file - data with Space at the beginning
3. CSV file - data with quotes
4. CSV file - data with custom Delimiters

13.6.1.1.1 ڈیفالٹ ڈلیمیٹر کوما(،) کے ساتھ CSV فائل (CSV file with default delimiter comma(,))

مندرجہ ذیل پروگرام، ایک فائل پڑھتا ہے جسے "sample1.csv" کہا جاتا ہے، جس میں ڈیفالٹ ڈیلیمیٹر کوما(،) اور صف بہ صف پرنٹ ہوتا ہے

```
#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
      print(row)
F.close()
```

OUTPUT

```
['SNO', 'NAME', 'CITY']
['12101', 'RAM', 'CHENNAI']
['12102', 'LAVANYA', 'TIRUCHY']
['12103', 'LAKSHMAN', 'MADURAI']
```

13.6.1.1.2 CSV فائلیں-شروع میں خالی جگہوں کے ساتھ ڈیٹا (CSV files-data with spaces at the beginning)

نوٹ پیڈ کے ذریعے کھولنے پر درج ذیل ڈیٹا پر مشتمل مندرجہ ذیل فائل "sample2.csv" پر غور کریں۔

```
Topic1,    Topic2,    Topic3,
    one,    two,    three
Example1, Example2, Example3
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:47 PM

مندرجہ ذیل پروگرام نے "csv.reader()" کا استعمال کرتے ہوئے پائیتھن کے ذریعے فائل پڑھی۔

```
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()
```

OUTPUT

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```

جیسا کہ آپ "sample2.csv" میں دیکھ سکتے ہیں ڈیلیمیٹر کے بعد خالی جگہیں ہیں جس کی وجہ سے آؤٹ پٹ بھی خالی جگہوں کے ساتھ ظاہر ہوتا ہے۔

csv ماڈیول کی csv.register_dialect ( ) کلاس کا استعمال کرتے ہوئے، نئی بولیوں (dealects) کو regrister کرکے ان وائٹ جگہوں کو بہتر بنایا جاسکتا ہے۔ ایک بولی csv فائل کی شکل لکھتی ہے جسے پڑھنا ہے۔ بولیوں میں پیرامیٹر "skipinitialspace" delimiter کے بعد وائٹ اسپیس کو ہٹانے کے لیے استعمال ہوتا ہے۔

نوٹ:

پہلے سے طے شدہ طور (default) پر "skipinitialspace" کی ایک قیمت false ہے۔

مندرجہ ذیل پروگرام "sample2.csv" فائل پڑھتا ہے، جس میں delimiter کے بعد خالی جگہیں ہیں۔

```
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()
```

OUTPUT

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:47 PM

نوٹ:

بولی csv ماڈیول کی ایک کلاس ہے جو csv پڑھنے اور لکھنے کے پیرامیٹرز کی وضاحت کرنے میں مدد کرتی ہے۔
یہ آپ کو اپنے ڈیٹا کے لیے مختلف فارمیٹنگ پیرامیٹرز بنانے، اسٹور کرنے اور دوبارہ استعمال کرنے کی اجازت دیتا ہے۔

13.6.1.1.3 quotes کے ساتھ CSV فائل ڈیٹا (CSV file -Data With Quotes)

آپ csv ماڈیول کی csv.register_dialect() کلاس کا استعمال کرتے ہوئے نئی بولیوں کو رجسٹر کر کے csv فائل کو کوٹس کے ساتھ پڑھ سکتے ہیں۔

یہاں، ہمارے پاس مندرجہ ذیل ڈیٹا کے ساتھ quotes.csv فائل ہے۔

```
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."
```

مندرجہ ذیل پروگرام "quotes.csv" فائل پڑھتا ہے، جہاں ڈیلیمیٹر کوما (،) ہے، لیکن وہ quotes (واوین) (" ") کے اندر ہیں۔

```
import csv
csv.register_dialect('myDialect',delimiter = ',',quoting=csv.QUOTE_ALL,
skipinitialspace=True)
f=open('c:\\pyprg\\quotes.csv','r')
reader = csv.reader(f, dialect='myDialect')
for row in reader:
    print(row)
```

OUTPUT

```
['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. ']
```

مذکورہ پروگرام میں، myDialect نام کے ساتھ ایک بولی رجسٹر کریں۔ پھر، ہم واوین (double quotes) کے بعد تمام حروف کو ظاہر کرنے کے لیے csv.QUOTE_ALL استعمال کیا۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:47 PM

(d) کسٹم ڈیلیمیٹرز کے ساتھ CSV فائلیں (CSV files with custom delimiters)

آپ csv.register_dialect() کی مدد سے ایک نئی بولی رجسٹر کر کے CSV فائل اپنی مرضی کے حد بندی کو پڑھ سکتے ہیں۔

درج ذیل فائل میں، جسے "sample4.csv" کہا جاتا ہے، ہر کالم کو | کے ساتھ الگ کیا گیا ہے۔ (pipe کی علامت)

Roll No	Name	City
12101	Arun	Chennai
12102	Meena	Kovai
12103	Ram	Nellai
103	Ayush	M
104	Abinandh	M

درج ذیل پروگرام نے صارف کی وضاحت شدہ ڈیلیمیٹر "|" کے ساتھ فائل 'sample4.csv" پڑھی۔

```
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()
```

OUTPUT

```
['RollNo', 'Name', 'City']
['12101', 'Arun', 'Chennai']
['12102', 'Meena', 'Kovai']
['12103', 'Ram', 'Nellai']
```

مذکورہ پروگرام میں، myDialect نامی ایک نئی بولیاں رجسٹرڈ ہیں۔ delimiter=| کا استعمال کریں، جس میں ایک پائپ(|) کالم جدا کار سمجھا جاتا ہے۔

13.6.2 ایک فائل میں ایک مخصوص کالم پڑھنا (Read a specific column In a File)

"sample5.csv" فائل میں صرف آئٹم کا نام اور منافع کے مخصوص کالم حاصل کرنے کے لیے۔

پھر آپ کو درج ذیل کام کرنے ہوں گے:

```
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

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:47 PM

	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

OUTPUT

Item Name Profit
Keyboard 1152
Monitor 10400
Mouse 2000

13.6.3 ایک CSV فائل پڑھنا اور اسے فہرست میں محفوظ کرنا (Read a CSV file and store it in a list)

اس عنوان میں آپ CSV فائل پڑھنے جا رہے ہیں اور فائل کے مندرجات کو بطور فہرست (list) محفوظ کرنا ہے۔

فہرست میں ذخیرہ کرنے کے لیے syntax یہ ہے۔

```
list = []                          # Start as the empty list
list.append(element)               # Use append() to add elements
```

مثال کے طور پر "sample.csv" فائل کی تمام صفوں کی قیمتیں، درج ذیل پروگرام کا استعمال کرتے ہوئے فہرست میں محفوظ ہیں۔

```
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

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:48 PM

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		

sample5.csv File with selected col

OUTPUT

['Topic1', 'Topic2', 'Topic3']
[' one', 'two', 'three']
['Example1', 'Example2', 'Example3']

نوٹ :

ایک List پائیتھن میں ایک ڈیٹاسٹرکچر ہے جو کہ ایک تغیر پذیر، یا بدلنے والے، ترتیب شدہ عناصر کی ترتیب ہے۔

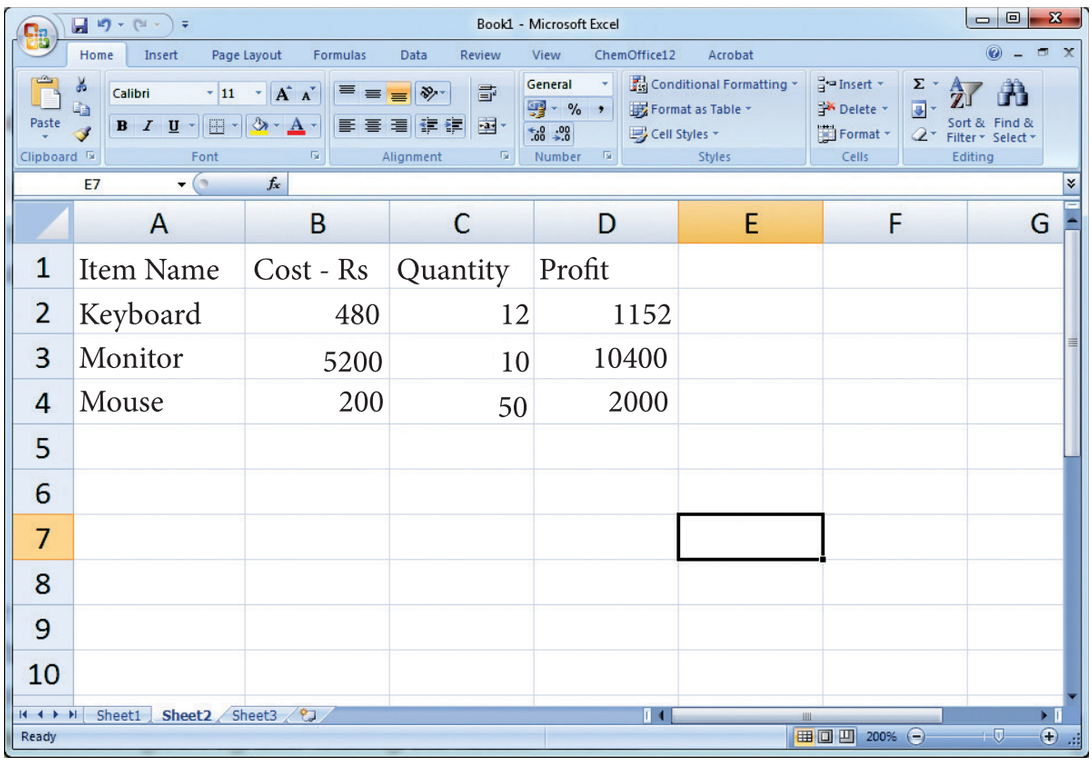
کیا تم جانتے ہو

لسٹ لٹرلز مربع بریکٹ میں لکھے جاتے ہیں [ ]۔ Lists، strings کی طرح کام کرتی ہیں۔

13.6.4 ایک CSV فائل کو پڑھنا اور چھانٹنے (sorting) کے لئے کالم کی قیمتوں کو محفوظ کرنا

اس پروگرام میں آپ صارف سے کالم نمبر حاصل کر کے "sample6.csv" فائل سے منتخب کالم پڑھیں گے اور مواد کو فہرست میں محفوظ کریں گے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:48 PM



خاکہ 13.6.4 CSV فائل چھانٹنے کے لیے منتخب کالم کے ڈیٹا کی CSV فائل

چونکہ صف کی سرخی بھی چھانٹی جاتی ہے، اس سے بچنے کے لیے کہ پہلی صف کو چھوڑ دیا جائے۔
یہ کمانڈ "next()" استعمال کر کے کیا جا سکتا ہے۔ فہرست چھانٹی گئی ہے اور ظاہر کی گئی ہے۔

```
# 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
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:48 PM

```
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 1 to 3:-"))
#  sorting a particular column-cost
for row in reader:
arrayValue.append(row[a])
arrayValue.sort()
for row in arrayValue:
      print (row)
F.close()
```

OUTPUT

```
Enter the column number 1 to 3:- 2
50
12
10
```



csv فائل میں ایک مخصوص کالم پڑھتا ہے اور اس کا نتیجہ بڑھتی ہوئی ترتیب میں دکھاتا ہے۔



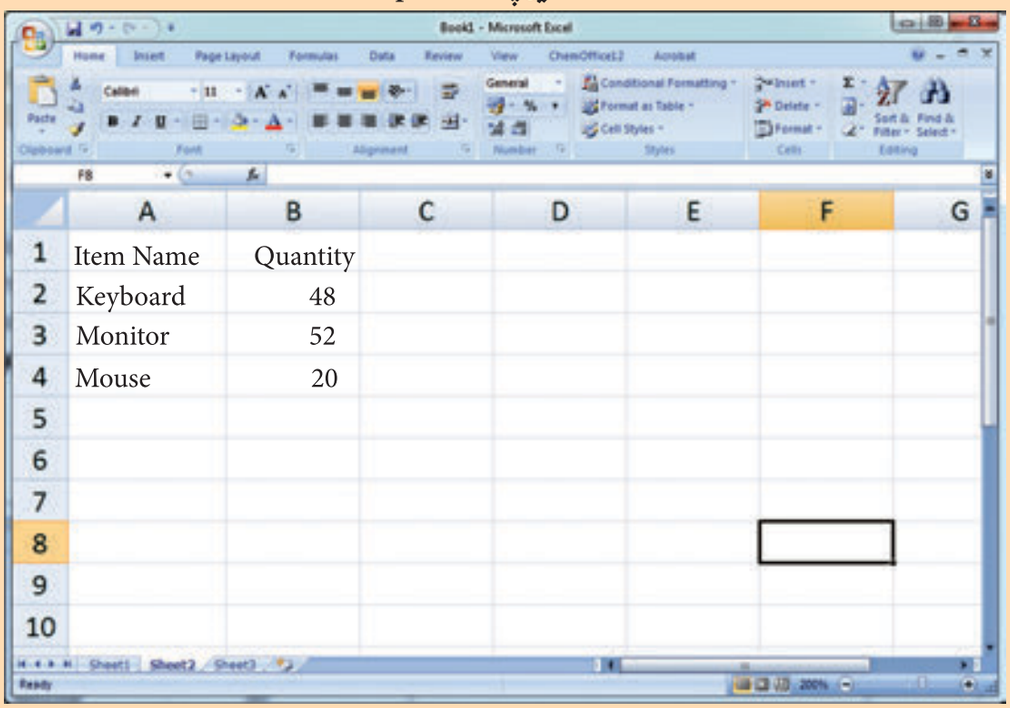
list_name.sort() کمانڈ، list کی قیمت کو بڑھتی ہوئی ترتیب (Ascending order) میں ترتیب دیتی ہے۔
list_name.sort(reverse-True) کمانڈ کو نزولی ترتیب (descending order) میں ترتیب دینے کے لیے استعمال کیا جاتا ہے۔

13.6.5 ایک مخصوص کالم میں ایک CSV فائل کو چھانٹنا (Sorting a CSV file with a specified column)

اس پروگرام میں آپ دیکھیں گے کہ "sample8.csv" فائل کا سارا مواد ایک فہرست میں منتقل ہو گیا ہے۔ پھر صفوں کی فہرست ترتیب دی جاتی ہے اور مقدار کے بڑھتی ہوئی ترتیب میں دکھائی جاتی ہے۔ ایک سے زیادہ کالموں کے مطابق ترتیب دینے کے لیے آپ متعدد اشاریوں کے ساتھ itemgetter(1,2). استعمال کر سکتے ہیں، "sample8.csv" کا مواد یہ ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:48 PM

sample8.csv اسکرین پر Excel



sample8.csv پر Notepad

```
ItemName ,Quantity
Keyboard, 48
Monitor,52
Mouse ,20
```

شکل 13.6.5 ایک list میں چھانٹنے کے لیے ایک CSV فائل میں ڈیٹا

مندرجہ ذیل پروگرام operator.itemgetter(col_no) کا استعمال کرتے ہوئے مذکورہ کام انجام دیتا ہے۔

```
#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)
OUTPUT
    ['Mouse ', '20']
    ['Keyboard ', '48']
    ['Monitor', '52']
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:48 PM

نوٹ:

sorted() طریقہ کسی مخصوص شے کے عناصر کو ایک مخصوص ترتیب میں ترتیب دیتا ہے- بڑھتی ہوئی یا اترتی ہوئی۔ sort() طریقہ اسی طرح انجام دیتا ہے جیسا کہ sorted() ۔ صرف فرق یہ ہے کہ، sort() طریقہ کوئی قیمت واپس نہیں کرتا اور اصل فہرست کو ہی بدل دیتا ہے۔

"sample8.csv" میں ایک اور کالم "cost" شامل کریں اور syntax کا استعمال کرتے ہوئے اسے لاگت کے نزولی ترتیب میں ترتیب دیں۔

```
sortedlist = sorted(data, key=operator.itemgetter(Col_number),reverse=True)
```

13.6.6 ایک dictionary میں CSV فائل پڑھنا

ایک CSV فائل کو dictionary میں پڑھنے کے لیے CSV ماڈیول کی DictReader کلاس کا استعمال کیا جا سکتا ہے، جو کہ reader() کلاس کی طرح کام کرتا ہے، لیکن ایک ایسی شے تخلیق کرتا ہے جو کسی dictionary میں ڈیٹا کا نقشہ بناتی ہے۔ keys بطور fieldname پیرامیٹرس کے طور پر دی گئی ہیں۔DictReader، CSV کی پہلی سطر پڑھ کر کام کرتا ہے اور اس لائن میں ہر comma سے الگ ہونے والی قیمت کو ڈکشنری کی key کے طور پر استعمال کرتا ہے۔ ہر بعد کی صف کے کالم پھر لغت کی اقدار کی طرح برتاؤ کرتے ہیں اور مناسب کلید (fieldname) کے ذریعے ان تک رسائی حاصل کی جا سکتی ہے۔

اگر آپ کے CSV کی پہلی صف میں آپ کے کالم کے نام شامل نہیں ہیں تو آپ لغت کی چابیاں دستی طور پر تفویض کرنے کے لیے فیلڈ نام پیرامیٹر کو DictReader کے کنسٹرکٹر میں منتقل کر سکتے ہیں۔

csv.reader() اور DictReader() کے درمیان بنیادی فرق سادہ الفاظ میں csv ہے۔ پڑھنے والا (reader) اور csv.writer list/tuple کے ساتھ کام کرتے ہیں، جبکہ csv.DictReader اور csv.DictWriter ڈکشنری کے ساتھ کام کرتے ہیں۔

csv.DictReader اور csv.DictWriter اضافی دلیل (additional argument) فیلڈ نام لیتے ہیں جو dictionary keys کے طور پر استعمال ہوتے ہیں۔

مثال کے طور پر "sample8.csv" فائل کو لغت میں پڑھنا

```
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
```

OUTPUT

```
{'ItemName ': 'Keyboard ', 'Quantity': '48'}
{'ItemName ': 'Monitor', 'Quantity': '52'}
{'ItemName ': 'Mouse ', 'Quantity': '20'}
```

مذکورہ پروگرام میں، DictReader() کو "sample8.csv" فائل پڑھنے اور Dictionary میں نقشہ بنانے کے لیے استعمال کیا جاتا ہے۔ پھر فنکشن dict() بغیر ترتیب کے ڈیٹا کو ڈکشنری کی شکل میں پرنٹ کرنے کے لئے استعمال ہوتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:49 PM

مندرجہ بالا پروگرام سے dict() فنکشن کو ہٹا دیں اور print(row) استعمال کرتا ہے۔ چیک کریں کہ آپ کو مندرجہ ذیل آؤٹ پٹ مل رہی ہے۔

```
OrderedDict([('ItemName ', 'Keyboard '), ('Quantity', '48')])
OrderedDict([('ItemName ', 'Monitor'), ('Quantity', '52')])
OrderedDict([('ItemName ', 'Mouse '), ('Quantity', '20')])
```

13.6.7 ایک لغت میں صارف کی وضاحت شدہ Delimiter کے ساتھ CSV فائل پڑھنا

(Reading CSV file with user defined into a dictionary)

آپ نئی بولیاں dialects بھی رجسٹر کر سکتے ہیں اور اسے DictReader() طریقوں میں استعمال کر سکتے ہیں۔ فرض کریں "sample8.csv" درج ذیل فارمیٹ میں ہے۔

ItemName	Quantity
Keyboard	48
Monitor	52
Mouse	20

پھر ایک نئی بولی dialect رجسٹر کر کے "sample8.csv" کو ڈکشنری میں پڑھا جا سکتا ہے۔

```
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()

OUTPUT
{'ItemName ,Quantity': 'Keyboard ,48'}
{'ItemName ,Quantity': 'Monitor,52'}
{'ItemName ,Quantity': 'Mouse ,20'}
```

نوٹ:

DictReader()، اپنے output میں OrderedDict، کو بطور ڈیفالٹ لیتا ہے۔ OrderedDict ڈکشنری کی ایک ذیلی کلاس ہے، جو اس ترتیب کو محفوظ کرتا ہے، جس میں اس کے مندرجات شامل کیے جاتے ہیں۔ OrderedDict کو ہٹانے کے لیے dict() استعمال کریں۔

13.7 CSV فائلوں میں مختلف اقسام میں ڈیٹا لکھنا (Writing Data into different types in CSV files)

جیسا کہ آپ جانتے ہیں کہ پائیتھن CSV فائل کے ساتھ کام کرنے کا ایک آسان طریقہ مہیا کرتا ہے اور CSV فائل میں ڈیٹا پڑھنے اور لکھنے کے لیے CSV ماڈیول رکھتا ہے۔ پچھلے عنوانات میں، آپ نے پائیتھن میں CSV فائلوں کو پڑھنا سیکھا ہے۔ اسی طرح، آپ پائیتھن میں ایک نئی CSV فائل لکھ سکتے ہیں یا موجودہ CSV فائلوں میں ترمیم کر سکتے ہیں۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:49 PM

1. Creating A New Normal CSV File
2. Modifying An Existing File
3. Writing On A CSV File with Quotes
4. Writing On A CSV File with Custom Delimiters
5. Writing On A CSV File with Lineterminator
6. Writing On A CSV File with Quotechars
7. Writing CSV File Into A Dictionary
8. Getting Data At Runtime And Writing In a File

13.7.1 ایک نئی عام CSV فائل بنانا (Creating a new normal CSV file)

جب آپ کے پاس ڈیٹا کا ایک مجموعہ ہوتا ہے جسے آپ CSV فائل کے اندر ذخیرہ کرنا چاہتے ہیں، اب وقت آ گیا ہے کہ اس کے برعکس کریں اور **write** فنکشن استعمال کریں۔

csv.writer() طریقہ ایک رائٹر آبجیکٹ لوٹاتا ہے، جو صارف کے ڈیٹا کو دی گئی فائل، جیسی آبجیکٹ پر **delimited strings** میں بدل دیتا ہے۔ **writerow()** طریقہ مخصوص فائل میں ڈیٹا کی ایک صف لکھتا ہے۔

csv.writer( ) کے لیے **syntax** ہے۔

```
csv.writer(fileobject,delimiter,fmtparams)
```

جس میں

fileobject	:	passes the path and the mode of the file.
delimiter	:	an optional parameter containing the standard dilects like , | etc can be omitted.
fmtparams	:	optional parameter which help to override the default values of the dialects like skipinitialspace,quoting etc. can be omitted.

writer() طریقہ سے CSV ماڈیول کے ساتھ ڈیفالٹ **delimiter**، کوما (,) استعمال کر کے، آپ ایک عام CSV فائل **create** کر سکتے ہیں۔

یہاں ایک مثال ہے۔

درج ذیل پئی تھان پروگرام، ڈیٹا کی **list** کو CSV فائل جسے **"pupil.csv"** کہتے ہیں، وہ **comma** کو قیمت کو الگ کرنے کے لئے استعمال کرتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:49 PM

```
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()
```

جب آپ ٹیکسٹ ایڈیٹر کے ساتھ "Pupil.csv" فائل کھولیں گے، تو یہ مواد کو اس طرح دکھائے گا۔

Student, Age
Dhanush, 17
Kalyani, 18
Ram, 15

مذکورہ پروگرام میں، csv.writer() کا طریقہ، فہرست "csvData" میں موجود تمام ڈیٹا کو string میں تبدیل کرتا ہے اور مواد کو بطور فائل بطور آبجیکٹ تخلیق کرتا ہے۔ writerrows() طریقہ، ایک نئی CSV file، "pupil.csv" میں موجود تمام ڈیٹا کو لکھ دیتا ہے۔

نوٹ:

writerrow( ) طریقہ، ایک ہی وقت میں ایک صف کو لکھتا ہے۔ اگر آپ کو تمام ڈیٹا لکھنا ہو تو آپ writerows() استعمال کر سکتے ہیں۔

13.7.2 ایک موجودہ فائل میں ترمیم (Modifying an existing file)

موجودہ فائل کے ڈیٹا میں کچھ تبدیلیاں کرنا یا مزید ڈیٹا شامل کرنا ترمیم کہلاتا ہے۔ مثال کے طور پر "student.csv" فائل میں درج ذیل ڈیٹا موجود ہے۔

Roll No, Name, City
1, Harshini, Chennai
2, Adhith, Mumbai
3, Dhuruv, Bangalore
4, Krishna, Tiruchy
5, Venkat, Madurai

مندرجہ ذیل پروگرام student.csv میں موجودہ صف کی قیمت میں ترمیم کر کے "student.csv" فائل میں ترمیم کرتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:50 PM

```
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()
```

جب ہم ٹیکسٹ ایڈیٹر کے ساتھ student.csv فائل کھولتے ہیں، تو یہ دکھائے گا:

Roll No	Name	City
1	Harshini,	Chennai
2	Adhith,	Mumbai
3	Dhuruv,	Bengaluru
4	egiste,	Tiruchy
5	Venkat,	Madurai

مذکورہ پروگرام میں، "student.csv" کی تیسری صف میں ترمیم اور محفوظ کیا گیا ہے۔ سب سے پہلے "student.csv" فائل کو csv.reader()فنکشن کے ذریعے پڑھا جاتا ہے۔ پھر، list() فائل کی ہر صف کو محفوظ کرتی ہے۔ بیان "lines[3] = row"، فائل کی تیسری صف کو "row" نے نئے مواد کے ساتھ تبدیل کر دیا۔ writerrows(lines) کا استعمال کرتے ہوئے آبجیکٹ رائٹر کی فائل list کی قیمتوں کو "student.csv" فائل میں لکھتی ہے۔

13.7.2.1 نئی صف شامل کرنا (Adding new row)

بعض اوقات، آپ کو موجودہ CSV فائل میں نئی قطاریں شامل کرنے کی ضرورت پڑ سکتی ہے۔ فائل کے اختتام پر ایک نئی صف شامل کرنے کو صف جوڑنا (appending a row) کہا جاتا ہے۔

مندرجہ ذیل پروگرام موجودہ "student.csv" فائل میں ایک نئی صف شامل کرتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:50 PM

```
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()
```

جب"student.csv" فائل text editor کے ساتھ کھولی جاتی ہے،تو یہ مندرجہ ذیل دکھاتا ہے۔

Roll No	Name	City
1	Harshini,	Chennai
2	Adhith,	Mumbai
3	Meena	Bengaluru
4	egiste,	Tiruchy
5	Venkat,	Madurai
6	Sajini ,	Madurai

مذکورہ پروگرام میں،ایک نئی صف کو"student.csv" میں شامل کیا گیا ہے۔اس کے لیے،صرف CSV فائل 'a' append mode میں کھولی جاتی ہے۔ "student.csvfile" کی آخری صف کے بعد append mode صف کی قیمت لکھتی ہے۔

کیا تم جانتے ہو!

'w' لکھنے کا موڈ ایک نئی فائل بناتا ہے۔اگر فائل پہلے سے موجود ہے تو 'w' موڈ اس کے اوپر لکھتا ہے۔جہاں 'a' اپینڈ موڈ کے طور پر فائل کے آخر میں ڈیٹا شامل کرتا ہے، وہاں اگر فائل پہلے سے موجود ہے تو ٹھیک ہے، ورنہ یہ ایک نئی فائل بناتا ہے۔

نوٹ

ایک فائل لکھنے کے لئے writerow() ایک ابعادی ڈیٹا (ایک صف) لیتا ہے،اور writerrows دوابعادی (کثیر صفوں) ڈیٹا لیتا ہے۔

13.7.3 کوٹس (واوین) کے ساتھ CSV فائلیں (CSV files with quotes)

csv ماڈیول کی کلاس، () csv.register_dialect استعمال کرتے ہوئے نئی بولیوں کو رجسٹر کر کے آپ csv فائل کو کوٹس (واوین) (quotes) کے ساتھ لکھ سکتے ہیں۔ مندرجہ ذیل پروگرام اس کی وضاحت کرتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:50 PM

```
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()
```

جب آپ "person.csv" فائل کھولتے ہیں تو ہمیں مندرجہ ذیل آؤٹ پٹ ملتا ہے۔

```
"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"
```

مذکورہ پروگرام میں، myDialect کے نام سے ایک بولی (registered(declared ہے۔

Quoting=csv.QUOTE_ALL تمام قیمتوں کے لئے واوین (double quote) " لکھنے کی اجازت دیتا ہے۔

13.7.4 CSV فائلیں کسٹم ڈیلیمیٹرز کے ساتھ (CSV files with custom delimiters)

ڈیلیمیٹر ایک string ہے، جو فیلڈز کو الگ کرنے کے لیے استعمال ہوتی ہے۔ ڈیفالٹ ویلیو کوما (،) ہے۔ آپ csv.register_dialect() کی مدد سے ایک نئی بولی رجسٹر کر کے CSV فائلوں میں کسٹم ڈیلیمیٹر رکھ سکتے ہیں۔ اپنی مرضی کے مطابق ڈیلیمیٹر پائپ کو استعمال کر کے لکھنے کے لئے یہ مثال (|) ہے۔

```
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()
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:51 PM

جب ہم "dob.csv" فائل کھولتے ہیں تو ہمیں درج ذیل آؤٹ پٹ ملتے ہیں۔

```
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
```

مندرجہ بالا پروگرام میں، ایک بولی بطور pipe(|) رجسٹرڈ ہے۔ پھر فہرست "info" CSV فائل "dob.csv" میں لکھی گئی ہے۔



نوٹ:

بولی (dialect) کا پیرامیٹر skipinitialspace، جب یہ True ہے تو ڈیلیمیٹر کے فوراً بعد خالی جگہ (white space) کو نظر انداز کر دیا جاتا ہے۔ اس کا ڈیفالٹ False ہے۔

13.7.5 لائن ٹرمینیٹر کے ساتھ ایک CSV فائل (CSV file with a line terminator)

ایک لائن ٹرمینیٹر ایک string ہے، جو writer کی تیار کردہ لائنوں کو ختم کرنے کے لیے استعمال ہوتی ہے۔ پہلے سے طے شدہ قیمت \r یا \n ہے۔ ہم csv ماڈیول کی csv.register_dialect() کلاس کا استعمال کرتے ہوئے نئی بولیوں کو رجسٹر کر کے پائیتھن میں لائن ٹرمینیٹر کے ساتھ csv فائل لکھ سکتے ہیں۔ مثال کے طور پر

```
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()
```

جب ہم line.csv فائل کھولتے ہیں تو ہمیں لائنوں کے درمیان وقفہ کے ساتھ درج ذیل آؤٹ پٹ ملتا ہے۔

```
Fruit|Quantity

Apple|5

Banana|7

Mango|8
```

مذکورہ کوڈ میں، نئی بولی "myDialect" اس کو استعمال کرتی ہے delimiter='|' جس میں | (pipe) کالم جدا کار (Column separator) سمجھا جاتا ہے۔ لائن ٹرمینیٹر lineterminator='\r\n\r\n' ہر صف کو الگ کرتا ہے اور ہر دو لائنوں کے بعد ڈیٹا دکھاتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:51 PM

نوٹ:

پائیتھن کا CSV ماڈیول صرف \r\n ، \n یا \r کو لائن ٹرمینیٹر کے طور پر قبول کرتا ہے۔

13.7.6 واوین کے علامات کے ساتھ CSV فائل (CSV file with quote characters)

آپ csv ماڈیول کی csv.register_dialect() کلاس کا استعمال کرتے ہوئے نئی بولیوں کو رجسٹر کر کے اپنی مرضی کے مطابق علامت (characers) کے ساتھ CSV فائل لکھ سکتے ہیں۔

```
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()
```

جب آپ نوٹ پیڈ میں "quote.csv" فائل کھولتے ہیں تو ہمیں مندرجہ ذیل آؤٹ پٹ ملتا ہے۔

```
"SNO"|"Items"
"1"|"Pen"
"2"|"Book"
"3"|"Pencil"
```

مذکورہ پروگرام میں، myDialect فائل کے اندر لکھنے کے لیے pipe(|) کو بطور ڈیلیمیٹر اور quotechar ڈبل کوٹ "" کے طور پر استعمال کرتا ہے۔

13.7.7 ایک ڈکشنری میں CSV فائل لکھنا (Writing CSV file into a dictionary)

csv ماڈیول کی DictWriter() کلاس کا استعمال کرتے ہوئے، ہم csv فائل کو ڈکشنری میں لکھ سکتے ہیں۔ یہ ایک ایسی شے بناتی ہے، جو ڈیٹا کو ڈکشنری میں نقشہ بناتی ہے۔ چابیاں فیلڈ نام پیرامیٹر کے ذریعہ دی گئی ہیں۔ درج ذیل پروگرام ڈکشنری کو فائل میں لکھنے میں مدد کرتا ہے۔

```
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()
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:51 PM

جب آپ نوٹ پیڈ میں "peak.csv" فائل کھولتے ہیں تو آپ کو درج ذیل آؤٹ پٹ ملتا ہے۔

```
MOUNTAIN,HEIGHT
Everest,8848
Anamudi,2695
Kanchenjunga,8586
```

مذکورہ پروگرام میں، فیلڈ ناموں کو csv فائل میں ہر کالم کے عنوان کے طور پر استعمال کریں۔ پھر، ڈکشنری ڈیٹا کو "peak.csv" فائل میں لکھنے کے لیے DictWriter() استعمال کریں۔

13.7.7.1 حسب ضرورت بولیوں کے ساتھ CSV فائل میں لغت لکھنا

(Writing Dictioniary 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")
```

جب ہم grade.csv فائل کھولتے ہیں، تو اس میں درج ذیل آؤٹ پٹ ہوں گے۔

```
"Name"|"Grade"
"Anu"|"B"
"Beena"|"A"
"Tarun"|"C"
```

مذکورہ پروگرام میں، ایک اپنی مرضی کی بولی جسے mydialect کہا جاتا ہے، pip(|) کے ساتھ بطور ڈیلیمیٹر فیلڈ ناموں کو ہر کالم کے عنوان کے طور پر csv فائل میں لکھنے کے لیے استعمال کرتا ہے۔ آخر میں، ہم لغت کا ڈیٹا "grade.csv" فائل میں لکھنے کے لیے ایک DictWriter() استعمال کرتے ہیں۔

13.7.8 رن ٹائم پر ڈیٹا حاصل کرنا اور اسے CSV فائل میں لکھنا

(Getting Data at runtime and writing it in a csv file)

یہاں تک کہ آپ کی بورڈ کے ذریعے ڈیٹا کو قبول کر سکتے ہیں اور CSV فائل میں لکھ سکتے ہیں۔ مثال کے طور پر درج ذیل پروگرام صارف سے ڈیٹا کو کلیدی بورڈ کے ذریعے قبول کرتا ہے اور اسے "dynamicfile.csv" نامی فائل میں محفوظ کرتا ہے۔ یہ فائل کا مواد بھی دکھاتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:51 PM

```
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()
```

OUTPUT

```
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			

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:52 PM

یادرکھنے کے لیے نکات Points to remember:

- ایک CSV فائل ایک انسان کے پڑھنے کے قابل ٹیکسٹ فائل ہے، جہاں ہر لائن کے کئی فیلڈز ہوتے ہیں، جو کوما یا کسی دوسرے ڈیلیمیٹر سے الگ ہوتے ہیں۔
- ایکسل ایک بائنری فائل ہے جبکہ CSV فارمیٹ ایک سادہ ٹیکسٹ فارمیٹ ہے۔
- CSV فائل کو پڑھنے کے دو طریقے csv.reader() فنکشن اور DictReader کلاس کا استعمال کر رہے ہیں۔
- پڑھنے اور لکھنے میں CSV فائل کا ڈیفالٹ موڈ، ٹیکسٹ موڈ ہے۔
- جب غیر ٹیکسٹ فائلوں جیسے امیج یا exe فائلوں سے نمٹنا ہو تو بائنری موڈ استعمال کیا جا سکتا ہے
- پیتھون کے پاس ایک ردی garbage جمع کرنے والا ہوتا ہے، جو غیر حوالہ شدہ اشیاء (unreferred objects) کو صاف کرتا ہے۔
- close() طریقہ ان وسائل کو آزاد کرے گا، جو فائل کے ساتھ بندھے ہوئے ہوتے ہیں۔
- ایکسل میں بطور ڈیفالٹ CSV فائلیں، خود بخود کھل جائیں۔
- CSV لائبریری میں CSV فائلوں سے ڈیٹا کو پڑھنے، لکھنے اور اس پر کارروائی کرنے کے لیے اشیاء اور دیگر کوڈ شامل ہیں۔
- "skipinitialspace" ڈیلیمیٹر کے بعد وائٹ اسپیس کو ہٹانے کے لیے استعمال کیا جاتا ہے۔
- ایک سے زیادہ کالم آپریٹر کے ذریعے ترتیب دینے کے لیے operator.itemgetter () استعمال کیا جا سکتا ہے۔
- csv ماڈیول کی کلاس DictReader( ) ایک شے بناتی ہے جو ڈیٹا کو ڈکشنری میں نقشہ سازی (map) کرتی ہے۔
- csv.register_dialect( ) کی مدد سے کسٹم ڈیلیمیٹر والی CSV فائل پڑھی جاتی ہے۔
- ایک سے زیادہ کالم کے ذریعے ترتیب دینے کے لیے itemgetter( ) متعدد اشاریوں کے ساتھ استعمال کیا جاتا ہے۔
- csv.reader اور csv.writer , tuple/list کے ساتھ کام کرتے ہیں۔
  جبکہ csv.DictReader اور csv.DictWriter ڈکشنری کے ساتھ کام کرتے ہیں۔
- csv.DictReader اور csv.DictWriter اضافی دلیل فیلڈ نام لیتے ہیں جو ڈکشنری کی keys کے طور پر استعمال ہوتے ہیں۔
- فنکشن dict () ڈیٹا کو لغت کی شکل میں بغیر آرڈر کے پرنٹ کرنے کے لیے استعمال ہوتا ہے۔
- csv.writer( ) طریقہ ایک رائٹر آبجیکٹ لوٹاتا ہے, جو صارف کے ڈیٹا کو حد بندی کی اسٹرنگ میں بدل دیتا ہے۔
- writerrow ( ) طریقہ ایک وقت میں ایک صف لکھتا ہے۔ Writerows() طریقہ ایک ساتھ تمام ڈیٹا لکھنے کے لیے استعمال کیا جاتا ہے۔
- فائل کے اختتام پر ایک نئی صف شامل کرنے کو صف جوڑنا (appending a row) کہا جاتا ہے۔

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:52 PM

Hands on Experience عملی مشق

1۔ درج ذیل Namelist.csv فائل کو پڑھنے کے لیے ایک پائیتھن کا پروگرام لکھیں، اور ناموں کو حروف تہجی کے مطابق فہرست میں ترتیب دیں اور آؤٹ پٹ دکھائیں۔

	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 طالب علموں کے نام اور پانچ مضامین کے مارکس کو قبول کرنے کے لیے ایک پائیتھن کا پروگرام لکھیں . کل مارکس معلوم کریں اور طلباء کی تمام تفصیلات CSV فائل میں محفوظ کریں۔

MQDPW

Evaluation محاسبہ

I - حصہ

صحیح جواب کا انتخاب کریں۔ (1 مارکس)

1۔ ایک CSV فائل کو بھی کہا جاتا ہے۔

(A) Flat File (B) 3D File

(C) String File (D) Random File

2۔ CRLF کی توسیع (پھیلاؤ) یہ ہے۔

(A) Control Return and Line Feed

(B) Carriage Return and Form Feed

(C) Control Router and Line Feed

(D) Carriage Return and Line Feed

3۔ درج ذیل میں سے کون سا ماڈیول پائیتھن نے CSV فائلوں پر کئی آپریشن کرنے کے لیے فراہم کیا ہے ؟

(A) py (B) xls

(C) csv (D) os

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:52 PM

4۔ تصویر یا exe فائلوں جیسی غیر ٹیکسٹ فائلوں سے نمٹنے کے لیے مندرجہ ذیل میں سے کون سا موڈ استعمال کیا جاتا ہے ؟

(A) Text mode (B) Binary mode

(C) xls mode (D) csv mode

5۔ ایک CSV فائل میں ایک صف کو چھوڑنے (skip) کے لیے استعمال ہونے والی کمانڈ ہے۔

(A) next() (B) skip()

(C) omit() (D) bounce()

6۔ CSV ماڈیول کے writer()method طریقہ کار کے ذریعہ تیار کردہ لائنوں کو ختم کرنے کے لیے مندرجہ ذیل میں سے کونسی string استعمال کی جاتی ہے ؟

(A) Line Terminator (B) Enter key

(C) Form feed (D) Data Terminator

7۔ درج ذیل پروگرام کا output کیا ہے ؟

```
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
```

A) chennai,mylapore (B) mumbai,andheri

(C) chennai (D) chennai,mylapore

mumbai mumbai,andheri

8۔ درج ذیل میں سے کونسا ڈکشنری میں ڈیٹا کو map کرنے کے لئے استعمال ہوتا ہے۔

(A) listreader() (B) reader()

(C) tuplereader() (D) DicReader ()

9۔ موجودہ فائل کے ڈیٹا میں کچھ تبدیلیاں کرنا یا مزید ڈیٹا شامل کرنے کا عمل یہ کہلاتا ہے۔

(A)Editing (B) Appending

(C)Modification (D) Alteration

10۔ درج ذیل پروگرام import csv کا استعمال کرتے ہوئے فائل test.csv کے اندر کیا لکھا جائے گا۔

```
import csv
D = [['Exam'],['Quarterly'],['Halfyearly']]
csv.register_dialect('M',lineterminator = '\n')
```


12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:52 PM

```
with open('c:\pyprg\ch13\line2.csv', 'w') as f:
wr = csv.writer(f,dialect='M')
wr.writerows(D)
f.close()
```

(A) Exam Quarterly Halfyearly	(B) Exam Quarterly Halfyearly
(C) E Q H	(D) Exam, Quarterly, Halfyearly

حصہ - II

درج ذیل سوالات کے جوابات لکھئے۔ (2 مارکس)

1۔ CSV فائل کیا ہے؟

2۔ پائیتھن کا استعمال کرتے ہوئے CSV فائل پڑھنے کے دو طریقے بتائیں۔

3۔ فائل کے Default modes کا ذکر کریں۔

4۔ next() فنکشن کا استعمال کیا ہے؟

5۔ آپ csv فائل سے ایک سے زیادہ کالم کیسے چھانٹیں (sort) گے؟ ایک مثال کا بیان دیں۔

حصہ - III

درج ذیل سوالات کے جوابات لکھئے۔ (3 مارکس)

1۔ پائیتھن کے open() فنکشن پر نوٹ لکھیں۔ دونوں طریقوں میں کیا فرق ہے؟

2۔ ایک موجودہ فائل میں ترمیم کرنے کے لیے ایک پائیتھن کا پروگرام لکھیں۔

3۔ ڈیفالٹ ڈیلیمیٹر کوما (،) والی CSV فائل پڑھنے کے لیے پائیتھن کا پروگرام لکھیں۔

4۔ لکھنے کے موڈ (write mode) اور اپینڈ موڈ (append mode) میں کیا فرق ہے؟

5۔ reader() اور DictReader() فنکشن میں کیا فرق ہے؟

حصہ - IV

درج ذیل سوالات کے جوابات لکھئے۔ (5 مارکس)

1۔ ایکسل فائل اور CSV فائل میں فرق کریں۔

2۔ مختلف موڈ کو اس کے معنی کے ساتھ جدول بندی کریں۔

3۔ پائیتھن میں فائل پڑھنے کے مختلف طریقے لکھیں۔

4۔ اپنی مرضی کے کوٹ (custom quotes) کے ساتھ ایک CSV فائل لکھنے کے لیے ایک پائیتھن کا پروگرام لکھیں۔

5۔ CSV فائل میں ڈیٹا کو فارمیٹ کرنے کے لیے کن اصولوں پر عمل کیا جائے، ان کو لکھیں۔

حوالہ جات 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***

12 CS UNIT 13 CDR Set
Monday, April 17, 2023 9:32:53 PM

GW2R6

سیکھنے کے مقاصد

اس باب کی تکمیل کے بعد، طالب علم قابل ہو جائے گا۔

- سمجھیں کہ ریپنگ (wrapping) کیا ہے۔
- پائیتھن پروگراموں میں C++ افعال اور کلاس درآمد کرنے کے قابل۔
- دونوں زبانوں کے ساتھ کام کرنے کے لیے ماحول بنائیں۔
- پائیتھن کے پروگراموں کو چلائیں اور ڈیبگ (debug) کریں۔

14.1 تعارف Introduction

پائیتھن اور C++ عام مقصد کے پروگرامنگ زبان ہے۔تاہم، پائیتھن C++ سے بالکل مختلف ہے۔

شمار عدد	PYTHON	C++
-1	Python ایک ترجمانی زبان ہے۔	C++ ایک مرتبی زبان ہے۔
-2	Python ایک محرک زبان ہے۔	C++ شماریاتی قسم کی زبان ہے۔
-3	جب متغیر کو ظاہر کیا جاتا ہے تو ڈیٹا کے قسم کی ضرورت نہیں پڑتی۔	متغیر کو ظاہر کئے جانے پر Datatype کی ضرورت پڑتی ہے۔
-4	اسے اسکرپٹنگ اور عام مقصد کی زبان کے طور پر استعمال کیا جاتا ہے۔	یہ ایک عام مقصد کی زبان ہے۔

پھر بھی یہ دونوں زبانیں ایک دوسرے کی مکمل تکمیل کرتی ہیں۔python زیادہ تر اسکرپٹنگ یا "glue"، زبان کے طور پر استعمال ہوتا ہے۔ یعنی، اعلی سطحی پروگرام زیادہ تر C یا C++ میں لکھے گئے معمولات کو کال کرتا ہے۔ یہ اُس وقت مفید ہے جب منطق، موجودہ کوڈ کے لحاظ سے لکھی جاسکتی ہے۔ (مثال کے طور پر C++ میں لکھا ہوا پروگرام) لیکن اسے python پروگرام کے ذریعے بلایا اور اسے برتا جاسکتا ہے۔

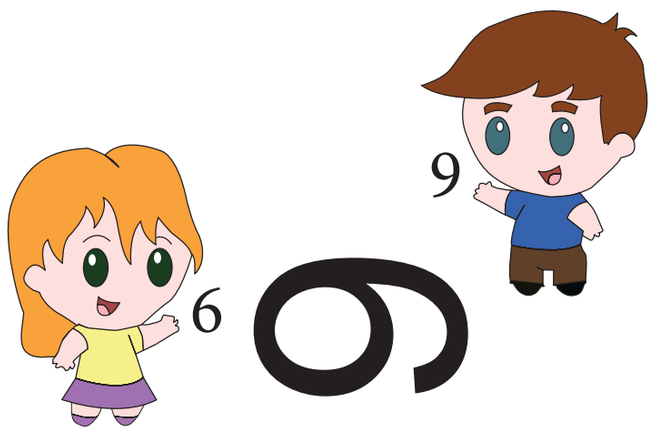
12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:04 PM

14.2 اسکرپٹ لینگوئج (Script language)

اسکرپٹ لینگوئج ایک پروگرامنگ لینگوئج ہے جو دوسری پروگرامنگ لینگوئجز کے ساتھ رابطہ قائم کرنے کے لیے بنائی گئی ہے۔ سب سے زیادہ استعمال ہونے والی اسکرپٹنگ زبانیں جاوا اسکرپٹ، وی بی ایس سکرپٹ، PHP، Perl، Python، Ruby، ASP اور TCl ہیں۔ چونکہ ایک سکرپٹ لینگوئج عام طور پر کسی اور پروگرامنگ لینگوئج کے ساتھ مل کر استعمال ہوتی ہے، وہ اکثر HTML، C++ کے ساتھ پائی جاتی ہیں۔

14.2.1 اسکرپٹنگ اور پروگرامنگ زبانوں میں فرق

سکرپٹ لینگوئج اور پروگرامنگ لینگوئج مندرجہ ذیل تصویر کی طرح دکھائی دیتی ہے۔



بنیادی طور پر، تمام اسکرپٹنگ زبانیں پروگرامنگ زبانیں ہیں۔ دونوں کے مابین نظریاتی فرق یہ ہے کہ سکرپٹ زبانوں کو تالیف کے مرحلے کی ضرورت نہیں ہوتی ہے، بلکہ ان کی تشریح کی جاتی ہے۔ مثال کے طور پر، عام طور پر، C++ پروگرام کو چلانے سے پہلے compile کرنے کی ضرورت ہوتی ہے، جبکہ جاوا اسکرپٹ یا پیتھان جیسی اسکرپٹنگ زبان کو مرتب کرنے کی ضرورت نہیں ہوتی ہے۔ ایک سکرپٹ لینگوئج کے لیے ایک مترجم (interpreter) کی ضرورت ہوتی ہے جبکہ ایک پروگرامنگ لینگوئج کو ایک compiler کی ضرورت ہوتی ہے۔ دی گئی زبان کو اسکرپٹنگ یا پروگرامنگ لینگوئج کہا جاسکتا ہے ۔ یہ اس پر منحصر ہے کہ وہ کس ماحول میں استعمال ہوتے ہیں۔

14.3 اسکرپٹنگ زبانوں کے اطلاقات (Applications of Scripting languages)

1۔ کسی پروگرام میں کچھ کاموں کو خودکار (automate) کرنا۔

2۔ ڈیٹا سیٹ سے معلومات نکالنا۔

3۔ روایتی پروگرامنگ زبان کے مقابلے میں کم کوڈ کا استعمال۔

4۔ ایپلی کیشنز میں نئے افعال لا سکتے ہیں اور پیچیدہ نظاموں کو ایک ساتھ جوڑ سکتے ہیں۔

پیتھون دراصل ایک تشریح شدہ، اعلیٰ سطحی، عام مقصد والی پروگرامنگ زبان ہے جو کسی بھی جدید کمپیوٹر آپریٹنگ سسٹم پر استعمال کی جاسکتی ہے۔ اس کا استعمال ٹیکسٹ، نمبرز، امیجز، سائنسی ڈیٹا اور کسی بھی دوسری چیز کے لیے کیا جاسکتا ہے جسے آپ کمپیوٹر پر محفوظ کر سکتے ہیں۔ آج کل، بڑی ایپلی کیشنز تقریبا خصوصی طور پر پیتھون میں لکھی جاتی ہیں۔

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:05 PM

14.4 C++ پر python کی خصوصیات۔

- پیتھون اپنے آپ ردّی (garbage) جمع کرتا ہے، جبکہ C++ نہیں کرتا۔
- C++ ایک مستحکم ٹائپ شدہ زبان ہے، جبکہ پیتھون متحرک طور پر ٹائپ کردہ زبان ہے۔
- پیتھون ایک مترجم interpreter کے ذریعے چلتا ہے، جبکہ C++ پہلے سے مرتب Compiled ہوتا ہے۔
- پیتھون کوڈ C++ میں لکھے گئے کوڈ سے 5 سے 10 گنا چھوٹا ہوتا ہے۔
- پیتھون میں، واضح طور پر اقسام کے اعلان کرنے کی ضرورت نہیں ہے، جیسا کہ C++ میں کیا جانا چاہئے۔
- پیتھون میں، ایک فنکشن کسی بھی قسم کی دلیل (argument) کو قبول کرسکتا ہے، اور بغیر کسی قسم کے اعلان کے کئی اقدار واپس کرسکتا ہے۔ جبکہ C++ میں واپسی کا بیان صرف ایک قدر واپس کرسکتا ہے۔

نوٹ:

پیتھون میموری کی جگہ کو خالی کرنے کے لیے غیر پسندیدہ اشیاء (بلٹ ان اقسام یا کلاس مثالیں) خود بخود حذف کر دیتی ہے۔ وہ عمل جس کے ذریعے پیتھون وقتاً فوقتاً میموری کے بلاکس کو آزاد اور دوبارہ حاصل کرتا ہے، جو اب استعمال میں نہیں ہے اسے ردی جمع کرنا کہا جاتا ہے۔

14.5 پیتھون میں C++ فائل درآمد کرنا (Importing C++ files in python)

ایک پیتھون پروگرام میں C++ پروگرام درآمد کرنے کو پیتھون میں C++ کو سمیٹنا کہا جاتا ہے۔ C++ پروگراموں کے لیے python کے انٹرفیس کو لپیٹنا یا بنانا کئی طریقوں سے کیا جاتا ہے۔ عام طور پر استعمال ہونے والے انٹرفیس ہیں۔



- Python-C-API جو C (Application Programming Interface) پروگراموں کے لئے انٹرفیس مہیا کرتا ہے۔
- ctypes (C پروگراموں کے ساتھ مداخلت کے لیے)
- SWIG (Simplified Wrapper Interface Generator) C اور C++ دونوں کے لئے
- Cython (Cython اور پیتھون دونوں ایک جیسی زبانیں ہیں جو C-Extension لکھنے کے لیے استعمال کی جاتی ہیں۔)
- Boost۔ پیتھون (پیتھون اور C++ کو انٹرفیس کرنے کا ایک فریم ورک)
- MinGW (ونڈوز کے لیے کم سے کم GNU)

14.5.1 MinGW انٹرفیس

MinGW رن ٹائم ہیڈر فائلوں کے ایک سیٹ سے مراد ہے، جو ونڈوز آپریٹنگ سسٹم پر چلائے جانے والے C، C++ اور فورٹران کے کوڈ کو مرتب کرنے اور جوڑنے میں استعمال ہوتا ہے۔

MinGW-W64 (MinGW کا ورژن)، ونڈوز پر C++ کے لیے بہترین مرتب (compiler) ہے۔ C++ پروگرام کو مرتب کرنے اور اس پر عمل کرنے کے لیے، آپ کو ونڈوز کے لیے 'g++' کی ضرورت ہے۔

MinGW کا استعمال کرتے ہوئے python پروگرام کے ذریعے متحرک طور پر C++ پروگرام مرتب اور چلانے کی اجازت دیتا ہے۔ python کا پروگرام، جس میں C++ کوڈنگ ہوتی ہے، کمانڈ پرامپٹ کے ذریعے یا رن ٹرمینل کا استعمال کر کے عمل میں لایا جاسکتا ہے۔

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:05 PM

g++ ایک پروگرام ہے جو GCC (GNU C Compiler) کو کال کرتا ہے اور خودکار طور پر مطلوبہ C++ لائبریری فائلوں کو آبجیکٹ کوڈ سے جوڑتا ہے۔

g++ is a program that calls GCC (GNU C Compiler) and automatically links the required C++ library files to the object code.

ضمیمہ-2 میں MinGW کی تنصیب کا حوالہ دیں۔

14.5.2 ازگر کے ذریعے C++ پروگرام پر عمل درآمد (Executing C++ program through python)

1۔ کمانڈ پرامپٹ یا رن ٹرمینل پر ڈبل کلک کریں۔

2. Go to the folder where the Python software is located (Python.exe) is located. In this example “Python folder” is located in

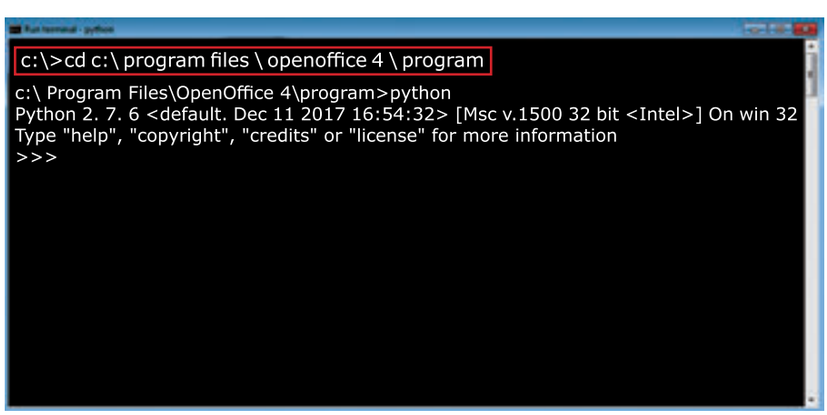
   For example here “Python” is located in

   C:\Program Files\OpenOffiice 4\Python.

The syntax to change from c:\> to the folder where Python is located is

cd <absolute path>

where “cd” command refers to change directory and absolute path refers to the complete path where Python is installed.



خاکہ 14.1

اس مثال میں اشارہ "C:\>" دکھاتا ہے۔ اوپر والی ونڈو میں وہ نمایاں علاقہ دیکھیں۔ ڈائریکٹری کو تبدیل کرنے کے لیے 'cd' کمانڈ استعمال کی جاتی ہے۔ مثال کے طور پر ڈائریکٹری .pyprg کو حاصل کرنے کے لیے، کمانڈ پرامپٹ میں 'cd pyprg' کمانڈ ٹائپ کریں۔

مثال پر غور کریں pycpp.py ایک python کا پروگرام ہے، جو C++ پروگرام Pali.cpp کو پڑھے گا۔"Pali.cpp" پروگرام ایک عدد کو قبول کرتا ہے اور ظاہر کرتا ہے کہ یہ "Palindrome"(الٹی ترتیب میں پڑھنے پر بھی الفاظ کا نہیں بدلنا) ہے یا نہیں۔ مثال کے طور پر داخل کیا گیا ان پٹ نمبر 232 ہے، جو آؤٹ پٹ ظاہر ہوتا ہے وہ 'Palindrome" ہوگا۔

C++ پروگرام کے نوٹ پیڈ میں pali ٹائپ کیا جاتا ہے اور pali.cpp کے طور پر محفوظ ہوتا ہے۔ اسی طرح python پروگرام pycpp.py کوڈ بھی نوٹ پیڈ میں ٹائپ کیا جاتا ہے اور pycpp.py کے طور پر محفوظ کیا جاتا ہے۔

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:05 PM

3۔ ہمارے پروگرام کو چلانے کے لیے رن ٹرمینل پر ڈبل کلک کریں۔ python فولڈر کے مقام کا راستہ تبدیل کریں۔
python پروگرام پر عمل کرنے کے لیے syntax یہ ہے۔

Python <filename.py> -i <C++ filename without cpp extension>

جس میں

Python پیتھون	کمانڈ لائن سے پیتھون پروگرام کو چلانے کے لیے keyword ہے۔
filename.py	عملدرآمد کرنے کے لیے پیتھون پروگرام کا نام
- i	ان پٹ موڈ input mode
cpp توسیع کے بغیر C++ فائل کا نام C++ filename without cpp extension	C++ فائل کا نام جسے مرتب کر کے عمل درآمد کرنا ہے۔

مثال کے طور پر کمانڈ پرامپٹ میں Python pycpp.py –i pali ٹائپ کریں اور enter key دبائیں۔ اگر تالیف کامیاب ہے تو آپ کو مطلوبہ آؤٹ پٹ ملے گا۔ ورنہ error ظاہر کرے گا۔

نوٹ:

عملدرآمد کمانڈ میں، ان پٹ فائل کو اس کی توسیع کی ضرورت نہیں ہے۔ مثال کے طور پر "pali.cpp" کے بجائے صرف "pali" نام کا ذکر کرنا کافی ہے۔

اب ہم اپنی مثال pycpp.py اور pali.cpp کے ذریعے عملدرآمد دیکھیں گے۔ یہ دونوں پروگرام فولڈر c:\cyprg میں محفوظ ہیں۔
اگر پروگرام ایک ہی فولڈر میں موجود نہیں ہیں تو پھر عمل کے دوران فائلوں کے لیے مکمل راستہ متعین کرنا ضروری ہے۔ آؤٹ پٹ نیچے دکھایا گیا ہے۔

```
C:\Program Files\migw-w64\i686-8.1.0-posix-dwarf-rt_v6-rev0>echo off
Microsoft windows [Version 6.1.7601]
copy right <c> 2009 Microsoft Corporation. All rights reserved.

C:\>cd program files\openoffice 4 program

C:\Program Files\openoffice 4\program> Python c:\pyprg\pycpp.py-i c:\pyprg\pali
compailing c:\pyprg\pali.cpp
Running c:\pyprg\pali.exe
----------------------
Enter apositive number:232
  The reverse of the number is:232
  The number is a palindrome
C:\Program jFiles\Openoffice 4\program>Python c:\pyprg\pycpp.py -i c:\pyprg\pali
Compiling c:\pyprg\pali.cpp
Running c:\pyprg\pali.exe
----------------------
Enter a positive number:234
  The reverse of the number is:432
  The number is not a palindrome
C:\Program Files\Openoffice 4\program>_
```

Fig 14.2

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:06 PM

نوٹ

کمانڈ ونڈو میں سکرین clear کرنے کے لیے cls کمانڈ استعمال کریں۔

اب آیئے دیکھتے ہیں کہ C++ کوڈ مرتب کرنے کے لیے Python پروگرام کیسے لکھا جائے۔

14.6 C++ درآمد کرنے کے لیے پیتھون پروگرام (Python program to import C++)

پیتھان کئی ماڈیولز پر مشتمل ہے۔ ایک مسئلے کے لیے پیتھان پروگرامرز کو اپنی سہولت کے مطابق مختلف ماڈیول استعمال کرنے میں لچک کی اجازت دیتا ہے۔ پیتھان پروگرام جو ہم نے لکھا ہے، اس میں کچھ نئے احکامات ہیں جو کہ ہم بنیادی پیتھان پروگرام میں نہیں آئے ہیں۔ چونکہ ہمارا پروگرام دو مختلف زبانوں کا انضمام ہے، اس لیے ہمیں ماڈیولز، جیسے os، sys اور getopt درآمد کرنا ہوں گے۔

14.6.1 ماڈیول (Module)

ماڈیولر پروگرامنگ ایک سافٹ ویئر ڈیزائن ٹیکنالوجی ہے جو آپ کے کوڈ کو الگ الگ حصوں میں تقسیم کرتی ہے۔ ان حصوں کو ماڈیول (Module) کہا جاتا ہے۔ اس علیحدگی میں ایسا مرکز ہونا چاہئے، جس میں دوسرے ماڈیولز پر کوئی انحصار نہ ہو، یا صرف چند انحصار ہوں۔ دوسرے الفاظ میں : انحصار کو کم کرنا مقصد ہے۔

لیکن ہم پیتھون میں ماڈیول کیسے بناتے ہیں؟ ماڈیولز ایک ایسی فائل کا حوالہ دیتے ہیں جس میں پیتھون بیانات اور تعریفیں ہوں۔ ایک فائل جس میں پیتھون کوڈ ہے، مثال کے طور پر factorial.py کو ماڈیول کہا جاتا ہے اور اس کے ماڈیول کا نام فیکٹوریئل ہوگا۔ ہم بڑے پروگراموں کو چھوٹے انتظام اور منظّم فائلوں میں توڑنے کے لیے ماڈیول استعمال کرتے ہیں۔ مزید برآں، ماڈیولز کوڈ دوبارہ استعمال کرنے کے لئے فراہم کرتے ہیں۔ ہم اپنے سب سے زیادہ استعمال ہونے والے افعال کو ماڈیول میں بیان کر سکتے ہیں اور ان کی تعریفیں مختلف پروگراموں میں نقل کرنے کے بجائے اسے درآمد کر سکتے ہیں۔

مثال :

```
def fact(n):
        f=1
        if n == 0:
        return 0
elif n == 1:
        return 1
        else:
                for i in range(1, n+1):
            f*i  f=
                (f)print
```

Output:

```
>>>fact (5)
120
```

مندرجہ بالا مثال کو factorial.py کا نام دیا گیا ہے۔

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:06 PM

14.6.2 پیتھون میں ماڈیول کیسے درآمد کریں؟ (How to import modules in Python?)

ہم ایک ماڈیول کے اندر تعریفیں دوسرے ماڈیول میں درآمد کر سکتے ہیں۔ ہم ایسا کرنے کے لیے کلیدی لفظ **import** استعمال کرتے ہیں۔ ہمارے پہلے بیان کردہ ماڈیول **factorial** کو درآمد کرنے کے لیے ہم مندرجہ ذیل **python prompt** میں ٹائپ کرتے ہیں۔

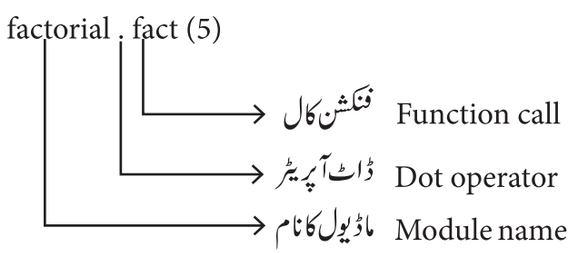
```
>>> import factorial
```

ماڈیول کے نام کا استعمال کرتے ہوئے ہم ماڈیول کے اندر متعین افعال تک رسائی حاصل کر سکتے ہیں۔ ڈاٹ (.) آپریٹر افعال تک رسائی کے لیے استعمال ہوتا ہے۔ ماڈیول سے افعال تک رسائی کے لیے **syntax** یہ ہے۔

<module name> . <function name>

مثال کے طور پر:

```
>>> factorial.fact(5)
120
```



پیتھون کے پاس معیاری (بلٹ ان) ماڈیولز ہیں۔ معیاری ماڈیول اسی طرح درآمد کیے جا سکتے ہیں جس طرح ہم اپنے صارف کے متعین کردہ ماڈیول درآمد کرتے ہیں۔ اب ہم معیاری ماڈیولز دیکھنے جا رہے ہیں جو ہمارے پروگرام **C++** کوڈ کو چلانے کے لیے درکار ہیں۔

14.6.2.1 پیتھان کا **sys** ماڈیول

یہ ماڈیول مترجم کے استعمال کردہ بلٹ ان متغیرات تک رسائی فراہم کرتا ہے۔ **sys** ماڈیول میں متغیر میں سے ایک **argv** ہے۔

sys.argv

sys.argv کمانڈ لائن دلائل (**argument**) کی فہرست ہے جو کہ پیتھون پروگرام کو دی گئی ہے۔ **argv** میں وہ تمام اشیاء شامل ہیں، جو کمانڈ لائن ان پٹ کے ذریعے آتی ہیں۔ یہ بنیادی طور پر ایک فہرست ہے جو پروگرام کے کمانڈ لائن دلائل رکھتی ہے۔

sys.argv استعمال کرنے کے لیے، **import sys** استعمال کیا جانا چاہیے۔ پہلی دلیل، **sys.argv[0]** میں پیتھون پروگرام (مثال **pali.py**) کا نام ہے، اور **sys.argv[1]** پروگرام کو دی گئی اگلی دلیل ہے۔ (یہاں یہ **C++** فائل ہے)۔ جو دلیل **main()** سے گزرے گی۔ مثال کے طور پر :

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:07 PM

main(sys.argv[1])	ان پٹ فائل (C++ فائل) اس کے راستے کے ساتھ argv[1] کا استعمال کرتے ہوئے بطور فہرست (صف) بھیجی جاتی ہے۔ argv[0] میں پیتھون پروگرام شامل ہے، جسے پاس کرنے کی ضرورت نہیں ہے کیونکہ بطور ڈیفالٹ __main__ میں سورس کوڈ کا حوالہ دیتا ہے۔

14.6.2.2 پیتھون کا OS ماڈیول

پیتھون میں OS ماڈیول آپریٹنگ سسٹم پر منحصر فعالیت کو استعمال کرنے کا ایک طریقہ فراہم کرتا ہے۔

OS ماڈیول آپ کو ونڈوز آپریٹنگ سسٹم کے ساتھ انٹرفیس کی اجازت دیتا ہے، جہاں پیتھون چل رہا ہے۔

os.system() : C++ کو چلائیں، مرتب کرنے والی کمانڈ (ایک string میں Unix، C کمانڈ ہے جو C++ کمانڈ کو بھی سپورٹ کرتی ہے). (یہ کمانڈ ونڈو ہے)۔ مثال کے طور پر C++ پروگرام مرتب کرنے کے لیے g++ compilr مرتب کرنے والے کو طلب کرنا چاہیے۔ ایسا کرنے کے لیے درج ذیل کمانڈ استعمال کیا جاتا ہے۔
جہاں ہر دلیل ان پر مشتمل ہے۔

os.system ('g++' + <varaiable_name1> '-<mode>' + <variable_name2>

جس میں

os.system :-	function system() defined in os module
g++ :-	General compiler to compile C++ program under Windows Operating system.
variable_name1:-	Name of the C++ file without extension .cpp in string format
mode :-	To specify input or output mode. Here it is o prefixed with hyphen.
variable_name2:-	Name of the executable file without extension .exe in string format

مثال کے طور پر C++ پروگرام کو مرتب کرنے اور اس پر عمل کرنے کی کمانڈ ذیل میں دی گئی ہے۔

os.system('g++ ' + cpp_file + ' -o ' + exe_file)	g++ compiler compiles the file cpp_file and –o (output) send to exe_file

نوٹ

os.system() میں '+' اس بات کی نشاندہی کرتا ہے کہ تمام string کو ایک string کے طور پر جوڑا گیا ہے. اس کے بعد مذکورہ بالا دلیل کے لیے ہر لفظ کے بعد ایک جگہ دیں۔ مثال کے طور پر 'g++' + cpp_file + ' -o ' + exe_file

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:07 PM

Python getopt module 14.6.2.3.3

python کا getopt ماڈیول آپ کو کمانڈ لائن کے اختیارات اور دلائل کو parse الگ کرنے میں مدد کرتا ہے۔ یہ ماڈیول getopt() طریقہ فراہم کرتا ہے تاکہ کمانڈ لائن، دلیل parsing کو فعال کیا جا سکے۔

getopt.getopt طریقہ

یہ طریقہ کمانڈ لائن آپشنز اور پیرامیٹر لسٹ کو parse کرتا ہے۔ اس طریقے کے لیے syntax درج ذیل ہے۔

<opts>,<args>=getopt.getopt(argv, options, [long_options])

یہاں پیرامیٹرز کی تفصیل ہے۔

- **argv** – یہ تجزیہ کی جانے والی قیمتوں کی دلیل کی فہرست ہے (تقسیم)۔ ہمارے پروگرام میں مکمل کمانڈ بطور لسٹ منظور کی جائے گی۔
- **Options** – یہ آپشن حروف کی ایک سٹرنگ ہے جسے پیتھان کا پروگرام ان پٹ یا آؤٹ پٹ کے طور پر تسلیم کرتا ہے، جس میں آپشنز (جیسے 'i' یا 'o') جس کے بعد بڑا کولون (:) ہوتا ہے۔ یہاں موڈ کو ظاہر کرنے کے لیے کولون : استعمال ہوتا ہے۔
- **long_options** – اس میں اسٹرنگ کی list ہے۔ طویل اختیارات کی دلیل (arguments) کے بعد ایک مساوی نشان ('=') ہونا چاہیے۔ ہمارے پروگرام میں C++ فائل کا نام اور اس کے path کو سٹرنگ کے طور پر 'i' ہوتا ہے۔ i کو بھی ان پٹ فائل کے طور پر ظاہر کرنے کے لیے پاس کیا جائے گا۔

getopt() طریقہ دو عناصر پر مشتمل قیمت لوٹاتا ہے۔ ان قیمتوں میں سے ہر ایک کو الگ الگ دو مختلف فہرستوں (arrays) opts اور args میں محفوظ کیا جاتا ہے۔ args میں اہوتی ہے، اگر کوئی تبصرہ غلط راستے یا موڈ کے ساتھ دیا گیا ہو۔ اگر کوئی error نہیں ہے تو arg ایک خالی فہرست ہوگی۔

مثال کے طور پر python کوڈ، جو C++ کے filep4 کو کمانڈ لائن میں چلانے والا ہے۔ اس میں getopt() طریقہ مندرجہ ذیل کی طرح ہوگا۔

opts, args = getopt.getopt (argv, "i:",['ifile='])

where **opts** contains	[('-i', 'c:\\pyprg\\p4')]
-i :-	**option** nothing but **mode should be followed by :**
'c:\\pyprg\\p4'	**value** nothing but the **absolute path of C++ file.**

ہماری مثالوں میں چونکہ پوری کمانڈ لائن کمانڈ parse کی گئی ہے اور کوئی بچی ہوئی دلیل (argument) نہیں ہے، second args empty[] argument خالی ہوں گی۔ اگر print() کمانڈ کا استعمال کرتے ہوئے args دکھائے جاتے ہیں تو یہ آؤٹ پٹ کو [] کے طور پر دکھاتا ہے۔

```
>>>print(args)
[]
```


12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:07 PM

نوٹ

آپ python کے معیاری ماڈیولز کی مکمل فہرست اور وہ کس چیز کے لیے ہیں چیک کر سکتے ہیں۔ یہ فائلیں Lib ڈائریکٹری میں اس جگہ کے اندر ہیں جہاں python نصب ہے۔

C++ کوڈ کو لپیٹنے کے لیے کچھ اور کمانڈ (some more command for wrapping C++ code)

```
if __name__=='__main__':
    main(sys.argv[1:])
```

python میں __name__ (ایک خاص متغیر)

چونکہ python میں کوئی main() فنکشن نہیں ہے، جب پیتھون پروگرام چلانے کی کمانڈ مترجم (interpreter) کو دی جاتی ہے تو، کوڈ جو کہ سطح 0 انڈینٹیشن پر ہوتا ہے۔ (پروگرام کی ٹاپ لازمی لائن) پر عملدرآمد کرنا ہوتا ہے۔ تاہم، ایسا کرنے سے پہلے، ترجمان کچھ خاص متغیرات کی وضاحت کرے گا۔ __name__ ایک ایسا خاص متغیر ہے جو بطور ڈیفالٹ فائل کا نام محفوظ کرتا ہے۔ اگر source فائل کو مرکزی پروگرام کے طور پر چلایا جاتا ہے تو، مترجم __name__ متغیر کو سورس فائل کا نام اس کی قیمت کے طور پر متعین کرتا ہے۔ ڈیفالٹ کے ذریعہ "__main__" میں سورس فائل کا نام ہے (پیتھان پروگرام)

__name__ ایک بلٹ ان متغیر ہے، جو موجودہ ماڈیول کے نام کا اندازہ کرتا ہے۔

اس طرح یہ چیک کرنے کے لیے استعمال کیا جا سکتا ہے کہ موجودہ سکرپٹ از خود چل رہا ہے یا نہیں۔

مثال کے طور پر درج ذیل پر غور کریں۔

```
if __name__ == '__main__':
    main (sys.argv[1:])
```

اگر کمانڈ لائن پیتھان کا پروگرام خود پہلے چلانے والا ہے، تو __main__ میں اس پیتھان پروگرام کا __name__ اور python کا خاص متغیر نام بھی شامل ہے۔ python پروگرام کے نام پر مشتمل ہے۔ اگر حالت true ہے تو یہ مین کو کال کرتی ہے جسے C++ فائل کے ساتھ بطور دلیل (argument) منظور کیا جاتا ہے۔

نوٹ

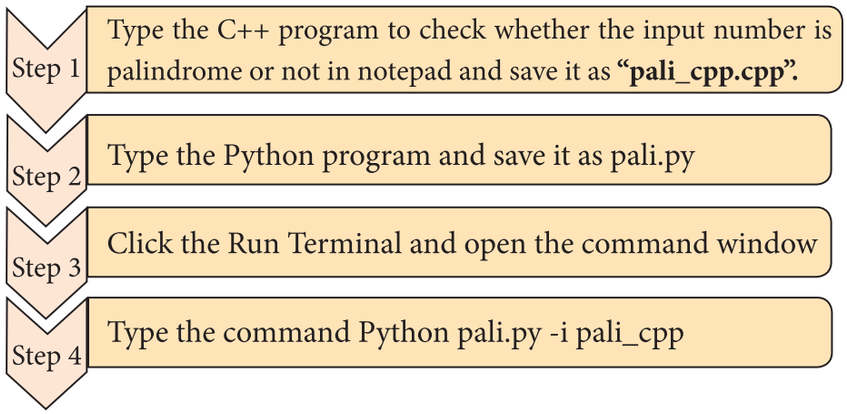
sys.argv[1:] - اسکرپٹ کے نام (فائل کا نام) کے بعد ہر چیز حاصل کریں۔

یاد رکھیں "string slicing" جس کا آپ نے باب 8 میں مطالعہ کیا ہے۔

14.7 Python کا پروگرام کنٹرول اسٹیٹمنٹ استعمال کرتے ہوئے C++ پروگرام پر عمل درآمد

اب آیئے ایک C++ کوڈنگ کو پڑھنے اور اس کے نتائج پر عمل کرنے کے لیے ایک python کا پروگرام لکھیں۔ دیے گئے نمبر کو چیک کرنے کے لیے C++ پروگرام پر عمل کرنے کے لیے اقدامات نیچے دیے گئے ہیں۔

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:08 PM



مثال:-14.7.1 while loop - استعمال کرتے ہوئے،کوئی بھی نمبر داخل کرنے کے لیے C++ پروگرام لکھیں اور چیک کریں کہ آیا نمبر پیلینڈروم (حروف کے الٹنے سے لفظ کا نہ بدلنا) یا نہیں۔

```
/*. 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
```


12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:08 PM

```
#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
```


12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:08 PM

Python code	How does it works
import sys, os, getopt	include sys , os and getopt modules to use the required function
def main(argv):	Function main() is defined and 'argv' contains the 'input mode and the c++ program file' in the form of list i.e ['-i', 'c:\ pyprg \pali_cpp']
opts, args = getopt.getopt(argv, "i:")	getopt() splits the command as option and argument. 'opts' contains [('-i', 'c:\pyprg\pali_cpp')]. Since no error 'args' shows []
for o, a in opts:	'o' contains the mode and 'a' contains the path of c++ program i.e 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)	Calls the function run() passed along with the c++ program
def run(a):	Definition of run() function begins here
inp_file=a+'.cpp'	Variable 'inp_file' contains the joined c++ program name and .cpp print(inp_file) shows c:\pyprg\pali_cpp.cpp
exe_file=a+'.exe'	Variable 'exe_file' contains the joined c++ program name and .exe print(exe_file) shows c:\pyprg\pali_cpp.exe
os.system('g++ ' + inp_file + ' -o ' + exe_file)	g++ compiler compiles the c++ program in inp_file and store the executable file in exe_file
os.system(exe_file)	Executes the exe file
if __name__=='__main__':	__name__ stores name of the python program __ main__ also stores the name of the python program.
main(sys.argv[1:])	If 'name' and 'main' are equal then main() is called and passed with the command line argument omitting the python program name argv[1:] contains –i c:\pyprg\pali_cpp

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:08 PM

پیتھان سکرپٹ (پروگرام) بنیادی طور پر C++ فائل کے ساتھ موڈ کی قسم کے ساتھ 'i'/'o' پڑھنے کے لیے استعمال ہوتا ہے۔'getopt()' کمانڈ لائن کی ہر قیمت کو پارس (شق) کرتی ہے اور اختیارات (اقدار) کو list کے طور پر 'opt' کرتا ہے اور چونکہ کوئی error نہیں ہے، اس لئے 'args' empty list[] نہیں بناتی۔ 'for loop' استعمال کرتے ہوئے list کے tupule کو کھولا جاتا ہے۔'o' موڈ کو محفوظ کرتا ہے اور 'a' نام کو C++ فائل کے راستے کے ساتھ محفوظ کرتا ہے۔

متغیر 'inp_file'، C++ فائل کو اس کی توسیع exe. کے ساتھ محفوظ کرتا ہے اور 'exe_file' توسیع کے ساتھ قابل عمل فائل کو محفوظ کرتا ہے۔ اس پروگرام میں 'usd'+' فائل کے نام کو ایکسٹینشن کے ساتھ جوڑنے میں مدد کرتا ہے۔ 'os.system()' کے ساتھ 'g++' فائل inp_file مرتب کرتا ہے۔ موڈ 'o' قابل عمل فائل کو 'exe_file' کو بھیجتا ہے۔

'__name__' متغیر پروگرام کو python اسکرپٹ کے آغاز سے شروع کرنے کی ہدایت کرتا ہے۔ (صفر ویں لائن) "main()" کی وضاحت پارسنگ اور run() کو کال کرتی ہے۔ "run()" , "g++" کو ابھارتا ہے۔ مرتب کرنے والے کی درخواست کرتا ہے اور exe فائل بناتا ہے۔ "os" ماڈیول کا نظام () عمل کرتا ہے۔ exe. فائل اور مطلوبہ آؤٹ پٹ آؤٹ پٹ سکرین پر ظاہر ہوگا۔ فائل کی توسیع python اسکرپٹ کے ذریعہ شامل کی گئی ہے۔ لہذا C پروگراموں پر عملدرآمد ممکن ہے۔

14.8 python C++ میں غلطیوں کو کس طرح سنبھال رہا ہے (How python is handling the errors in C++)

python نہ صرف کامیاب C++ پروگرام پر عمل درآمد کرتا ہے، یہ تالیف کے دوران اگر C++ بیان میں کوئی error بھی ظاہر کرنے میں مدد کرتا ہے۔ مثال کے طور پر درج ذیل C++ پروگرام میں ایک error ہے۔ آیئے دیکھتے ہیں کہ جب آپ python کے ذریعے مرتب کرتے ہیں تو کیا ہوتا ہے۔

مثال 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)
```


12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:09 PM

```
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:7:19: 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.
```

نوٹ

مذکورہ پروگرام میں,python ، C++ میں error ظاہر کرنے میں مدد کرتا ہے۔خرابی اس کے لائن نمبر کے ساتھ ظاہر ہوتی ہے، لائن نمبر python اسکرپٹ سے شروع ہوتا ہے۔

یاد رکھنے کے نکات: Points to remember:

- C++ ایک مرتب compiler پر مبنی زبان ہے جبکہ python ایک مترجم interpreter پر مبنی زبان ہے۔
- C++ مستحکم طور پر مرتب کیا گیا ہے، جب کہ پیتھان کی تشریح متحرک طور پر کی گئی ہے۔
- ایک مستحکم ٹائپ شدہ زبان جیسے C++، پروگرامر کو کمپیوٹر کو واضح طور پر بتانے کی ضرورت ہوتی ہے کہ ہر ڈیٹا کی قیمت کے لئے کون سا "datatype" استعمال کرے گی۔
- python جیسی متحرک ٹائپ شدہ زبان، ڈیٹا کے لیے ڈیٹا ٹائپ کو واضح طور پر دینے کی ضرورت نہیں ہے۔ python، قیمت کی قسم کی بنیاد پر متغیر کو جوڑتا ہے۔
- اسکرپٹ لینگویج ایک پروگرامنگ لینگویج ہے جو دیگر پروگرامنگ زبانوں کے ساتھ مربوط کرنے اور بات چیت کرنے کے لیے بنائی گئی ہے۔
- MinGW سے مراد رن ٹائم ہیڈر فائلوں کا ایک مجموعہ ہے، جو ونڈوز آپریٹنگ سسٹم پر چلائے جانے والے C، C++ اور FORTRAN کے کوڈ کو مرتب اور جوڑنے میں استعمال ہوتا ہے

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:09 PM

Points to remember: یادرکھنے کے نکات:

- dot(.) آپریٹر درآمد شدہ ماڈیول کے افعال تک رسائی کے لیے استعمال ہوتا ہے۔
- sys ماڈیول مترجم کے ذریعہ استعمال ہونے والے کچھ متغیرات تک اور ان افعال تک رسائی فراہم کرتا ہے جو مترجم کے ساتھ تعامل کرتے ہیں۔
- python میں OS ماڈیول آپریٹنگ سسٹم پر مخصر فعالیت کو استعمال کرنے کا ایک طریقہ فراہم کرتا ہے۔
- پیتھان میں getopt ماڈیول آپ کو کمانڈ لائن کے اختیارات اور دلائل کو parse (شق) کرنے میں مدد کرتا ہے۔



Hands on Experience عملی مشق

1۔ درج ذیل تفصیلات کے ساتھ student نامی کلاس بنانے کے لیے ایک C++ پروگرام لکھیں۔

Protected member

Rno integer

Public members

void Readno(int); to accept roll number and assign to Rno

void Writeno(); To display Rno.

The class Test is derived Publically from the Student class contains the following details

Protected member

Mark1 float

Mark2 float

Public members

void Readmark(float, float); To accept mark1 and mark2

void Writemark(); To display the marks

Create a class called Sports with the following detail

Protected members

score integer

Public members

void Readscore(int); To accept the score

void Writescore(); To display the score

The class Result is derived Publically from Test and Sports class contains the following details

Private member

Total float

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:09 PM

Public member

void display() assign the sum of mark1, mark2, score in total.

invokeWriteno(), Writemark() and Writescore(). Display the total also.

C++ پروگرام کو **hybrid** نامی فائل میں محفوظ کریں۔ **hybrid.cpp** پر عمل کرنے کے لیے ایک پیتھان کا پروگرام لکھیں۔

2۔ ایک میٹرکس کے حدود کے عناصر کو پرنٹ کرنے کے لیے ایک C++ پروگرام لکھیں اور فائل کو **border.cpp** کا نام دیں۔ **Border.cpp** پر عمل کرنے کے لیے ایک پیتھان کا پروگرام لکھیں۔

Evaluation محاسبہ

حصہ - I

صحیح جواب منتخب کیجئے۔ (1 مارکس)

1۔ مندرجہ ذیل میں سے کون سی سکرپٹ زبان نہیں ہے؟

(A) JavaScript (B) PHP (C) Perl (D) HTML

2۔ ایک از گر پروگرام میں C++ پروگرام درآمد کرنے کو کہتے ہیں۔

(A) wrapping (B) Downloading

(C) Interconnecting (D) Parsing

3۔ API کی توسیع یہ ہے۔

(A) Application Programming Interpreter

(B) Application Programming Interface

(C) Application Performing Interface

(D) Application Programming Interlink

4۔ Python اور C++ کو انٹرفیس کرنے کا ایک فریم ورک یہ ہے۔

(A) Ctypes (B) SWIG (C) Cython (D) Boost

5۔ آپ کے کوڈ کو الگ الگ حصوں میں تقسیم کرنے کے لیے مندرجہ ذیل میں سے کون سا سافٹ ویئر ڈیزائن ٹکنیک ہے؟

(A) Object oriented Programming

(B) Modular programming

(C) Low Level Programming

(D) Procedure oriented Programming

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:09 PM

6۔ وہ ماڈیول جو آپ کو ونڈوز آپریٹنگ سسٹم کے ساتھ انٹرفیس کی اجازت دیتا ہے۔

(A) OS module (B) sys module (c) csv module (d) getopt module

7۔ اگر اسٹرنگز کو تقسیم کرنے میں کوئی error نہ ہو تو ()getopt ایک خالی صف واپس کرے گا۔

(A) argv variable (B) opt variable (c)args variable (d) ifile variable

8۔ درج ذیل ٹکڑے (snippet) میں فنکشن کال اسٹیٹمنٹ کی شناخت کریں۔

```
if __name__ =='__main__':
main(sys.argv[1:])
```

(A) main(sys.argv[1:]) (B) __name__ (C) __main__ (D) argv

9۔ درج ذیل میں سے کون سا ٹیکسٹ، نمبرز، امیجز اور سائنسی ڈیٹا پراسیسنگ کے لیے استعمال کیا جا سکتا ہے؟

(A) HTML (B) C (C) C++ (D) PYTHON

10۔ __name__ میں کیا شامل ہے؟

(A) c++ filename (B) main() name

(C) python filename (D) os module name

حصہ - II

درج ذیل سوالات کے جوابات لکھئے۔ (2 مارکس)

1۔ اسکرپٹ لینگویج اور دیگر پروگرامنگ لینگویج کے درمیان نظریاتی فرق کیا ہے؟

2۔ مرتب (compiler) اور مترجم (interpreter) میں فرق کریں۔

3۔ (i) SWIG اور (ii) MinGW کی توسیع لکھیں۔

4۔ ماڈیولز کا استعمال کیا ہے ؟

5۔ cd command کا استعمال کیا ہے؟ ایک مثال دیں۔

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:09 PM

حصہ - III

درج ذیل سوالات کے جوابات لکھئے۔ (3 مارکس)

1۔ PYTHON اور C++ میں فرق کریں۔

2۔ اسکرپٹ لینگویج کے اطلاقات کیا ہیں؟

3۔ MinGW کیا ہے ؟ اس کا کیا فائدہ ہے ؟

4۔ مندرجہ ذیل welcome.display() کے لیے module، operator، definition نام کی شناخت کریں۔

5۔ sys.argv کیا ہے؟ اس میں کیا ہوتا ہے؟

حصہ-IV

درج ذیل سوالات کے جوابات لکھئے۔ (5 مارکس)

1۔ python کی کوئی 5 خصوصیات لکھیں۔

2۔ درج ذیل کمانڈ کے ہر لفظ کی وضاحت کریں۔

3۔ Python.Explain میں sys، os، getopt ماڈیول کا مقصد کیا ہے؟

4۔ getopt() کے لیے syntax لکھیں اور اس کے دلائل (arguments) اور واپسی کی قیمتوں کی وضاحت کریں۔

5۔ درج ذیل C++ کوڈنگ پر عمل کرنے کے لیے python کا پروگرام لکھیں۔

```
#include <iostream>
using namespace std;
int main()
{ cout<<“WELCOME”;
return(0);
}
```

The above C++ program is saved in a file welcome.cpp

REFERENCES حوالہ جات

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*

12 CS UNIT 14 CDR Set
Monday, April 17, 2023 10:01:10 PM

CHAPTER 15 باب 15

DATA MANIPULATION THROUGH SQL SQL کے ذریعے ڈیٹا کو برتنا

V

سیکھنے کے مقاصد

PNEX0

اس باب کی تکمیل کے بعد، طالب علم پائیتھن کا سکرپٹ لکھ سکے گا۔

- ایک ٹیبل بنائیں اور ڈیٹا بیس میں نئی صفیں شامل کرنا۔
- ٹیبل میں ریکارڈ اپ ڈیٹ اور ڈیلیٹ کرنا۔
- جدول سے استفسار (Query) کرنا۔
- ایک CSV فائل میں سوال لکھنا۔

15.1 تعارف

ڈیٹا بیس ڈیٹا کا منظم مجموعہ ہے۔ اصطلاح "database" دونوں ڈیٹا کو خود یا ڈیٹا بیس مینجمنٹ سسٹم کا حوالہ دے سکتی ہے۔ ڈیٹا بیس مینجمنٹ سسٹم صارفین اور ڈیٹا بیس کے درمیان تعامل کے لیے ایک سافٹ ویئر ایپلی کیشن ہے۔ صارفین کو انسانی صارفین ہونے کی ضرورت نہیں ہے۔ وہ دوسرے پروگرام اور ایپلی کیشنز بھی ہو سکتے ہیں ۔ہم سیکھیں گے کہ کس طرح پائیتھن پروگرام SQL ڈیٹا بیس کے صارف کے طور پر بات چیت کر سکتا ہے۔

15.2 SQLite

SQLite ایک سادہ رشتہ دار ڈیٹا بیس سسٹم ہے، جو اپنے ڈیٹا کو باقاعدہ ڈیٹا فائلوں میں یا کمپیوٹر کی اندرونی میموری میں محفوظ کرتا ہے۔ اسے ایک الگ ڈیٹا بیس سرور پروگرام جیسے MySQL یا ORACLE کے استعمال کے بجائے ایپلی کیشنز میں سرایت کرنے کے لیے ڈیزائن کیا گیا ہے۔ SQLite تیز، سخت جانچ کردہ اور لچکدار ہے، جس سے کام کرنا آسان ہوتا ہے۔ پائیتھن SQLite کے لیے ایک مقامی لائبریری ہے۔

SQLite استعمال کرنے کے لیے :

مرحلہ 1 sqlite3 کو درآمد کرنا

مرحلہ 2 connect() طریقہ استعمال کرتے ہوئے کنکشن بنائیں اور ڈیٹا بیس فائل کا نام پاس کریں۔

مرحلہ 3 کرسر آبجیکٹ سیٹ کریں۔ cursor = connection.cursor()

- مرحلہ 2 میں ڈیٹا بیس سے جڑنے کا مطلب ہے، ڈیٹا بیس کے نام تک رسائی حاصل کرنا۔ اگر ڈیٹا بیس پہلے سے موجود ہے تو کنکشن اسی طرح کھل جائے گا۔ بصورت دیگر، پائیتھن مخصوص نام کے ساتھ ایک نئی ڈیٹا بیس فائل کھولے گا۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:02 AM

- مرحلہ 3 میں کرسر : ایک کنٹرول ساخت ہے جو ڈیٹا بیس کے ریکارڈ کو عبور کرنے اور لانے کے لیے استعمال ہوتا ہے۔
- پائیتھن کے ساتھ کام کرنے میں کرسر کا اہم کردار ہے۔ تمام احکامات صرف کرسر آبجیکٹ کا استعمال کرتے ہوئے عمل میں لائے جائیں گے۔

ڈیٹا بیس میں ٹیبل بنانے کے لیے ایک شے بنائیں اور اس میں SQL کمانڈ لکھیں۔

مثال: **Example:-** sql_comm = "SQL statement"

کمانڈ پر عمل کرنے کے لیے کرسر طریقہ استعمال کریں اور بطور پیرامیٹر مطلوبہ sql کمانڈ پاس کریں۔ بہت ساری کمانڈز sql_comm میں محفوظ کی جاسکتی ہیں اور ایک کے بعد ایک عمل درآمد کی جاسکتی ہیں۔ ریکارڈ کی قیمتوں میں کی جانے والی کسی بھی تبدیلی کو "Table connection" کو بند کرنے سے پہلے تعریف "comment" کے ذریعے محفوظ کیا جانا چاہیے۔

SQLite کا استعمال کرتے ہوئے ڈیٹا بیس بنانا
15.3 Creating a Database using SQLite

مندرجہ ذیل مثال 15.3 بتاتی ہے کہ کس طرح ایک ڈیٹا بیس سے Python sqlite3 کے ذریعے کنکشن بنایا جائے۔

```
# 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()
```

مندرجہ بالا مثال میں "Academy" کے نام سے ایک ڈیٹا بیس بنایا جائے گا۔ یہ sql کمانڈ کی طرح ہے۔
"CREATE DATABASE Academy;" to SQL server."sqlite3.connect('Academy.db')" بعض پروگرام میں دوبارہ استعمال ہوتا ہے۔ "connect" کمانڈ، پہلے ہی سے تیار شدہ database کو کھولتا ہے۔

15.3.1 ٹیبل بنانا (Creating a Table)

خالی ڈیٹا بیس بنانے کے بعد، آپ شاید اس ڈیٹا بیس میں ایک یا زیادہ جدولوں کو شامل کریں گے۔ ڈیٹا بیس "Academy" میں ٹیبل "student" بنانے کے لیے sql syntax اس طرح لگتا ہے۔

```
CREATE TABLE Student (
RollnoINTEGER, SnameVARCHAR(20), GradeCHAR(1), gender CHAR(1),
Average float(5.2), birth_date DATE, PRIMARY KEY (Rollno) );
```


12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:02 AM

یہ طریقہ ہے، کوئی اسے sql کمانڈ شیل پر کرسکتا ہے۔ ہم چاہتے ہیں کہ اسے براہ راست پائیتھن سے کریں۔"SQL"، یا SQLite کو کمانڈ بھیجنے کے قابل ہونے کے لیے، ہمیں ایک کرسر آبجیکٹ چاہئے۔ عام طور پر cursor ایک SQL اور ڈیٹا بیس میں ایک cursor ایک ساخت ہوتی ہے، جو ڈیٹا بیس میں ریکارڈ کو عبور کرتا ہے۔ لہذا یہ نتائج حاصل کرنے کے لیے استعمال ہوتا ہے۔

نوٹ

cursor تمام SQL کمانڈز کو انجام دینے کے لیے استعمال کیا جاتا ہے۔

کرسر آبجیکٹ کو cursor() کنکشن کے طریقے کو کال کر کے بنایا گیا ہے۔ کرسر رزلٹ سیٹ سے ریکارڈ کو عبور کرنے کے لیے استعمال ہوتا ہے۔ آپ پائیتھن میں" triple quoted string کے ساتھ SQL کمانڈ کی وضاحت کر سکتے ہیں۔ "" کے پیچھے کی وجہ یہ ہے کہ بعض اوقات ٹیبل کی قیمتوں میں سنگل ' یا ڈبل " کوٹس ہو سکتے ہیں۔

مثال 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);"""
```

مندرجہ بالا مثال میں Emp_no فیلڈ بطور"INTEGER PRIMARY KEY" ۔ ایک کالم جس پر اس طرح کا لیبل لگا ہوا ہے خود بخود SQLite3 میں خود بخود بڑھ جائے گا۔ اسے دوسرے لفظوں میں ڈالنے کے لیے: اگر کسی ٹیبل کے کالم کو INTEGER PRIMARY KEY قرار دیا جاتا ہے، تو جب بھی اس کالم کے لیے ایک NULL ان پٹ کے طور پر استعمال کیا جائے گا، NULL خود بخود ایک سالم عدد میں تبدیل ہو جائے گا، جو اس کالم میں اب تک استعمال ہونے والی سب سے زیادہ قیمت سے ایک بڑا ہوگا۔ اگر ٹیبل خالی ہے تو، قیمت 1 استعمال کی جائے گی۔

15.3.2 ریکارڈ شامل کرنا (Adding records)

جدول بھرتی کرنے کے لئے"INSERT" کمانڈ SQLite کو منتقل کی جاتی ہے۔"execute" طریقہ کچھ کارروائی کرنے کے لیے SQL کمانڈ پر عمل کرتا ہے۔ مندرجہ ذیل مثال 15.3.2 ایک مکمل کام کرنے والی مثال ہے۔ پروگرام چلانے کے لیے آپ کو یا تو فائل Academy.db کو ہٹانا پڑے گا، یا SQL کمانڈ میں "DROP TABLE" لائن کو uncomment کریں۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:02 AM

Example مثال

```
import sqlite3
connection = sqlite3.connect ("Academy.db")
cursor = connection.cursor()
# delete
cursor.execute ("""DROP TABLE Student;""")
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")
```

OUTPUT

```
    STUDENT TABLE CREATED
```

یقیناً، زیادہ تر معاملات میں، آپ لفظی طور پر SQL ٹیبل میں ڈیٹا داخل نہیں کریں گے۔ آپ کے پاس کچھ پائیتھن ڈیٹا ٹائپ کے اندر بہت زیادہ ڈیٹا ہوگا جیسے ایک ڈکشنری یا ایک لسٹ، جسے داخل کرنے والے **statement** کے ان پٹ کے طور پر استعمال کرنا پڑتا ہے۔

مندرجہ ذیل ورکنگ مثال، فرض کرتی ہے کہ آپ کے پاس پہلے سے موجود ڈیٹا بیس **Academy.db** اور جدول **student** ہے۔ ہمارے پاس افراد کے ڈیٹا کے ساتھ ایک فہرست ہے، جسے **INSERT statement** میں استعمال کی جائے گی:

مندرجہ بالا پروگرام میں {gr} قیمت حاصل کرنے کے لیے ایک **place holder** (متغیر) ہے۔ **format_str.format()** ایک فنکشن ہے، جو مطلوبہ ڈیٹا ٹائپ میں ویلیو کو فارمیٹ کرنے کے لیے استعمال ہوتا ہے۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:03 AM

Example مثال

```
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 ")
```

OUTPUT

```
RECORDS ADDED TO STUDENT TABLE
```

15.4 SQL Query Using Python پائیتھن کا استعمال کرتے ہوئے SQL استفسار

اب وقت آ گیا ہے کہ آخر میں ہمارے "student" ٹیبل سے استفسار (query) کریں۔ ریکارڈ سے ڈیٹا لانا اتنا ہی آسان ہے جتنا ان کو داخل کرنا۔ عمل کرنے کا طریقہ جدول سے تمام ڈیٹا حاصل کرنے کے لیے SQL کمانڈ استعمال کرتا ہے۔

15.4.1 SELECT Query

"SELECT" SQL میں سب سے زیادہ استعمال ہونے والا بیان ہے ۔SQL میں SELECT بیان ایک ڈیٹا بیس میں ٹیبل سے ڈیٹا کو بازیافت کرنے کے لیے استعمال کیا جاتا ہے۔اس بیان کو استعمال کرنے کے لیے syntax یہ ہے۔"Select * from table_name" ہے اور تمام ٹیبل ڈیٹا کو ایک شئے کی شکل میں list of lists کی طرح حاصل کیا جاسکتا ہے۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:03 AM

اگر آپ اس پروگرام کو "sql_Academy_query.py" کی طرح محفوظ کرتے ہیں تو، آپ کو اصل ڈیٹا کی بنیاد پر درج ذیل نتیجہ ملے گا۔

واضح رہے کہ جو ڈیٹا بیس فائل بنائی جائے گی وہ اسی فولڈر میں ہوگی جیسا کہ پائیتھن کی فائل۔ اگر ہم فائل کا path تبدیل کرنا چاہتے ہیں تو فائل کھولتے وقت path تبدیل کریں۔

Example **مثال 15.4.1 -1**

```
#save the file as “sql_Academy_query.py”
# cursor object
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() کا استعمال کرتے ہوئے ریکارڈ دکھانا

fetchall() طریقہ database کی جدول سے تمام صفوں کو لاتا ہے۔

Example **مثال 15.4.1.1 - 2**

```
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)
OUTPUT
    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')
```


12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:03 AM

نوٹ

cursor.fetchall() طریقہ ڈیٹا بیس ٹیبل سے تمام صفوں کو لاتا ہے۔

cursor.fetchone() طریقہ سوال کے نتائج کی اگلی صف کو واپس کرتا ہے یا کوئی نہیں اگر کوئی صف باقی نہ ہو۔

cursor.fetchmany() طریقہ جو رزلٹ سیٹ کی اگلی تعداد کے صف (n) کو لوٹاتا ہے۔

15.4.1.2-1 fetchone() کا استعمال کرتے ہوئے تمام ریکارڈ دکھانا

while لوپ اور fetchone() طریقہ استعمال کرتے ہوئے ہم ٹیبل سے تمام ریکارڈ سپلے کر سکتے ہیں۔

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("\nfetch one:")
res = cursor.fetchone()
print(res)
```

OUTPUT

```
fetch one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
```

15.4.1.3 fetchone() کا استعمال کرتے ہوئے تمام ریکارڈ دکھانا

while لوپ اور fetchone() طریقہ استعمال کرتے ہوئے ہم ٹیبل سے تمام ریکارڈ سپلے کر سکتے ہیں۔

Example **مثال 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()
```

OUTPUT

```
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')
```


12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:03 AM

15.4.1.4 fetchmany() استعمال کرتے ہوئے دکھانا

ریکارڈ کی مخصوص تعداد دکھانے کے لئے fetchmany() کا استعمال کیا جاتا ہے۔ یہ طریقہ رزلٹ سیٹ کی اگلی تعداد صفوں (n) کو لوٹاتا ہے۔

مثال 15.4.1.4-1 fetchmany() استعمال کرتے ہوئے tuples میں موجود مواد کو دکھانے کے لئے پروگرام

```
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)
```

OUTPUT

```
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')]
```

مثال 15.4.1.4-2 لوپ کا استعمال کئے بغیر نئی لائن کے tuples میں موجود مواد کو دکھانے کے لئے پروگرام

```
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")
```

OUTPUT

```
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')
```

نوٹ

علامت کا استعمال تمام عناصر کی فہرست کو space کے ساتھ ایک لائن میں پرنٹ کرنے کے لیے کیا جاتا ہے۔ تمام عناصر کو نئی لائنوں میں پرنٹ کرنے یا space سے الگ کرنے کے لیے بالترتیب sep= "\n" or sep= "," استعمال کریں۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:04 AM

15.4.2 SQL میں شقیں (Clauses in SQL)

SQL مختلف شقیں فراہم کرتا ہے، جو SELECT بیانات میں استعمال کی جاسکتی ہیں۔ یہ شقیں پائیتھن کے اسکرپٹ کے ذریعے کہی جاسکتی ہیں۔ تقریباً تمام شقیں SQLite کے ساتھ کام کریں گی۔ مندرجہ ذیل اکثر استعمال شدہ شقوں پر یہاں بحث کی جاتی ہے۔

- DISTINCT
- WHERE
- GROUP BY
- ORDER BY.
- HAVING

15.4.2.1 SQL DISTINCT CLAUSE

مخصوص شق مددگار رہوتی ہے جب کسی مخصوص کالم/ٹیبل میں موجود ڈپلیکیٹ قیمتوں سے بچنے کی ضرورت ہو۔ جب ہم الگ الگ کلیدی الفاظ استعمال کرتے ہیں تو صرف منفرد قیمتیں حاصل کی جاتی ہیں۔ اس مثال میں ہم طالب علموں کے "student table" سے حاصل کردہ مختلف درجات کو ظاہر کرنے جارہے ہیں۔

مثال 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)
```

OUTPUT

```
[('B',), ('A',), ('C',), ('D',)]
```

مذکورہ بالا مثال میں کلیدی الفاظ "distinct" کے بغیر 4 کے بجائے 7 ریکارڈ دکھاتا ہے، کیونکہ اصل ٹیبل میں اصل میں 7 ریکارڈ ہیں اور کچھ ڈپلیکیٹ اقدار کے ساتھ ہیں۔

15.4.2.2 SQL WHERE CLAUSE

WHERE شق صرف ان ریکارڈوں کو نکالنے کے لیے استعمال کی جاتی ہے جو ایک مخصوص شرط کو پورا کرتے ہیں۔ اس مثال میں ہم "student table" سے مرد طلباء کے حاصل کردہ مختلف مارکس کو ظاہر کرنے جارہے ہیں۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:04 AM

```
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")
```

OUTPUT

```
('B',)
('A',)
('C',)
('D',)
```

SQL Group By Clause 15.4.2.3

SELECT بیان کو GROUP BY شق کے ساتھ استعمال کیا جاسکتا ہے۔ GROUP BY شق گروپوں کو **summary rows** میں ریکارڈ کرتا ہے۔ یہ ہر گروپ کے لیے ایک ریکارڈ لوٹاتا ہے۔ یہ اکثر مجموعی افعال (AVG،SUM،MIN،MAX،COUNT) کے ساتھ استعمال کیا جاتا ہے، تاکہ ایک یا زیادہ کالموں کے ذریعہ رزلٹ سیٹ کو گروپ کیا جاسکے۔ مندرجہ ذیل مثال طلباء کی جدول سے مرد اور خواتین کی تعداد شمار کر کے نتیجہ ظاہر کرتی ہے۔

Example **مثال 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")
```

OUTPUT

```
('F', 2)
('M', 5)
```

SQL ORDER BY Clause 15.4.2.4

ORDER BY Clause کو SELECT بیان کے ساتھ استعمال کیا جاسکتا ہے تاکہ مخصوص فیلڈز کے ڈیٹا کو ترتیب سے ترتیب دیا جاسکے۔ اس کا استعمال رزلٹ سیٹ کو چڑھتی ہوئی یا اترتی ہوئے ترتیب میں کرنے کے لیے کیا جاتا ہے۔ اس مثال میں طالب علموں کے نام اور **Rollno** کو حروف تہجی کے ناموں کے ساتھ دکھایا گیا ہے۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:04 AM

Example

مثال 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")
```

OUTPUT

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(6, 'PRIYA')
(4, 'SAJINI')
(7, 'TARUN')
(5, 'VARUN')
```

SQL HAVING Clause 15.4.2.5

Having clause کا استعمال گروپ کے افعال پر مبنی ڈیٹا کو فلٹر کرنے کے لیے کیا جاتا ہے۔ یہ WHERE حالت کی طرح ہے، لیکن صرف (گروپ افعال) group functions کے ساتھ استعمال کیا جا سکتا ہے۔ گروپ افعال WHERE clause میں استعمال نہیں ہو سکتے، لیکن HAVING clause میں استعمال ہو سکتے ہیں۔

مثال 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)
```

OUTPUT

```
['gender', 'COUNT(GENDER)']
[('M', 5)]
```

15.5 The SQL AND, OR and NOT Operators

WHERE شق کو AND، OR اور NOT آپریٹرز کے ساتھ ملایا جا سکتا ہے۔ AND اور OR آپریٹرز ایک سے زیادہ شرائط پر مبنی ریکارڈ کو فلٹر کرنے کے لیے استعمال ہوتے ہیں۔ اس مثال میں آپ ان طلباء کی تفصیلات ظاہر کرنے جا رہے ہیں، جنہوں نے 'A' سے "student table" یا 'B' کے علاوہ دوسرے نمبر حاصل کیے ہیں۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:04 AM

WHERE WITH NOT operator کے لئے مثال

مثال 15.5-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student where grade<>'A' and Grade<>'B'")
result = cursor.fetchall()
print(*result,sep="\n")
```

OUTPUT

```
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

WHERE WITH AND operator کے لئے مثال

اس مثال میں ہم طالب علموں کے **name**، **roll no** اور **average** ظاہر کرنے جا رہے ہیں جنہوں نے اوسط **80** سے **90** فیصد کے درمیان اسکور کیا ہے (دونوں حدیں شامل ہیں)۔

مثال 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")
```

OUTPUT

```
(1, 'Akshay', 87.8)
(5, 'VARUN', 80.6)
```

WHERE WITH OR operator کے لئے مثال

اس مثال میں ہم طالب علموں کے **name**، **roll no** اور **average** ظاہر کرنے جا رہے ہیں جنہوں نے **60** سے **70** فیصد کے درمیان اوسط حاصل نہیں کیا ہے۔

100
95
75
25
5
0

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:04 AM

مثال 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")
```

OUTPUT

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(4, 'SAJINI')
(5, 'VARUN')
(6, 'PRIYA')
```

تاریخ والے کالم سے استفسار کرنا
15.6 Querying A Date Column

اس مثال میں ہم سال 2001 میں پیدا ہونے والے طلباء کا نام اور مارکس ظاہر کرنے جا رہے ہیں۔

مثال 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' ANDBirth_date<='2001-12-01')")
result = cursor.fetchall()
print(*result,sep="\n")
```

OUTPUT

```
(5, 'VARUN')
```


12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:05 AM

مجموعی افعال
Aggregate Functions 15.7

یہ افعال کالم کی قیمتوں سے آپریشن کرنے کے لیے استعمال ہوتے ہیں اور ایک ہی قیمت لوٹائی جاتی ہے۔

- COUNT()
- AVG()
- SUM()
- MAX()
- MIN()

COUNT() function 15.7.1

SQLCOUNT() فنکشن، ٹیبل میں صفوں کی تعداد لوٹاتا ہے، جو WHERE شق میں بیان کردہ معیار کو پورا کرتا ہے۔COUNT()، 0 لوٹاتا ہے اگر کوئی صف میچ نہ ہوتی ہو۔

مثال 15.7.1-1

مثال 1 : اس مثال میں ہم ریکارڈز (صفوں) کی تعداد گنتے جا رہے ہیں۔

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(*)  FROM student ")
result = cursor.fetchall()
print(result)
```

Output:

```
[(7,)]
```

مثال 15.7.1-2

مثال 2 : اس مثال میں ہم کالم کو مخصوص کر کے ریکارڈ کی تعداد گنتے جا رہے ہیں۔

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

Output:

```
[(7,)]
```

نوٹ

NULL قیمتیں شمار نہیں کی جاتی ہیں۔ اگر ہم طالب علموں کے جدول میں ریکارڈ میں سے کسی ایک کو مثال کے طور پر اوسط فیلڈ میں null قرار دیتے ہیں تو output کی قیمت 6 ہوگی۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:05 AM

15.7.2 AVG():

پائیتھن پروگرام میں درج ذیل SQL بیان تمام طلباء کا اوسط نمبر تلاش کرتا ہے۔

مثال 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)
```

OUTPUT

```
[(84.65714285714286,)]
```



نوٹ

NULL قیمتوں کو نظر انداز کیا جاتا ہے۔

15.7.3 SUM():

پائیتھن پروگرام میں درج ذیل SQL بیان "student table" کے اوسط کے فیلڈ میں تمام اوسط کا مجموعہ تلاش کرتا ہے۔

مثال 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)
```

OUTPUT

```
[(592.6,)]
```



نوٹ

NULL قیمتوں کو نظر انداز کیا جاتا ہے۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:05 AM

15.7.4 MAX() اور MIN() فنکشنس

MAX() فنکشن منتخب کالم کی سب سے بڑی قیمت لوٹاتا ہے۔

MIN() فنکشن منتخب کالم کی سب سے چھوٹی قیمت لوٹاتا ہے۔

مندرجہ ذیل مثال طالب علموں کے نام کا سب سے بڑا اور چھوٹا اوسط بتاتا ہے۔

مثال 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)
```

OUTPUT

```
Displaying the name of the Highest Average
[('PRIYA', 98.6)]
Displaying the name of the Least Average
[('TARUN', 62.3)]
```

15.8 Updating A Record ایک ریکارڈ کو اپ ڈیٹ کرنا

یہاں تک کہ آپ پائیتھن کے سکرپٹ کے ذریعے ٹیبل میں ریکارڈ (tuple) کو اپ ڈیٹ کر سکتے ہیں۔ درج ذیل مثال نے "priya" کا نام تبدیل کر کے "student table" میں ریکارڈ میں "priyanka" رکھ دیا

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:05 AM

مثال 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()
```

OUTPUT

```
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')
```

نوٹ

اس باب میں یاد رکھیں کہ ہم نے **student table** بنایا ہے، اسے SQL سوالات کی وضاحت کے لیے مثال کے طور پر لیا گیا ہے۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:05 AM

ڈیلیشن آپریشن
Deletion Operation 15.9

ایک ریکارڈ کو حذف کرنے کے لیے Sql کمانڈ کی طرح، پائیتھن بھی ایک ریکارڈ کو حذف کرنے کی اجازت دیتا ہے۔ مندرجہ ذیل مثال Rollno 2 کے مواد کو 'student table" سے حذف کرنے کے لئے ہے۔

مثال 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()
```

OUTPUT

```
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')
```


12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:05 AM

صارف کے ذریعہ ڈیٹا ان پٹ

15.10 Data input by User

اس مثال میں ہم رن ٹائم کے دوران پائیتھن input() کمانڈ کا استعمال کرتے ہوئے ڈیٹا کو قبول کرنے جا رہے ہیں اور پھر "Person" نامی ٹیبل میں لکھنے جا رہے ہیں۔

مثال 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' )
```


12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:05 AM

OUTPUT

```
        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)
```

یہاں تک کہ آپ پہلے سے موجود ٹیبل میں ریکارڈ بھی شامل کر سکتے ہیں جیسا کہ "student" فیلڈ نام میں مناسب ترمیم کے ساتھ مذکورہ کوڈنگ کا استعمال کرتے ہوئے۔ ایسا کرنے کے لیے آپ کو **create table** بیان پر تبصرہ کرنا چاہیے۔

نوٹ

Execute(sql[,parameters]) : یہ ایک اکیلے **sql** بیان پر عمل درآمد کرتا ہے۔ **sql** بیان کو پیرامیٹرائز کیا جاتا ہے۔ یعنی (**sql literals**) کی بجائے **placeholders** کو استعمال کرتا ہے۔ **sqlite3** ماڈیول دو قسم کے **placeholders** کی حمایت کرتا ہے۔ : سوالیہ نشان **?** ("qmark syle") اور نام والے **name** ("named style") ۔**placeholders**

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:06 AM

15.11 استفسار کے لیے ایک سے زیادہ جدول کا استعمال (Using multiple table for Querying)

پائیتھن ان میں شامل ہوکر ایک سے زیادہ ٹیبل سے استفسار کرنے کی اجازت دیتا ہے۔ مندرجہ ذیل مثال میں ایک نیا جدول جسے "Appointments" کہا جاتا ہے جس میں طلباء کے Rollno، Duty، Age وغیرہ تیار کئے جاتے ہیں۔ جدولیں "student table" اور "Appointments" شامل ہیں اور کالم کے عنوانات کے ساتھ نتیجہ ظاہر کرتی ہیں۔

مثال 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)
```

OUTPUT

```
['Rollno', 'Sname', 'Duty', 'age']
(1, 'Akshay', 'Prefect', 17)
(2, 'Aravind', 'Secretary', 16)
```

نوٹ:

cursor تفصیل ہر کالم کے عنوانات کی تفصیلات پر مشتمل ہے۔ یہ ایک ٹپ tuple کے طور پر محفوظ کیا جائے گا اور پہلا جو 0 (صفر) انڈیکس ہے، کالم کے نام سے مراد ہے۔ انڈیکس 1 سے کالم (ریکارڈز) کی اقدار کا حوالہ دیا جاتا ہے۔ اس کمانڈ کا استعمال کرتے ہوئے آپ ٹیبل کے فیلڈ نام ظاہر کر سکتے ہیں۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:06 AM

15.12 سوال کو Csv فائل کے ساتھ مربوط کرنا (Integrating Query with CSV file)

یہاں تک کہ آپ استفسار (query) کے نتیجہ کو CSV فائل میں محفوظ کر سکتے ہیں۔ یہ استفسار آؤٹ پٹ کو ٹیبلر فارمیٹ میں ظاہر کرنے کے لیے مفید ہوگا۔ مندرجہ ذیل مثال میں (مثال 15.12-1) پائیتھن اسکرپٹ کا استعمال کرتے ہوئے طالب علم کی جدول کو "gender" کے لحاظ سے ترتیب وار ترتیب دیا جاتا ہے اور پھر حروف تہجی کے مطابق ریکارڈ ترتیب دیا جاتا ہے۔ اس استفسار کی پیداوار ایک CSV فائل میں لکھی جائے گی، جسے "SQL.CSV" کہا جاتا ہے، دوبارہ مواد کو CSV فائل سے پڑھا جاتا ہے اور نتیجہ دکھایا جاتا ہے۔

مثال 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()
```

OUTPUT

```
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
```


12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:06 AM

مثال 15.12 –2 MS-Excel کے ذریعہ ("sqlexcel.csv") فائل کو کھولنا اور نتیجہ دیکھنا
(پروگرام مثال 15.12 -1 اسکرپٹ کی طرح ہے۔)

```
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()
```

نوٹ

CSV فائل میں لکھتے ہوئے by default ہر ریکارڈ \n(newline) کے ساتھ ختم ہوتا ہے تا کہ اس نئی لائن کو ختم کیا جا سکے۔
replace() فائل کو پڑھنے کے دوران استعمال کیا جاتا ہے۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:06 AM

OUTPUT

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

جدول کی فہرست
Table List 15.13

ایک ڈیٹا بیس میں بنائے گئے جدولوں کی list ظاہر کرنے کے لیے درج ذیل پروگرام (مثال 15.3-1) استعمال کیا جاسکتا ہے۔

مثال 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())
```

OUTPUT

[('Student',), ('Appointment',), ('Person',)]

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:06 AM

مذکورہ پروگرام (مثال 15.3-1) اکیڈمی میں بنائے گئے تمام جدولوں کے نام ظاہر کرتا ہے۔

'Academy.db' ماسٹر ٹیبل آپ کے ڈیٹا بیس ٹیبلز کے بارے میں کلیدی معلومات رکھتا ہے اور اسے sqlite_master کہا جاتا ہے۔

اب تک، آپ پائیتھن کے اسکرپٹس میں Structured Query Language (SQL) استعمال کر رہے ہیں۔ اس باب میں بہت سے بنیادی SQL احکامات کا احاطہ کیا گیا ہے۔ تقریباً تمام SQL کمانڈ پائیتھن SQLite ماڈیول کے ذریعے عمل میں لائی جاسکتی ہیں۔ یہاں تک کہ آپ SQL باب میں زیر بحث دیگر احکامات کو بھی آزما سکتے ہیں۔

یاد رکھنے کے نکات:

- database ڈیٹا بیس ڈیٹا کا منظّم مجموعہ ہے۔
- database ڈیٹا بیس کے صارفین انسانی استعمال کے لئے، دوسرے پروگرام یا ایپلی کیشنز ہو سکتے ہیں۔
- SQLite QL ایک سادہ تعلقی ڈیٹا بیس سسٹم ہے، جو اپنا ڈیٹا باقاعدہ ڈیٹا فائلوں میں محفوظ کرتا ہے۔
- Cursor ایک کنٹرول ساخت ہے، جو ڈیٹا بیس کے ریکارڈ کو عبور کرنے اور لانے کے لیے استعمال ہوتا ہے۔ تمام SQL کمانڈ صرف کرسر آبجیکٹ کا استعمال کرتے ہوئے عمل میں لائے جائیں گے۔
- جیسا کہ ایک ٹیبل میں ڈیٹا، سنگل ' یا ڈبل کوٹس " پر مشتمل ہو سکتا ہے، پائیتھن میں SQL کمانڈز کو ٹرپل کوٹڈ سٹرنگ """ کے طور پر بیان کیا جاتا ہے۔
- "SELECT" SQL میں سب سے زیادہ استعمال ہونے والا بیان ہے۔
- SQL میں SELECT بیان ایک ڈیٹا بیس میں ٹیبل سے ڈیٹا بازیافت کرنے کے لیے استعمال کیا جاتا ہے۔
- GROUP BY شق گروپوں کو summary rows میں ریکارڈ کرتا ہے۔
- ORDER BY شق SELECT بیان کے ساتھ مخصوص فیلڈز کے ڈیٹا کو ترتیب دینے کے لیے استعمال کیا جاسکتا ہے
- گروپ کے افعال پر مبنی ڈیٹا کو فلٹر کرنے کے لیے شق کا استعمال کیا جاتا ہے۔
- WHERE شق کے ساتھ 'Group by' استعمال نہیں کی جاسکتی۔
- WHERE شق کو AND، OR اور NOT آپریٹرز کے ساتھ ملایا جاسکتا ہے۔
- 'AND' اور 'OR' آپریٹرز ایک سے زیادہ شرائط پر مبنی ریکارڈ کو فلٹر کرنے کے لیے استعمال ہوتے ہیں۔
- functions مجموعی افعال کالم کی قیمتوں سے آپریشن کرنے کے لیے استعمال کیے جاتے ہیں اور ایک ہی قیمت واپس کی جاتی ہے۔
- COUNT() فنکشن ٹیبل میں صفوں کی تعداد لوٹاتا ہے۔
- AVG() فنکشن ٹیبل میں صفوں کے منتخب کالم کی اوسط حاصل کرتا ہے۔
- SUM() فنکشن ٹیبل میں قطاروں کے منتخب کالم کا مجموعہ حاصل کرتا ہے۔
- MAX() فنکشن منتخب کالم کی سب سے بڑی قیمت لوٹاتا ہے۔
- MIN() فنکشن منتخب کالم کی سب سے چھوٹی قیمت لوٹاتا ہے۔
- sqlite_master ماسٹر ٹیبل ہے جو آپ کے ڈیٹا بیس ٹیبلز کے بارے میں کلیدی معلومات رکھتا ہے۔
- پیتھان میں فائل کی path کو '/' یا '\' کے استعمال سے نمائندگی کی جاتی ہے۔ مثال کے طور پر path کو اس طرح 'c:/pyprg/sql.csv' یا 'c:\pyprg\sql.csv' کی طرح بیان کیا جاسکتا ہے۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:07 AM



عملی مشق (Hands on experience)

1۔ صارف سے تفصیلات کو قبول کرنے کے لیے ایک انٹرایکٹو پروگرام بنائیں اور اسے درج ذیل ٹیبل کے لیے پائیتھن کا استعمال کرتے ہوئے csv فائل میں محفوظ کریں۔

Table name : 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۔ درج ذیل ٹیبل GAMES پر غور کریں۔ سوال (i) سے (iv) کے ریکارڈ ظاہر کرنے کے لیے ایک پائیتھن کا پروگرام لکھیں اور SQL سوالات (v) سے (viii) کے لیے آؤٹ پٹ دیں۔

Table: 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) تمام گیمز کا نام ان کے Gcodes کے ساتھ ان کے شیڈول کی تاریخ کے نزولی ترتیب میں ظاہر کرنا۔

(ii) ان گیمز کی تفصیلات دکھا، جن کے پاس پرائز منی 7000 سے زیادہ ہے۔

(iii) gamename کے بڑھتے ہوئے ترتیب میں کھلاڑیوں کا نام اور گیم نام ظاہر کرنا۔

(iv) حصہ لینے والے گروپوں میں سے ہر ایک کے لیے پرائز منی کی رقم ظاہر کرنا (جیسا کہ کالم نمبر 4 میں دکھایا گیا ہے)

(v) گیم نام پر مبنی تمام ریکارڈ دکھانا۔

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:07 AM

Evaluation محاسبہ

حصہ 1

صحیح جواب کا انتخاب کریں۔ (1 مارک)

1۔ مندرجہ ذیل میں سے کون سا ڈیٹا کا منظم مجموعہ ہے؟

(A) Database (B) DBMS (C) Information (D) Records

2۔ SQLite کس ڈیٹا بیس سسٹم کے تحت آتا ہے؟

(A) Flat file database system (B) Relational Database system

(C) Hierarchical database system (D) Object oriented Database system

3۔ مندرجہ ذیل میں سے کون سی ڈیٹا ساخت ہے جو ڈیٹا بیس کے ریکارڈ کو عبور کرنے اور لانے کے لیے استعمال ہوتا ہے؟

(A) Pointer (B) Key

(C) Cursor (D) Insertion point

4۔ ریکارڈ کی قیمتوں میں کی جانے والی کسی بھی تبدیلی کو کمانڈ کے ذریعے محفوظ کیا جانا چاہیے۔

(A) Save (B) Save As (C) Commit (D) Oblige

5۔ مندرجہ ذیل میں سے کون سا عمل کرنے کے لیے SQL کمانڈ پر عمل کرتا ہے؟

(A) Execute() (B) Key() (C) Cursor() (D) run()

6۔ مندرجہ ذیل میں سے کون سا فنکشن ٹیبل میں قطاروں کے منتخب کالم کی اوسط حاصل کرتا ہے؟

(A) Add() (B) SUM() (C) AVG() (D) AVERAGE()

7۔ وہ فنکشن جو منتخب کالم کی سب سے بڑی قدر لوٹاتا ہے۔

(A) MAX() (B) LARGE()

(C) HIGH() (D) MAXIMUM()

8۔ مندرجہ ذیل میں سے کس کو ماسٹر ٹیبل کہا جاتا ہے؟

(A) sqlite_master (B) sql_master

(C) main_master (D) master_main

9۔ SQL میں سب سے زیادہ استعمال ہونے والا بیان یہ ہے۔

(A) cursor (B) select (C) execute (D) commit

10۔ درج ذیل میں سے کون سی شق نقل (duplicate) سے بچاتی ہے؟

(A) Distinct (B) Remove (C) Where (D) GroupBy

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:07 AM

حصہ II

مندرجہ ذیل سوالات کے جوابات دیں۔ (2 نمبرات)

1۔ ڈیٹا بیس استعمال کرنے والے صارفین کا ذکر کریں۔

2۔ ڈیٹا بیس کو جوڑنے کے لیے کون سا طریقہ استعمال کیا جاتا ہے؟ ایک مثال دیں۔

3۔ کالم کو "INTEGER PRIMARY KEY" قرار دینے کا کیا فائدہ ہے؟

4۔ جدول میں ریکارڈ کو بھرتی کرنے کے لیے کمانڈ لکھیں۔ ایک مثال دیں۔

5۔ ڈیٹا بیس ٹیبل سے تمام صفوں لانے کے لیے کون سا طریقہ استعمال کیا جاتا ہے؟

حصہ III

درج ذیل سوالات کے جوابات دیں۔ (3 نمبرات)

1۔ SQLite کیا ہے؟ اس کا فائدہ کیا ہے؟

2۔ fetchone() اور fetchmany() کے درمیان فرق کا ذکر کریں۔

3۔ WHERE شق کا استعمال کیا ہے۔ WHERE شق کا استعمال کرتے ہوئے ایک پائیتھن کا بیان دیں۔

4۔ درج ذیل تفصیلات پڑھیں۔ اس کی بنیاد پر department wise ریکارڈوں کو بتانے کے لئے ایک پیتھان اسکرپٹ لکھیں۔

database name :- organization.db
Table name :- Employee
Columns in the table :- Eno, EmpName, Esal, Dept

5۔ درج ذیل تفصیلات پڑھیں۔اس کی بنیاد پر ریکارڈوں کو نزولی ترتیب میں ظاہر کرنے کے لئے ایک پیتھان اسکرپٹ لکھیں۔

Eno
database name :- organization.db
Table name :- Employee
Columns in the table :- Eno, EmpName, Esal, Dept

حصہ IV

مندرجہ ذیل سوالات کے جوابات دیں۔ (5 نمبرات)

1۔ SQLite اور اس کے استعمال کے لیے استعمال کیے جانے والے اقدامات کے بارے میں مختصر طور پر لکھیں۔

2۔ fetchmany() کا استعمال کرتے ہوئے درج ذیل ٹیبل کے تمام ریکارڈز کو ظاہر کرنے کے لیے پائیتھن کا سکرپٹ لکھیں

Icode	ItemName	Rate
1003	Scanner	10500
1004	Speaker	3000
1005	Printer	8000
1008	Monitor	15000
1010	Mouse	700

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:08 AM

3۔ HAVING شق کا استعمال کیا ہے ۔ پائیتھن اسکرپٹ کی ایک مثال دیں۔

4۔ مندرجہ ذیل تفصیلات کے ساتھ ITEM نامی ٹیبل بنانے کے لیے ایک پائیتھن کا اسکرپٹ لکھیں۔ جدول پر ایک ریکارڈ شامل کریں۔

Add one record to the table.
Name of the database :- ABC
Name of the table :- Item
Column name and specification :-

Icode	:-	integer and act as primary key
Item Name	:-	Character with length 25
Rate	:-	Integer
Record to be added	:-	1008, Monitor,15000

5۔ مندرجہ ذیل ٹیبل سپلائر اور آئٹم پر غور کریں۔ (i) سے (ii) کے لیے پائیتھن کی اسکرپٹ لکھیں۔

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) ان سپلائرز کے نام، شہر اور آئٹم نام دکھائیں جو دہلی میں نہیں رہتے۔

ii) Akila کی سپلائی میں 40 کے اضافہ کے لئے

حوالہ جات

1. *The Definitive Guide to SQLite by Michael Owens*
2. *Programming for Beginners: 2 Manuscripts: SQL & Python by Byron Francis*
3. *Tutorialspoint.com*

12 CS UNIT 15 CDR set
Tuesday, April 18, 2023 6:19:08 AM

باب 16

PYPLOT کا استعمال کرتے ہوئے ڈیٹا کو دیکھنا: خطی خاکے ، پئی خاکے اور ستونی خاکے
DATA VISUALIZATION USING PYPLOT: LINE CHART, PIE CHART AND BAR CHART

سیکھنے کے مقاصد

8XCYS

اس باب کا مطالعہ کرنے کے بعد، سیکھنے والے اس قابل ہو جائیں گے۔

- اصطلاح Data visualization کی وضاحت کر سکیں گے۔
- Data visualization کے اقسام کی فہرست بنا سکیں گے۔
- Data visualization کے استعمال کی فہرست بنا سکیں گے۔
- Matplotlib میں Visualizations کے اقسام کی فہرست بنا سکیں گے۔
- Matplotlib انسٹال کرنے کی مشق کر سکیں گے۔
- Matplotlib کی جانکاری حاصل کر سکیں گے۔
- Data visualization plots کی اقسام کی درجہ بندی کر سکیں گے۔
- Matplotlib کا استعمال کرتے ہوئے مختلف قسم کے پلاٹ بنانے کی مشق کر سکیں گے۔

16.1 ڈیٹا ویزولائزیشن کی تعریف (Data Visualization Definition)

ڈیٹا ویزولائزیشن معلومات اور ڈیٹا کی تصویری نمائندگی ہے۔ ڈیٹا ویژولائزیشن کا مقصد صارفین کو معلومات بصری طور پر پہنچانا ہے۔ اس کے لیے ڈیٹا ویزولائزیشن شماریاتی گرافکس (Statistical graphics) استعمال کرتی ہے۔ عددی ڈیٹا کو نقطوں، لائنوں یا ستونوں کا استعمال کرتے ہوئے انکوڈ کیا جا سکتا ہے، تاکہ مقداری پیغام کو بصری طور پر پہنچایا جا سکے۔

ڈیٹا ویزولائزیشن کی عمومی اقسام

- چارٹس
- جدولیں
- گراف
- نقشے
- انفوگرافکس
- ڈیش بورڈز

ڈیٹا کو دیکھنا - استعمالات (Data Visualization - uses)

- ڈیٹا ویزولائزیشن صارفین کو ڈیٹا کا آسانی سے تجزیہ اور تشریح کرنے میں مدد کرتا ہے۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:33 AM

- یہ پیچیدہ ڈیٹا کو قابل فہم اور قابل استعمال بناتا ہے۔
- ڈیٹا ویژولائزیشن میں مختلف چارٹس ایک یا زیادہ متغیرات کے لیے ڈیٹا میں تعلق دکھانے میں مدد کرتے ہیں۔

کیا تم جانتے ہو

انفوگرافکس (Inforgraphics)-ایک انفوگرافک (انفارمیشن گرافک) گرافک فارمیٹ میں معلومات کی نمائندگی ہے۔
ڈیش بورڈ (Dashboard)- ڈیش بورڈ وسائل کا ایک مجموعہ ہے، جو ایک واحد بصری ڈسپلے بنانے کے لیے جمع ہوتا ہے۔ ڈیٹا ویزولائزیشن اور ڈیش بورڈ پیچیدہ خیالات اور تصورات کو سادہ بصری شکل میں ترجمہ کرتے ہیں۔ پیٹرن اور تعلقات جو متن میں ناقابل شناخت ہیں، ڈیش بورڈ کا استعمال کرتے ہوئے ایک نظر میں قابل شناخت ہیں۔

Matplotlib کا تعارف - پائیتھن میں ڈیٹا ویزولائزیشن

Matplotlib پائیتھن کی سب سے مشہور ڈیٹا ویزولائزیشن لائبریری ہے۔ یہ آپ کو کوڈ کی چند لائنوں میں چارٹ بنانے کی اجازت دیتا ہے۔

Matplotlib میں Visualizations کی اقسام

Matplotlib کے تحت Visualizations کی کئی اقسام ہیں۔ ان میں سے کچھ یہ ہیں:

- لائن پلاٹ
- بکھرنے والا پلاٹ (Scatter plot)
- ہسٹوگرام
- باکس پلاٹ
- ستونی چارٹ
- پائی چارٹ

کیا تم جانتے ہو

سیکٹر پلاٹ (Sector plot) :سیکٹر پلاٹ ایک قسم کا پلاٹ ہے جو ڈیٹا کو پوائنٹس کے مجموعے کے طور پر دکھاتا ہے۔ ایک نقطہ کی پوزیشن اس کی دو ابعادی قیمت پر منحصر ہے، جہاں ہر ایک قیمت افقی یا عمودی ابعاد پر ایک پوزیشن میں ہوتی ہے۔
باکس پلاٹ (Box plot): باکس پلاٹ پانچ نمبر کے خلاصے پر مبنی ڈیٹا کی تقسیم کو ظاہر کرنے کا ایک معیاری طریقہ ہے: کم سے کم، پہلا چوتھائی، درمیانی، تیسرا چوتھائی اور زیادہ سے زیادہ۔

Matplotlib انسٹال کرنا

آپ pip کا استعمال کرتے ہوئے matplotlib انسٹال کر سکتے ہیں۔ Pip ، python package انسٹال کرنے کے لیے ایک مینجمنٹ سافٹ ویئر ہے۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:33 AM

نوٹ

انسٹالیشن کا تفصیلی طریقہ کار ضمیمہ میں دیا گیا ہے۔

16.2 شروع کرنا (Getting started)

Matplotlib انسٹال کرنے کے بعد، ہم کمانڈ کا استعمال کرتے ہوئے Matplotlib درآمد کر کے کوڈنگ شروع کریں گے:

```
import matplotlib.pyplot as plt
```

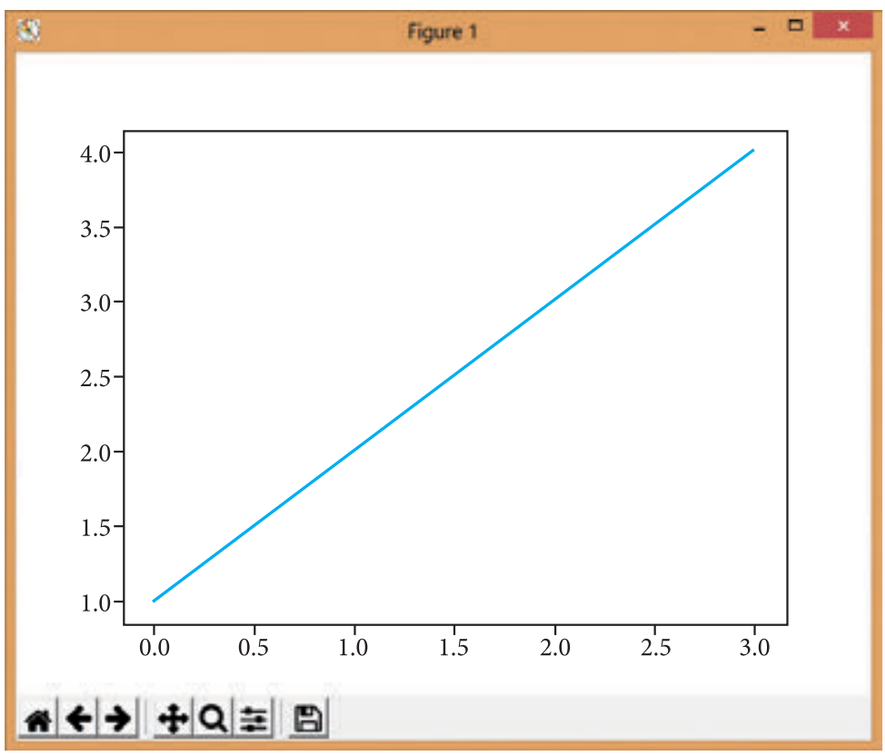
اب آپ نے اپنے کام کی جگہ پر Matplotlib درآمد کیا ہے۔ آپ کو پلاٹس دکھانے کی ضرورت ہے۔ پائیتھن اسکرپٹ کے اندر سے Matplotlib کا استعمال کرتے ہوئے، آپ کو اپنے پلاٹ کو ظاہر کرنے کے لیے فائل کے اندر plt.show( ) طریقہ شامل کرنا ہوگا۔

مثال

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.show()
```

Output

ونڈو ایک matplotlib ونڈو ہے، جو آپ کو اپنا گراف دکھاتا ہے۔ تم گراف کو استعمال کر کے نچلے دائیں جانب محدد دیکھ سکتے ہیں۔



خاکہ 16. 1

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:33 AM

آپ سوچ رہے ہوں گے کہ X محور 0-3 اور y محور 1-4 کے حدود میں ہے۔ اگر آپ plot( ) کمانڈ کو ایک فہرست یا صف (array) میں فراہم کرتے ہیں تو، matplotlib فرض کرتا ہے کہ یہ y کی قیمتوں کا تسلسل ہے، اور خود بخود آپ کے لیے x کی قیمتیں پیدا کرتا ہے۔ چونکہ پائیتھن کی حدیں 0 سے شروع ہوتی ہیں، پہلے سے طے شدہ x ویکٹر کی لمبائی y کے مساوی ہوتی ہے، لیکن 0 سے شروع ہوتی ہے۔

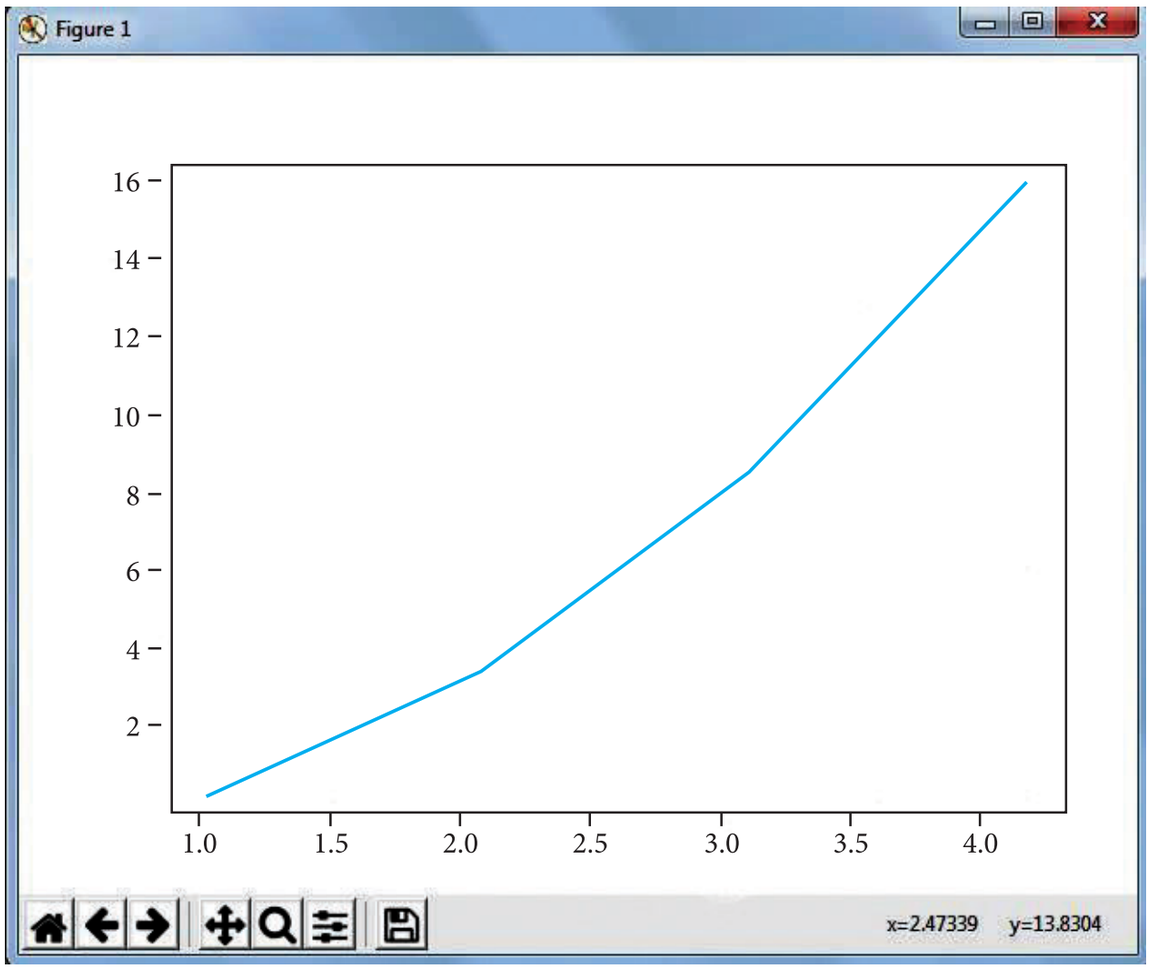
plot( ) ایک ورسٹائل کمانڈ ہے، اور دلائل (arguments) کی تعداد من مانی (arbitrary) لے گا۔

پروگرام

مثال کے طور پر، x بمقابلہ y کو پلاٹ کرنے کے لیے، آپ یہ کمانڈ جاری کر سکتے ہیں:

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16])
plt.show()
```

یہ پلاٹ کئی پیرامیٹرس لے گا، مگر یہاں موجود پہلے دو x اور y محددیں ہیں۔ اس کا یہ مطلب ہے کہ تمہارے پاس اس لسٹ کے حساب سے 4 محددیں ہیں۔ (1,1)، (2,4)، (3,9)، (4,16)۔



خاکہ 16.2

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:33 AM

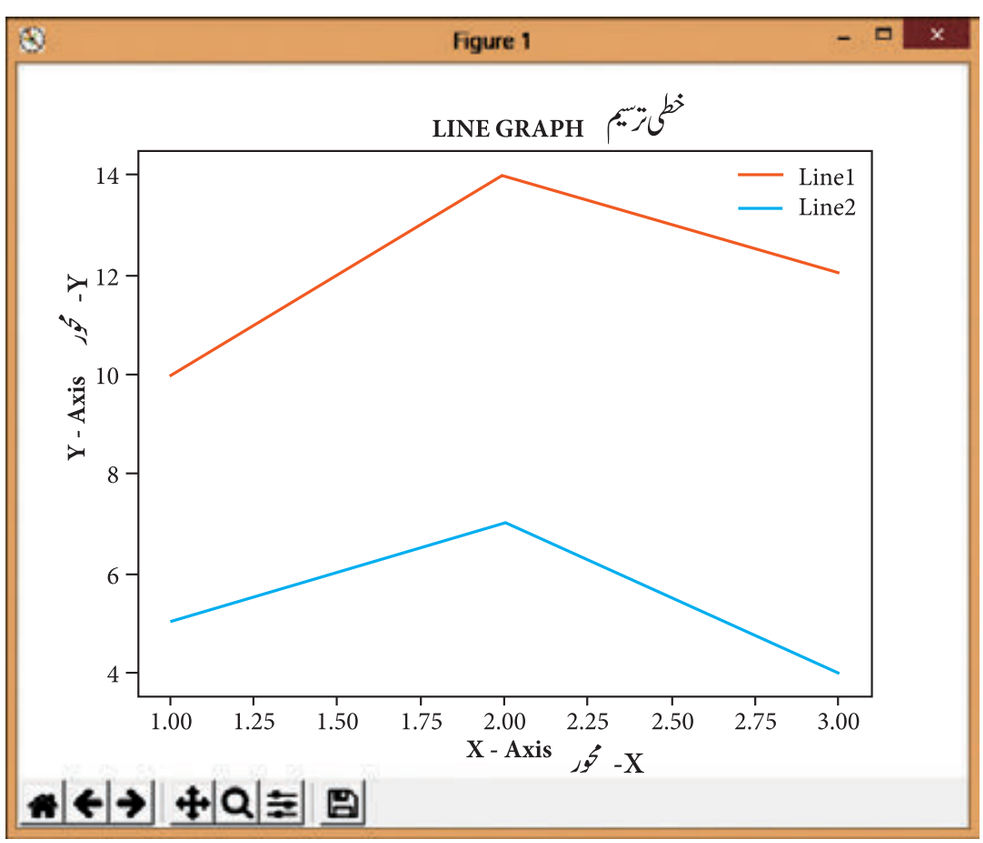
دوخطوط کو مرتسم کرنا (Plotting Two lines)

دوخطوط کو مرتسم کرنے کے لیے درج ذیل کوڈ استعمال کریں:

```
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()
```

Output

plt.xlabel اور plt.ylabel کے ساتھ، آپ ان متعلقہ محور کو لیبل تفویض کر سکتے ہیں۔ اگلا، آپ پلاٹ کا ٹائٹل plt.title کے ساتھ تفویض کر سکتے ہیں، اور پھر آپ plt کے ساتھ ڈیفالٹ legend کو plt.legend() کی طرح طلب کر سکتے ہیں۔

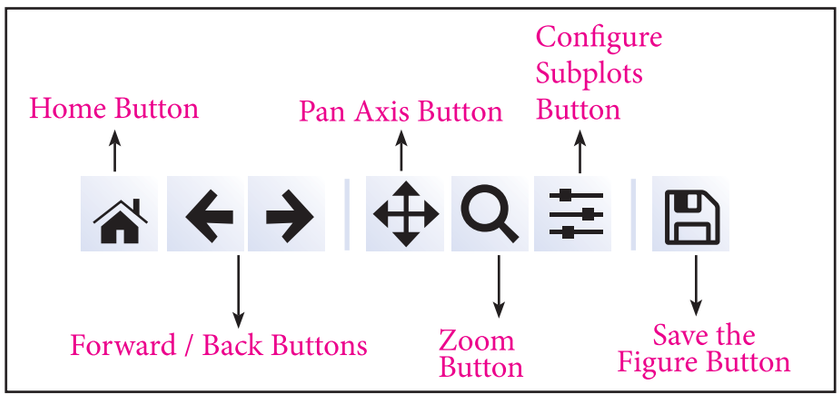


خاکہ 16.3

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:33 AM

آؤٹ پٹ کے بٹن (Buttons in the output)

آؤٹ پٹ کے خاکہ میں، آپ نیچے بائیں کونے میں چند بٹن دیکھ سکتے ہیں۔ آیئے ان بٹنوں کا استعمال دیکھیں۔



خاکہ 16.4

ہوم بٹن (Home Button) ← جیسے ہی آپ نے ہوم بٹن دبایا، یہ اپنے چارٹ پر لے جائے گا۔ اگر آپ کو اصل منظر پر واپس جانا چاہتے ہیں، تو آپ اس پر کلک کر سکتے ہیں۔

فارورڈ/ بیک بٹن (Forward/Back buttons) ← یہ بٹن آپ کے براؤزر میں فارورڈ اور بیک بٹن کی طرح استعمال کیے جا سکتے ہیں۔ آپ ان پر کلک کر کے پچھلے مقام پر واپس جا سکتے ہیں، یا پھر آگے بڑھا سکتے ہیں۔

پین محور (Pan Axis) ← یہ کراس دکھائی دینے والا بٹن آپ کو اس پر کلک کر کے اسے اپنے گراف کو گھمانے کی اجازت دیتا ہے۔

زوم (Zoom) ← زوم بٹن آپ کو اس پر کلک کرنے دیتا ہے، پھر کلک کریں اور ایک مربع کو گھسیٹیں جس میں آپ خاص طور پر زوم کرنا چاہتے ہیں۔ زوم کرنے کے لیے بائیں کلک اور ڈریگ کی ضرورت ہوگی۔ آپ متبادل طور پر دائیں کلک اور ڈریگ سے زوم آؤٹ کر سکتے ہیں۔

Subplots ترتیب دیں (Configure Subplots) ← یہ بٹن آپ کو اپنے خاکہ اور پلاٹ کے ساتھ مختلف اسپیسنگ آپشنز کو configure کرنے کی اجازت دیتا ہے۔

تصویر محفوظ کریں (Save figure) ← یہ بٹن آپ کو اپنی شکل کو مختلف شکلوں میں محفوظ کرنے کی اجازت دے گا۔

16.3 پلاٹ کی خصوصی اقسام (Special Plot Types)

Matplotlib آپ کو مختلف قسم کے پلاٹ بنانے کی اجازت دیتا ہے، جس میں ہسٹوگرام اور سکیٹر پلاٹس (Scatter plots) سے لے کر بارگراف اور بارچارٹ شامل ہیں۔

خطی چارٹ (Line Chart)

ایک خطی چارٹ یا خطی ترسیم ایک قسم کا چارٹ ہے جو معلومات کو ڈیٹا پوائنٹس کی ایک سیریز کے طور پر دکھاتا ہے جسے 'markers' کہا جاتا ہے جو سیدھی لائن کے حصوں سے جڑا ہوتا ہے۔ ایک لائن چارٹ اکثر وقت کے وقفوں کے دوران ڈیٹا میں رجحان کو دیکھنے کے لیے استعمال کیا جاتا ہے۔ ایک ٹائم سیریز۔ اس طرح یہ لائن اکثر ترتیب وار کھینچی جاتی ہے۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:34 AM

مثال:لائن پلاٹ۔

```
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()
```

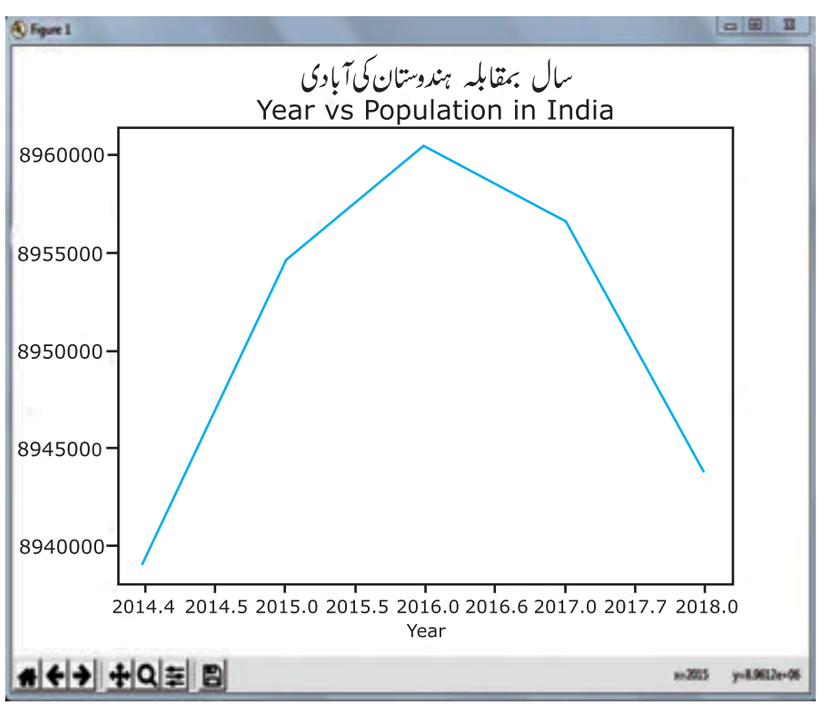
In this program,

Plt.title() → specifies title to the graph

Plt.xlabel() → specifies label for X-axis

Plt.ylabel() → specifies label for Y-axis

Output



خاکہ 16.5

بارچارٹ (ستونی چارٹ) (Bar chart)

ایک بار پلاٹ (ستونی چارٹ) پلاٹ کی سب سے عام قسم میں سے ایک ہے۔ یہ ایک عددی متغیر اور ایک غیر مشروط متغیر (Categorical variable) کے درمیان تعلق کو ظاہر کرتا ہے۔

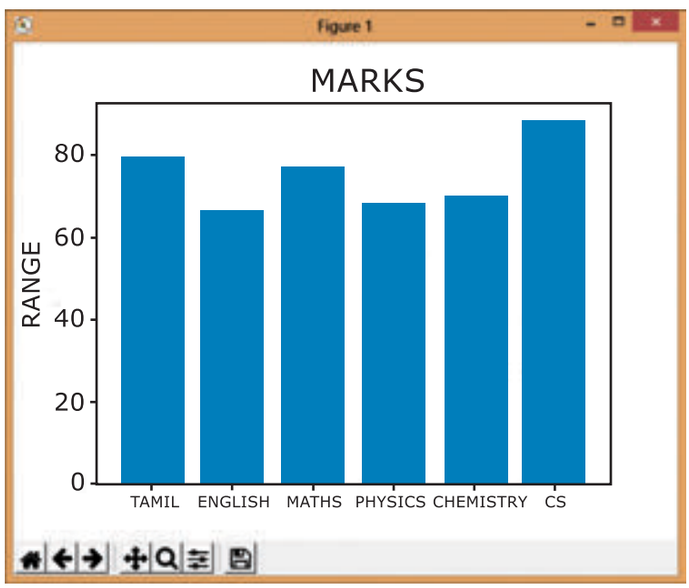
بار چارٹ مستطیل ستونوں کے ساتھ غیر مشروط ڈیٹا کی نمائندگی کرتا ہے۔ ہر بار کی اونچائی اس قدر سے ملتی ہے جس کی وہ نمائندگی کرتی ہے۔ سلاخوں کو عمودی یا افقی طور پر پلاٹ کیا جا سکتا ہے۔ یہ اس وقت کام آتے ہیں، جب ہم مختلف زمروں میں دی گئی عددی قیمتوں کا موازنہ کرنا چاہتے ہیں۔ Matplotlib کے ساتھ بارچارٹ بنانے کے لیے ہم plt.bar() فنکشن استعمال کر سکتے ہیں۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:34 AM

مثال

```
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()
```

Output



خاکہ 16.6

مندرجہ بالا کوڈ مندرجہ ذیل کی نمائندگی کرتا ہے:

Labels ← ستونوں کے لیبل کو مخصوص کرتا ہے۔

Usgae ← مخصوص کردہ لیبلز کو قیمتیں تفویض کرتا ہے۔

Xticks ← نمائندگی کردہ قیمتوں کو x-axis پر ٹک کے نشانات دکھاتا ہے اور ئیں۔ پھر ہر ٹک مارک کے لیبل کی وضاحت کرتا ہے۔

Range ← نمبروں کی ترتیب تیار کرتا ہے۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:34 AM

کیا تم جانتے ہو؟

ڈیٹا کو خاکہ کی شکل میں ظاہر کرنے کے دو طریقے ستونی گراف اور ہسٹوگرام (مستطیلی ترسیم) ہیں۔

ہسٹوگرام (مستطیلی خاکہ) اور ستونی ترسیم کے درمیان اہم فرق

ہسٹوگرام اور بارگراف کے درمیان فرق مندرجہ ذیل ہے۔

1۔ ہسٹوگرام سے مراد ترسیمی نمائندگی ہے۔ جو ڈیٹا کی تعداد کو ظاہر کرنے کے لیے مستطیل کے ذریعے ڈیٹا دکھاتا ہے۔
ستونی ترسیم ڈیٹا کی ایک تصویری نمائندگی ہے، جو ڈیٹا کے مختلف زمروں کا موازنہ کرنے کے لیے ستونوں کا استعمال کرتی ہے۔

2۔ ایک ہسٹوگرام مسلسل متغیرات کی تعداد تقسیم کی نمائندگی کرتا ہے۔ اس کے برعکس، ایک مستطیلی گراف مجرد متغیرات کا ایک خاکہ نگاری موازنہ ہے۔

3۔ ہسٹوگرام عددی اعداد و شمار پیش کرتا ہے، جبکہ بارگراف مختلف ڈیٹا کو ظاہر کرتا ہے۔

4۔ ہسٹوگرام اس طرح کھینچا گیا ہے کہ مستطیلوں کے درمیان کوئی خلا نہیں ہے۔ اس کے ہاتھ پر، ایک بار مستطیلوں کے درمیان مناسب وقفہ ہے، جو بند ہونے کی نشاندہی کرتا ہے۔

5۔ ہسٹوگرام کی اشیاء اعداد ہیں، جو اعداد کی حدود کی نمائندگی کے لیے ایک ساتھ درجہ بندی کی جاتی ہیں۔ ستونی ترسیم کے برعکس، اشیاء کو انفرادی ہستیاں سمجھا جاتا ہے۔

6۔ ستونی ترسیم کے معاملے میں، بلاکس کو دوبارہ ترتیب دینا بہت عام ہے، سب سے اونچے سے نیچے تک۔ لیکن ہسٹوگرام کے ساتھ، یہ نہیں کیا جا سکتا، کیونکہ وہ کلاسوں کی ترتیب میں دکھائے گئے ہیں۔

7۔ ہسٹوگرام میں موجود مستطیلی چوڑائی ایک جیسی ہو سکتی ہے یا نہیں، جبکہ بارگراف میں ستونوں کی چوڑائی ہمیشہ ایک جیسی ہوتی ہے۔

پائی چارٹ (Pie Chart)

پائی چارٹ شاید چارٹ کی سب سے عام قسم میں سے ایک ہے۔ یہ ایک دائرہ نما ترسیم ہے، جس میں عددی تناسب کو واضح کرنے کے لیے ٹکڑوں میں تقسیم کیا گیا ہے۔ ایک پائی چارٹ کا نقطہ یہ ہے کہ حصوں کا تعلق پورے سے ظاہر کیا جائے۔

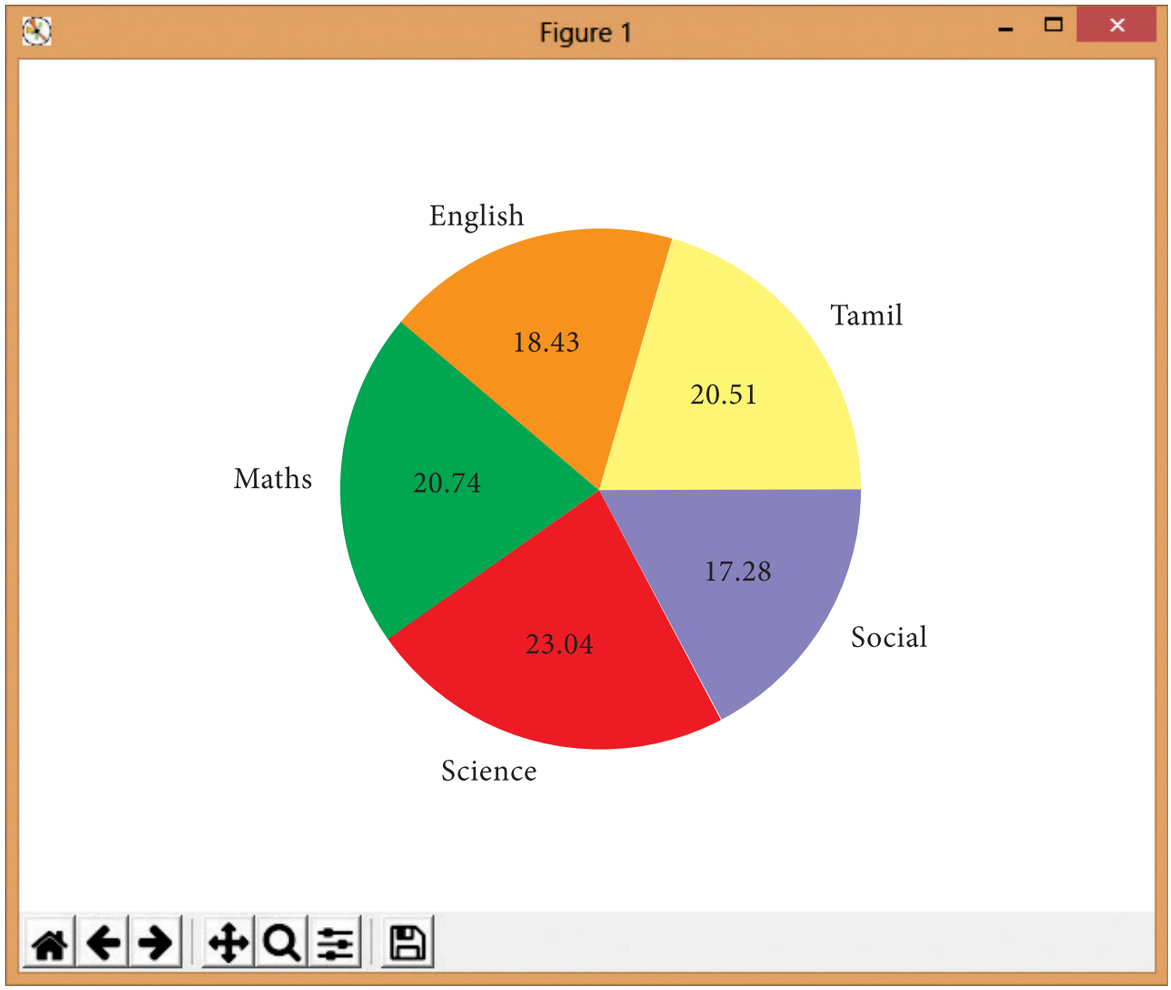
Matplotlib کے ساتھ پائی چارٹ بنانے کے لیے ہم ***plt.pie()*** فنکشن استعمال کر سکتے ہیں۔ **autopct** پیرامیٹر ہمیں پائیتھن سٹرنگ فارمیٹنگ کا استعمال کرتے ہوئے فی صد ویلیو ظاہر کرنے کی اجازت دیتا ہے۔

مثال

```
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()
```


12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:35 AM

Output



خاکہ 16 7 .

Hands on Practice عملی مشق

1۔ ایک پلاٹ بنائیں۔دونوں محوروں کے لیے عنوان، x اور y لیبل مقرر کریں۔

2۔ حالیہ امتحان میں اپنے نمبروں کے لیے پائی چارٹ بنائیں۔

3۔ پچھلے 10 سالوں سے کمپیوٹر سائنس میں 12 ویں کلاس کے طلباء کی تعلیمی کارکردگی پر ایک لائن (خطی) چارٹ بنائیں۔

4۔ ایک ہفتے میں کمپیوٹر سائنس کی گھنٹیوں کی تعداد کے لیے بار (ستونی) چارٹ بنائیں۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:35 AM

Evaluation محاسبہ

حصہ - I

صحیح جواب منتخب کیجئے۔ (مارک 1)

1۔ 2 D گرافکس کے لیے استعمال ہونے والا پائیتھن کا پیکج کون سا ہے ؟

a. matplotlib.pyplot
b. matplotlib.pip
c. matplotlib.numpy
d. matplotlib.plt

2۔ پائیتھن پیکجوں، یا ماڈیولز کے لیے پیکج مینیجر (package manager) کی شناخت کریں۔

a. Matplotlib
b. PIP
c. plt.show()
d. python package

3۔ ڈیٹا اور اطلاعات کو ترسیمی طور پر نمائندگی کرنے کے لئے کونسا فیچر استعمال ہوتا ہے؟

a. Data List
b. Data Tuple
c. Classes and Objects
d. Data Visualization

4۔ ایک اکہرے اور متحدہ بصری ڈسپلے کی تیاری کے لئے جمع کر کے ترتیب دئے جانے والے ذرائع کا مجموعہ۔۔۔۔۔۔۔۔ہے۔

a. Interface
b. Dashboard
c. Objects
d. Graphics

5۔ پئی تھان میں ڈیٹا اور اطلاعات کو دیکھنے کے لئے ذیل میں سے کونسے ماڈیول کو درآمد (import) کرنا پڑتا ہے؟

a. csv
b. getopt
c. mysql
d. matplotlib

6۔ قطاع خط کے ذریعہ سلسلہ وار ڈیٹا کے نقاط کو جوڑ کر اطلاعات کو ظاہر کرنے والے چارٹ کی ایک قسم۔۔۔۔۔۔۔۔۔ہے۔

a. csv
b. Pie chart
c. Bar chart
d. All the above

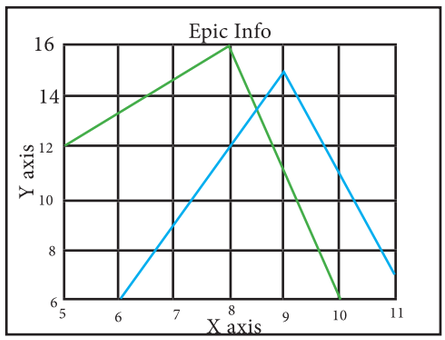
7۔ کوڈ کو پڑھیں :

```
a. import matplotlib.pyplot as plt
b. plt.plot(3,2)
c. plt.show()
```

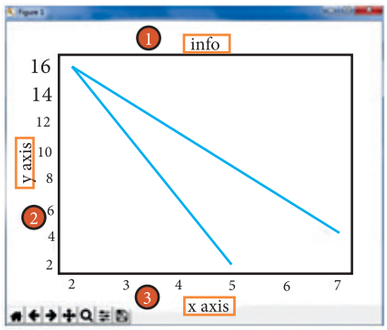
مندرجہ بالا کوڈنگ کے آؤٹ پٹ کی شناخت کریں۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:35 AM

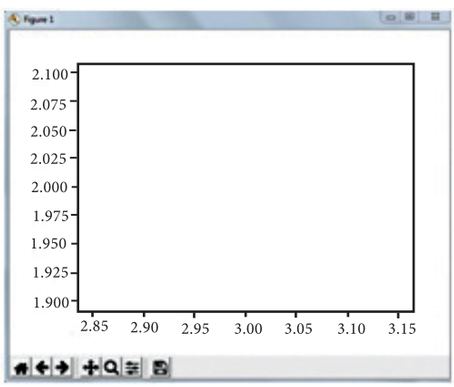
a.



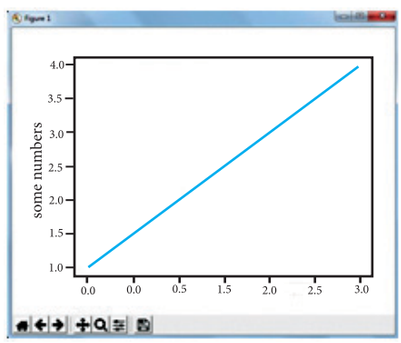
b.



c.



d.



8۔ درج ذیل اشارے کا استعمال کرتے ہوئے صحیح قسم کے چارٹ کی شناخت کریں۔

اشارہ 1 : یہ چارٹ اکثر وقت کے وقفوں پر ڈیٹا میں رجحان کو دیکھنے کے لیے استعمال کیا جاتا ہے۔

اشارہ 2 : اس قسم کے چارٹ میں خط (line) اکثر ترتیب وار (Chronologically) کھینچی جاتی ہے۔

a. Line chart　　b. Bar chart

c. Pie chart　　d. Scatter plot

9۔ نیچے دیئے گئے بیانات پڑھیں۔ پائی چارٹ کے لیے درج ذیل میں سے صحیح آپشن کی شناخت کریں۔

بیان A : Matplotlib کے ساتھ پائی چارٹ بنانے کے لیے، ہم plt.pie() فنکشن استعمال کر سکتے ہیں۔

بیان B : autopct پیرامیٹر ہمیں پائتھن سٹرنگ فارمیٹنگ کا استعمال کرتے ہوئے فی صد قیمت کو ظاہر کرنے کی اجازت دیتا ہے۔

a۔ بیان A درست ہے۔　　b۔ بیان B درست ہے۔

c۔ دونوں بیانات درست ہیں۔　　d۔ دونوں بیانات غلط ہیں۔

حصہ - II

مختصر جوابات لکھئے۔

(2 Marks)

1۔ وضاحت کیجئے: ڈیٹا ویزولائزیشن (Define : Data Visualization)۔

2۔ Data Visualization کی عام اقسام کی فہرست بنائیں۔

3۔ Matplotlib میں Visualizations کی اقسام کی فہرست بنائیں۔

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:35 AM

4۔ آپ Matplotlib کیسے انسٹال کریں گے ؟

5۔ درج ذیل افعال کے درمیان فرق لکھیں :

plt.plot ([1،2،3،4]), plt.plot([1،2،3،4], [1،4،9،16]).

حصہ - III

درج ذیل سوالات کے جوابات لکھئے۔ (3 مارکس)

1۔ درج ذیل ڈیٹا ویزولائزیشن پلاٹ کے لیے آؤٹ پٹ چھڑائیں۔

```
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۔ ڈیٹا ویزولائزیشن (Data visualisation) کے کوئی تین استعمالات لکھیں۔

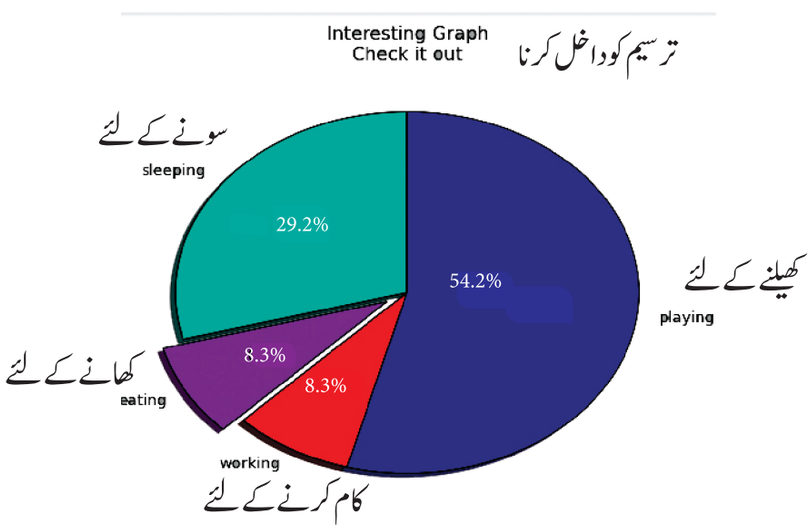
3۔ درج ذیل کے لیے کوڈنگ لکھیں:

a۔ یہ چیک کرنے کے لیے کہ آیا آپ کے کمپیوٹر میں PIP انسٹال ہے۔

b۔ اپنے کمپیوٹر میں نصب PIP کا ورژن چیک کرنے کے لیے۔

c۔ matplotlib میں پیکجوں کی فہرست بنانے کے لیے۔

4۔ مندرجہ ذیل پائی چارٹ آؤٹ پٹ کے لیے پلاٹ لکھیں۔



12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:36 AM

حصہ - IV

درج ذیل سوالات کے جوابات لکھئے۔ (5 مارکس)

1۔ Matplotlib کا استعمال کرتے ہوئے pyplots کی اقسام کی تفصیل سے وضاحت کریں۔

2۔ ایک matplotlib ونڈو کے مختلف بٹنوں کی وضاحت کریں۔

3۔ درج ذیل افعال کے مقصد کی وضاحت کریں:

a. plt.xlabel

b. plt.ylabel

c. plt.title

d. plt.legend()

e. plt.show()

Reference کتب حوالہ

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.

12 CS UNIT 16 CDR Set
Tuesday, April 18, 2023 6:53:36 AM



Terminology	Meaning
Access control	*security technique that regulates who or what can view or use resources in a computing environment*
Access modifiers	*Private , Protected and Public*
Alternative	*One of two choices*
append()	*Used to add an element in a list*
Argument	*Argument is the actual value of this variable that gets passed to function.*
argv	*An array containing the values passed through command line argument*
Attribute	*Data items that makes up an object*
Authorization	*Giving permission or access*
Block	*Set of Statements*
Boolean	*means Logical*
break	*Exit the control*
c = sqlite3.connect('test. db')	*create a database connection to the SQLite database 'test.db'. You can also supply the special name :memory: to create a database in RAM.*
c.close()	*To release the connection of the database*
c.commit()	*To save the changes made in the table*
c.execute()	*Executes all SQL commands .Accepts two kinds of placeholders: question marks ? ("qmark style") and named placeholders :name ("named style").*
Cartesian product	*Cartesian operation is helpful to merge columns from two relations*
cd	*cd command refers to change directory*
Class	*Template of creating objects.*
Class variable	*An ordinary variable declared inside a class*
cls	*To clear the screen in command window*
Comma(,)	*Comma is used to separate each data in a csv file*

12th Std - CS EM Glossary Pages.indd 321 23-12-2022 11:35:04

compiler	*Scans the entire program and translates it as a whole into machine code. It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.*
Conjunction	*Concurrence, coincidence*
Constraint	*Restriction or limitation*
Constructor	*A special function get execution automatically when an object enter into scope.*
continue	*To skip the remaining part and start with next iteration.*
CRUD	*Create, Read, Update and Delete*
csv.reader()	*The reader function is designed to take each line of the file and make a list of all columns*
csv.register_dialect()	*A dialect describes the format of the csv file that is to be read*
CsvQuote All	*If quoting is set to csvquote all, then writerow() will quote all fields.*
cur = c.cursor()	*Creating cursor object*
cur.fetchall()	*method to get a list of the matching rows.*
cur.fetchmany()	*method that returns the next number of rows (n) of the result set*
cur.fetchone()	*method to retrieve a single matching row*
CWI	*Centrum Wiskunde & Informatica*
DBA	*DataBase Adminstrator*
DBMS	*Database Management System*
def	*This keyword is used to define function.*
Destructor	*A special function get execution automatically when an object exit from its scope.*
dict()	*It is used to print the data in dictionary format without orderdict*
Dictionary	*Collection of Key-Value pairs*
DictReader()	*Works by reading in the first line of the CSV and using each column comma separated value in this line as a dictionary key.*
Dictwriter()	*Write dictionary data into a CSV file*
elif	*else…if*
Embedded	*Firmly attached*
Enter key	*Enter key or newline is used to create rows in a csv file*

12th Std - CS EM Glossary Pages.indd 322 23-12-2022 11:35:04

eval()	*This function is used to evaluate the value of a string.*
Father of Relational Database	*Dr. Edgar Frank Codd*
g++	*compiler to compile c++ program*
GIS	*Geographic Information System*
global Scope	*A variable, with global scope can be used anywhere in the program.*
Glue language	*You do not write the complete application in the language, but rather, you use the language to orchestrate(organize) modules written in (possibly many different) other languages, making them work together to form the application. A glue language makes it easy to do that (convenient syntax, good support for inter-process communication and data managing, no compilation step etc).*
id ()	*It returns the memory address of the given object.*
IDLE	*Integrated Development Environment*
immutable	*unchangeable*
Implementation	*Implementation carries out the operation declared in the interfac*
import	*Import in python is similar to #include header_file in C++. Python modules can get access to code from another module by importing the file/function using "import" statement.*
Impure Functions	*Any function that changes the internal state of one of its arguments or the value of some external variable is an impure function.*
Instantiation	*Process of creating an object*
Integrity	*Whole and undivided*
Interactive Mode	*A way of using the Python interpreter by typing command and expressions at the prompt.*
Interface	*Interface defines what an object can do, but doesn't actually do it*
interpreter	*Translates program one statement at a time. It continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.*
Intersection	*Intersection defines a relation consisting of a set of all tuple that are in both A and B.*
Key	*Data that is mapped to a value in a dictionary*

12th Std - CS EM Glossary Pages.indd 323 23-12-2022 11:35:04

lambda	*Lambda function is mostly used for creating small and one- time anonymous function.*
LEGB rule	*Local → Enclosed → Global → Built-in scope*
List	*Mutable ordered collection of values*
local Scope	*A variable declared inside the function's body or in a block is called local scope.*
Looping	*Repetition*
Mapping	*The process of binding a variable name with an object*
Method	*A function declared and defined inside a class.*
module	*A module is a file containing Python definitions and statements. The file name is the module name with the suffix .py appended. Within a module, the module's name (as a string) is available as the value of the global variable name .*
Namespaces	*containers for mapping names of variables to objects*
Nested Block	*A block within a block is called nested block.*
next()	*The next() function returns the next item from the iterator. It can also be used to skip a row of the csv file*
Object	*Collection of Data and Functions.*
Object Oriented Programming	*Computer Programming concept based on real world objects.*
operator.itemgetter(col_ no)	*To sort by more than one column from a csv file*
os.system()	*Used to execute system command and here in our python program is used to compile the c++ program using g++*
Parameter	*Parameter is variable in the declaration of function definition.*
parse	*To split an input into pieces of data that can be easily stored or manipulated.*
pass	*Can be used as placeholder in functions and loops.*
Projection (π)	*The projection eliminates all attributes of the input relation but those mentioned in the projection list*
Prompt	*Character (<<<) displayed by the interpreter to indicate that it is ready to take input from the user.*
Pure Functions	*Pure functions always returns the same result if the same arguments are passed in*
Python prompt	>>>

12th Std - CS EM Glossary Pages.indd 324 23-12-2022 11:35:04

Queue	*Queue is an abstract data structure, somewhat similar to Stacks. Unlike stacks, a queue is open at both its ends. One end is always used to insert data(enqueue) and the other is used to remove data(dequeue). Queue follows First-In-First-Out methodology, i.e., the data item stored first will be accessed first.*
RDBMS	*Relational Database Management System*
recursion	*When a function calls itself is known as recursion.*
Redundant	*Duplication of data*
Routines	*routines are otherwise called as functions or methods. In Python it is also called as definition*
Schema	*Structure or model*
Scope	*Visibility of variables, parameters and functions in one part of a program to another part of the same program.*
Script	*A Python program stored in a file.*
Script Mode	*A way of using the Python interpreter by typing command and expressions at the prompt.*
Select (σ)	*The SELECT operation is used for selecting a subset of the tuples according to a given selection condition*
Selectors	*Functions that retrieve information from the data type.*
Sequential	*One after another*
Set difference(-)	*(-) symbol denotes it. The result of A-B is a relation which includes all tuples that are in A but not in B.*
skipinitialspace=true	*When true, whitespace immediately following the delimeter is ignored. The default is false*
slicing	*cut*
Stack	*A stack (sometimes called a "push-down stack") is an ordered collection of items where the addition of new items and the removal of existing items always takes place at the same end. This end is commonly referred to as the "top." The end opposite the top is known as the "base.This ordering principle is sometimes called LIFO, last- in first-out.*
stride	*a long step*
string	*sequence of letters, numbers or symbols*
subscript	*an index number*

12th Std - CS EM Glossary Pages.indd 325 23-12-2022 11:35:05

Syntax	*The structure of a program*
Syntax Error	*An error in a program that makes it impossible to parse.*
Token	*One of the basic elements of the syntactic structure of a program.*
Tuple	*It is a sequence of immutable(not changeable) objects. Tuples are sequences, just like lists.Tuples are defined by having values between parentheses ( ).*
Union operation(U)	*Union is symbolized by symbol. It includes all tuples that are in tables A or in B.*
variable	*Memory box to store values*
writer ow()	*Method to write a single row of data in a file*
writer ows()	*Method to write multiple rows of data in a file*
FALSE	*Logical value 0*
TRUE	*Logical value 1*

12th Std - CS EM Glossary Pages.indd 326 23-12-2022 11:35:05

ANNEXURE - 1

Python کے افعال کی فہرست List of Python Functions

I. Built-in Functions

Function	Description
abs()	returns absolute value of a number
all()	returns true when all elements in iterable is true
any()	Checks if any Element of an Iterable is True
ascii()	Returns String Containing Printable Representation
bin()	converts integer to binary string
bool()	Converts a Value to Boolean
bytearray()	returns array of given byte size
bytes()	returns immutable bytes object
callable()	Checks if the Object is Callable
chr()	Returns a Character (a string) from an Integer
classmethod()	returns class method for given function
compile()	Returns a Python code object
complex()	Creates a Complex Number
delattr()	Deletes Attribute From the Object
dir()	Tries to Return Attributes of Object
divmod()	Returns a Tuple of Quotient and Remainder
enumerate()	Returns an Enumerate Object
eval()	Runs Python Code Within Program
exec()	Executes Dynamically Created Program
filter()	constructs iterator from elements which are true
float()	returns fl ating point number from number, string
format()	returns formatted representation of a value
getattr()	returns value of named attribute of an object
globals()	returns dictionary of current global symbol table
hasattr()	returns whether object has named attribute
hash()	returns hash value of an object
help()	Invokes the built-in Help System
hex()	Converts to Integer to Hexadecimal
id()	Returns Identify of an Object
isinstance()	Checks if a Object is an Instance of Class
issubclass()	Checks if a Object is Subclass of a Class
iter()	returns iterator for an object
len()	Returns Length of an Object
locals()	Returns dictionary of a current local symbol table
map()	Applies Function and Returns a List
max()	returns largest element
memoryview()	returns memory view of an argument
min()	returns smallest element
next()	Retrieves Next Element from Iterator
object()	Creates a Featureless Object
oct()	converts integer to octal

12th Std - CS EM Annexure Pages.indd 327 23-12-2022 13:16:59

open()	Returns a File object
ord()	returns Unicode code point for Unicode character
pow()	returns x to the power of y
print()	Prints the Given Object
property()	returns a property attribute
range()	return sequence of integers between start and stop
repr()	returns printable representation of an object
reversed()	returns reversed iterator of a sequence
round()	rounds a fl ating point number to ndigits places.
set()	returns a Python set
setattr()	sets value of an attribute of object
slice()	creates a slice object specifi d by range()
sorted()	returns sorted list from a given iterable
staticmethod()	creates static method from a function
str()	returns informal representation of an object
sum()	Add items of an Iterable
super()	Allow you to Refer Parent Class by super
type()	Returns Type of an Object
vars()	Returns __dict__ attribute of a class
__import__()	Advanced Function Called by import

II. String Functions

Function	Description
capitalize()	Converts fi st character to Capital Letter
casefold()	converts to casefolded strings
center()	Pads string with specifi d character
count()	returns occurrences of substring in string
encode()	returns encoded string of given string
endswith()	Checks if String Ends with the Specifi d Suffix
expandtabs()	Replaces Tab character With Spaces
find()	Returns the index of fi st occurrence of substring
format()	formats string into nicer output
format_map()	Formats the String Using Dictionary
index()	Returns Index of Substring
input()	reads and returns a line of string
int()	returns integer from a number or string
isalnum()	Checks Alphanumeric Character
isalpha()	Checks if All Characters are Alphabets
isdecimal()	Checks Decimal Characters
isdigit()	Checks Digit Characters
isidentifier()	Checks for Valid Identifie
islower()	Checks if all Alphabets in a String are Lowercase
isnumeric()	Checks Numeric Characters
isprintable()	Checks Printable Character
isspace()	Checks Whitespace Characters

12th Std - CS EM Annexure Pages.indd 328 23-12-2022 13:16:59

istitle()	Checks for Titlecased String
isupper()	returns if all characters are uppercase characters
join()	Returns a Concatenated String
ljust()	returns left- ustifi d string of given width
lower()	returns lowercased string
lstrip()	Removes Leading Characters
maketrans()	returns a translation table
partition()	Returns a Tuple
replace()	Replaces Substring Inside
rfind()	Returns the Highest Index of Substring
rindex()	Returns Highest Index of Substring
rjust()	returns right-justifi d string of given width
rpartition()	Returns a Tuple
rsplit()	Splits String From Right
rstrip()	Removes Trailing Characters
slice()	creates a slice object specifi d by range()
split()	Splits String from Left
splitlines()	Splits String at Line Boundaries
startswith()	Checks if String Starts with the Specifi d String
strip()	Removes Both Leading and Trailing Characters
swapcase()	swap uppercase characters to lowercase; vice versa
title()	Returns a Title Cased String
translate()	returns mapped charactered string
upper()	returns uppercased string
zfill()	Returns a Copy of The String Padded With Zeros

III. List Functions

Function	Description
append()	Add Single Element to The List
clear()	Removes all Items from the List
copy()	Returns Shallow Copy of a List
count()	returns occurrences of element in a list
extend()	Add Elements of a List to Another List
index()	returns smallest index of element in list
insert()	Inserts Element to The List
list() Function	creates list in Python
pop()	Removes Element at Given Index
remove()	Removes Element from the List
reverse()	Reverses a List
slice()	creates a slice object specifi d by range()
sort()	sorts elements of a list

12th Std - CS EM Annexure Pages.indd 329 23-12-2022 13:16:59

IV. Tuple Functions

Function	Description
count()	returns occurrences of element in a tuple
index()	returns smallest index of element in tuple
slice()	creates a slice object specifi d by range()
tuple() Function	Creates a Tuple
zip()	Returns an Iterator of Tuples

V. Set Functions

Function	Description
add()	adds element to a set
clear()	remove all elements from a set
copy()	Returns Shallow Copy of a Set
difference()	Returns Difference of Two Sets
difference_update()	Updates Calling Set With Intersection of Sets
discard()	Removes an Element from The Set
frozenset()	returns immutable frozenset object
intersection()	Returns Intersection of Two or More Sets
intersection_update()	Updates Calling Set With Intersection of Sets
isdisjoint()	Checks Disjoint Sets
issubset()	Checks if a Set is Subset of Another Set
issuperset()	Checks if a Set is Superset of Another Set
pop()	Removes an Arbitrary Element
remove()	Removes Element from the Set
set()	returns a Python set
symmetric_difference()	Returns Symmetric Difference
symmetric_difference_update()	Updates Set With Symmetric Difference
union()	Returns Union of Sets
update()	Add Elements to The Set.

VI. Dictionary Functions

Function	Description
clear()	Removes all Items
copy()	Returns Shallow Copy of a Dictionary
dict()	Creates a Dictionary
fromkeys()	creates dictionary from given sequence
get()	Returns Value of The Key
items()	returns view of dictionary's (key, value) pair
keys()	Returns View Object of All Keys
pop()	removes and returns element having given key
popitem()	Returns & Removes Element From Dictionary
setdefault()	Inserts Key With a Value if Key is not Present
update()	Updates the Dictionary
values()	returns view of all values in dictionary

12th Std - CS EM Annexure Pages.indd 330 23-12-2022 13:16:59

عملی مشقیں

PRACTICAL EXERCISES

(پریکٹکل)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:31 PM

Practical Guide پریکٹکل کے لئے رہ نما

ہدایات:

1. پریکٹکل کلاسوں میں پئی تھان سے آٹھ مشقیں اور MySQL میں سے دو مشقوں کو کرنا چاہئے۔
2. اندرونی اختیار کی بنیاد پر پئی تھان میں سے ایک سوال لیا جانا چاہئے۔
3. مارکس کی تقسیم۔

I۔ اندرونی تشخیص	5 مارکس
ریکارڈ بک کے لئے	5 مارکس
II۔ بیرونی تشخیص	**5 مارکس**
(a) پئی تھان پروگرامنگ کوڈنگ	10 مارکس
(b) ایگزیکیوشن اور آؤٹ پٹ	5 مارکس

جملہ	20 مارکس

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:31 PM

INDEX فہرست

صفحہ نمبر Page Number	پروگرام کا نام Program Name	سوال نمبر Question Number	شمار عدد S No
334	(a) Calculate Factorial (a) فیکٹوریل کو محسوب کرنا (b) Sum of Series (b) سلسلوں کا حاصل جمع	PY1	1
335	(a) Odd or Even (a) طاق ہے یا جفت (b) Reverse the String (b) اسٹرنگ کو اُلٹنا	PY2	2
336	قیمتیں حاصل کرنا اور طاق اعداد کو نکالنا Generate values and remove odd numbers	PY3	3
337	اعداد اولی کو حاصل کرنا اور ان پر اعمال متعین کرنا Generate prime numbers and set operations	PY4	4
338	کلاس کے ذریعہ اسٹرنگ کے عناصر کو ظاہر کرنا Display a string elements - using class	PY5	5
340	MySQL - ملازم کی جدول MySQL - Employee table	DB6	6
344	MySQL - طلباء کے لئے جدول MySQL - Student table	DB7	7
349	CSV کے ساتھ پئی تھان Python with CSV	PY8	8
351	SQL کے ساتھ پئی تھان Python with SQL	PY9	9
353	PIP کے ساتھ پئی تھان گرافکس Python Graphics with PIP	PY10	10

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:31 PM

PY1(a) – Calculate Factorial PY1(a) فیکٹوریل محسوب کرنا

1(a) for loop کے استعمال سے دئے گئے عدد کے لئے فیکٹوریل محسوب کرنے کے لئے ایک پروگرام لکھئے۔
(built-in function factorial کا استعمال مت کیجئے)

مقصد for loop کے استعمال سے دئے گئے عدد کے لئے فیکٹوریل محسوب کرنا

کوڈنگ

```
num = int(input('Enter a Number: '))
fact = 1
for i in range(1,num+1):
        fact = fact * i
print("Factorial of ", num, " is ", fact)
```

اوٹ پٹ :

```
Enter a Number: 5
Factorial of  5  is  120
```

نتیجہ

فیکٹوریل محسوب کرنے پئی تھان پروگرام استعمال کیا گیا اور نتیجہ کی جانچ کی گئی۔

PY1 (b) - Sum of Series PY1 (b) سلسلہ کا حاصل جمع

1(b) سلسلہ $1^1/1 + 2^2/2 + 3^3/3 + \ldots\ldots n^n/n$ کے حاصل جمع کے لئے ایک پروگرام لکھئے۔

مقصد سلسلہ $1^1/1 + 2^2/2 + 3^3/3 + \ldots\ldots n^n/n$ کا حاصل جمع محسوب کرنا۔

کوڈنگ

```
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)
```

اوٹ پٹ :

ایک قیمت n : 4 درج کیجئے۔

نتیجہ

اس سلسلہ کا حاصل جمع 76 ہے۔

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:31 PM

PY2(a) – Odd or Even PY2 (a) طاق ہے یا جفت

2(a) تفاعل کے استعمال سے یہ جانچئے کہ دیا گیا عدد طاق ہے یا جفت۔

مقصد واضح تفاعل کے استعمال سے یہ جانچنا کہ دیا گیا عدد طاق ہے یا جفت .

کوڈنگ

```
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))
```

آؤٹ پٹ :

```
Enter a number: 7
The given number is Odd
Enter a number: 6
The given number is Even
```

نتیجہ

پائتھان پروگرام کے ذریعہ یہ جانچ کی گئی کہ دیا گیا عدد طاق ہے یا جفت اور اس کے آؤٹ پٹ کی بھی تصدیق کی گئی۔

PY2(b) – Reverse the string اسٹرنگ کو اُلٹنا

2(b) دئے گئے اسٹرنگ کو اُلٹا کرنے کے لئے ایک پروگرام لکھئے۔ مثال کے طور پر (stride operation کے دوران string slice کا استعمال مت کیجئے۔

مقصد دئے گئے اسٹرنگ کو اُلٹنا

کوڈنگ

```
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))
```


12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:31 PM

اوٹ پٹ :

```
Enter a String: school
The reverse of the given string is:  loohcs
```

نتیجہ

پئی تھان پروگرام کے ذریعہ دی گئی اسٹرنگ کو اُلٹا گیا اور اس کے اوٹ پٹ کی بھی تصدیق کی گئی۔

PY 3 - قیمتیں حاصل کرنا اور طاق اعداد کو خارج کرنا PY3 – Generate values and remove odd numbers

3 1 سے 10 تک کے اعداد حاصل کرنا اور اس فہرست میں سے تمام طاق اعداد کو الگ کرنے کے لئے پروگرام لکھئے۔

مقصد 1 سے 10 تک کے اعداد حاصل کرنا اور اس فہرست میں سے تمام طاق اعداد کو الگ کرنا۔

کوڈنگ

```
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)
```

اوٹ پٹ :

```
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]
```

نتیجہ

پئی تھان پروگرام کے ذریعہ 1 سے 10 تک کے اعداد حاصل کئے گئے اور اس فہرست میں سے تمام طاق اعداد کو الگ کیا گیا اور نتیجہ کی تصدیق کی گئی۔

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:32 PM

4. ایک پروگرام لکھئے جس میں اعدادِاولیٰ کا ایک سٹ اور طاق اعداد کا ایک سٹ حاصل ہو۔ ان کے درمیان اتحاد، تقاطع، فرق، اورتشاکلی فرق کے اعمال بتایئے۔

مقصد

اعدادِاولیٰ کا ایک سٹ اور طاق اعداد کا ایک سٹ حاصل کرنا۔ ان کے درمیان اتحاد، تقاطع، فرق، اورتشاکلی فرق کے اعمال کرنا۔

کوڈنگ

```
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))
```

اوٹ پٹ:

```
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}
```

نتیجہ

پئی تھان پروگرام کے ذریعہ اعدادِاولیٰ کا ایک سٹ اور طاق اعداد کا ایک سٹ حاصل کیا گیا اور ان کے درمیان اتحاد، تقاطع، فرق، اورتشاکلی فرق کے اعمال کرکے ان کے نتیجہ کی تصدیق کی گئی۔

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:32 PM

5. کلاس کو استعمال کر کے دی گئی اسٹرنگ میں موجود جلی حروف (upper case)، خفی حروف (lower case)، حروف علت (Vowels)، صحیح حروف (consonants) اور خالی جگہوں (spaces) کی تعداد کو ایک اسٹرنگ کر کے print کرنے کے لئے ایک پروگرام لکھنا۔

مقصد

کلاس کو استعمال کر کے دی گئی اسٹرنگ میں موجود جلی حروف (upper case)، خفی حروف (lower case)، حروف علت (Vowels)، صحیح حروف (consonants) اور خالی جگہوں (spaces) کی تعداد کو ایک اسٹرنگ کے ذریعہ قبول کرنا۔

کوڈنگ

```
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)
```


12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:32 PM

```
S = String()
S.getstr()
S.execute()
S.display()
```

آؤٹ پٹ :

```
Enter a String: Welcome to Computer Science
The given string contains...
3 Uppercase letters
21 Lowercase letters
10 Vowels
14 Consonants
3 Spaces
```

نتیجہ

پئی تھان پروگرام کے ذریعہ کلاس کو استعمال کر کے ایک اسٹرنگ کو ظاہر کیا گیا اور نتیجہ کی تصدیق کی گئی۔

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:32 PM

6. Empno, Empname, Desig, Dept, Age اور place جیسے fields کواستعمال کرکے مزدوروں کی ایک جدول بنایئے۔
جدول میں مزید دو اندراجوں کو شامل کیجئے۔
date of joining نامی ایک field کو شامل کرکے جدول کی ساخت کو تبدیل کیجئے۔
کسی بھی رکارڈ میں doj میں Null value کی جانچ کیجئے۔
01/01/2018 کے بعد شامل ہونے والے مزدوروں کی فہرست بنایئے۔

مقصد

(1) Empno, Empname, Desig, Dept, Age اور place جیسے fields کواستعمال کرکے مزدوروں کی ایک جدول بنانا۔
(2) پانچ رِکارڈوں کو داخل کرنا۔
(3) مزید دو رِکارڈوں کو شامل کیجئے۔
(4) جدول کی ساخت کو تبدیل کیجئے۔
(5) NULL value کی جانچ کیجئے۔
(6) رِکارڈوں کی درکار subset فہرست بنایئے۔

SQL کے سوالات اور اوٹ پٹ

(i) مزدوروں کے لئے جدول بنانا۔

```
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) جدول کی ساخت کو دیکھنا۔

```
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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:32 PM

(iii) Inserting Data into Table: (iii) ڈیٹا کو جدول میں داخل کرنا۔

```
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: (iv) تمام رکارڈوں کا انتخاب کرنا۔

```
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: (v) مزید دو رِکارڈوں کو شامل کرنا۔

```
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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:32 PM

(vi) Adding one more Field: (vi) ایک اور ریکارڈ کو شامل کرنا۔

```
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: (vii) ہر ایک مزدور کے ساتھ **date of joining** کو داخل کرنا۔

```
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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

(viii) Checking null value in doj (viii) doj میں null value کی جانچ کرنا۔

```
mysql>  select * from emp where empno is null;
```

Empty set (0.00 sec)

(ix) 01/01/2018 کے بعد شامل ہونے والے مزدوروں کی فہرست بنانا۔

(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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

7 درج ذیل fields کو شامل کر کے ایک جدول بنائیے اور نیچے کی جدول میں دئے گئے ڈیٹا کو داخل کیجئے۔

Field Name	Type	Size
Reg_No	char	5
Sname	varchar	15
Age	int	2
Dept	varchar	10
Class	char	3

Data to be entered : داخل کیا جانے والا ڈیٹا

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 : درج ذیل کے سوالات (query) پوچھئے۔

(i) "Computer Science" کے شعبہ کے تمام طالب علموں کی فہرست بنائیے۔

(ii) mechanical department کے طلباء اور 20 سال کی عمر سے زیادہ طلباء کی فہرست بنائیے۔

(iii) طلباء کی شعبہ کے مطابق فہرست بنائیے۔

(iv) class M2 کو class M1 میں ترمیم کیجئے۔

(v) رجسٹر نمبر کے عجیب خصوصیت کو ظاہر کیجئے۔

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

(1) Creating Table - Student (1) جدول کو بنانا - طالب علم

```
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: جدول کی ساخت کو دیکھنا

```
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: (2) جدول میں ڈیٹا کو داخل کرنا

```
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 : تمام رکارڈوں کو دیکھنا

```
mysql>  select * from Student;
```


12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

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:

(3) دیگرسوالات

(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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

(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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

(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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

8 10 کھلاڑیوں کے نام اوران کے اسکور کے لئے ایک پروگرام لکھئے۔ ایک csv فائل میں input لکھئے۔ پئی تھان کو استعمال کر کے مزید ایک player کے نام کو شامل کیجئے۔ csv فائل کو پڑھئے اور اس کے نام اور اسکور کو بتایئے۔ اگر کھلاڑی کا نام موجود نہ ہو تو مناسب message دیجئے۔

مقصد

10 کھلاڑی اوران کے اسکور کے نام حاصل کرنے کے لئے ایک پروگرام لکھنا۔ ایک csv file کے لئے input لکھنا۔ پئی تھان استعمال کر کے ایک کھلاڑی کا نام شامل کرنا۔ اگر کھلاڑی کا نام موجود نہ ہو تو مناسب message دینا۔

کوڈنگ

```
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:
```


12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:33 PM

```
        print(row)
        q+=1
        if(q==0):
            print("string not found")
f.close()
f.close()
```

آؤٹ پٹ :

```
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']
```


12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:34 PM

9 پی تھان استعمال کر کے ایک 10 ناموں اور ان کی عمروں کو شامل کیجئے۔ ان کی عمروں کی اترتی ہوئی ترتیب میں ترتیب دیجئے اور ڈسپلے کیجئے۔

مقصد پی تھان استعمال کر کے ایک 10 ناموں اور ان کی عمروں کو شامل کرنا۔ ان کی عمروں کی اترتی ہوئی ترتیب میں ترتیب دیجئے اور ڈسپلے کرنا۔

کوڈنگ

```
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' )
```


12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:34 PM

اوٹ پٹ:

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)

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:34 PM

10 فہرست کے ذریعہ پانچ مارکس حاصل کرنے کے لئے ایک پروگرام لکھئے اوران مارکس کوایک پئی چارٹ میں ڈسپلے کیجئے۔

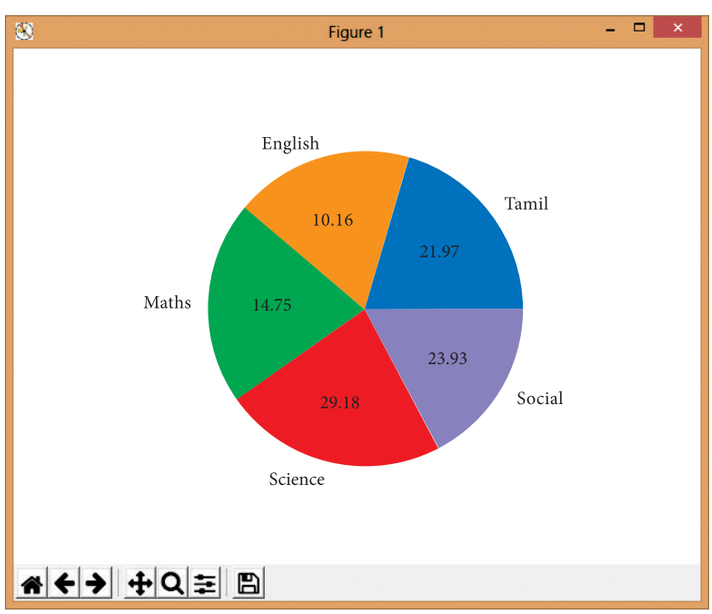
مقصد فہرست کے ذریعہ پانچ مارکس حاصل کرنے کے لئے ایک پروگرام لکھنااوران مارکس کوایک پئی چارٹ میں ڈسپلے کرنا۔

کوڈنگ

```
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()
```

Output:

```
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
```



12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:34 PM

Higher Secondary Second Year
Computer Science
(Urdu Medium)
List of Translators and Reviewers

Academic Head and Chief Co-ordinator:

Dr. Mrs. S. Shameem,
Deputy Director, SCERT, Chennai.

Translation Team :

Dr. Md. Nayeemur Rahman, Vice Chairman
Tamil Nadu State Urdu Academy
Dr. Mohd. Tayyab Ali, Head, Dept. of Urdu
The New College, Chennai-600 014.
Syed Shabbeer Hussain, Asst. Prof. of Urdu
The New College, Chennai-600 014.
Sajid Hussain,Asst. Prof. of Urdu
The New College, Chennai-600 014.
Rehana Yasmin, Former PG Asst. (Comp.Sc.)
Anjuman Matriculation HSS, Chennai-600 017
Reshma Begam, B.T. Asst.,
MWA Matriculation HSS, Chennai-600 086.
Ayesha Tasneem, Headmistress,
New Hope Public School, Yadgir, Karnataka.
Mahmooda Tamkeen, Former Asst. Professor,
AMS College of Engg., Avadi, Chennai-600 055.

Reviewers

P. Baseer Ahmed,
P. G. Assistant, (computer Science)
Islamiah Boys' Hr. Sec. School, Vaniyambadi, Tirupattur Dist.
.Mohamed Jaweed Akram,
P.G. Asst., Islamiah Boys' Hr. Sec. School, Vaniyambadi,
Tirupattur District.
Mohamed Irshad Khan .J
B.T. Asst., Islamiah Boys' Hr. Sec. School, Vaniyambadi,
Tirupattur District.

Co-ordinators :

Dr. R. Anbazhagan
Principal, DIET, Ranipet District.
M. Suresh Kumar
Lecturer, DIET Ranipet
A. Sivakumar
Lecturer, DIET Ranipet

Urdu Layout

TAJ Printers, Vaniyambadi

In-House QC

Yogesh B
Rajesh Thangappan

Wrapper Design

Kathir Arumugam

Coordination

Ramesh Munisamy

Ths book has been printed on 80 GSM
Elegant Maplitho paper

Printed by offset at:

ہائرسکنڈری سالِ دوم
کمپیوٹر سائنس

تعلیمی سربراہ اور معاون

ڈاکٹر مسز یس۔شمیم
ڈپٹی ڈائرکٹر، یس.ٹی.ای.آر.ٹی، چنئی۔

مترجمین

ڈاکٹر محمد نعیم الرحمٰن، وائس چیرمین
تمل ناڈو اسٹیٹ اردو اکیڈمی۔
ڈاکٹر محمد طیب علی، صدر شعبہ اردو،
دی نیو کالج، چنئی-600 014۔
سید شبیر حسین، پروفیسر آف اردو
دی نیو کالج، چنئی-600 014۔
ساجد حسین، پروفیسر آف اردو
دی نیو کالج، چنئی-600 014۔
ریحانہ یاسمین، سابق پی.جی اسسٹنٹ
انجمن مٹریکولیشن ہائرسکنڈری اسکول، چنئی-17۔
ریشمہ بیگم، بی ٹی اسسٹنٹ،
MWA مٹریکولیشن ہائرسکنڈری اسکول، چنئی-86۔
عائشہ تسنیم، میر مدرس،
نیو ہوپ پبلک اسکول، یادگیر، کرناٹک،
محمودہ تمکین، سابق اسسٹنٹ پروفیسر،
عالم محمد صالح کالج آف انجینئرنگ، آوڈی، چنئی-55۔

نظر ثانی

پی۔بصیر احمد
پی۔جی۔اسسٹنٹ، شعبہ کمپیوٹر سائنس، اسلامیہ بائس ہائرسکنڈری اسکول، وانم باڑی۔
محمد جاوید اکرم
پی.جی.اسسٹنٹ، اسلامیہ بائس ہائرسکنڈری اسکول، وانم باڑی۔
یم۔ارشاد خان
بی۔ٹی۔اسسٹنٹ، اسلامیہ بائس ہائرسکنڈری اسکول، وانم باڑی۔

معاونین

ڈاکٹر آر. انبوگن
پرنسپال، DIET، رانی پیٹ
یم.بریش کمار
اسسٹنٹ پروفیسر، DIET، رانی پیٹ
اے۔ سوا کمار
اسسٹنٹ پروفیسر، DIET، رانی پیٹ

ترتیب وتزئین

تاج پرنٹرس، وانم باڑی۔

12 CS Unit 18 practical
Tuesday, April 18, 2023 4:42:35 PM