DIRECTORATE OF EDUCATION
Govt. of NCT, Delhi

SUPPORT MATERIAL
(2025-2026)

COMPUTER SCIENCE
CLASS: XII

Under the Guidance of

Sh. Pandurang K. Pole
Secretary (Education)

Ms. Veditha Reddy
Director (Education)

Dr. Rita Sharma
Addl. DE (School & Exam.)

Coordinators

Mr. Parvinder Kumar	Mrs. Ritu Singhal	Mr. Krishan Kumar	Mr. Tushar Saluja
DDE (Exam)	OSD (Exam)	OSD (Exam)	OSD (Exam)

Production Team

Published at Delhi Bureau of Text Books, 25/2, Institutional Area, Pankha Road, New Delhi-58 by **BiJender Kumar**, Secretary, Delhi Bureau of Text Books and **Printed at:** Palak printers, 6, Mohkampur Phase-II, Delhi Road, Meerut -250004 (UP.)

पांडुरंग के. पोले, भा.प्र.से
सचिव (शिक्षा)

PANDURANG K. POLE, IAS
SECRETARY (Education)

राष्ट्रीय राजधानी क्षेत्र, दिल्ली सरकार
पुराना सचिवालय, दिल्ली–110054
दूरभाष: 011–23890187, 23890119
Government of National Capital Territory of Delhi
Old Secretariat, Delhi-110054
Phone: 23890187, 23890119
E-mail : sccyedu@nic.in

D.O. NO. : DE.5/228/Exam/Message/S.M/248/249
Date : 07/11/2025

MESSAGE

The Directorate of Education remains steadfast in its vision to achieve excellence in the academic domain and its commitment to develop meaningful, engaging, and child-friendly learning content.

Each year, the Directorate carefully reviews and updates the Support Material to ensure alignment with the latest CBSE guidelines and emerging academic developments.

The Support Material provides comprehensive academic support through well-structured practice questions and exercises that strengthen conceptual understanding and exam readiness and aims to nurture students' critical thinking, analytical abilities, and problem-solving skills. Through such sustained efforts, the Directorate of Education continues to guide students towards academic excellence and holistic growth.

This Support Material is intended to bridge classroom learning and examination preparation, enabling students to consolidate knowledge through systematic practice. It has been thoughtfully designed for students, with the belief that its effective use will strengthen their understanding and support them in achieving their learning goals with confidence.

I appreciates the dedication and collaborative effort of all those involved in the development of this material and extends my best wishes to all students —may this Support Material serve as an essential academic aid, enhancing students' confidence and preparedness for examinations.

Best Wishesh.

(Pandurang K. Pole)

VEDITHA REDDY, IAS
Director, Education & Sports

सत्यमेव जयते

Directorate of Education
Govt. of NCT of Delhi
Room No. 12, Old Secretariat
Near Vidhan Sabha,
Delhi-110054
Ph.: 011-23890172
E-mail :diredu@nic.in

DE-5/228/Exam/Message/S-M/2018/402
dated - 09/05/25

MESSAGE

Education is the cornerstone of a progressive society, and providing students with the right learning resources is essential for their academic and personal growth. Keeping this in mind, the Directorate of Education, GNCT of Delhi, develops comprehensive Support Material every year for various subjects of Classes IX to XII.

The support material serves as an additional study resource to supplement textbooks by offering clear and easy-to-understand explanation of complex topics. Our dedicated team of expert faculty members has meticulously reviewed and updated this material, aligning it with the latest CBSE syllabus, question paper pattern and assessment guidelines. Our effort is to simplify difficult concepts and make them more accessible to students, helping them save time and effort with ready references for effective preparation.

As Ruskin Bond beautifully said, "Education must inspire the spirit of inquiry, Creativity and Joy" True learning goes beyond memorisation-it encourages curiosity, fosters creativity, and makes the learning process meaningful and enjoyable.

In alignment with the vision of NEP 2020, the CBSE framework now places emphases on competency-based assessments for 50% of the evaluation, highlighting the need for students to develop critical thinking and problem-solving skills. The Support Material is designed to help students analyse concepts deeply, think innovatively, and apply their knowledge affectively, ensuring they are well-prepared not only for exams but also for real-life challenges.

I appreciate the dedicated efforts of the entire team of subject experts in developing this valuable learning resource. I am confident that both teachers and students will make the best use of these material to enhance learning and academic success.

Wishing all students great success in their exam and a bright, fulfilling future ahead.

Veditha

(VEDITHA REDDY, IAS)

Dr. RITA SHARMA
Additional Director of Education
(School/Exam)

सत्यमेव जयते

Govt. of NCT of Delhi
Directorate of Education
Old Secretariat, Delhi-110054
Ph.: 23890185

D.O. No. DE.5/228/Exam/Message/SM/2018/570
Dated: .. *02/07/2025*

MESSAGE

"Children are not things to be molded, but are people to be unfolded." - Jess Lair

In line with this insightful quote, the Directorate of Education, Delhi, has always made persistent efforts to nurture and unfold the inherent potential within each student. This support material is a testimony to this commitment.

The support material serves as a comprehensive tool to facilitate a deeper understanding of the curriculum. It is crafted to help students not only grasp essential concepts but also apply them effectively in their examinations. We believe that the thoughtful and intelligent utilization of these resources will significantly enhance the learning experience and academic performance of our students.

Our expert faculty members have dedicated themselves to the support material to reflect the latest CBSE guidelines and changes. This continuous effort aims to empower students with innovative approaches, fostering their problem-solving skills and critical thinking abilities.

I extend my heartfelt congratulations to the entire team for their invaluable contribution to creating a highly beneficial and practical support material. Their commitment to excellence ensures that our students are well-prepared to meet the challenges of the CBSE examinations and beyond.

Wishing you all success and fulfilment in your educational journey.

(Dr. Rita Sharma)

भारत का संविधान

उद्देशिका

हम, भारत के लोग, भारत को एक [1]**[संपूर्ण प्रभुत्व-संपन्न समाजवादी पंथनिरपेक्ष लोकतंत्रात्मक गणराज्य]** बनाने के लिए, तथा उसके समस्त नागरिकों को :

सामाजिक, आर्थिक और राजनैतिक **न्याय,**
विचार, अभिव्यक्ति, विश्वास, धर्म
और उपासना की **स्वतंत्रता,**
प्रतिष्ठा और अवसर की **समता**
प्राप्त कराने के लिए,
तथा उन **सब में**
व्यक्ति की गरिमा और [2]**[राष्ट्र की एकता
और अखंडता]** सुनिश्चित करने वाली **बंधुता**
बढ़ाने के लिए

दृढ़संकल्प होकर **अपनी इस संविधान सभा में** आज तारीख 26 नवंबर, 1949 ई. को **एतद्द्वारा इस संविधान को अंगीकृत, अधिनियमित और आत्मार्पित करते हैं।**

[1] संविधान (बयालीसवां संशोधन) अधिनियम, 1976 की धारा 2 द्वारा (3.1.1977 से) ''प्रभुत्व-संपन्न लोकतंत्रात्मक गणराज्य'' के स्थान पर प्रतिस्थापित।

[2] संविधान (बयालीसवां संशोधन) अधिनियम, 1976 की धारा 2 द्वारा (3.1.1977 से) ''राष्ट्र की एकता'' के स्थान पर प्रतिस्थापित।

THE CONSTITUTION OF INDIA

PREAMBLE

WE, THE PEOPLE OF INDIA, having solemnly resolved to constitute India into a [1]**[SOVEREIGN SOCIALIST SECULAR DEMOCRATIC REPUBLIC]** and to secure to all its citizens :

JUSTICE, social, economic and political;

LIBERTY of thought, expression, belief, faith and worship;

EQUALITY of status and of opportunity; and to promote among them all

FRATERNITY assuring the dignity of the individual and the [2][unity and integrity of the Nation];

IN OUR CONSTITUENT ASSEMBLY this twenty-sixth day of November, 1949 do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

[1] Subs. by the Constitution (Forty-second Amendment) Act, 1976, Sec.2, for "Sovereign Democratic Republic" (w.e.f. 3.1.1977)

[2] Subs. by the Constitution (Forty-second Amendment) Act, 1976, Sec.2, for "Unity of the Nation" (w.e.f. 3.1.1977)

भारत का संविधान

भाग 4क

नागरिकों के मूल कर्तव्य

अनुच्छेद 51 क

मूल कर्तव्य – भारत के प्रत्येक नागरिक का यह कर्तव्य होगा कि वह –

(क) संविधान का पालन करे और उसके आदर्शों, संस्थाओं, राष्ट्रध्वज और राष्ट्रगान का आदर करे;

(ख) स्वतंत्रता के लिए हमारे राष्ट्रीय आंदोलन को प्रेरित करने वाले उच्च आदर्शों को हृदय में संजोए रखे और उनका पालन करे;

(ग) भारत की संप्रभुता, एकता और अखंडता की रक्षा करे और उसे अक्षुण्ण बनाए रखे;

(घ) देश की रक्षा करे और आह्वान किए जाने पर राष्ट्र की सेवा करे;

(ङ) भारत के सभी लोगों में समरसता और समान भ्रातृत्व की भावना का निर्माण करे जो धर्म, भाषा और प्रदेश या वर्ग पर आधारित सभी भेदभावों से परे हो, ऐसी प्रथाओं का त्याग करे जो महिलाओं के सम्मान के विरुद्ध हों;

(च) हमारी सामासिक संस्कृति की गौरवशाली परंपरा का महत्त्व समझे और उसका परिरक्षण करे;

(छ) प्राकृतिक पर्यावरण की, जिसके अंतर्गत वन, झील, नदी और वन्य जीव हैं, रक्षा करे और उसका संवर्धन करे तथा प्राणिमात्र के प्रति दयाभाव रखे;

(ज) वैज्ञानिक दृष्टिकोण, मानववाद और ज्ञानार्जन तथा सुधार की भावना का विकास करे;

(झ) सार्वजनिक संपत्ति को सुरक्षित रखे और हिंसा से दूर रहे;

(ञ) व्यक्तिगत और सामूहिक गतिविधियों के सभी क्षेत्रों में उत्कर्ष की ओर बढ़ने का सतत् प्रयास करे, जिससे राष्ट्र निरंतर बढ़ते हुए प्रयत्न और उपलब्धि की नई ऊँचाइयों को छू सके; और

(ट) यदि माता-पिता या संरक्षक है, छह वर्ष से चौदह वर्ष तक की आयु वाले अपने, यथास्थिति, बालक या प्रतिपाल्य को शिक्षा के अवसर प्रदान करे।

Constitution of India

Part IV A (Article 51 A)

Fundamental Duties

It shall be the duty of every citizen of India —

(a) to abide by the Constitution and respect its ideals and institutions, the National Flag and the National Anthem;

(b) to cherish and follow the noble ideals which inspired our national struggle for freedom;

(c) to uphold and protect the sovereignty, unity and integrity of India;

(d) to defend the country and render national service when called upon to do so;

(e) to promote harmony and the spirit of common brotherhood amongst all the people of India transcending religious, linguistic and regional or sectional diversities; to renounce practices derogatory to the dignity of women;

(f) to value and preserve the rich heritage of our composite culture;

(g) to protect and improve the natural environment including forests, lakes, rivers, wildlife and to have compassion for living creatures;

(h) to develop the scientific temper, humanism and the spirit of inquiry and reform;

(i) to safeguard public property and to abjure violence;

(j) to strive towards excellence in all spheres of individual and collective activity so that the nation constantly rises to higher levels of endeavour and achievement;

*(k) who is a parent or guardian, to provide opportunities for educationto his child or, as the case may be, ward between the age of sixand fourteen years.

Note: The Article 51A containing Fundamental Duties was inserted by the Constitution (42nd Amendment) Act, 1976 (with effect from 3 January 1977).

*(k) was inserted by the Constitution (86th Amendment) Act, 2002 (with effect from 1 April 2010).

List of Group Leader and Subject- Experts for Preparation/ Review of Support Material

Class – XII (2025-26)

Subject: Computer Science

1. Ms. Monika Batra	Principal RPVV SEC 5 Dwarka (1821286)
2. Ms. Preeti Dalal	Lecturer Computer Science, Govt Co-ed SSS, Sec 2 Dwarka (1821029)
3. Ms. Ruby Rana	Lecturer Computer Science, SBV Ashok Nagar (1515002)
4. Mr. Anuj Kumar Dagar	Lecturer Computer Science, G Co.ed SV, Jaffarpur kalan (1822055)
5. Mr. Rohit Ahlawat	Lecturer Computer Science, GBSSS, Hastsal (1618058)
6. Mr. Sunny Dagar	Lecturer Computer Science, GBSSS No 3, Najafgarh (1822057)
7. Mr. Vinay Gahlot	Lecturer Computer Science, RPVV Sector-19, Dwarka (1821285)

Computer Science (2025-26)
CLASS XII Code No. 083

1. Prerequisites

Computer Science- Class XI

2. Learning Outcomes

Student should be able to

a) apply the concept of function.

b) explain and use the concept of file handling.

c) use basic data structure: Stacks

d) explain basics of computer networks.

e) use Database concepts, SQL along with connectivity between Python and SQL.

3. Distribution of Marks:

Unit No.	Unit Name	Marks
1	Computational Thinking and Programming - 2	40
2	Computer Networks	10
3	Database Management	20
	Total	**70**

Unit 1: Computational Thinking and Programming – 2

- Revision of Python topics covered in Class XI.
- Functions: types of function (built-in functions, functions defined in module, user defined functions), creating user defined function, arguments and parameters, default parameters, positional parameters, function returning value(s), flow of execution, scope of a variable (global scope, local scope)
- Exception Handling: Introduction, handling exceptions using try-except-finally blocks
- Introduction to files, types of files (Text file, Binary file, CSV file), relative and absolute paths
- Text file: opening a text file, text file open modes (r, r+, w, w+, a, a+), closing a text file, opening a file using with clause, writing/appending data to a text file using write() and writelines(), reading from a text file using read(), readline() and readlines(), seek and tell methods, manipulation of data in a text file

- Binary file: basic operations on a binary file: open using file open modes (rb, rb+, wb, wb+, ab, ab+), close a binary file, import pickle module, dump() and load() method, read, write/create, search, append and update operations in a binary file
- CSV file: import csv module, open / close csv file, write into a csv file using writer(),writerow(),writerows() and read from a csv file using reader()
- Data Structure: Stack, operations on stack (push & pop), implementation of stack using list.

Completion of the Mid-Term Syllabus till 06/09/2025

- **MID TERM EXAM**

Unit 2: Computer Networks

- Evolution of networking: introduction to computer networks, evolution of networking (ARPANET, NSFNET, INTERNET)
- Data communication terminologies: concept of communication, components of data communication (sender,receiver, message, communication media, protocols), measuring capacity of communication media (bandwidth, data transfer rate), IP address, switching techniques (Circuit switching, Packet switching)
- Transmission media: Wired communication media (Twisted pair cable, Co-axial cable, Fiber-optic cable), Wireless media (Radio waves, Micro waves, Infrared waves)
- Network devices (Modem, Ethernet card, RJ45, Repeater, Hub, Switch, Router, Gateway, WIFI card)
- Network topologies and Network types: types of networks (PAN, LAN, MAN, WAN), networking topologies (Bus, Star, Tree)
- Network protocol: HTTP, FTP, PPP, SMTP, TCP/IP, POP3, HTTPS, TELNET, VoIP
- Introduction to web services: WWW, Hyper Text Markup Language (HTML), Extensible Markup Language (XML), domain names, URL, website, web browser, web servers, web hosting

Unit 3: Database Management

- Database concepts: introduction to database concepts and its need
- Relational data model: relation, attribute, tuple, domain, degree, cardinality, keys (candidate key, primary key, alternate key, foreign key)
- Structured Query Language: introduction, Data Definition Language and Data Manipulation Language, data type (char(n), varchar(n), int, float, date), constraints (not null, unique, primary key), create database, use database, show databases, drop database, show tables, create table, describe table, alter table (add and remove an attribute, add and remove primary key), drop table, insert, delete, select, operators (mathematical, relational and logical), aliasing, distinct clause, where clause, in, between, order by, meaning of null, is null, is not null, like, update command, delete command, aggregate functions (max, min, avg, sum, count), group by, having clause, joins: cartesian product on two tables, equi-join and natural join
- Interface of python with an SQL database: connecting SQL with Python, performing insert, update, delete queries using cursor, display data by using connect(), cursor(), execute(), commit(), fetchone(), fetchall(), rowcount, creating database connectivity applications, use of %s format specifier or format() to perform queries

- **Complete the Annual Syllabus till 06/12/2025**
- **PRE BOARD EXAM**
- **Complete Syllabus will be covered in Annual Exam**

4. Practical

S.No	Unit Name	Marks (Total=30)
1	Lab Test: 1. Python program (60% logic + 20% documentation + 20% code quality)	8
	2. SQL queries (4 queries based on one or two tables)	4
2	Report file: • Minimum 15 Python programs. • SQL Queries – Minimum 5 sets using one table / two tables. • Minimum 4 programs based on Python – SQL connectivity	7
3	Project (using concepts learnt in Classes 11 and 12)	8
4	Viva voce	3

5. Suggested Practical List:

Python Programming

- Read a text file line by line and display each word separated by a #.
- Read a text file and display the number of vowels/consonants/uppercase/lowercase characters in the file.
- Remove all the lines that contain the character 'a' in a file and write it to another file.
- Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.
- Create a binary file with roll number, name and marks. Input a roll number and update the marks.
- Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).
- Write a Python program to implement a stack using list.
- Create a CSV file by entering user-id and password, read and search the password for given userid.

Database Management

- Create a student table and insert data. Implement the following SQL commands on the student table:
    - o ALTER table to add new attributes / modify data type / drop attribute
    - o UPDATE table to modify data
    - o ORDER By to display data in ascending / descending order
    - o DELETE to remove tuple(s)
    - o GROUP BY and find the min, max, sum, count and average
- Similar exercise may be framed for other cases.
- Integrate SQL with Python by importing suitable module.

6. Suggested Reading Material

- NCERT Textbook for COMPUTER SCIENCE (Class XII)
- Support Materials on the CBSE website.

7. Project

The aim of the class project is to create something that is tangible and useful using Python file handling/ Python-SQL connectivity. This should be done in groups of two to three students and should be started by students at least 6 months before the submission deadline. The aim here is to find a real world problem that is worthwhile to solve.

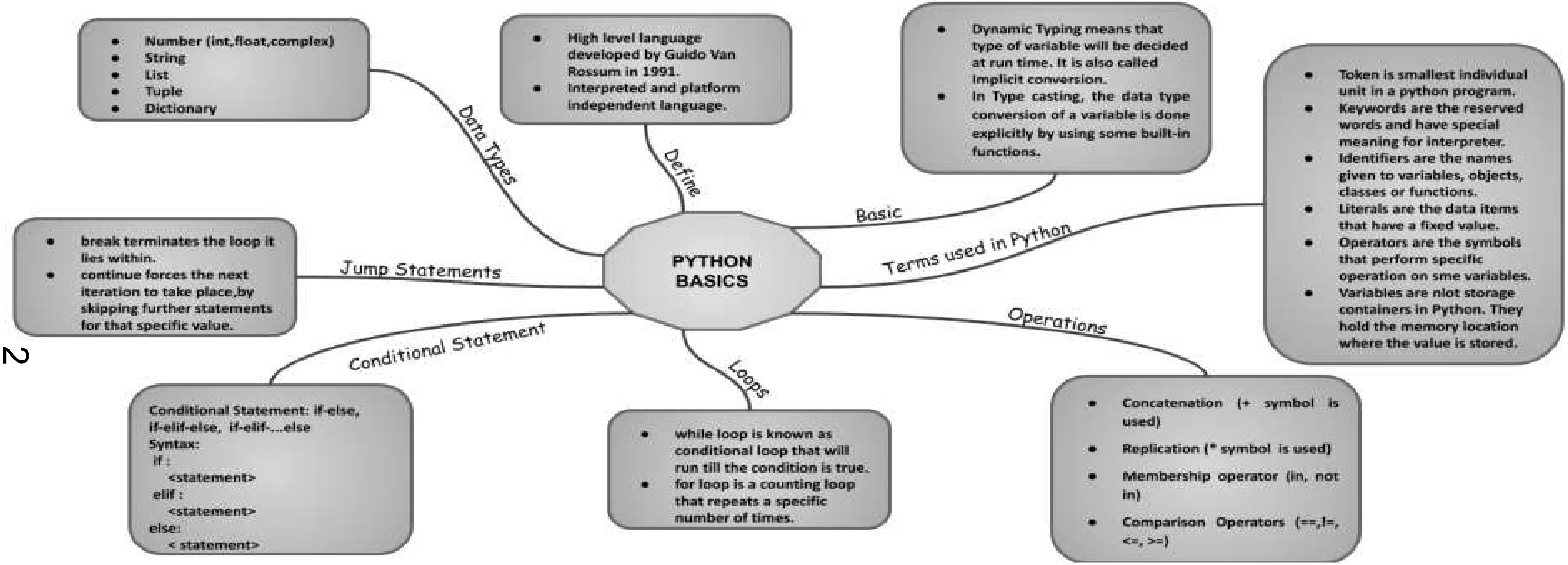
Students are encouraged to visit local businesses and ask them about the problems that they are facing. For example, if a business is finding it hard to create invoices for filing GST claims, then students can do a project that takes the raw data (list of transactions), groups the transactions by category, accounts for the GST tax rates, and creates invoices in the appropriate format. Students can be extremely creative here. They can use a wide variety of Python libraries to create user friendly applications such as games, software for their school, software for their disabled fellow students, and mobile applications, of course to do some of these projects, some additional learning is required; this should be encouraged. Students should know how to teach themselves.

The students should be sensitized to avoid plagiarism and violations of copyright issues while working on projects. Teachers should take necessary measures for this.

Table of Contents

S.No	Chapters	Page No.
1	Python revision tour	1-21
2	Functions	22-48
3	Exception Handing	49-61
4	File Handling	62-101
5	Data Structures	102-113
6	Computer Networks	114-139
7	Database Management system	140-202
8	Practice set 1 (unsolved)	203-210
9	Practice set 2 (unsolved)	211-219
10	Answer Keys	220-248
11	QR Code Of Previous Year Question Papers and Marking Scheme	249-250

PYTHON REVISION TOUR



PYTHON BASICS

Data Types

- Number (int,float,complex)
- String
- List
- Tuple
- Dictionary

Define

- High level language developed by Guido Van Rossum in 1991.
- Interpreted and platform independent language.

Basic

- Dynamic Typing means that type of variable will be decided at run time. It is also called Implicit conversion.
- In Type casting, the data type conversion of a variable is done explicitly by using some built-in functions.

Terms used in Python

- Token is smallest individual unit in a python program.
- Keywords are the reserved words and have special meaning for interpreter.
- Identifiers are the names given to variables, objects, classes or functions.
- Literals are the data items that have a fixed value.
- Operators are the symbols that perform specific operation on sme variables.
- Variables are nlot storage containers in Python. They hold the memory location where the value is stored.

Jump Statements

- break terminates the loop it lies within.
- continue forces the next iteration to take place,by skipping further statements for that specific value.

Conditional Statement

Conditional Statement: if-else, if-elif-else, if-elif-...else
Syntax:

```
 if :
   <statement>
 elif :
   <statement>
else:
   < statement>
```

Loops

- while loop is known as conditional loop that will run till the condition is true.
- for loop is a counting loop that repeats a specific number of times.

Operations

- Concatenation (+ symbol is used)
- Replication (* symbol is used)
- Membership operator (in, not in)
- Comparison Operators (==,!=, <=, >=)

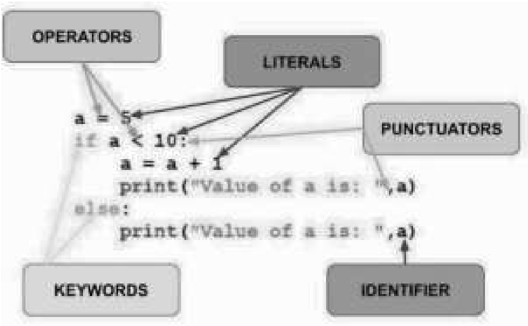
CHAPTER- 1 PYTHON REVISION TOUR

ABOUT PYTHON:

1. Python is a high-level programming language developed by Guido Van Rossum. The language was released in February 1991 and got its name from a BBC comedy series “Monty Python’s Flying Circus”.
2. It is an interpreted and platform-independent language i.e. the same code can run on any operating system.
3. It can be used to follow both procedural and object-oriented approaches to programming.
4. It is free to use and based on two programming languages: ABC language and Modula-3.

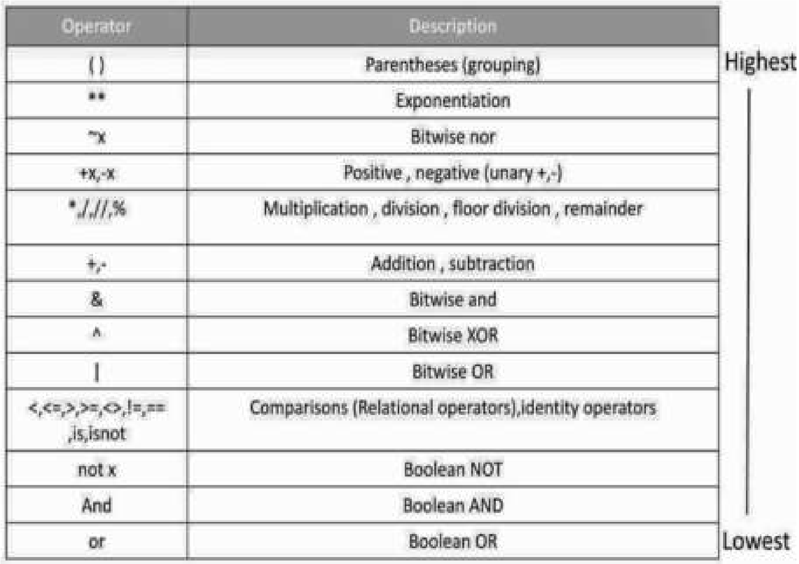
BASIC TERMS USED IN PYTHON:

1. **Token / Lexical Unit:** The smallest individual unit in a Python program is known as Token or Lexical Unit. A token has a specific meaning for a Python interpreter. Examples of tokens are Keywords, Identifiers, Literals, Operators and Punctuators.



```
a = 5
if a < 10:
    a = a + 1
    print("Value of a is: ",a)
else:
    print("Value of a is: ",a)
```

2. **Keywords:** Keywords are reserved words and have special meanings for Python Interpreters. Each keyword can be used only for the purpose to which it has been assigned. Examples: and, while, del, with, True, None, False, return, try, etc.
3. **Identifiers:** These are the names given to variables, objects, classes, functions, etc. There are some predefined rules for forming identifiers that should be followed, or else the program will raise Syntax errors.
4. **Literals:** The data items that have a fixed value are called Literals. If a data item holds numeric values it will be known as Numeric Literal, if it contains String values it will be known as String Literal, and so on. It means the type of value stored by the data item will decide the type of Literal. Python has one special literal ‘None’ to indicate the absence of value.
5. **Operators:** These are the symbols that perform specific operations on some variables. Operators operate on operands. Some operators require two operands and some require only one operand to operate. The operator precedence in Python is as follows:



Operator	Description
( )	Parentheses (grouping)
**	Exponentiation
~x	Bitwise nor
+x,-x	Positive , negative (unary +,-)
*,/,//,%	Multiplication , division , floor division , remainder
+,-	Addition , subtraction
&	Bitwise and
^	Bitwise XOR
|	Bitwise OR
<,<=,>,>=,<>,!=,== ,is,isnot	Comparisons (Relational operators),identity operators
not x	Boolean NOT
And	Boolean AND
or	Boolean OR

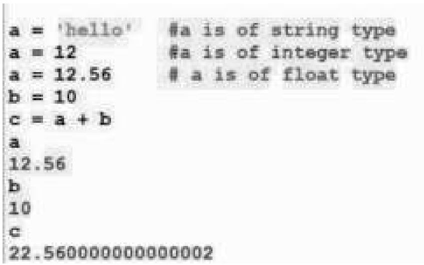
Highest

Lowest

NOTE: When we compare two variables pointing to the same value, then both Equality (==) and identity (is) will return True. But when the same value is assigned to different objects, then the == operator will return True and ‘is’ operator will return False.

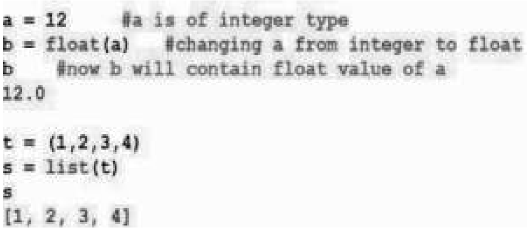
6. **Punctuators:** These are the symbols that are used to organize sentence structure in programming languages. Common punctuators are: ‘ ” # $ @ [] {} = : ; () , .
7. **Variables:** In Python, unlike other programming languages, variables are the temporary memory locations used to store values which will be used in the program further. Each time we assign a new value to a variable it will point to a new memory location where the assigned value is stored. Here, we do not specify the size and type of variable, besides these are decided as per the value we assign to that variable.
8. **Data Type:** It specifies the type of data we will store in the variable according to which memory will be allocated to that variable and it will also specify the type of operations that can be performed on that variable. Examples: integer, string, float, list etc.
9. **Dynamic Typing:** It means that it will be decided at the runtime which type of value the variable will store. It is also called implicit conversion. For example,

   Here, we need not specify the type of value a will store besides we can assign any type of value to a directly. Similarly, the data type of **c** will be decided at run time based on the value of **a** and **b.**



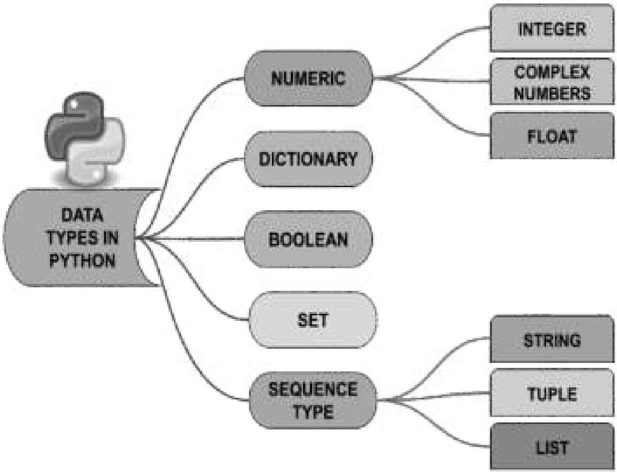
```
a = 'hello'    #a is of string type
a = 12         #a is of integer type
a = 12.56      # a is of float type
b = 10
c = a + b
a
12.56
b
10
c
22.560000000000002
```

10. **Type Casting:** In Type casting, the data type conversion of a variable is done explicitly by using some built-in functions. Here, we can say that we force the variable by applying a built-in function to change the data type and it is not done at run time. Some common type casting functions are int(), float(), str(), list(), tuple(), dict() etc.



```
a = 12     #a is of integer type
b = float(a)    #changing a from integer to float
b    #now b will contain float value of a
12.0

t = (1,2,3,4)
s = list(t)
s
[1, 2, 3, 4]
```

DATA TYPES IN PYTHON:

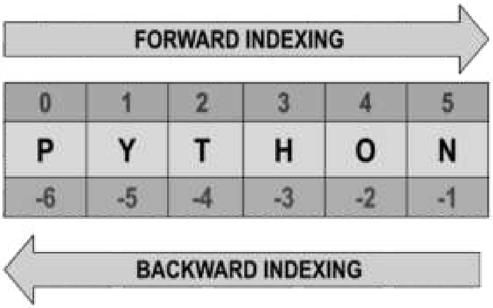


The above data types are classified into two basic categories: Mutable data types and Immutable data types. **Mutable data types** are those data types whose value can be changed without creating a new object. It means mutable data types hold a specific memory location and changes are made directly to that memory location. **Immutable data types** are those data types whose values cannot be changed after they are created. It means that if we make any change in the immutable data object then it will be assigned a new memory location.

1. In **Numeric data types**, Integers allow storing whole numbers only which can be positive or negative. Floating point numbers are used for storing numbers having fractional parts like temperature, area etc. In Python, Floating point numbers represent double precision i.e. 15-digit precision. Complex numbers are stored in python in the form of A + Bj where A is the real part and B is the imaginary part of complex numbers.
2. A **Dictionary** is an unordered set of comma-separated values where each value is a **key: value** pair. We represent the dictionary using curly brackets {}. Keys in the dictionary should be unique and they cannot be changed while the values of the keys can be changed.
3. **Boolean** allows to store only two values True and False where True means 1 and False means 0 internally.
4. **Sequence data types** store a collection or set of values in an ordered manner. We can traverse the values/elements using indexing. The sequence data types are

STRING:

- In Python, a string is a sequence having Unicode characters and is immutable. Any value written/assigned in " " or ' ' is considered as a string in python.
- Each character is at a particular place called Index having a starting value of 0 if traversing forward. Indexing can also be done in a backward direction starting from the last element/character of the string where the starting value will be -1 in the backward direction.
- Joining operation in a string can be done between two strings only. We cannot join a number and a string using '+'. Concatenation operation in a string can be done between a string and a number only using '*'.
- Slicing is defined as extracting a part of a string from the main string using unique index positions starting from 0. In slicing we can specify start, stop and step values to extract a substring from the given string.



0	1	2	3	4	5
P	Y	T	H	O	N
-6	-5	-4	-3	-2	-1

LIST:

- Lists can hold multiple elements of the same or different data types.
- Lists are mutable, which means the values can be updated without assigning a new memory location. It is denoted by square brackets [].
- We can add (join) a list with another list only and not with int, float, or string type. Joining of 2 or more can be done using '+'. We can concatenate a list with an integer only. Concatenation (Replication) can be done in a list using '*'.

TUPLE:

- Tuples can also hold multiple elements of the same or different data types like lists but **tuples are immutable**. These are denoted by round brackets ().
- If multiple values are assigned to a single variable the default type of variable will be tuple.
- Tuples can be traversed in the same way as Strings using a 'for' loop.
- Slicing, Concatenation (Replication) and Joining operations can also be performed on Tuples in the same way as Strings.

WIDELY USED BUILT-IN FUNCTIONS IN PYTHON:

STRING FUNCTIONS:

len (string)	It returns the number of characters in any string including spaces. **e.g. if s = 'hello', then len(s) will result in 5.**
capitalize()	It is used to convert the first letter of sentences into capital letters. **e.g. s = 'python', then s.capitalize() will result in 'Python'.**
title()	It is used to convert the first letter of every word in a string into capital letters. **e.g. s = 'hello python', then s.title() will result in 'Hello Python'.**

upper()	It is used to convert the entire string into capital case letters. **e.g. s = ‘python’, then s.upper() will result in ‘PYTHON’.**
lower()	It is used to convert entire strings into small case letters. **e.g. s = ‘PYTHON’, then s.lower() will result in ‘python’.**
count(substring, [start], [end])	It is used to find the number of occurrences of a substring in a string. We can also specify the starting and ending index to specify a range for searching substring. **e.g. s = ‘he is playing’, then s.count(‘i’) will result in 2.**
find(substring, [start],[end])	This function returns the starting index position of the substring in the given string. Like count(), we can specify the range for searching using the starting and ending index. **It returns -1 if the substring is not found. e.g. s = ‘hello python’, then s.find(‘p’) will return 6 and s.index(‘z’) will return -1**
index(substring)	It returns the starting index position of the substring. If the substring is not found then it will return an error **“Substring not found”. e.g. s = ‘hello python’, then s.index(‘p’) will return 6 and s.index(‘z’) will return ValueError:‘Substring not found’.**
isalnum()	It is used to check if all the elements in the string are alphanumeric or not. It returns either True or False. **e.g. s = ‘abc1’, then s.isalnum() will return True.**
islower()	It returns True if all the elements in the string are in lower case, otherwise returns False. **e.g. if s = ‘python’, then s.islower() will return True.**
isupper()	It returns True if all the elements in the string are in upper case, otherwise returns False. **e.g. if s = ‘PYTHON’, then s.isupper() will return True.**
isspace()	It returns True if all the elements in the string are spaces, otherwise returns False. **e.g. if s = ’ ‘, then s.isspace() will return True.**
isalpha()	It returns True if all the elements in the string are alphabets, otherwise returns False. **e.g. if s = ‘python’, then s.isalpha() will return True.**
isdigit()	It returns True if all the elements in the string are digits, otherwise returns False. **e.g. if s = ‘1234’, then s.isdigit() will return True.**
split([sep])	This function is used to split the string based on the delimiter /separator value which is space by default. It returns a list of n elements where the value of n is based on delimiter. The delimiter is not included in the output. **e.g. if s = ‘hello python world’, then s.split() will return [‘hello’,’ python’, ’world’]**
partition(sep)	It divides the string into three parts: head, separator and tail, based on the **sep** value which acts as a delimiter in this function. It will always return a tuple of 3 elements. The delimiter/separator will be included as the 2nd element of the tuple in the output. **e.g. if s = ‘hello world python’, then s.partition(‘py’) will return (‘hello world ‘, ’py’, ‘thon’)**

replace(old, new)	It is used to replace an old substring inside the string with a new value. **e.g. if s = ‘Good morning’ then s.replace(‘mor’, ’eve’) will return ‘Good evening’.**
strip([chars])	It returns a copy of a string after removing leading and trailing white spaces by default. We can also provide **chars** value if we want to remove characters instead of spaces. If **chars** are given then all possible combinations of given characters will be checked and removed. **e.g. s = ‘good morning good’, then s.strip(‘dg oo’) will return mornin**
lstrip([chars])	It returns a copy of the string after removing leading white spaces. If the **chars** value is given then characters will be removed. **e.g. s = ‘good morning good’, then s.lstrip(‘dgoo’) will return ‘ morning good’**
rstrip([chars])	It returns a copy of a string after removing trailing white spaces. If the **chars** value is given then characters will be removed. **e.g. s = ‘good morning good’, then s.rstrip(‘dgoo’) will return ‘good morning’**

LIST FUNCTIONS:

index()	Used to get the index of the first matched item from the list. It returns the index value of an item to search. If the item is not found, it will return ValueError: n is not in the list. **e.g. x = [10,20,30,40], then x.index(20) will be 1.**
append()	Used to add items to the end of the list. It will add the new item but not return any value. **e.g. x = [10,20,30], then x.append(40) will be [10,20,30,40].**
extend()	Used for adding multiple items. With extend we can add multiple elements but only in the form of a list to any list. Even if we want to add a single element it will be passed as an element of a list. **e.g. x = [10,20,30] then x.extend([40,50]) will be [10,20,30,40,50]**
insert()	Used to add elements to a list at a position of our choice i.e. we can add new elements anywhere in the list. **e.g. x = [10,20,30] then x.insert([2,40]) will be [10,20,40,30].**
pop()	Used to remove an item from a list. It raises an exception if the list is already empty. By default, the last item will be deleted from the list. If an index is provided then the given indexed value will be deleted. **e.g. x = [10,20,30,40,50] then x.pop() will be 50.**

remove()	Used to remove an element when the index is not known and we want to delete it by providing the element value itself. It will remove the first occurrence of a given item from the list and return an error if there is no such item in the list. It will not return any value. **e.g. x = [1,2,3,4,5] then x.remove(4) will be [1,2,3,5].**
clear()	Use to remove all the items of a list at once and the list will become empty. **e.g. x = [10,20,30] then x.clear() will be [ ].**
del	del statement is used to delete the structure of an existing list. **e.g. x = [10,20,30] then del x will delete the structure of x and will return nothing. While trying to access x we will get NameError.**
count()	Used to count the number of occurrences of the item we passed as argument. If an item does not exist in the list, it returns Zero. **e.g. x = [1,2,3,4,2,1,3,4,4] then x.count(4) will return 3.**
reverse()	Used to reverse the items of a list. It made the changes in the original list and did not return anything. **e.g. x = [1,2,3,4] then x.reverse() will be [4,3,2,1]**
sort()	Used to sort the items of a list. It made the changes in the original list and sort the items in increasing order by default. We can specify the reverse argument as True to sort in decreasing order. **e.g. x = [1,5,2,3,7,4] then x.sort() will be [1,2,3,4,5,7].**
sorted()	Used to sort the items of a **sequence data type** and returns a list after sorting in increasing order by default. We can specify the reverse argument as True to sort in decreasing order. **e.g. x = [1,5,2,3,7,4] then sorted(x) will return a sorted list as[1,2,3,4,5,7]**

TUPLE FUNCTIONS:

len()	Returns the number of elements in a tuple. **e.g. x = (1,2,3,4,5,6) then len(x) will be 6.**
max()	Returns the element having a maximum value in the tuple. **e.g. x = (100,30,250,40,110) then max(x) will be 250.**
min()	Returns the element having minimum value **e.g. x = (100,30,250,40,110) then min(x) will be 30.**

index()	Returns index value of given element in the tuple. If the item doesn’t exist, it will raise a ValueError exception. **e.g. x = (10,20,30,40) then x.index(30) will be 2.**
count()	It returns the number of occurrences of the item passed as an argument. If not found, it returns Zero. **e.g. x = (10,20,30,10,10,20) then x.count(20) will be 2.**
sorted()	Used to sort the items of a **sequence data type** and returns a list after sorting in increasing order by default. We can specify the reverse argument as True to sort in decreasing order. **e.g. x = (100,30,250,40,110) then sorted(x) will return a sorted list as [30,40,100,110,250].**

Dictionary Functions:

clear()	Used to remove all items from the dictionary. **e.g. d = {1:‘Delhi’, 2:‘Mumbai’, 3: ‘Kolkata’, 4: ‘Chennai’} then d.clear() will clear all the elements of d and d will be empty {}**.
get()	Used to access the value of a given key, if the key is not found it raises an exception. **e.g. d = {1: ‘Delhi’, 2: ‘Mumbai’, 3: ‘Kolkata’, 4: ‘Chennai’} then d.get(4) will return ‘Chennai’.**
items()	Used to return all the items of a dictionary in the form of tuples.**e.g. d = {1: ‘Delhi’, 2: ‘Mumbai’, 3: ‘Kolkata’, 4: ‘Chennai’} then d.items() will return [(1, Delhi),(2, Mumbai),(3, Kolkata),(4, Chennai)]**
keys()	Used to return all the keys in the dictionary as a sequence of keys. **e.g. d = {1: ‘Delhi’, 2: ‘Mumbai’, 3: ‘Kolkata’, 4: ‘Chennai’} then d.keys() will return [(1,2,3,4)]**
values()	Used to return all the values in the dictionary as a sequence of values. **e.g. d = {1: ‘Delhi’, 2: ‘Mumbai’, 3: ‘Kolkata’, 4: ‘Chennai’} then d.values() will return [(‘Delhi, ’Mumbai, ’Kolkata, ’Chennai’)]**
update()	Merges the key: value pair from the new dictionary into the original dictionary. The key: value pairs will be added to the original dictionary, if any key already exists, the new value will be updated for that key. **e.g. d = {1: ‘Delhi’, 2: ‘Mumbai’} and d1 = {3: ‘Kolkata’, 4: ‘Chennai’} then d.update(d1) will return {1: ‘Delhi’, 2: ‘Mumbai’, 3: ‘Kolkata’, 4: ‘Chennai’}**
fromkeys()	Returns a new dictionary with the given set of elements as the keys of the dictionary. **e.g. x = (1,2,3,4) , y = ‘Delhi’ then d = dict.from keys(x,y) will result in {1: 'Delhi', 2: 'Delhi', 3: 'Delhi', 4: 'Delhi'}**

copy()	It will create a copy of the dictionary. **e.g. x = d.copy()**
popitem()	Used to remove the last added dictionary item (key: value pair) **e.g. d = {1: 'Delhi', 2: 'Mumbai', 3: 'Kolkata', 4: 'Chennai'} then d.popitem() will return d.popitem() will return (4, 'Chennai)**
sorted()	Used to sort the key: value pair of dictionary in either ascending or descending order based on the keys. **e.g. d = {3: 'Kolkata',1: 'Delhi', 2: 'Mumbai', 4: 'Chennai'} then sorted(d) will return [1,2,3,4]**

Statements in Python:

Instructions given to computers to perform any task are called statements. Python, we have 3 types of statements:

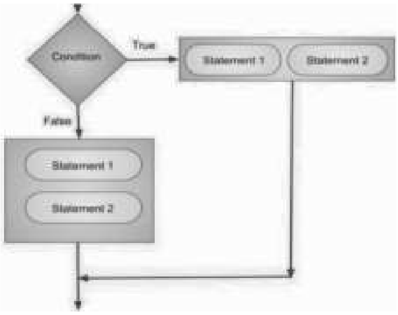
1. **EMPTY STATEMENTS:** When a statement is required as per syntax but we don't want to execute anything or do not want to take any action we use the ***pass*** keyword.
   Whenever a ***pass*** is encountered, the Python interpreter will do nothing and control will move to the next statement in the flow of control.

2. **SIMPLE STATEMENT:** All the single executable statements in Python are Simple Statements.
3. **COMPOUND STATEMENTS:** A group of statements executed as a unit are called compound statements. Compound statements have a Header which begins with a keyword and ends with a colon (:). There can be at least one or more statements in the body of the compound statement, all indented at the same level.

Conditional statements in Python:

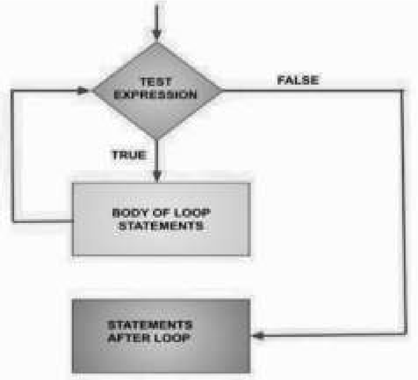
When the execution of any statement depends on some condition then such statements are considered as Conditional Statements. In Python, we use the *if* keyword for Conditional statements.

- It must contain a valid condition which evaluates to either True or False. The condition must be followed by Colon (:), it is mandatory. The statement inside 'if' must be at the same indentation level.
- if statement can be of many forms:
- if without a false statement
- if with else
- if with elif
- Nested if



Iterative statements in Python:

In Python, to perform repetition we have two keywords, *for* (Counting Loop) and *while*
(Conditional Loop)



- *for* loop in Python is used when we know the number of iterations.
- *for* loop is used to create a loop where we are working on sequence data types.
- To repeat the loop **n number of times** in the *for* loop we use the range function in which we can specify the lower limit and upper limit.
- **range function** will generate a set of values from the lower limit to an upper limit. We can also specify the step value which is +1 by default.
- Step value can be in -ve also to generate a set of numbers in reverse orders.
- *while* loop in Python is also called as entry-controlled loop in which entry in loop is allowed only if the condition is true.
- There are 4 main elements of the *while* loop:
- **Initialization:** In the while loop, we cannot use a variable in the test condition without initializing it with a starting value. So, we will specify the starting value of the variable to be used in the loop.
- **Test Condition:** We will specify a test condition in the header of the loop. If this condition is True then only the body of the loop will be executed else the loop will not get executed at all.
- **Body of loop:** We will write the statements to be executed if the test condition will be True.
- **Update Statement:** In the while loop, we have to increase and decrease the value of the variable used in the test condition else the loop will result in an infinite loop.
- **break** keyword is used to take control out of the loop for any given condition. When *break* is encountered in a loop the flow of control jumps to the very next statement after the loop.
- **continue** keyword is used to skip the execution of remaining statements inside the loop and takes control to the next iteration.

PART A (1 Mark questions)

MULTIPLE CHOICE QUESTIONS

1 Find the invalid identifier from the following

a) none
b) address
c) Name
d) pass

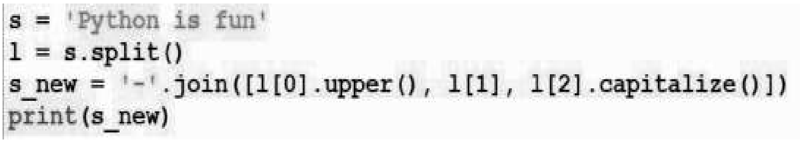
2 Consider a declaration L = (1, 'Python', '3.14').
Which of the following represents the data type of L?

a) List
b) Tuple
c) Dictionary
d) String

3 Identify the valid arithmetic operator in Python from the following.

a) ?
b) <
c) **
d) And

4 Select the correct output of the code:



```
s = 'Python is fun'
l = s.split()
s_new = '-'.join([l[0].upper(), l[1], l[2].capitalize()])
print(s_new)
```

a) PYTHON-IS-Fun
b) PYTHON-is-Fun
c) Python-is-fun
d) PYTHON-Is –Fun

5 State True or False.
"Identifiers are names used to identify a variable or function in a program".

6 One of the following statements will raise an error. Identify the statement that will raise the error.

a) x,y = 20
b) a, b = 6, 7*9
c) a = b = c = 35
d) None of the above will raise an error

7 Given the following Tuple

Tup= (10, 20, 30, 50)

Which of the following statements will result in an error?

a) print (Tup [0])

b) print (Tup [1:2])

c) Tup.insert (2,3)

d) print(len (Tup))

8 Consider the given expression : 5<10 and 12>7 or not 7>4

Which of the following will be the correct output, if the given expression is evaluated?

a) True

b) False

c) NULL

d) NONE

9 Which of the following will give output as [5,14,6] if lst=[1,5,9,14,2,6]?

a) print(lst[0::2])

b) print(lst[1::2])

c) print(lst[1:5:2])

d) print(lst[0:6:2])

10 Consider a tuple tup1 = (10, 15, 25, 30). Identify the statement that will result in an error.

a) print(tup1[2])

b) tup1[2] = 20

c) print(min(tup1))

d) print(len(tup1))

11 Which of these about a dictionary is false?

a) The values of a dictionary can be accessed using keys

b) The keys of a dictionary can be accessed using values

c) Dictionaries aren’t ordered

d) Dictionaries are mutable

12 Which is the correct form of declaration of a dictionary?

a) Day={1:’monday’,2:’tuesday’,3:’wednesday’}

b) Day=(1;’monday’,2;’tuesday’,3;’wednesday’)

c) Day=[1:’monday’,2:’tuesday’,3:’wednesday’]

d) Day={1’monday’,2’tuesday’,3’wednesday’]

13 What will be the output of the following statement:

print(3-2**2**3+99/11)

a) 244

b) 244.0

c) -244.0

d) Error

14 What is the output of the following code:
T=(100); print(T*2)

a) Syntax error

b) (200,)

c) 200

d) (100,100)

15 Identify the output of the following Python statements:
x = [[10.0, 11.0, 12.0],[13.0, 14.0, 15.0]]
y = x[1][2]
print(y)

a) 12.0

b) 13.0

c) 14.0

d) 15.0

16 Select the correct output of the code :
S= "Amrit Mahotsav @ 75"; A=S.partition (" "); print (A)

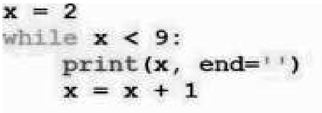
a) ('Amrit Mahotsav', '@', '75')

b) ['Amrit', 'Mahotsav', '@', '75']

c) ('Amrit', 'Mahotsav @ 75')

d) ('Amrit', '', 'Mahotsav @ 75')

17 Identify the output of the following Python statements.



```
x = 2
while x < 9:
    print(x, end='')
    x = x + 1
```

a) 12345678

b) 123456789

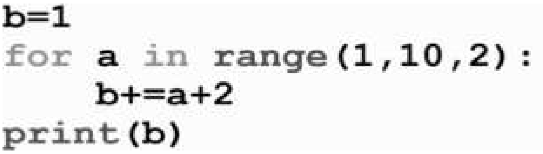
c) 2345678

d) 23456789

18 A tuple is declared as T = (2,5,6,9,8). What will be the value of sum(T)?

a) 27 b) 29 c) 30 d) 32

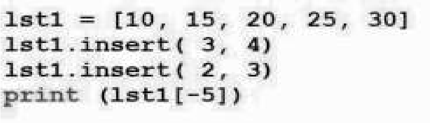
19 Identify the output of the following Python statements.



```
b=1
for a in range(1,10,2):
    b+=a+2
print(b)
```

a) 31
b) 33
c) 36
d) 39

20 Identify the output of the following Python statements.



```
lst1 = [10, 15, 20, 25, 30]
lst1.insert( 3, 4)
lst1.insert( 2, 3)
print (lst1[-5])
```

a) 2
b) 3
c) 4
d) 20

21 Evaluate the following expression and identify the correct answer.

16 - (4 + 2) * 5 + 2**3 * 4

a) 54
b) 46
c) 18
d) 32

22 Fill in the blank.

________________function is used to arrange the elements of a list in ascending order.

a) sort()
b) ascending()
c) arrange()
d) asort()

23 Which of the following will delete the key-value pair for key = "Red" from a dictionary D1?

a) delete D1("Red")
b) del D1["Red"]
c) del.D1["Red"]
d) D1.del["Red"]

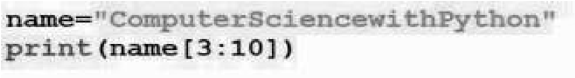
24 Identify the valid declaration of L:
L = [‘Mon’, ‘23’, ‘hello’, ’60.5’]

a) dictionary
b) string
c) tuple
d) list

25 Given a Tuple tup1= (10, 20, 30, 40, 50, 60, 70, 80, 90).
What will be the output of print (tup1 [3:7:2])?

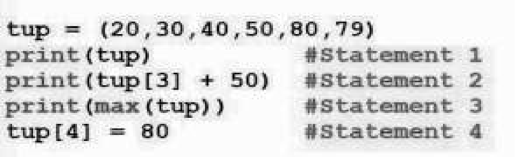
a) (40,50,60,70,80)
b) (40,50,60,70)
c) [40,60]
d) (40,60)

26 If the following code is executed, what will be the output of the following code?



```
name="ComputerSciencewithPython"
print(name[3:10])
```

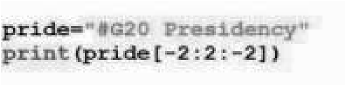
27 Which of the following statement(s) would give an error during the execution of the following code?



```
tup = (20,30,40,50,80,79)
print(tup)              #Statement 1
print(tup[3] + 50)      #Statement 2
print(max(tup))         #Statement 3
tup[4] = 80             #Statement 4
```

a) Statement 1
b) Statement 2
c) Statement 3
d) Statement 4

28 Consider the statements given below and then choose the correct output from the given options:

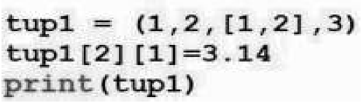


```
pride="#G20 Presidency"
print(pride[-2:2:-2])
```

a) ndsr
b) ceieP0
c) ceieP
d) yndsr

29 Given is a Python list declaration:
names=["Aman", "Ankit", "Ashish", "Rajan", "Rajat"]
What will be the output of:
print (names [-1:-4:-1])

30 What will be the output of the following code?



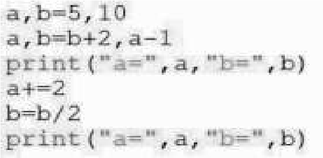
```
tup1 = (1,2,[1,2],3)
tup1[2][1]=3.14
print(tup1)
```

a) (1,2,[3.14,2],3)
b) (1,2,[1,3.14],3)
c) (1,2,[1,2],3.14)
d) Error Message

31 Which of the following statements is/are not Python keywords?

a) False
b) Math
c) WHILE
d) break

32 What will be the result of following the Python code?

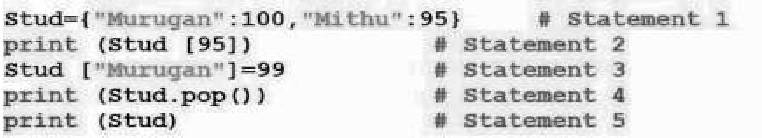


```
a,b=5,10
a,b=b+2,a-1
print("a=",a,"b=",b)
a+=2
b=b/2
print("a=",a,"b=",b)
```

33 For a string S declared as S= ”PYTHON”, Which of the following is incorrect:

a) N=len(s)
b) T=S
c) “T” in S
d) S[0]=”M”

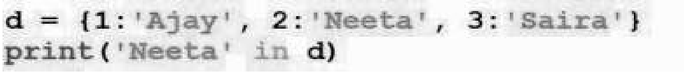
34 Which of the following statement(s) would give an error after executing the following code?



```
Stud={"Murugan":100,"Mithu":95}     # Statement 1
print (Stud [95])                   # Statement 2
Stud ["Murugan"]=99                 # Statement 3
print (Stud.pop())                  # Statement 4
print (Stud)                        # Statement 5
```

a) Statement 2
b) Statement 4
c) Statement 3
d) Statements 2 and 4

35 What will be the output of the following Python code?



```
d = {1:'Ajay', 2:'Neeta', 3:'Saira'}
print('Neeta' in d)
```

a) False

b) True

c) Error

d) None

36 Write a statement in Python to declare a dictionary whose keys are 1, 2, and 3 and whose values are Monday, Tuesday and Wednesday respectively.

37 Assertion(A): List is an immutable data type
Reasoning(R): When an attempt is made to update the value of an immutable variable, the old variable is destroyed and a new variable is created by the same name in memory.

a) Both A and R are true and R is the correct explanation for A

b) Both A and R are true and R is not the correct explanation for A

c) A is True but R is False

d) A is false but R is True

38 Assertion (A): Python Standard Library consists of various modules.
Reasoning(R): A function in a module is used to simplify the code and avoid repetition.

a) Both A and R are true and R is the correct explanation for A

b) Both A and R are true and R is not the correct explanation for A

c) A is True but R is False

d) A is false but R is True

38. Assertion (A): A list can not become a key in a dictionary.
Reasoning(R): Only integers can be keys in a dictionary.

a) Both A and R are true and R is the correct explanation of A

b) Both A and R are true but R is not the correct explanation of A

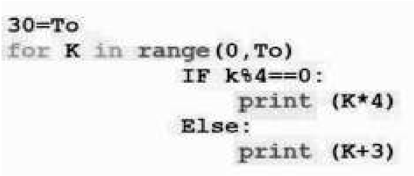
c) A is True but R is false

d) R is true but A is false

PART B (2 Mark Questions)

1

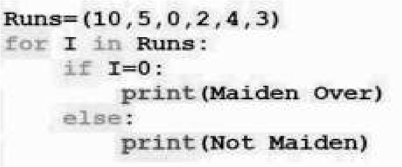
Rewrite the following code in Python after removing all syntax error(s).



```
30=To
for K in range(0,To)
              IF k%4==0:
                  print (K*4)
              Else:
                  print (K+3)
```

2

Rewrite the following code after removing all syntax error(s):



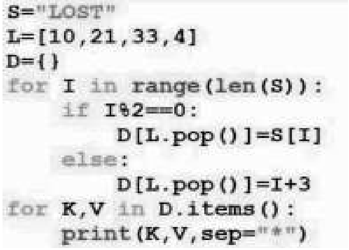
```
Runs=(10,5,0,2,4,3)
for I in Runs:
    if I=0:
        print(Maiden Over)
    else:
        print(Not Maiden)
```

3 Evaluate the following Python expression:

a) 2*3+4**2-5//2

b) 6<12 and not (20>15) or (10>5)

Predict the output of the following code:



```
S="LOST"
L=[10,21,33,4]
D={}
for I in range(len(S)):
    if I%2==0:
        D[L.pop()]=S[I]
    else:
        D[L.pop()]=I+3
for K,V in D.items():
    print(K,V,sep="*")
```

4

5

Write the Python statement for each of the following tasks using BUILT-IN functions/methods only:

(i) To insert an element 200 at the third position, in the list L1.

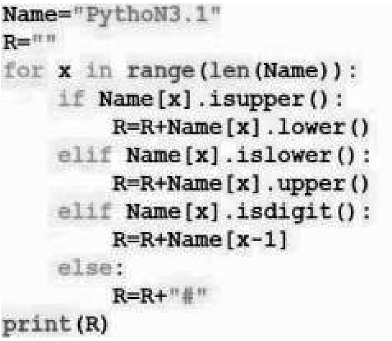
(ii) To check whether a string named, message ends with a full stop/period or not.

6

A list named studentAge stores the age of students in a class. Write the Python command to import the required module and display (using the built-in function) the most common age value from the given list.

7

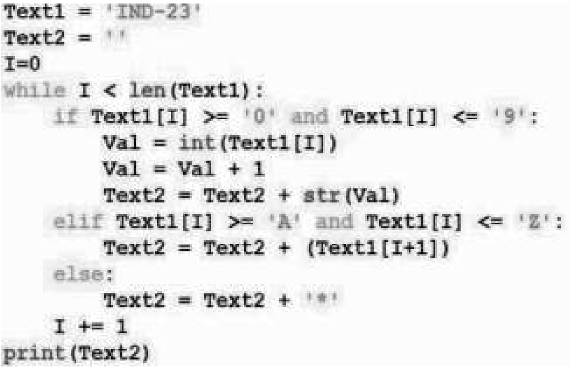
Find the output of the following code:



```
Name="PythoN3.1"
R=""
for x in range(len(Name)):
    if Name[x].isupper():
        R=R+Name[x].lower()
    elif Name[x].islower():
        R=R+Name[x].upper()
    elif Name[x].isdigit():
        R=R+Name[x-1]
    else:
        R=R+"#"
print(R)
```

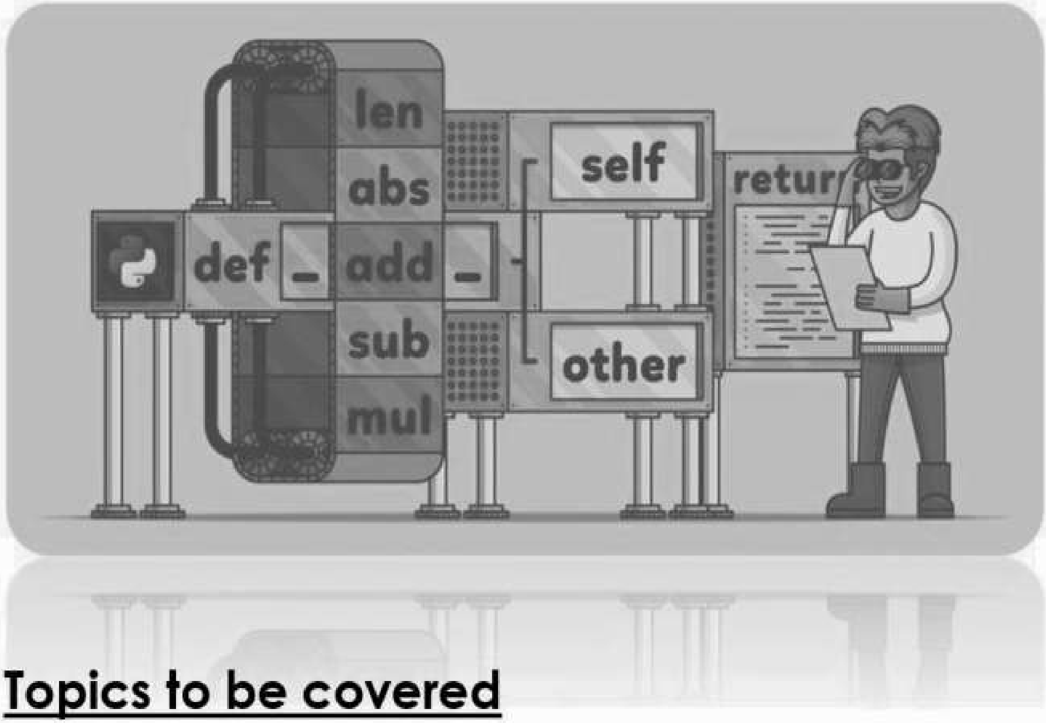
8

Predict the output of the Python code given below:



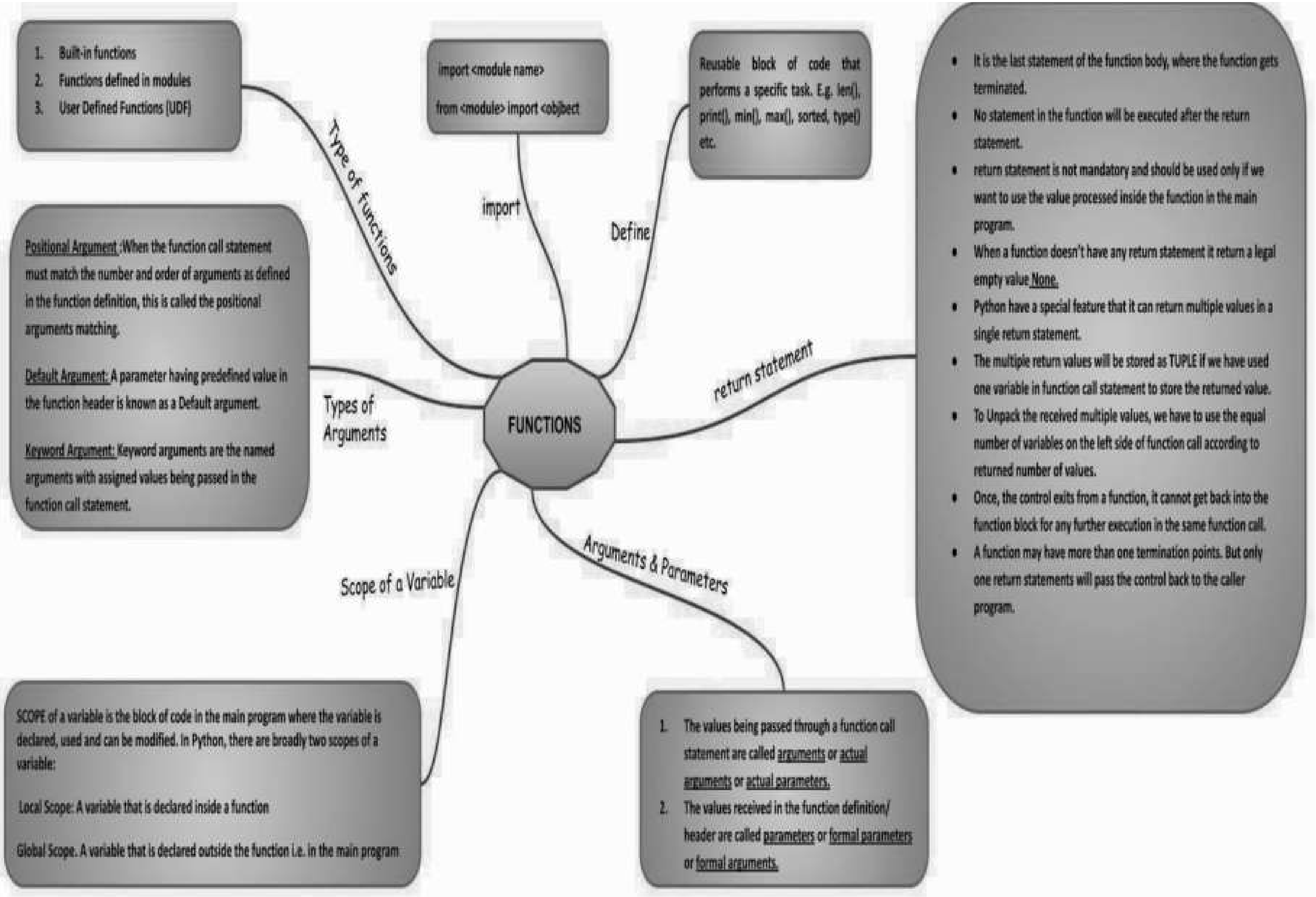
```
Text1 = 'IND-23'
Text2 = ''
I=0
while I < len(Text1):
    if Text1[I] >= '0' and Text1[I] <= '9':
        Val = int(Text1[I])
        Val = Val + 1
        Text2 = Text2 + str(Val)
    elif Text1[I] >= 'A' and Text1[I] <= 'Z':
        Text2 = Text2 + (Text1[I+1])
    else:
        Text2 = Text2 + '*'
    I += 1
print(Text2)
```

FUNCTION



Topics to be covered

- Introduction
- Types of Functions
- Create User User-defined Function
- Arguments And Parameters
- Default Parameters
- Positional Parameters
- Function Returning Values
- Flow of Execution
- Scope of Variable



1. Built-in functions
2. Functions defined in modules
3. User Defined Functions (UDF)

import <module name>

from <module> import <objbect

Reusable block of code that performs a specific task. E.g. len(), print(), min(), max(), sorted, type() etc.

- It is the last statement of the function body, where the function gets terminated.
- No statement in the function will be executed after the return statement.
- return statement is not mandatory and should be used only if we want to use the value processed inside the function in the main program.
- When a function doesn't have any return statement it return a legal empty value None.
- Python have a special feature that it can return multiple values in a single return statement.
- The multiple return values will be stored as TUPLE if we have used one variable in function call statement to store the returned value.
- To Unpack the received multiple values, we have to use the equal number of variables on the left side of function call according to returned number of values.
- Once, the control exits from a function, it cannot get back into the function block for any further execution in the same function call.
- A function may have more than one termination points. But only one return statements will pass the control back to the caller program.

Positional Argument :When the function call statement must match the number and order of arguments as defined in the function definition, this is called the positional arguments matching.

Default Argument: A parameter having predefined value in the function header is known as a Default argument.

Keyword Argument: Keyword arguments are the named arguments with assigned values being passed in the function call statement.

SCOPE of a variable is the block of code in the main program where the variable is declared, used and can be modified. In Python, there are broadly two scopes of a variable:

Local Scope: A variable that is declared inside a function

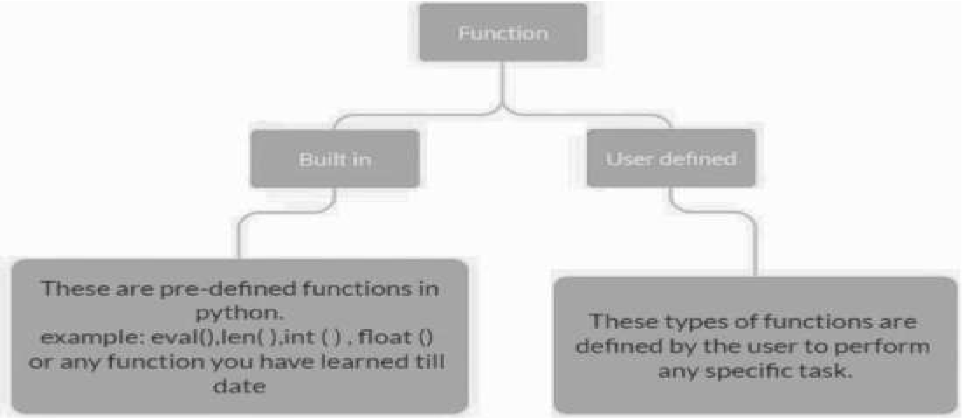
Global Scope. A variable that is declared outside the function i.e. in the main program

1. The values being passed through a function call statement are called arguments or actual arguments or actual parameters.
2. The values received in the function definition/ header are called parameters or formal parameters or formal arguments.

Function

Reusable block of code that performs a specific task. For example: len(), print(), min(), max(), sorted () , type() etc.

Types of Functions



These are pre-defined functions in python.
example: eval(),len( ),int ( ) , float ()
or any function you have learned till date

These types of functions are defined by the user to perform any specific task.

Defining a function in Python:

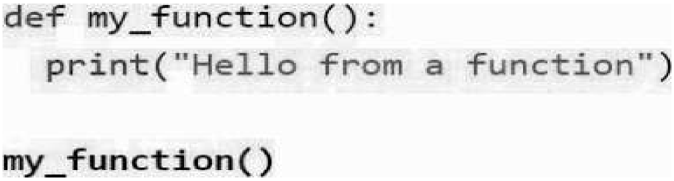
Name the function and specify what to do when the function is called. Python interpreter ignores the function definition until the function is called.

Calling a function:

Calling the function performs the specified actions with the indicated parameters

Function Definition in Python

In Python, a function is defined using the def keyword



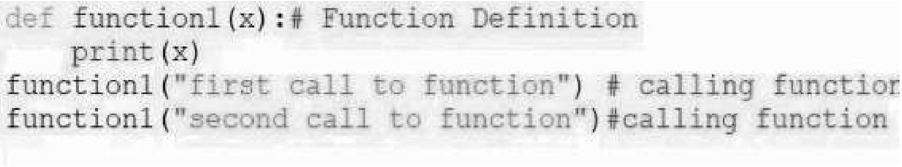
```
def my_function():
  print("Hello from a function")

my_function()
```

- **Arguments:** Information can be passed into functions as arguments. Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.
- Actual Parameters (Arguments) are values supplied to the function when it is invoked/called

- Formal Parameters are variables specified in function definitions to receive values from arguments passed during function calls.

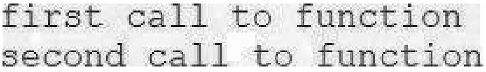
Example 1:

Observe the following code:



```
def function1(x):# Function Definition
    print(x)
function1("first call to function") # calling function
function1("second call to function")#calling function
```

Output:



```
first call to function
second call to function
```

In the above example, a user-defined function “function1” has been defined that receives one argument. Once the function is defined, it can be called any number of times with different arguments.

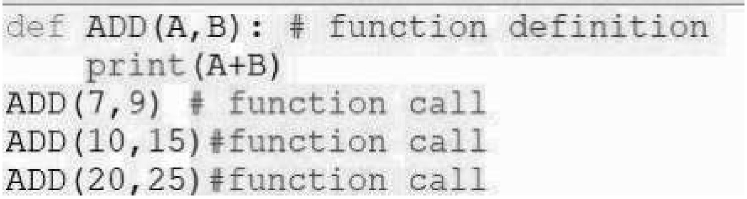
Formal argument: x

Actual argument:

“first call to function “ passed in the first call

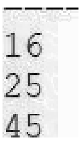
“second call to function” passed in the second call

Example 2: Write a function ADD(A, B) that receives two integer arguments and prints their sum.



```
def ADD(A,B): # function definition
    print(A+B)
ADD(7,9) # function call
ADD(10,15)#function call
ADD(20,25)#function call
```

Output:

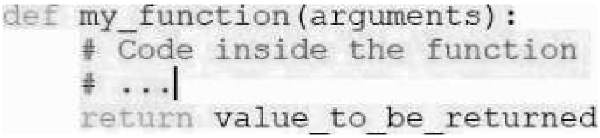


```
----
16
25
45
```

return keyword:

In Python, the ‘return’ keyword is used in functions to specify the value that the function will return when it is called. When a function is executed, it may perform some computations or operations, and the result can be sent back to the caller using the `return` statement.

The basic syntax for using the `return` statement is as follows:



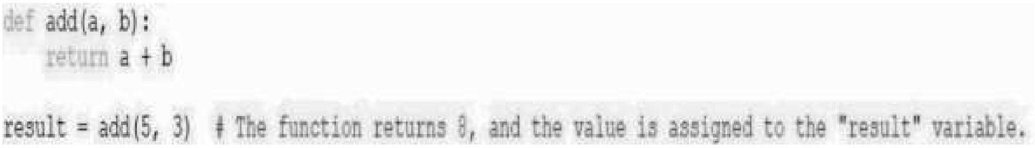
```
def my_function(arguments):
    # Code inside the function
    # ...|
    return value_to_be_returned
```

Here's what you need to know about the `return` statement:

1. **Returning a Value:**

   When you want to return a specific value from the function, you can use the `return` statement followed by the value you want to return.

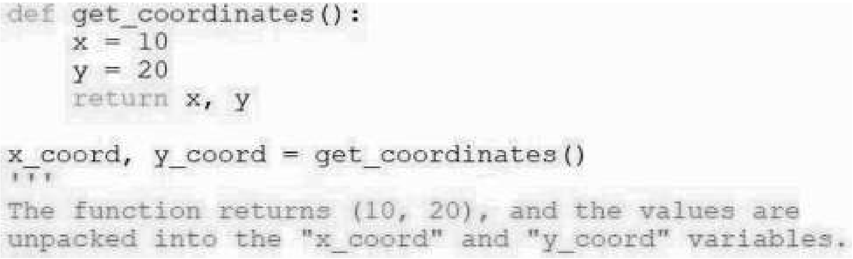
   Note: The function will stop executing immediately after the `return` statement is encountered, and the value will be passed back to the caller.



```
def add(a, b):
    return a + b

result = add(5, 3)  # The function returns 8, and the value is assigned to the "result" variable.
```

2. **Returning Multiple Values:**

   Python allows you to return multiple values from a function as a tuple. You can simply separate the values with commas after the `return` statement.

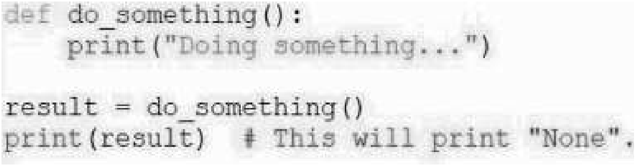


```
def get_coordinates():
    x = 10
    y = 20
    return x, y

x_coord, y_coord = get_coordinates()
'''
The function returns (10, 20), and the values are
unpacked into the "x_coord" and "y_coord" variables.
```

3. **Returning None:**

   If a function doesn't have a `return` statement or has a `return` statement without any value, it implicitly returns `None`.

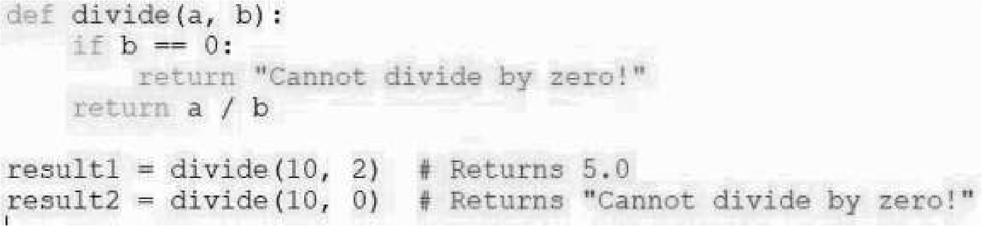
   `**None**` is a special constant in Python that represents the absence of a value.



```
def do_something():
    print("Doing something...")

result = do_something()
print(result)  # This will print "None".
```

4. **Early Exit with Return:**

You can use the `return` statement to exit a function early if certain conditions are met. This is useful when you want to terminate the function before reaching the end.



```
def divide(a, b):
    if b == 0:
        return "Cannot divide by zero!"
    return a / b

result1 = divide(10, 2)   # Returns 5.0
result2 = divide(10, 0)   # Returns "Cannot divide by zero!"
```

The `return` statement is a powerful tool that enables functions to produce results and pass data back to the calling code. Understanding how to use it correctly will help you design and implement effective functions in Python.

Scope of a variable:

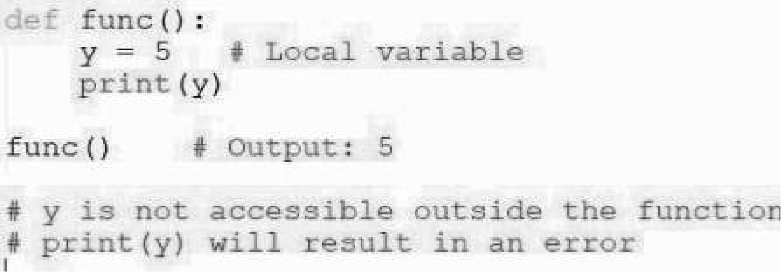
In Python, the scope of a variable refers to the region of the program where the variable is accessible. The scope determines where a variable is created, modified, and used.

Global Scope:

- Variables defined outside of any function or block have a global scope.
- They are accessible from anywhere in the code, including inside functions.

Local Scope:

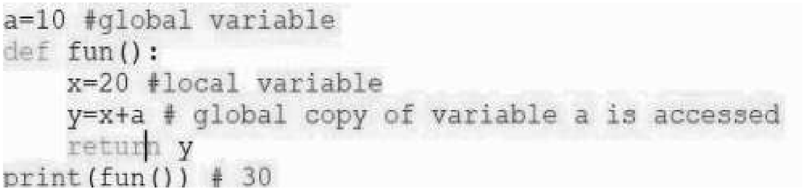
- Variables defined inside a function have a local scope.
- They are accessible only within the function where they are defined.
- Local variables are created when the function is called and destroyed when the function returns.



```
def func():
    y = 5    # Local variable
    print(y)

func()     # Output: 5

# y is not accessible outside the function
# print(y) will result in an error
```

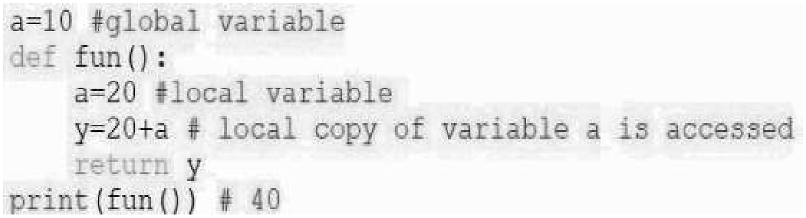
Points to be noted:

- When the local variable and global variable have different names: the global variable can be accessed inside the function



```
a=10 #global variable
def fun():
    x=20 #local variable
    y=x+a # global copy of variable a is accessed
    return y
print(fun()) # 30
```

- When local and global variables have the same name: priority is given to the local copy of the variable

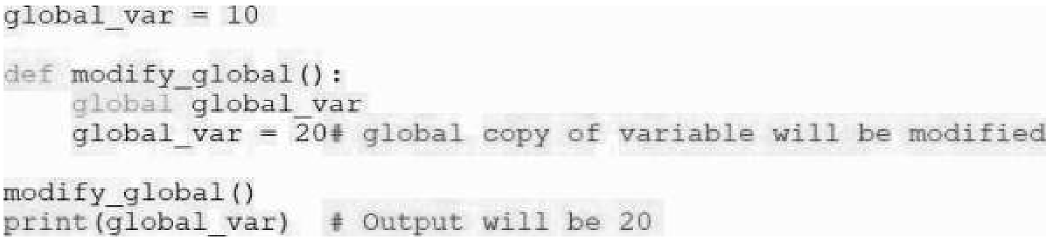


```
a=10 #global variable
def fun():
    a=20 #local variable
    y=20+a # local copy of variable a is accessed
    return y
print(fun()) # 40
```

global keyword

In Python, the ***global*** keyword is used to indicate that a variable declared inside a function should be treated as a global variable, rather than a local variable. When you assign a value to a variable inside a function, Python, by default, creates a local variable within that function's scope. However, if you need to modify a global variable within a function, you must use the global keyword to specify that you want to work with the global variable instead.

Syntax for using the global keyword:



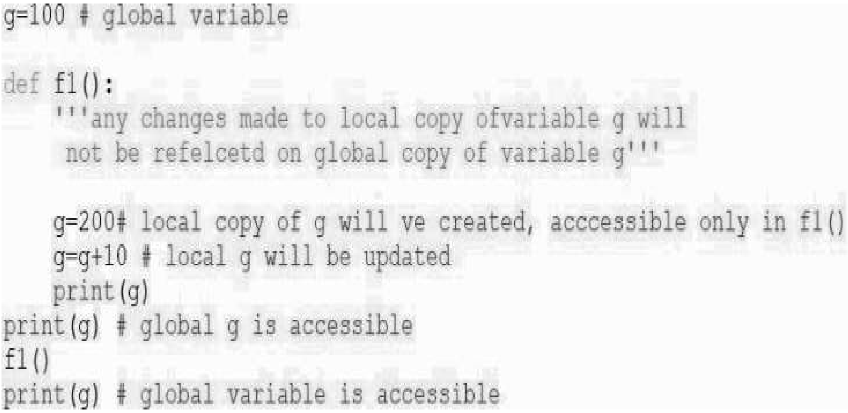
```
global_var = 10

def modify_global():
    global global_var
    global_var = 20# global copy of variable will be modified

modify_global()
print(global_var)  # Output will be 20
```

The lifetime of a variable:

The lifetime of a variable in Python depends on its scope. Global variables persist throughout the program's execution, whereas local variables within functions exist only during the function's execution.

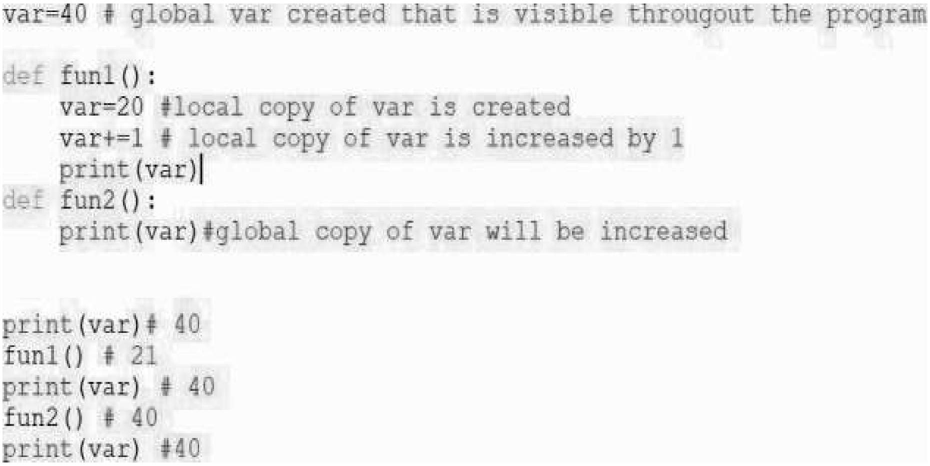
Study the following programs:

Example 1:



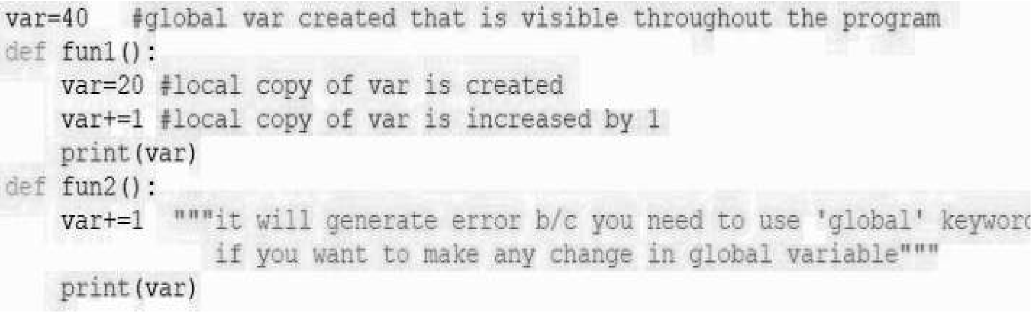
```
g=100 # global variable

def f1():
    '''any changes made to local copy ofvariable g will
     not be refelcetd on global copy of variable g'''

    g=200# local copy of g will ve created, acccessible only in f1()
    g=g+10 # local g will be updated
    print(g)
print(g) # global g is accessible
f1()
print(g) # global variable is accessible
```

Example 2:

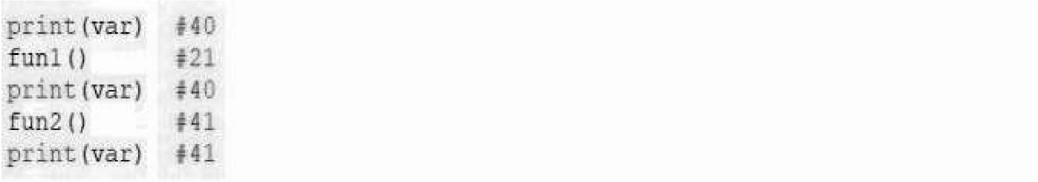


```
var=40 # global var created that is visible througout the program

def fun1():
    var=20 #local copy of var is created
    var+=1 # local copy of var is increased by 1
    print(var)
def fun2():
    print(var)#global copy of var will be increased


print(var)# 40
fun1() # 21
print(var) # 40
fun2() # 40
print(var) #40
```

Example 3:

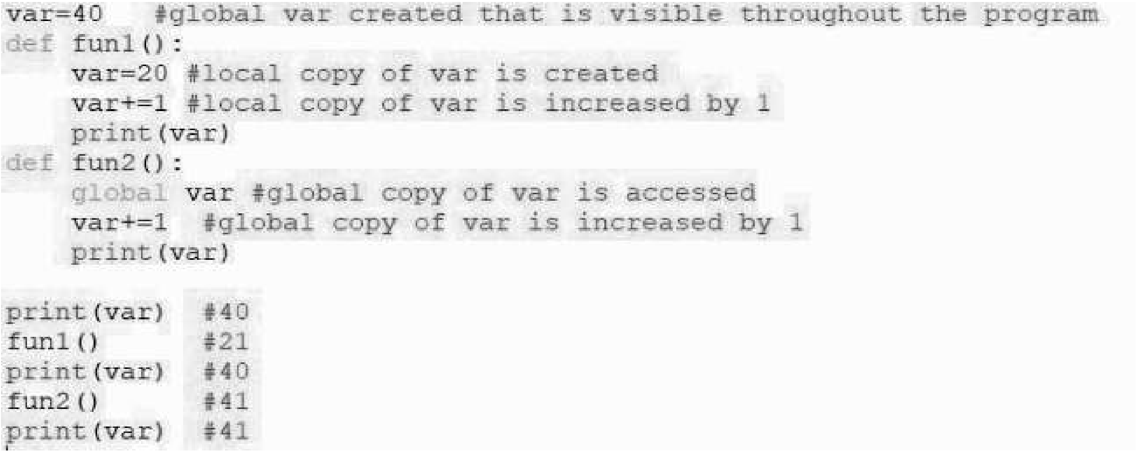


```
var=40   #global var created that is visible throughout the program
def fun1():
    var=20 #local copy of var is created
    var+=1 #local copy of var is increased by 1
    print(var)
def fun2():
    var+=1  """it will generate error b/c you need to use 'global' keyworc
                 if you want to make any change in global variable"""
    print(var)
```



```
print(var)  #40
fun1()      #21
print(var)  #40
fun2()      #41
print(var)  #41
```

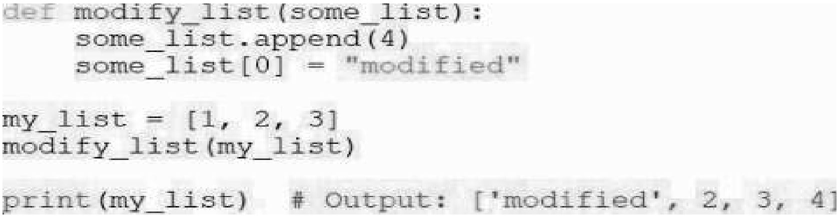
Example 4:



```
var=40   #global var created that is visible throughout the program
def fun1():
    var=20 #local copy of var is created
    var+=1 #local copy of var is increased by 1
    print(var)
def fun2():
    global var #global copy of var is accessed
    var+=1  #global copy of var is increased by 1
    print(var)

print(var)  #40
fun1()      #21
print(var)  #40
fun2()      #41
print(var)  #41
```

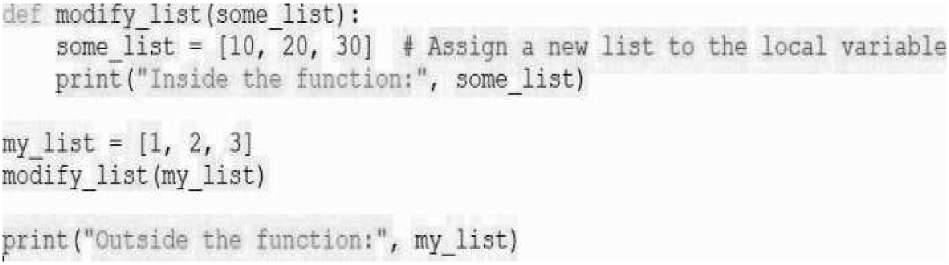
Passing list as an argument to the function:

Please note that when a list is passed as an argument, the original copy of the list is passed to the function i.e. if any change is made at any index in the list inside the function, it is reflected in the original list. That is because a list is a mutable datatype and in Python, when you pass a list as an argument to a function, you are actually passing a reference to the list rather than a copy of the list. This means that the function parameter will point to the same memory location as the original list. As a result, any changes made to the list within the function will be reflected in the original list outside the function.



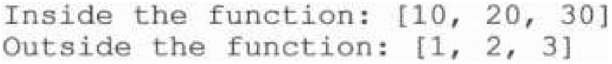
```
def modify_list(some_list):
    some_list.append(4)
    some_list[0] = "modified"

my_list = [1, 2, 3]
modify_list(my_list)

print(my_list)  # Output: ['modified', 2, 3, 4]
```

However, if you assign a different list to a variable inside a function in Python, it will create a new local variable that is separate from any variables outside the function. This local variable will only exist within the scope of the function, and changes made to it won't affect the original list outside the function.



```
def modify_list(some_list):
    some_list = [10, 20, 30]  # Assign a new list to the local variable
    print("Inside the function:", some_list)

my_list = [1, 2, 3]
modify_list(my_list)

print("Outside the function:", my_list)
```

Output:

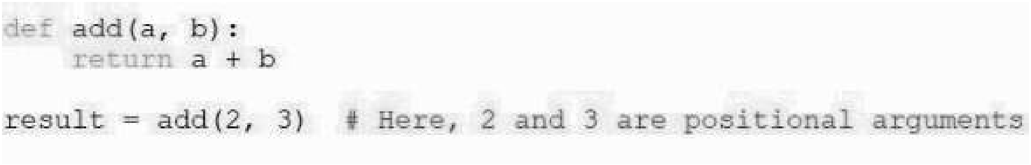


```
Inside the function: [10, 20, 30]
Outside the function: [1, 2, 3]
```

Types of arguments passed to a function:

Positional Arguments:

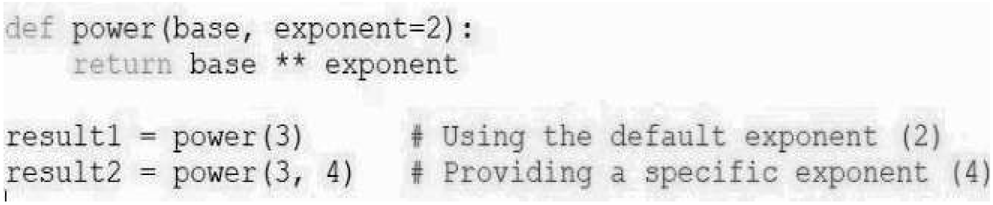
- These are the most common types of arguments and are matched to the function parameters based on their positions. The first argument corresponds to the first parameter, the second argument corresponds to the second parameter, and so on.
- The number and order of positional arguments must match the function's parameter list.



```
def add(a, b):
    return a + b

result = add(2, 3)  # Here, 2 and 3 are positional arguments
```

Default Arguments:

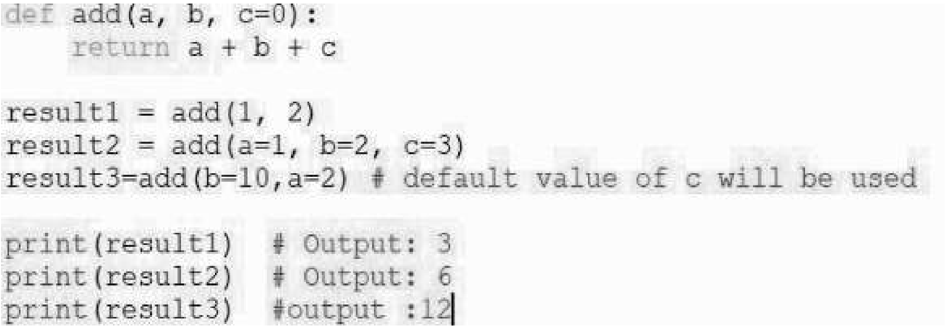
- Default arguments are used when a function is called with fewer arguments than there are parameters.
- The default values are specified in the function definition.
- If a value is not provided for a parameter during the function call, the default value is used.



```
def power(base, exponent=2):
    return base ** exponent

result1 = power(3)      # Using the default exponent (2)
result2 = power(3, 4)   # Providing a specific exponent (4)
```

Keyword Arguments:

- In this type, each argument is preceded by a keyword (parameter name) followed by an equal sign.
- The order of keyword arguments does not matter, as they are matched to the function parameters based on their names.
- These arguments provide flexibility to call a function with arguments passed in any order.



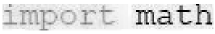
```
def add(a, b, c=0):
    return a + b + c

result1 = add(1, 2)
result2 = add(a=1, b=2, c=3)
result3=add(b=10,a=2) # default value of c will be used

print(result1)  # Output: 3
print(result2)  # Output: 6
print(result3)  #output :12
```

Python modules:

- In Python, a module is a file containing Python code that defines variables, functions, and classes.
- Modules allow you to organize and reuse code by breaking it into separate files, making it easier to maintain and understand complex programs.
- Python's standard library comes with a vast collection of built-in modules that cover various functionalities
- If needed, you can also create your own custom modules.
- To use a module in your Python code, you need to import it using the import statement.

Math module:

- The math module in Python is a built-in module that provides various mathematical functions and constants.
- It is part of the Python Standard Library i.e. it does not require any additional installation to use.
- To use the math module, you need to import it at the beginning of your Python script.



```
import math
```

- Once you've imported the module, you can access its functions and constants using the math prefix.

Here are some commonly used functions and constants provided by the math module:

Mathematical Constants:

- math.pi: Represents the mathematical constant π (pi).
- math.e: Represents the mathematical constant e (Euler's number).

Basic Mathematical functions :

- math.sqrt(x): Returns the square root of x.
- math.pow(x, y): Returns x raised to the power y.
- math.exp(x): Returns the exponential of x (e^x).
- math.log(x, base): Returns the logarithm of x to the specified base (default base is e).

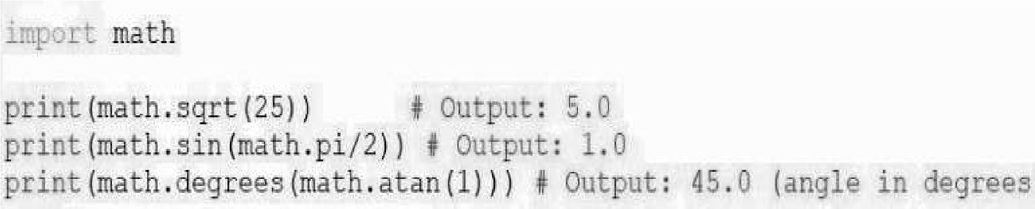
Trigonometric Functions (all angles are in radians):

- math.sin(x), math.cos(x), math.tan(x): Sine, cosine, and tangent of x, respectively.
- math.asin(x), math.acos(x), math.atan(x): Arcsine, arccosine, and arctangent

of x, respectively.

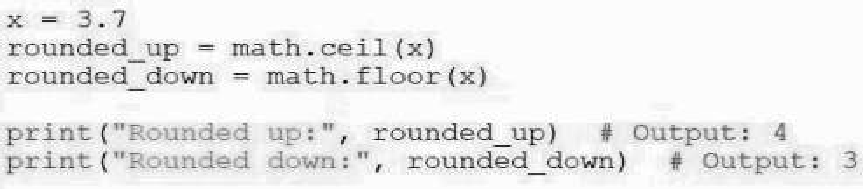
Study the following examples:

Example 1:



```
import math

print(math.sqrt(25))      # Output: 5.0
print(math.sin(math.pi/2)) # Output: 1.0
print(math.degrees(math.atan(1))) # Output: 45.0 (angle in degrees
```

Example 2:



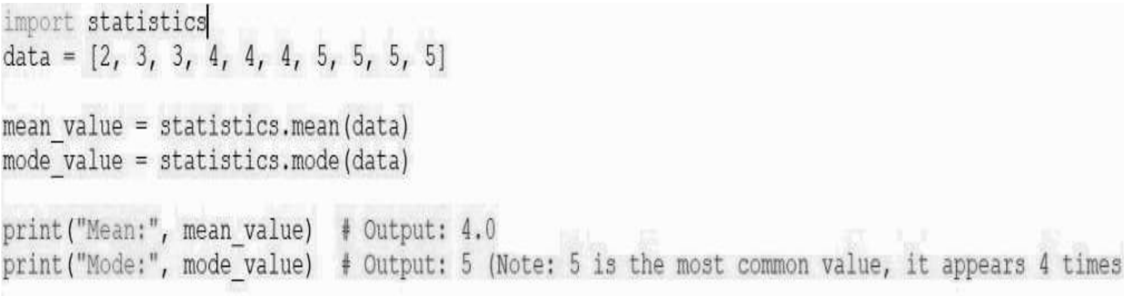
```
x = 3.7
rounded_up = math.ceil(x)
rounded_down = math.floor(x)

print("Rounded up:", rounded_up)  # Output: 4
print("Rounded down:", rounded_down)  # Output: 3
```

Statistics module:

- The statistics module in Python is another built-in module that provides functions for working with statistical data.
- It offers a variety of statistical functions to compute measures like mean, median, standard deviation, variance, etc.
- The statistics module is part of the Python Standard Library, so there's no need to install any additional packages to use it.

Here are some commonly used functions provided by the statistics module:

Function	Definition
satistics.mean(data)	Calculates the arithmetic mean (average) of the data.
statistics.median(data)	Computes the median value of the data.
statistics.mode(data)	Finds the mode (most common value) in the data.

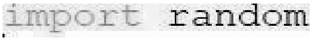


```
import statistics
data = [2, 3, 3, 4, 4, 4, 5, 5, 5, 5]

mean_value = statistics.mean(data)
mode_value = statistics.mode(data)

print("Mean:", mean_value)  # Output: 4.0
print("Mode:", mode_value)  # Output: 5 (Note: 5 is the most common value, it appears 4 times
```

random module

- The random module in Python is another built-in module that provides functions for generating random numbers, and sequences, and making random choices.

- It is commonly used for tasks such as random number generation, random shuffling, and random sampling.



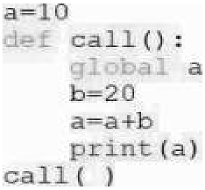
```
import random
```

Here are some commonly used functions provided by the random module:

- **random.random():** Generates a random float number in the range [0.0, 1.0).
- **random.randint(a, b):** Generates a random integer in the range [a, b] (inclusive).
- **random.randrange(a,b,c)**:Generates a random integer in the range [a, b] (excluding b), where c is the step value.

Assignment

1. What will be the output of the following code?



```
a=10
def call():
    global a
    b=20
    a=a+b
    print(a)
call( )
```

a) 10 b) 30 c) error d) 20

2. What is the scope of a variable defined inside a function?

a) Global scope b) Local scope c)Universal scope d)Function scope

3. In Python, can a function return multiple values simultaneously?

a) Yes b) No

4. What is the purpose of the "return" statement in a function?

a) It specifies the type of the function.
b) It defines the input parameters of the function.
c) It indicates the end of a function.

d) It returns a value from the function to the caller.

5. Which of the following module functions generates an integer?

a) randint() b) uniform() c) random() d) all of the above

6. The return type of the input() function is

a) String b) integer c) list d) tuple

7. The values being passed through a function call statement are called:

a) Actual parameter b) Formal parameter c) Default parameter d) None of these

8. Which of the following components are part of a function header in Python?

a) Function Name b) Return Statement c) Parameter List d) Both a and c

9. Which of the following function header is correct?

a) def cal_si(p=100, r, t=2) b) def cal_si(p=100, r=8, t)

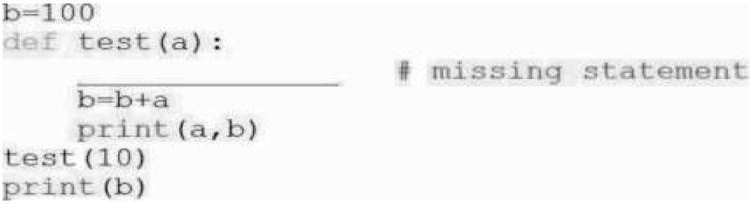
c) def cal_si(p, r=8, t) d) def cal_si(p, r=8, t=2)

10. Which of the following is the correct way to call a function?

a) my_func() b) def my_func() c) return my_func d) call my_func()

11. Consider the code given below:

Which of the following statements should be given in the blank for #Missing Statement, if the output produced is 110?



```
b=100
def test(a):
    ____________________        # missing statement
    b=b+a
    print(a,b)
test(10)
print(b)
```

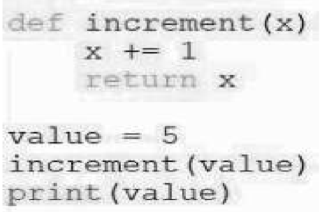
a. global a

b. global b=100

c. global b

d. global a=100

12. What will be the output?



```
def increment(x)
    x += 1
    return x

value = 5
increment(value)
print(value)
```

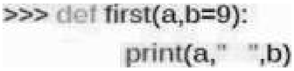
a. 5

b. 6

c. 4

d. This code will raise an error.

13. A function is defined as below, the function call parameters can be:



```
>>> def first(a,b=9):
        print(a,"  ",b)
```

a) Two tuples b) Two numbers c) One number d) All of the above

11. What is the correct syntax for defining a function in Python?

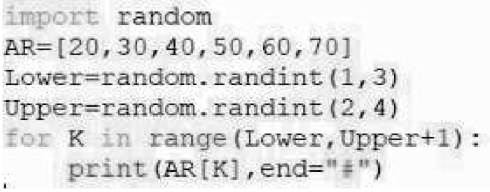
a) function myFunction( ): c) def myFunction( ):

b) def myFunction: d) define myFunction( ):

15. Statistics.mode([10,10,11,12,14,11,11,15,15,16,15]) will return (consider module is imported)

a) 10 b) 15 c) 11 d) Error

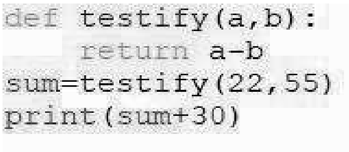
16. What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also, specify the maximum values that can be assigned to each of the variables Lower and Upper.



```
import random
AR=[20,30,40,50,60,70]
Lower=random.randint(1,3)
Upper=random.randint(2,4)
for K in range(Lower,Upper+1):
    print(AR[K],end="#")
```

a) 10#40#70# b) 30#40#50# c) 50#60#70# d) 40#50#70#

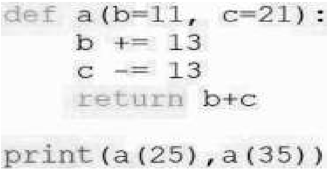
17. What will be the output of the Python code?



```
def testify(a,b):
    return a-b
sum=testify(22,55)
print(sum+30)
```

c) 33 b) -33 c) 3 d) -3

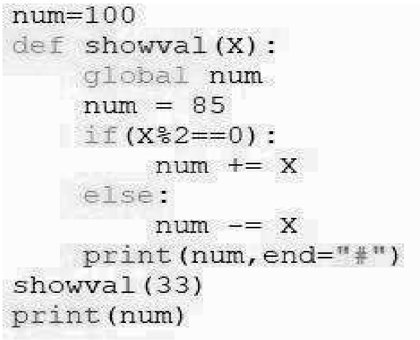
18. What will be the output of the following code?



```
def a(b=11, c=21):
    b += 13
    c -= 13
    return b+c

print(a(25),a(35))
```

a) 15 18 b) 46 56 c) 25 35 d) 13 12

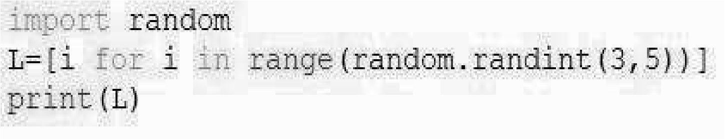
19. What will be the output of the following code?



```
num=100
def showval(X):
    global num
    num = 85
    if(X%2==0):
        num += X
    else:
        num -= X
    print(num,end="#")
showval(33)
print(num)
```

a) 100#52 b) 85#52 c) 85#33 d) 52#52

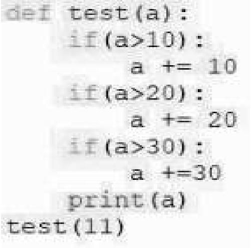
20. Find the impossible option from the following



```
import random
L=[i for i in range(random.randint(3,5))]
print(L)
```

a) [0, 1, 2, 3] b) [0, 1, 2, 3, 4] c) [0, 1, 2, 3, 4, 5] d) [0, 1, 2]

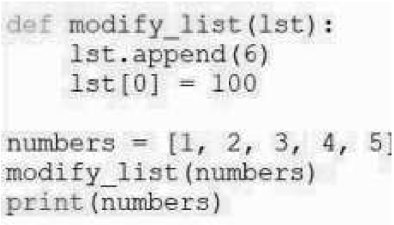
21. Look at the function definition and the function call and determine the correct output



```
def test(a):
    if(a>10):
        a += 10
    if(a>20):
        a += 20
    if(a>30):
        a +=30
    print(a)
test(11)
```

a) 21
b) 72
c) 61
d) 71

22. Predict output:



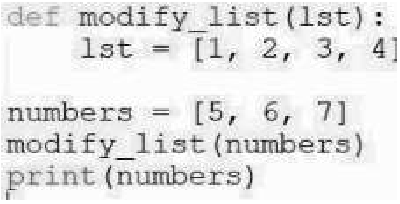
```
def modify_list(lst):
    lst.append(6)
    lst[0] = 100

numbers = [1, 2, 3, 4, 5]
modify_list(numbers)
print(numbers)
```

a) [1, 2, 3, 4, 5, 6] b) [100, 2, 3, 4, 5, 6]

c) [100, 2, 3, 4, 5]

d) [1, 2, 3, 4, 5]

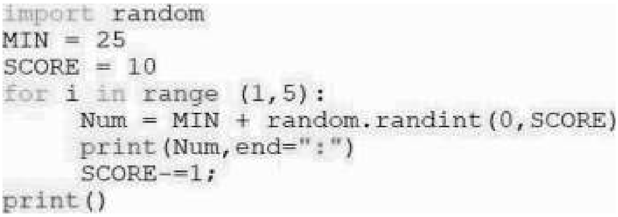
23. Predict output:



```
def modify_list(lst):
    lst = [1, 2, 3, 4]

numbers = [5, 6, 7]
modify_list(numbers)
print(numbers)
```

a) [1, 2, 3, 4]
b) [5, 6, 7]
c) [1, 2, 3, 4, 5, 6, 7]
d) This code will raise an error.

24. Which of the following outputs is not possible:

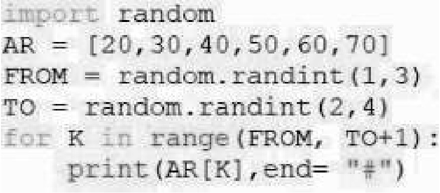


```
import random
MIN = 25
SCORE = 10
for i in range (1,5):
     Num = MIN + random.randint(0,SCORE)
     print(Num,end=":")
     SCORE-=1;
print()
```

Options:

a. 34:24:30:33:	b. 29:33:36:31:
c. 24:31:30:31:	d. 34:31:29:30:

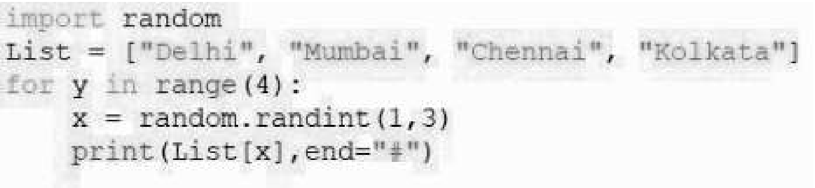
25. What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also, specify the maximum values that can be assigned to each of the variables FROM and TO.



```
import random
AR = [20,30,40,50,60,70]
FROM = random.randint(1,3)
TO = random.randint(2,4)
for K in range(FROM, TO+1):
    print(AR[K],end= "#")
```

a) 10#40#70#
b) 30#40#50#
c) 50#60#70
d) 40#50#70#

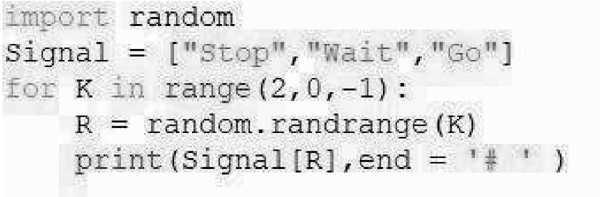
26. What will be the possible outcomes:



```
import random
List = ["Delhi", "Mumbai", "Chennai", "Kolkata"]
for y in range(4):
    x = random.randint(1,3)
    print(List[x],end="#")
```

a. Delhi#Mumbai#Chennai#Kolkata#	b. Mumbai#Chennai#Kolkata#Mumbai#
c. Mumbai#MumbaiMumbai# Delhi#	d. Mumbai#Mumbai Delhi#Mumbai

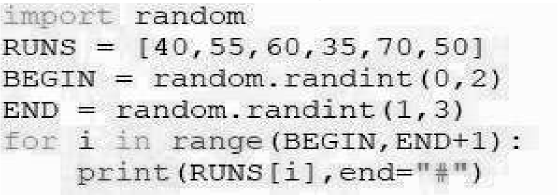
27. What is/are the possible outcome/(s) for the following code?



```
import random
Signal = ["Stop","Wait","Go"]
for K in range(2,0,-1):
    R = random.randrange(K)
    print(Signal[R],end = '# ' )
```

a) Stop # Wait # Go b) Wait # Stop # c) Go # Wait # d) Go # Stop #

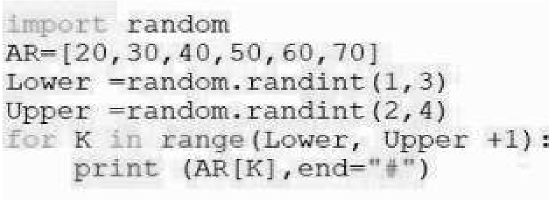
28. What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code?



```
import random
RUNS = [40,55,60,35,70,50]
BEGIN = random.randint(0,2)
END = random.randint(1,3)
for i in range(BEGIN,END+1):
    print(RUNS[i],end="#")
```

a) 40#35#
b) 60#35#70#50#
c) 35#70#50#
d) 40#55#60#

29. What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code?



```
import random
AR=[20,30,40,50,60,70]
Lower =random.randint(1,3)
Upper =random.randint(2,4)
for K in range(Lower, Upper +1):
    print (AR[K],end="#")
```

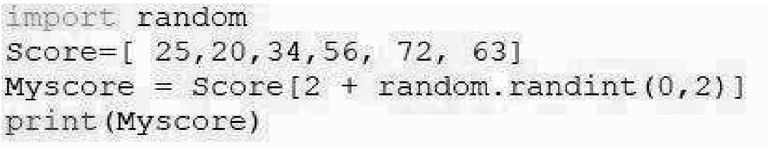
a) 10#40#70#
b) 30#40#50#

c) 50#60#70#

d) 40#50#70#

30. What are the possible outcomes/(s) :

i



```
import random
Score=[ 25,20,34,56, 72, 63]
Myscore = Score[2 + random.randint(0,2)]
print(Myscore)
```

Output Options :

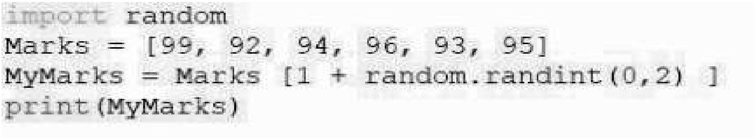
a) 25

b) 34

c) 20

d) None of the above

ii.



```
import random
Marks = [99, 92, 94, 96, 93, 95]
MyMarks = Marks [1 + random.randint(0,2) ]
print(MyMarks)
```

Output Options :

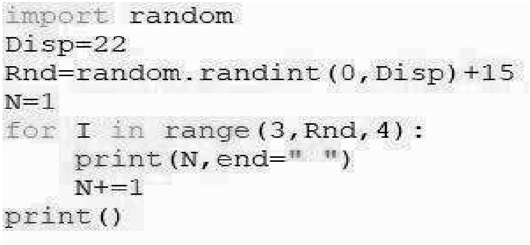
a) 99

b) 94

c) 93

d) None of the above

iii.



```
import random
Disp=22
Rnd=random.randint(0,Disp)+15
N=1
for I in range(3,Rnd,4):
    print(N,end=" ")
    N+=1
print()
```

Output Options:

a) 1

b) 1 2

c) 1 2 3 4

d) 1 2 3

31. Assertion and Reason type questions

a) Both A and R are true and R is the correct explanation for A

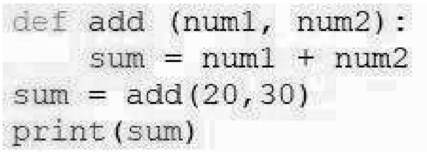
b) Both A and R are true and R is not the correct explanation for A

c) A is True but R is False

d) A is false but R is True

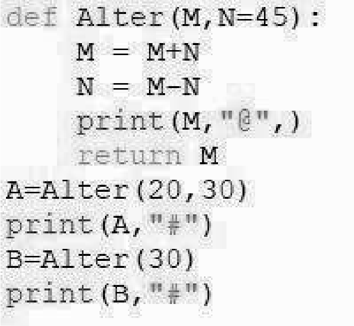
i.	Assertion (A): To use a function from a particular module, we need to import the module. Reason(R): import statement can be written anywhere in the program, before using a function from that module.
ii.	Assertion (A): In Python, functions can be defined inside other functions. Reason (R): Python supports nested functions which allow functions to be defined within other functions.
iii.	Assertion (A): In Python, a function must always return a value. Reason (R): If a return statement is not used in a function, it will return `None` by default.
iv.	Assertion (A): In Python, the `def` keyword is used to create a new function. Reason(R): The `def` keyword allows for the definition of both named and anonymous functions in Python.
v.	Assertion (A): In Python, function parameters can have default values. Reason(R): Default values for parameters are specified by assigning values in the function definition.

32. What will be the output of the following Python code?



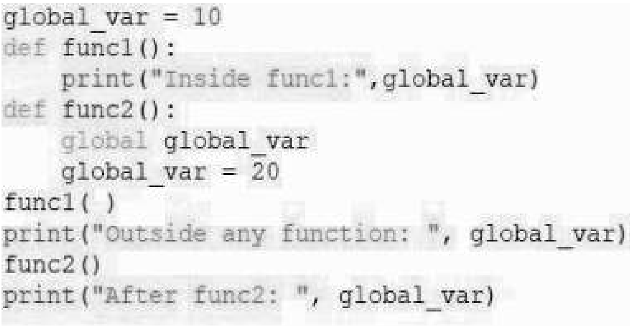
```
def add (num1, num2):
    sum = num1 + num2
sum = add(20,30)
print(sum)
```

33. Find and write the output of the following Python code:



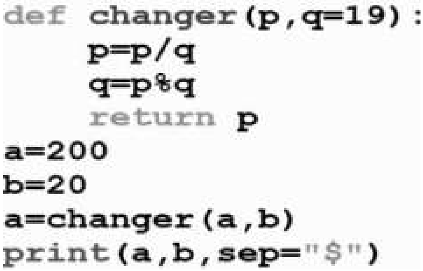
```
def Alter(M,N=45):
    M = M+N
    N = M-N
    print(M,"@",)
    return M
A=Alter(20,30)
print(A,"#")
B=Alter(30)
print(B,"#")
```

34. Predict output:



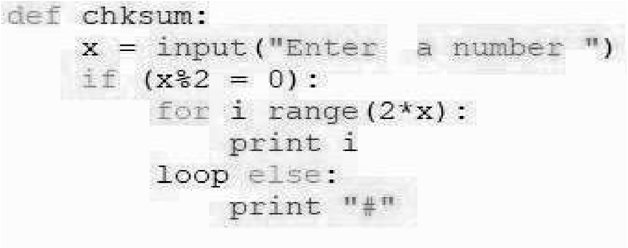
```
global_var = 10
def func1():
    print("Inside func1:",global_var)
def func2():
    global global_var
    global_var = 20
func1( )
print("Outside any function: ", global_var)
func2()
print("After func2: ", global_var)
```

35. Predict output:



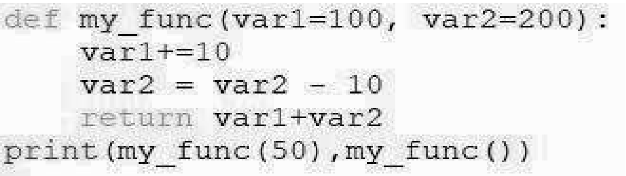
```
def changer(p,q=19):
    p=p/q
    q=p%q
    return p
a=200
b=20
a=changer(a,b)
print(a,b,sep="$")
```

36. Rewrite the following code after removing the syntactical errors (if any).



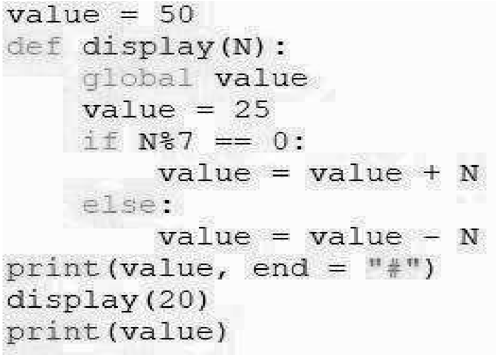
```
def chksum:
    x = input("Enter  a number ")
    if (x%2 = 0):
        for i range(2*x):
            print i
        loop else:
            print "#"
```

37. What will be the output of the following code?



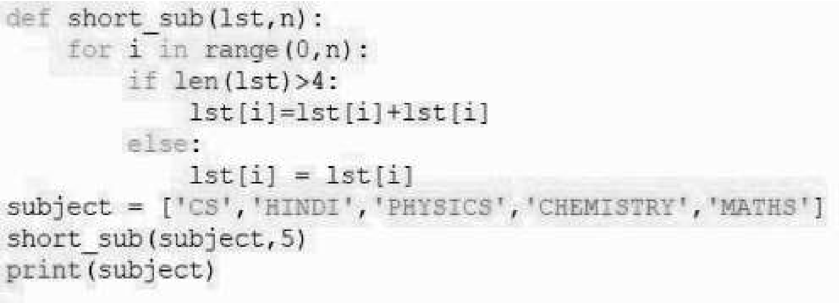
```
def my_func(var1=100, var2=200):
    var1+=10
    var2 = var2 - 10
    return var1+var2
print(my_func(50),my_func())
```

38. What will be the output of the following code?



```
value = 50
def display(N):
    global value
    value = 25
    if N%7 == 0:
        value = value + N
    else:
        value = value - N
print(value, end = "#")
display(20)
print(value)
```

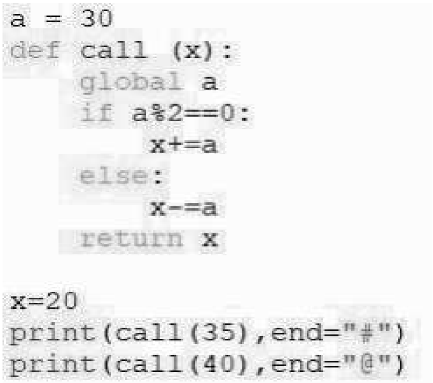
39. What will be the output of the following code?



```
def short_sub(lst,n):
    for i in range(0,n):
        if len(lst)>4:
            lst[i]=lst[i]+lst[i]
        else:
            lst[i] = lst[i]
subject = ['CS','HINDI','PHYSICS','CHEMISTRY','MATHS']
short_sub(subject,5)
print(subject)
```

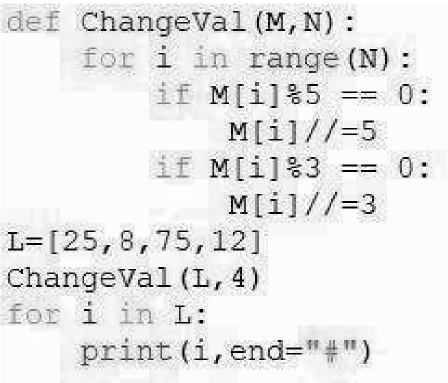
40. What is the output of the following code snippet?

i.



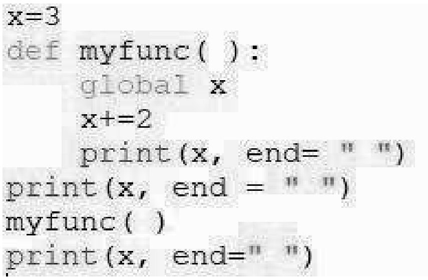
```
a = 30
def call (x):
    global a
    if a%2==0:
        x+=a
    else:
        x-=a
    return x

x=20
print(call(35),end="#")
print(call(40),end="@")
```

ii.



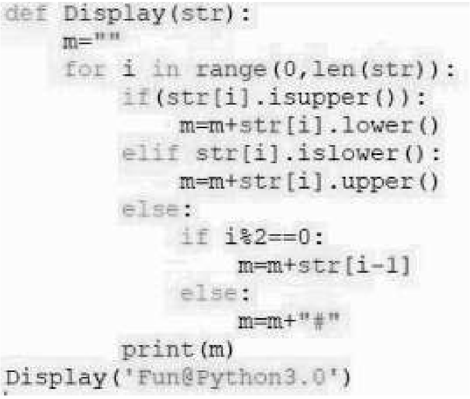
```
def ChangeVal(M,N):
    for i in range(N):
        if M[i]%5 == 0:
            M[i]//=5
        if M[i]%3 == 0:
            M[i]//=3
L=[25,8,75,12]
ChangeVal(L,4)
for i in L:
    print(i,end="#")
```

41. What will be the output of the following code?



```
x=3
def myfunc( ):
    global x
    x+=2
    print(x, end= " ")
print(x, end = " ")
myfunc( )
print(x, end=" ")
```

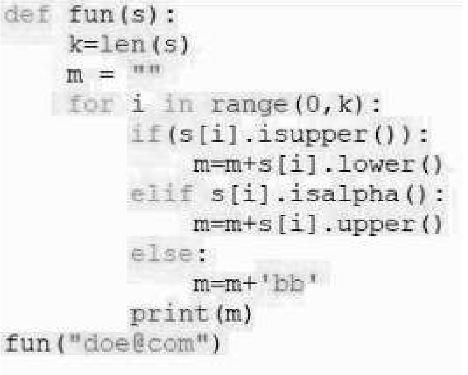
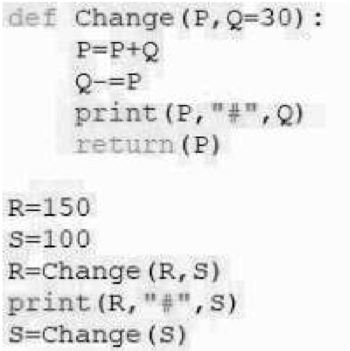
42. Find and write the output of the following Python Code:



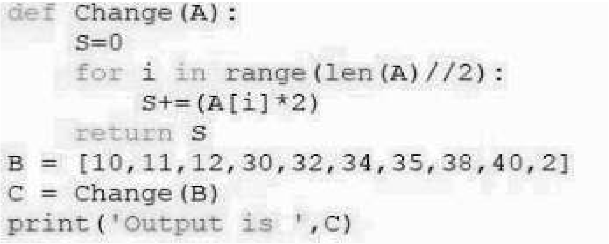
```
def Display(str):
    m=""
    for i in range(0,len(str)):
        if(str[i].isupper()):
            m=m+str[i].lower()
        elif str[i].islower():
            m=m+str[i].upper()
        else:
            if i%2==0:
                m=m+str[i-1]
            else:
                m=m+"#"
        print(m)
Display('Fun@Python3.0')
```

43. Find and write the output of the following Python code:

I	ii


def fun(s):
    k=len(s)
    m = ""
    for i in range(0,k):
        if(s[i].isupper()):
            m=m+s[i].lower()
        elif s[i].isalpha():
            m=m+s[i].upper()
        else:
            m=m+'bb'
        print(m)
fun("doe@com")
	
	def Change(P,Q=30):
	    P=P+Q
	    Q-=P
	    print(P,"#",Q)
	    return(P)
	
	R=150
	S=100
	R=Change(R,S)
	print(R,"#",S)
	S=Change(S)

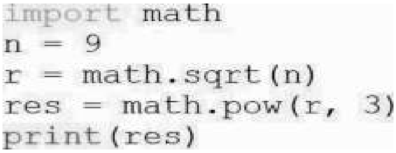
44. Write the output of the following Python code:



```
def Change(A):
    S=0
    for i in range(len(A)//2):
        S+=(A[i]*2)
    return S
B = [10,11,12,30,32,34,35,38,40,2]
C = Change(B)
print('Output is ',C)
```

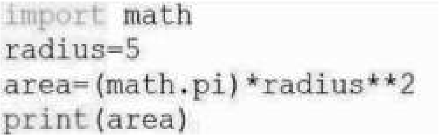
45. Predict the output of the following:

i.



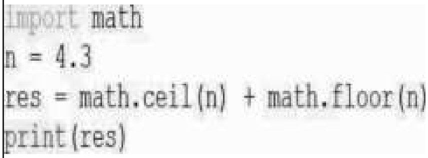
```
import math
n = 9
r = math.sqrt(n)
res = math.pow(r, 3)
print(res)
```

ii.



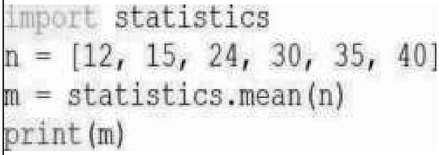
```
import math
radius=5
area=(math.pi)*radius**2
print(area)
```

iii.



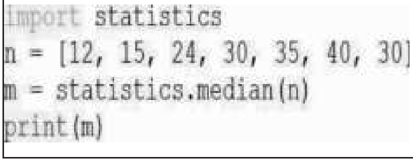
```
import math
n = 4.3
res = math.ceil(n) + math.floor(n)
print(res)
```

iv.



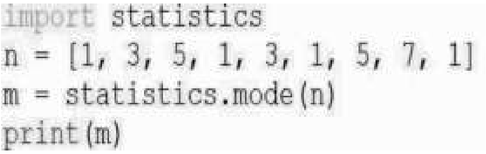
```
import statistics
n = [12, 15, 24, 30, 35, 40]
m = statistics.mean(n)
print(m)
```

v.



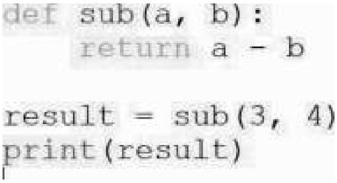
```
import statistics
n = [12, 15, 24, 30, 35, 40, 30]
m = statistics.median(n)
print(m)
```

vi



```
import statistics
n = [1, 3, 5, 1, 3, 1, 5, 7, 1]
m = statistics.mode(n)
print(m)
```

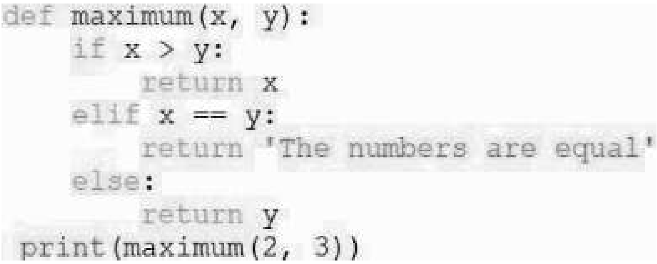
46. Predict output :



```
def sub(a, b):
    return a - b

result = sub(3, 4)
print(result)
```

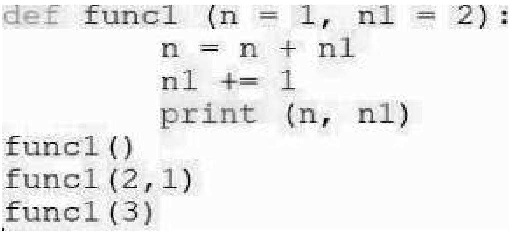
47. Predict output :

i.



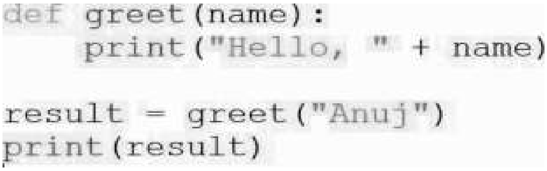
```
def maximum(x, y):
    if x > y:
        return x
    elif x == y:
        return 'The numbers are equal'
    else:
        return y
 print(maximum(2, 3))
```

ii.



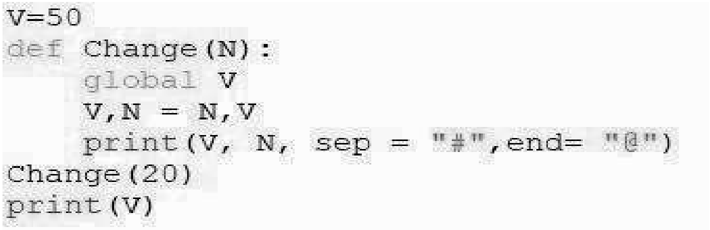
```
def func1 (n = 1, n1 = 2):
        n = n + n1
        n1 += 1
        print (n, n1)
func1()
func1(2,1)
func1(3)
```

iii.



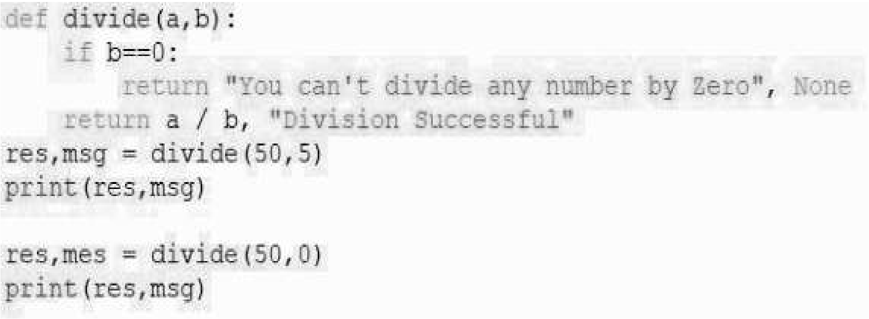
```
def greet(name):
    print("Hello, " + name)

result = greet("Anuj")
print(result)
```

48. What will be the output of the following Python Code?



```
V=50
def Change(N):
    global V
    V,N = N,V
    print(V, N, sep = "#",end= "@")
Change(20)
print(V)
```

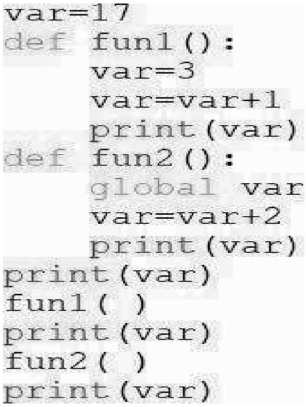
49. Predict output:



```
def divide(a,b):
    if b==0:
        return "You can't divide any number by Zero", None
    return a / b, "Division Successful"
res,msg = divide(50,5)
print(res,msg)

res,mes = divide(50,0)
print(res,msg)
```

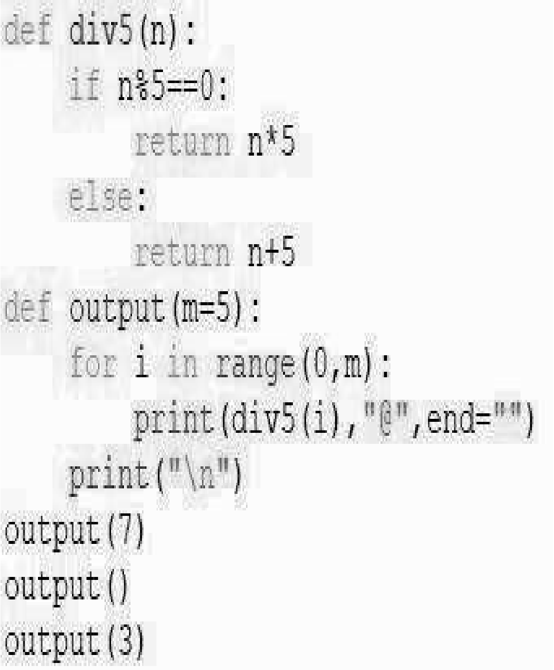
50. Predict output:

i



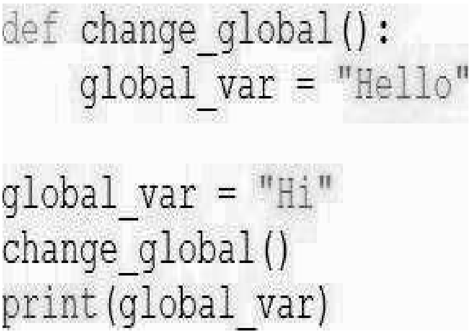
```
var=17
def fun1():
    var=3
    var=var+1
    print(var)
def fun2():
    global var
    var=var+2
    print(var)
print(var)
fun1( )
print(var)
fun2( )
print(var)
```

ii.



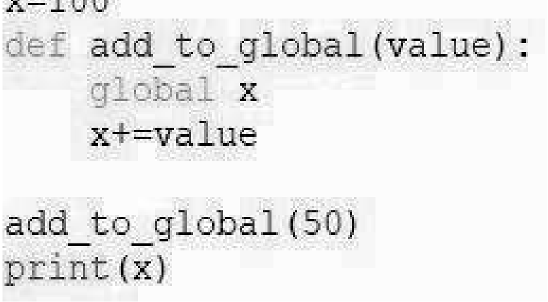
```
def div5(n):
    if n%5==0:
        return n*5
    else:
        return n+5
def output(m=5):
    for i in range(0,m):
        print(div5(i),"@",end="")
    print("\n")
output(7)
output()
output(3)
```

iii.



```
def change_global():
    global_var = "Hello"

global_var = "Hi"
change_global()
print(global_var)
```

iv.

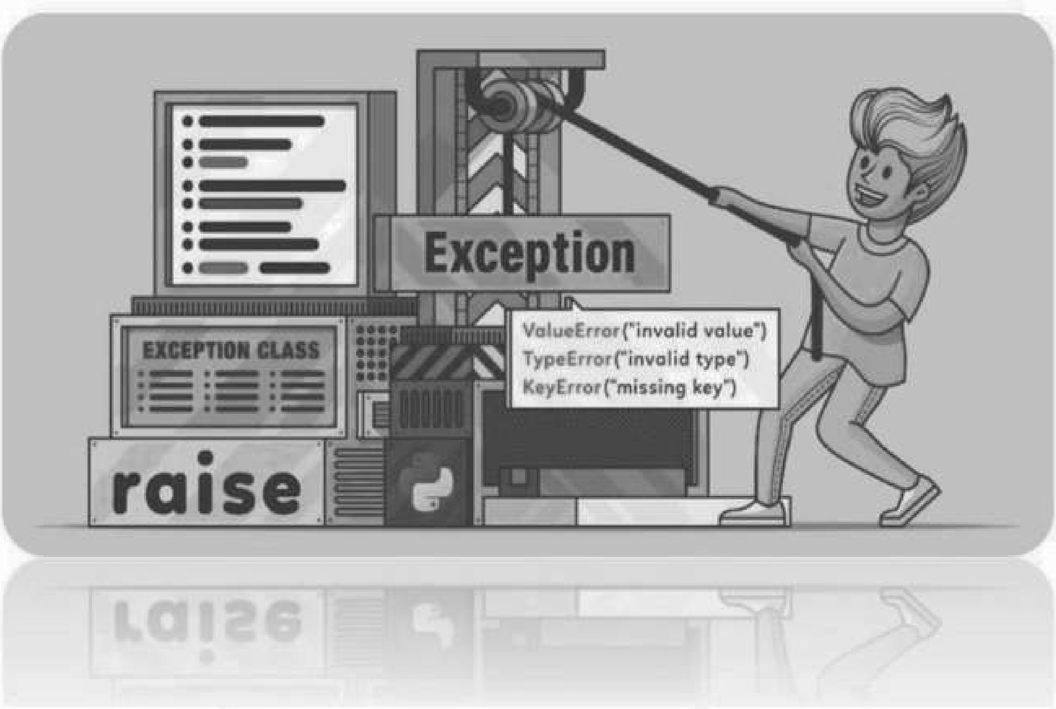


```
x=100
def add_to_global(value):
    global x
    x+=value

add_to_global(50)
print(x)
```

Lab Exercise

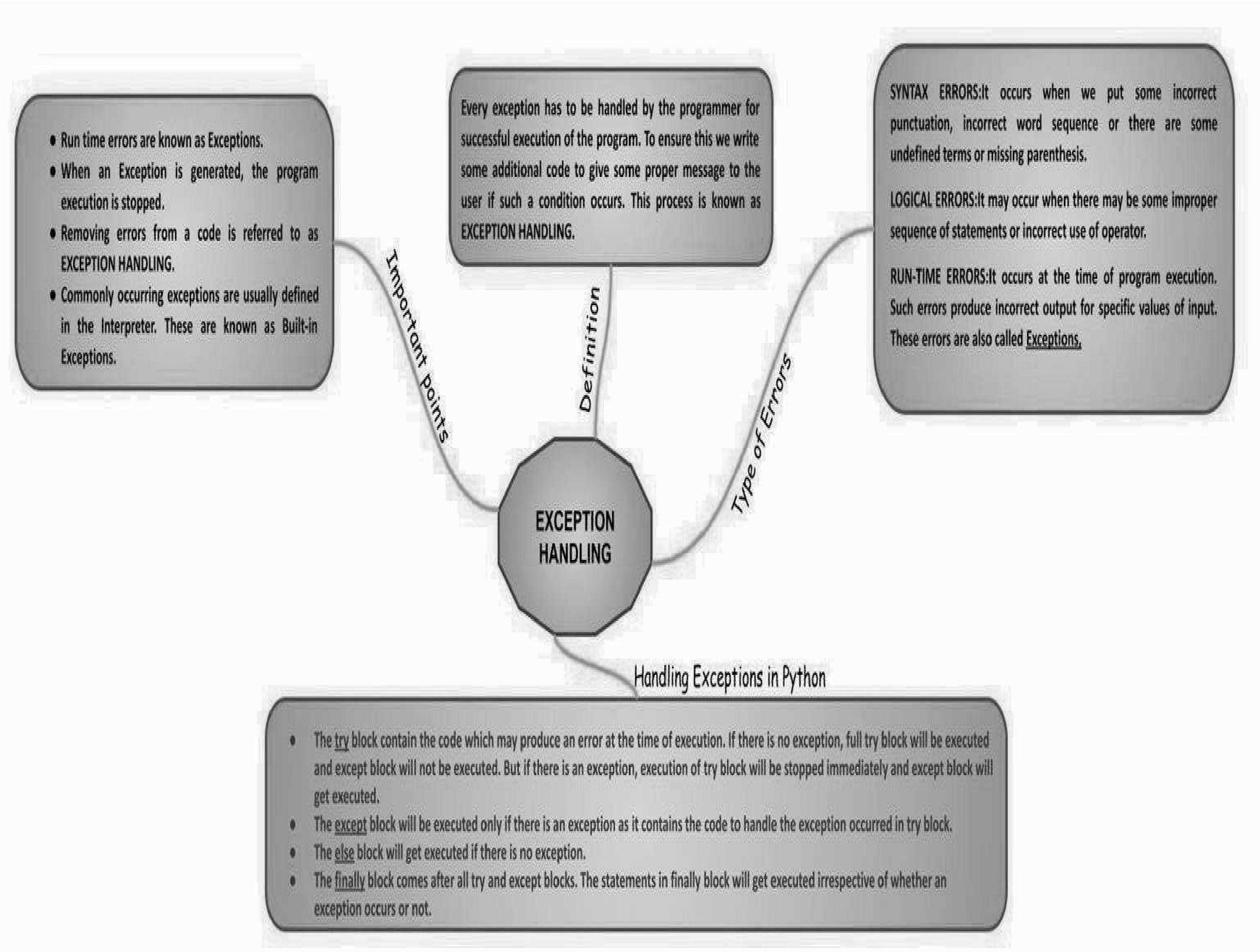
1.	Write a Python function named `calculate_gross_salary` that takes the basic salary (an integer) as an argument and returns the gross salary. The gross salary is calculated as follows: - If the basic salary is less than or equal to 10000, the gross salary is the basic salary plus 20% of the basic salary. - If the basic salary is more than 10000, the gross salary is the basic salary plus 25% of the basic salary. Write the function definition for `calculate_gross_salary` and use it to calculate the gross salary for a basic salary of 12000.
2.	Write a Python function called calculate_average that takes a list of numbers as input and returns the average of those numbers.
3.	Write a function **update_list(L)** that receives a list as arguments and increases the value of even elements by 2. For example: if the list is [2,4,6,7,9] after the execution of the function, the list should be: [4,6,8,7,9]
4.	Write a function **count_vowels(string)** that counts the number of vowels (a, e, i, o, u) in a string.
5.	Create a function **min_max_avg(numbers)** that returns the minimum, maximum, and average of a list of numbers.
6.	Write a function **sum_list(numbers)** that takes a list of numbers and returns their sum.

EXCEPTION HANDLING



Topics to be covered:

- Introduction
- Types of errors
- Handling Exceptions Using Try–Except–Finally Blocks



EXCEPTION HANDLING

Important points

- Run time errors are known as Exceptions.
- When an Exception is generated, the program execution is stopped.
- Removing errors from a code is referred to as EXCEPTION HANDLING.
- Commonly occurring exceptions are usually defined in the Interpreter. These are known as Built-in Exceptions.

Definition

Every exception has to be handled by the programmer for successful execution of the program. To ensure this we write some additional code to give some proper message to the user if such a condition occurs. This process is known as EXCEPTION HANDLING.

Type of Errors

SYNTAX ERRORS:It occurs when we put some incorrect punctuation, incorrect word sequence or there are some undefined terms or missing parenthesis.

LOGICAL ERRORS:It may occur when there may be some improper sequence of statements or incorrect use of operator.

RUN-TIME ERRORS:It occurs at the time of program execution. Such errors produce incorrect output for specific values of input. These errors are also called Exceptions.

Handling Exceptions in Python

- The try block contain the code which may produce an error at the time of execution. If there is no exception, full try block will be executed and except block will not be executed. But if there is an exception, execution of try block will be stopped immediately and except block will get executed.
- The except block will be executed only if there is an exception as it contains the code to handle the exception occurred in try block.
- The else block will get executed if there is no exception.
- The finally block comes after all try and except blocks. The statements in finally block will get executed irrespective of whether an exception occurs or not.

EXCEPTION HANDLING

While executing a Python program, it may happen that the program does not execute at all or it can generate unexpected output. This happens when there are syntax errors, run time errors, logical errors or any semantic errors in the code.

SYNTAX ERROR	LOGICAL ERROR	RUN TIME ERROR

It occurs when we put some incorrect punctuation, incorrect word sequence or there are some undefined terms or missing parenthesis. Syntax errors are also known as **Parsing errors.**
For example:
>>> p=2(num1+num2)
This statement is mathematically correct but the Python interpreter will raise a SYNTAX error as there is no sign present between 2 and parenthesis. The correct statement will be:
>>> p=2*(num1+num2)
	It may occur when there may be some improper sequence of statements or incorrect use of operators. It will not stop a program from executing but will produce incorrect output. It will generate incorrect output for every value of input.
	For example:
	If we want to find the sum of two numbers, write the following code:
	A, B = 10, 15
	C = A * B
	print (“Sum is: “, C)
	Here, the code will generate A * B but we wanted to find Sum. Hence it is a logical error.
		It occurs at the time of program execution. Such errors produce incorrect output for specific values of input. These errors are also called **Exceptions,** which occur when something unexpected happens leading to stopping the program execution.
		For example:
		1. Division by zero.
		2. Finding the square root of a negative number.
		3. Insufficient memory is available on the computer.
		4. Trying to open a file that does not exist.

EXCEPTIONS:

- Run time errors are known as Exceptions.
- When an Exception is generated, the program execution is stopped.
- Removing errors from a code is referred to as EXCEPTION HANDLING.
- Commonly occurring exceptions are usually defined in the Interpreter. These are known as Built-in Exceptions.

Some Built-in Exceptions are listed below:

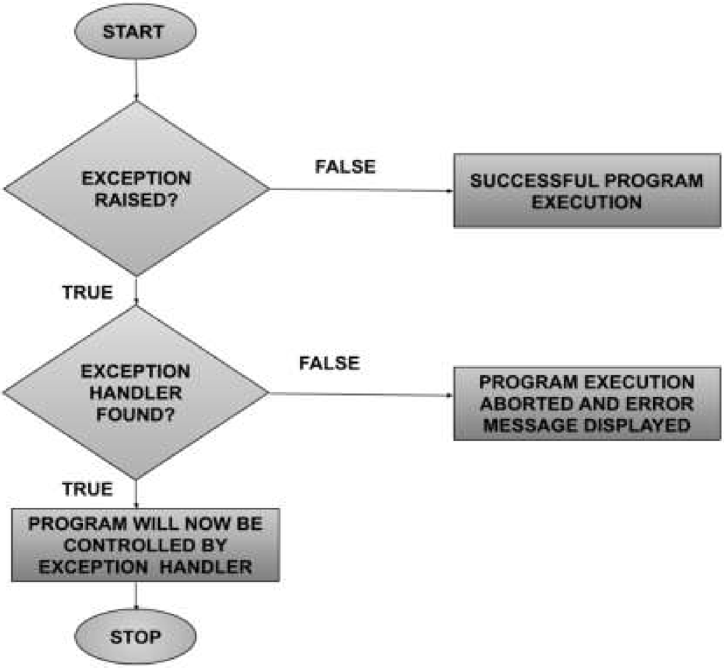
EXCEPTION	DESCRIPTION
ZeroDivisionError	It is raised when an expression or a value is getting divided by zero (0). For example: If c = 0, then p = b/c will result in 'ZeroDivisionError'.
NameError	It is raised when an identifier is not assigned any value earlier and is being used in some expression. For example: if p = a*b/c then it will result in 'NameError' when one or more variables are not assigned values.
TypeError	It is raised when variables used in any expression have values of different data types. For example: if p=(a+b)/c then it will result in 'TypeError' when the variables a, b and c are of different data types.
ValueError	It is raised when the given value of a variable is of the right data type but not appropriate according to the expression.
IOError	It is raised when the file specified in a program statement cannot be opened.
IndexError	It is raised when the index of a sequence is out of the range.
KeyError	It is raised when a key doesn't exist or is not found in a dictionary.

EXCEPTION HANDLING:

Every exception has to be handled by the programmer for the successful execution of the program. To ensure this, we write some additional codes to give a proper message to the user, if such a condition occurs. This process is known as **EXCEPTION HANDLING.**

Exception handlers separate the main logic of the program from the error detection and correction code. The segment of code where there is any possibility of error or exception is placed inside one block and the code to be executed in this case is placed inside another block. These statements for detection and reporting the execution do not affect the main logic of the program.

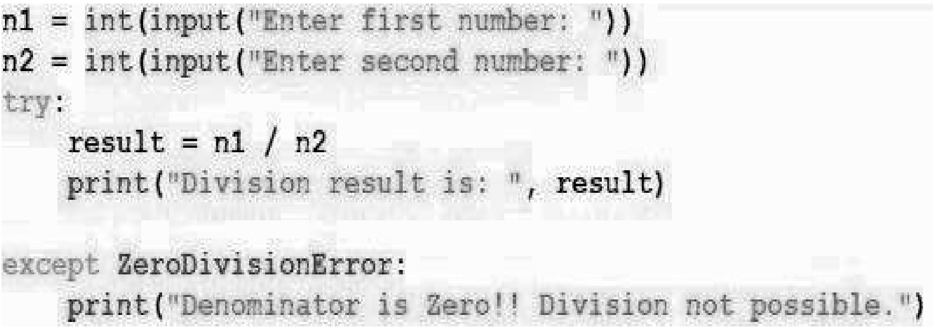
STEPS FOR EXCEPTION HANDLING:


START
EXCEPTION RAISED?
FALSE
SUCCESSFUL PROGRAM EXECUTION
TRUE
EXCEPTION HANDLER FOUND?
FALSE
PROGRAM EXECUTION ABORTED AND ERROR MESSAGE DISPLAYED
TRUE
PROGRAM WILL NOW BE CONTROLLED BY EXCEPTION HANDLER
STOP

In Python, exceptions, if any, are handled by using the try-except-finally block. While writing code, a programmer might doubt a particular part of code to raise an exception. Such suspicious lines of code are written inside a **try** block which will be followed by an **except** block. The code to handle every possible exception that may arise in the try block, will be written inside the **except** block. An exception is said to be caught when a code designed for handling it is executed.

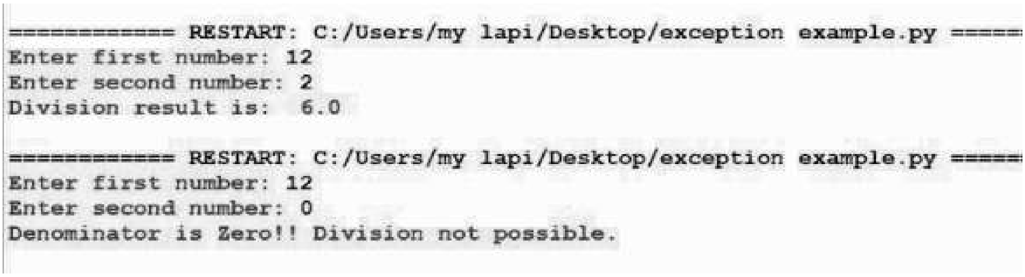
If no exception occurs during the execution of the program, the program produces the desired output successfully. But if an exception is encountered, further execution of the code inside the try block will be stopped and the control flow will be transferred to the except block.

Example 1:



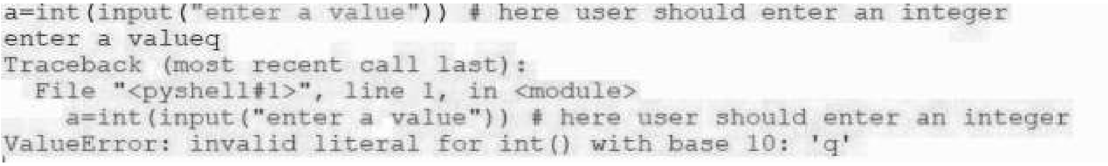
```
n1 = int(input("Enter first number: "))
n2 = int(input("Enter second number: "))
try:
    result = n1 / n2
    print("Division result is: ", result)

except ZeroDivisionError:
    print("Denominator is Zero!! Division not possible.")
```

The output will be:



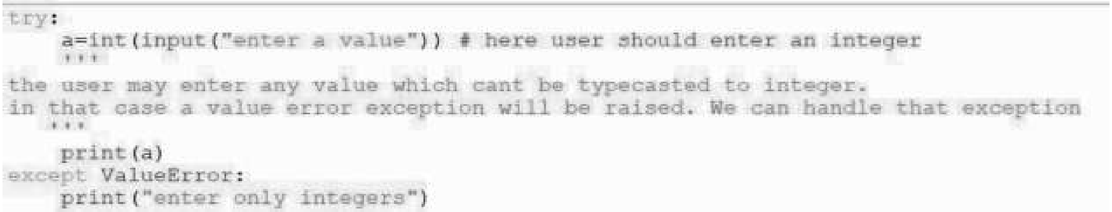
```
============ RESTART: C:/Users/my lapi/Desktop/exception example.py =====
Enter first number: 12
Enter second number: 2
Division result is:  6.0

============ RESTART: C:/Users/my lapi/Desktop/exception example.py =====
Enter first number: 12
Enter second number: 0
Denominator is Zero!! Division not possible.
```

Example 2:



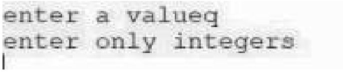
```
a=int(input("enter a value")) # here user should enter an integer
enter a valueq
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    a=int(input("enter a value")) # here user should enter an integer
ValueError: invalid literal for int() with base 10: 'q'
```

In the above example, the user entered a wrong value that raised ValueError. We can handle this exception by using the **ValueError exception.**



```
try:
    a=int(input("enter a value")) # here user should enter an integer
    '''
the user may enter any value which cant be typecasted to integer.
in that case a value error exception will be raised. We can handle that exception
   '''
    print(a)
except ValueError:
    print("enter only integers")
```

Result:

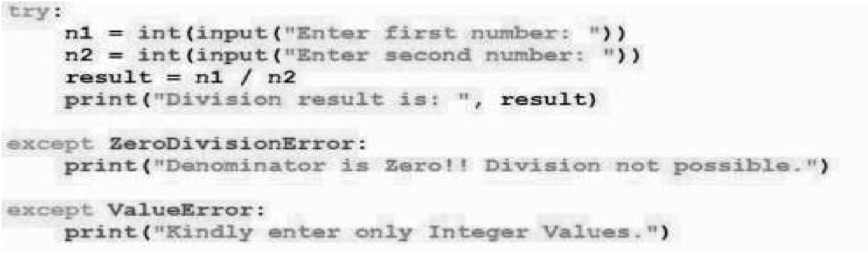


```
enter a valueq
enter only integers
```

Use of multiple "except" blocks:

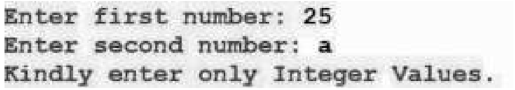
Sometimes, a single piece of code in a program may have more than one type of error. If such an event happens, we can use multiple **except** blocks for a single **try** block.

Example 1:



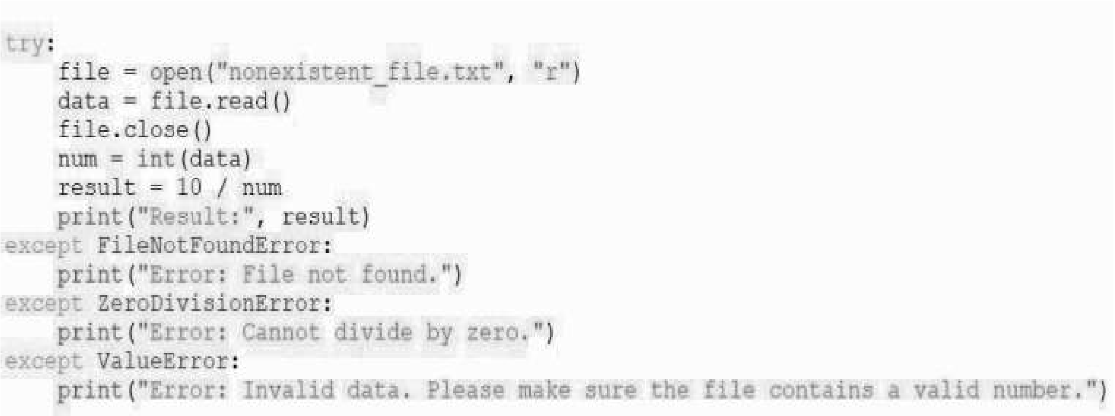
```
try:
    n1 = int(input("Enter first number: "))
    n2 = int(input("Enter second number: "))
    result = n1 / n2
    print("Division result is: ", result)

except ZeroDivisionError:
    print("Denominator is Zero!! Division not possible.")

except ValueError:
    print("Kindly enter only Integer Values.")
```

The output will be:



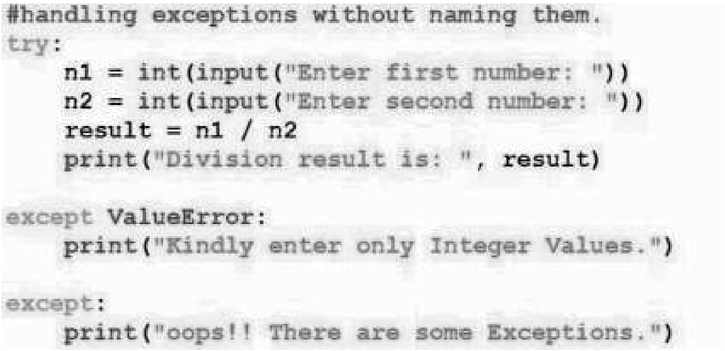
```
Enter first number: 25
Enter second number: a
Kindly enter only Integer Values.
```

Example 2:



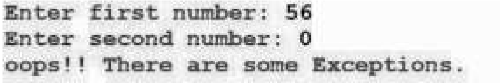
```
try:
    file = open("nonexistent_file.txt", "r")
    data = file.read()
    file.close()
    num = int(data)
    result = 10 / num
    print("Result:", result)
except FileNotFoundError:
    print("Error: File not found.")
except ZeroDivisionError:
    print("Error: Cannot divide by zero.")
except ValueError:
    print("Error: Invalid data. Please make sure the file contains a valid number.")
```

We can also handle exceptions without naming them.



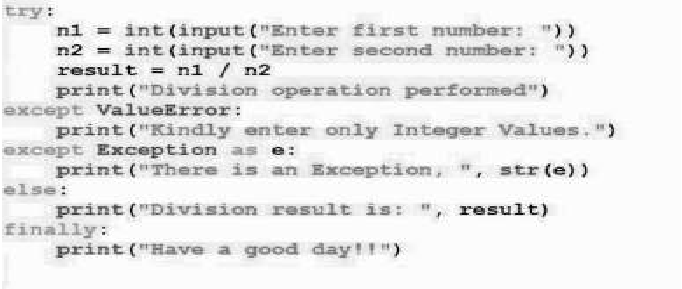
```
#handling exceptions without naming them.
try:
    n1 = int(input("Enter first number: "))
    n2 = int(input("Enter second number: "))
    result = n1 / n2
    print("Division result is: ", result)

except ValueError:
    print("Kindly enter only Integer Values.")

except:
    print("oops!! There are some Exceptions.")
```

The output will be:



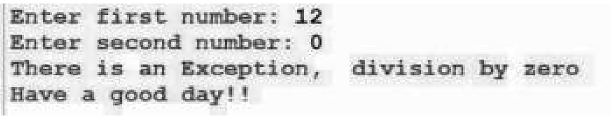
```
Enter first number: 56
Enter second number: 0
oops!! There are some Exceptions.
```

Default exception messages can also be displayed when we are not handling exceptions by name.



```
try:
    n1 = int(input("Enter first number: "))
    n2 = int(input("Enter second number: "))
    result = n1 / n2
    print("Division operation performed")
except ValueError:
    print("Kindly enter only Integer Values.")
except Exception as e:
    print("There is an Exception, ", str(e))
else:
    print("Division result is: ", result)
finally:
    print("Have a good day!!")
```

The output will be:

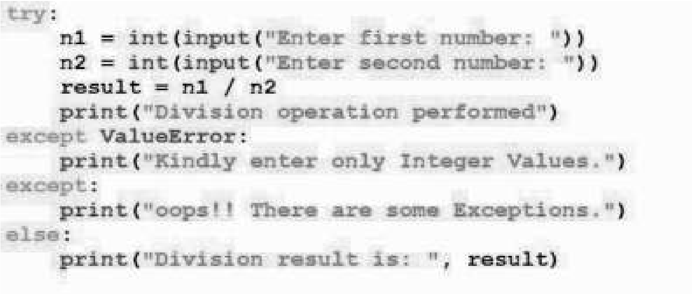


```
Enter first number: 12
Enter second number: 0
There is an Exception,  division by zero
Have a good day!!
```

try…except…else Clause:

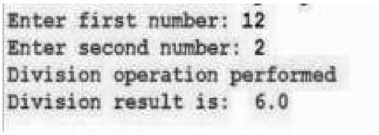
Just like Conditional and Iterative statements we can use an optional **else** clause along with the **try…except** clause. An **except** block will be executed only when some exceptions are raised in the try block. But if there is no error then except blocks will not be executed.

In this case, the else clause will be executed.



```
try:
    n1 = int(input("Enter first number: "))
    n2 = int(input("Enter second number: "))
    result = n1 / n2
    print("Division operation performed")
except ValueError:
    print("Kindly enter only Integer Values.")
except:
    print("oops!! There are some Exceptions.")
else:
    print("Division result is: ", result)
```

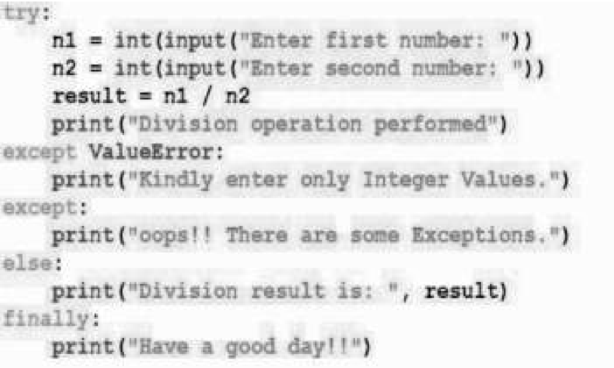
The output will be:



```
Enter first number: 12
Enter second number: 2
Division operation performed
Division result is:  6.0
```

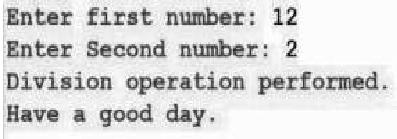
finally CLAUSE:

The **try…except…else** block in Python has an optional **finally** clause. The statements inside the **finally** block are always executed whether an exception has occurred in the try block or not. If we want to use the **finally** block, it should always be placed at the end of the clause i.e. after all except blocks and the else block.



```
try:
    n1 = int(input("Enter first number: "))
    n2 = int(input("Enter second number: "))
    result = n1 / n2
    print("Division operation performed")
except ValueError:
    print("Kindly enter only Integer Values.")
except:
    print("oops!! There are some Exceptions.")
else:
    print("Division result is: ", result)
finally:
    print("Have a good day!!")
```

The output will be:



```
Enter first number: 12
Enter Second number: 2
Division operation performed.
Have a good day.
```

Assignment

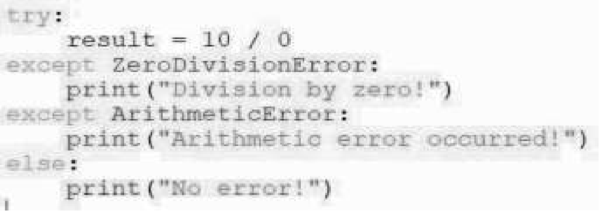
1 In a try-except block, can there be multiple 'except' clauses?

a) No, there can be only one 'except' clause.
b) Yes, but only if the exceptions are of the same type.
c) Yes, it allows handling different exceptions separately.
d) No, 'except' clauses are not allowed in a try-except block.

2 When might you use the 'finally' block in exception handling?

a) To handle exceptions that are expected to occur frequently.
b) To provide a resolution for every possible error.
c) To close resources that were opened in the 'try' block, regardless of whether an exception occurred or not.
d) To avoid having to use 'except' blocks.

3 What will be the output of the following code snippet?



```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Division by zero!")
except ArithmeticError:
    print("Arithmetic error occurred!")
else:
    print("No error!")
```

a) Division by zero!
b) An arithmetic error occurred!
c) No error!
d) This code will raise a syntax error.

4 Which of the following is NOT a standard built-in exception in Python?
a) ValueError
b) IndexError
c) ExceptionError
d) KeyError

5 What is an exception in programming?
a) An error that occurs during runtime
b) A warning message from the compiler
c) A comment in the code
d) A statement that terminates the program

6 What is the purpose of the "try" block in a try-except construct?
a) To handle the exception by executing specific code
b) To specify the type of exception to be thrown
c) To define a custom exception class
d) To ensure a specific block of code always executes

7 **Assertion (A):** In Python, the "try-except” block is used to handle exceptions.
Reasoning (R): In the "try" block, the program attempts to execute code that might result in an exception. If an exception occurs, it is handled in the corresponding "except" block.

A. Both A and R are true and R is the correct explanation of A
B. Both A and R are true but R is not the correct explanation of A
C. A is True but R is False
D. R is True but A is False

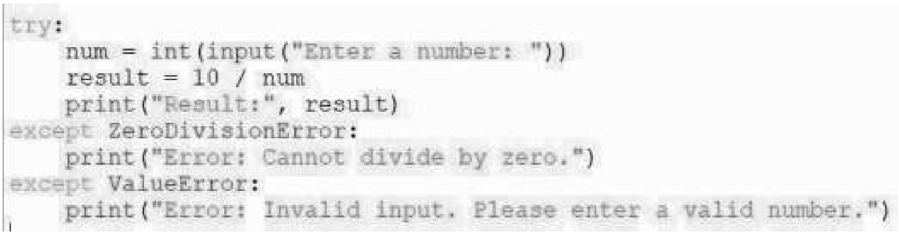
8 **Assertion (A):** The "finally" block is mandatory after the try block.
Reasoning (R): The "finally" block contains code that is guaranteed to execute, whether an exception occurs within the "try" block or not.

A. Both A and R are true and R is the correct explanation of A
B. Both A and R are true but R is not the correct explanation of A
C. A is True but R is False
D. R is True but A is False

9 **Assertion (A):** Python allows multiple "except" blocks to be used within a single "try" block to handle different exceptions.
Reasoning (R): By using multiple "except" blocks with different exception types, Python provides the flexibility to handle various types of exceptions separately.

A. Both A and R are true and R is the correct explanation of A
B. Both A and R are true but R is not the correct explanation of A
C. A is True but R is False
D. R is True but A is False

10 Code snippet:



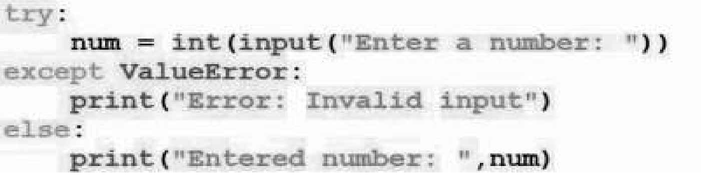
```
try:
    num = int(input("Enter a number: "))
    result = 10 / num
    print("Result:", result)
except ZeroDivisionError:
    print("Error: Cannot divide by zero.")
except ValueError:
    print("Error: Invalid input. Please enter a valid number.")
```

Predict the output when:

a) The user enters "0".
b) The user enters "5".
c) The user enters "abc".

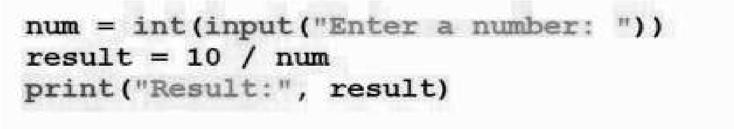
11 State whether the following statement is **True** or **False**:
An exception may be raised even if the program is syntactically correct.

12 What will be the output of the following code if the input is 2.2:



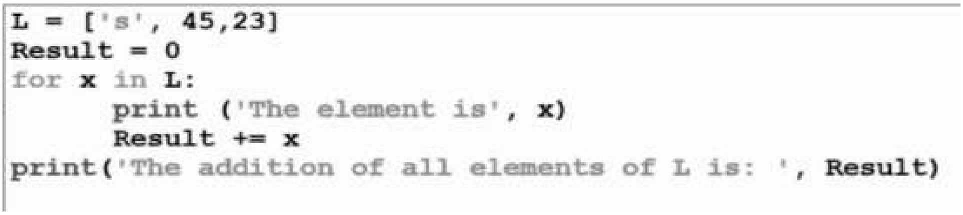
```
try:
    num = int(input("Enter a number: "))
except ValueError:
    print("Error: Invalid input")
else:
    print("Entered number: ",num)
```

13 Rewrite the following code after handling all possible exceptions.



```
num = int(input("Enter a number: "))
result = 10 / num
print("Result:", result)
```

14 Consider the code given below:



```
L = ['s', 45,23]
Result = 0
for x in L:
      print ('The element is', x)
      Result += x
print('The addition of all elements of L is: ', Result)
```

Which of the following errors will be raised by the given Python code?

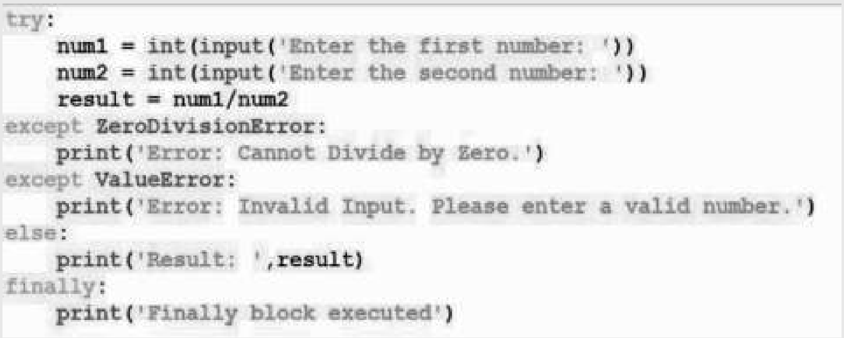
a) NameError

b) ValueError

c) TypeError

d) IOError

15 Code snippet:



```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    result = num1/num2
except ZeroDivisionError:
    print('Error: Cannot Divide by Zero.')
except ValueError:
    print('Error: Invalid Input. Please enter a valid number.')
else:
    print('Result: ',result)
finally:
    print('Finally block executed')
```

Predict the output when:

a) The user enters "10" for both numbers.

b) The user enters "5" for the first number and "0" for the second number.

c) The user enters "abc" for the first number.

16 Which of the following statements is true?

a). The standard exceptions are automatically imported into Python programs.

b). All raised standard exceptions must be handled in Python.

c). When there is a deviation from the rules of a programming language, a semantic error is thrown.

d). If any exception is thrown in the try block, the else block is executed.

17 Identify the statement(s) from the following options which will raise TypeError exception(s):

a) print('5') b) print(5 * 3) c) print('5' +3) d) print('5' + '3')

LAB EXERCISE

1 Create a simple calculator program that takes two numbers and an operator (+, -, *, /) as input. Implement exception handling to handle cases like division by zero and invalid operators.

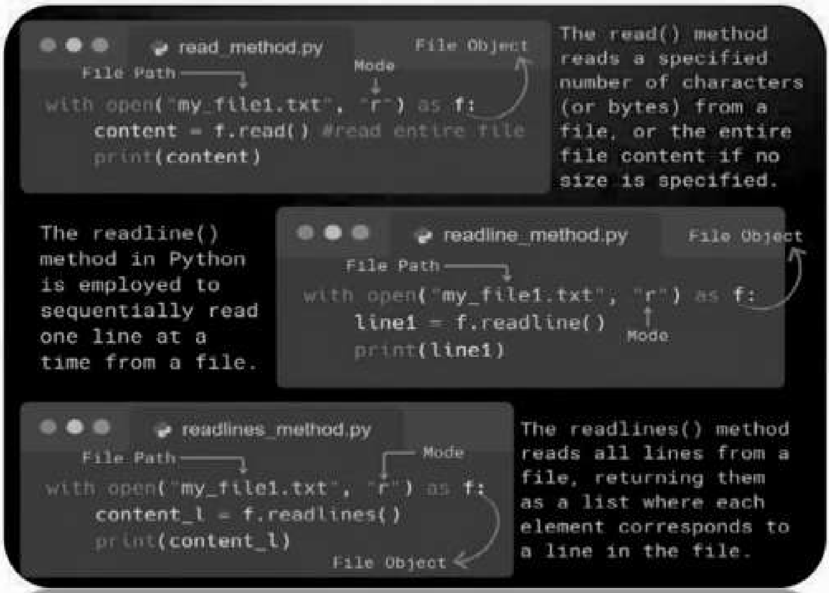
2 Create a program that asks the user for an integer input. Use exception handling to ensure that the input is a valid integer. If the user provides invalid input, prompt them to retry until a valid integer is entered.

3 Create a program that connects to a database and performs database operations (e.g., insert, update, delete). Use exception handling to deal with database-related exceptions, such as connection errors or SQL syntax errors.

4 Create a program that reads data from a file specified by the user. Implement exception handling to catch and handle the "FileNotFoundError" exception if the file does not exist.

5 Define a dictionary with some key-value pairs. Ask the user for a key and attempt to access the corresponding value from the dictionary. Handle the "KeyError" exception and display a message if the key is not found.

FILE HANDLING



```
with open("my_file1.txt", "r") as f:
    content = f.read() #read entire file
    print(content)
```

The read() method reads a specified number of characters (or bytes) from a file, or the entire file content if no size is specified.

The readline() method in Python is employed to sequentially read one line at a time from a file.

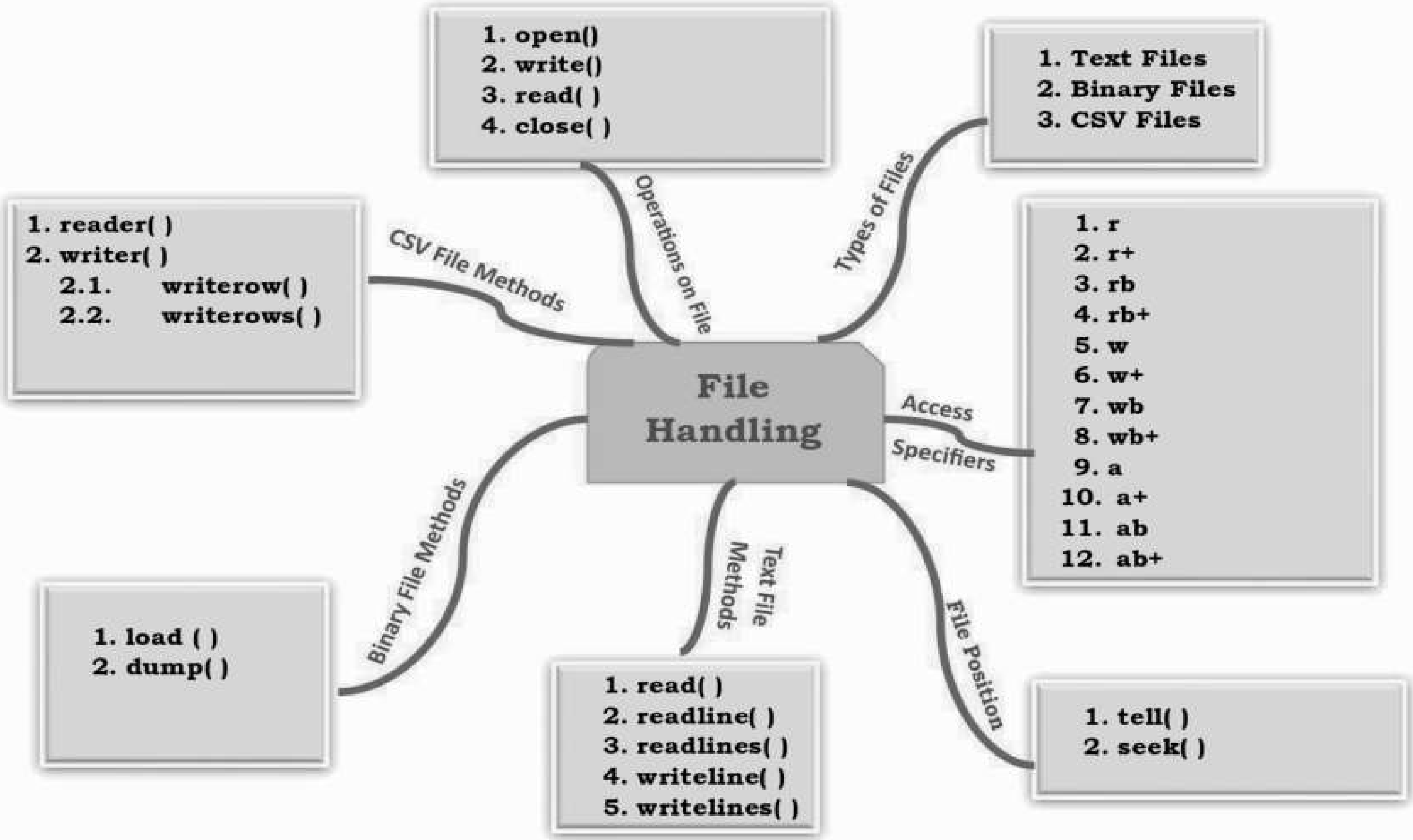
```
with open("my_file1.txt", "r") as f:
    line1 = f.readline()
    print(line1)
```

```
with open("my_file1.txt", "r") as f:
    content_l = f.readlines()
    print(content_l)
```

The readlines() method reads all lines from a file, returning them as a list where each element corresponds to a line in the file.

Topic to be covered:

- Introduction
- Types of files
- Access Specifiers in the file
- seek ( ) and tell( )
- Text files and its methods
- Binary files and its methods
- CSV files and its methods
- Absolute and Relative Paths



File Handling

Operations on File

1. open()
2. write()
3. read( )
4. close( )

Types of Files

1. Text Files
2. Binary Files
3. CSV Files

CSV File Methods

1. reader( )
2. writer( )
   2.1. writerow( )
   2.2. writerows( )

Access Specifiers

1. r
2. r+
3. rb
4. rb+
5. w
6. w+
7. wb
8. wb+
9. a
10. a+
11. ab
12. ab+

Binary File Methods

1. load ( )
2. dump( )

Text File Methods

1. read( )
2. readline( )
3. readlines( )
4. writeline( )
5. writelines( )

File Position

1. tell( )
2. seek( )

FILE HANDLING

Files are used to store data permanently and can be retrieved later.

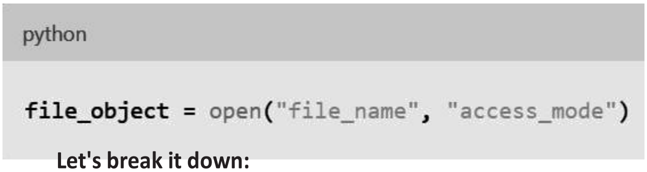
Type of Files

1. Text Files
2. Binary Files
3. CSV Files

Steps for File Handling:

1. Open File
2. Processing file i.e Read or Write from/to file
3. Close File

1. Opening Files:



```
file_object = open("file_name", "access_mode")
```

Let's break it down:

I. **file_object:** This is a variable that will hold the file object returned by the `open()` function. You'll use this variable to perform operations like reading from or writing to the file.

II. **open():** This is the built-in function used to open files in Python.

III. **access_mode:** This specifies the mode in which you want to open the file.

There are two ways of opening a file in Python

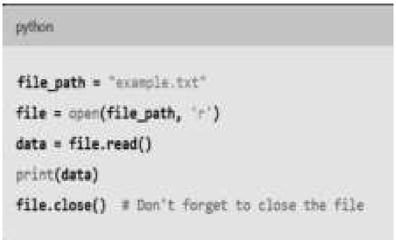
a. open()

b. Open using ‘with’ statement.

open() Function:	with Statement:
● It's like turning the key to open a door. ● You use it to open a file and get access to its contents.	● It's like having a helper who opens and closes the door for you automatically. ● You tell the helper what you want to do with the door (read, write,

<table>
<tr>
<td>

- After you're done reading or writing, you have to remember to close the file yourself, just like locking the door when you're done.



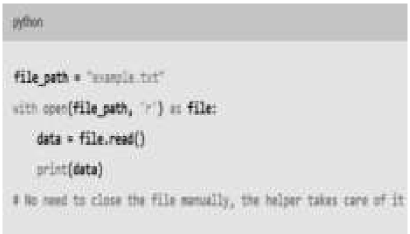
python

```python
file_path = "example.txt"
file = open(file_path, 'r')
data = file.read()
print(data)
file.close()  # Don't forget to close the file
```

</td>
<td>

etc.), and they handle everything for you.

- They make sure the door is properly closed even if something unexpected happens, like a sudden gust of wind.



python

```python
file_path = "example.txt"
with open(file_path, 'r') as file:
    data = file.read()
    print(data)
# No need to close the file manually, the helper takes care of it
```

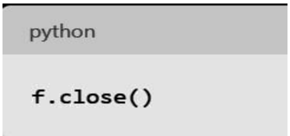
</td>
</tr>
</table>

Best Choice:

- Using `with` is usually better because it's safer and saves you from having to remember to close the file.
- It's like having a reliable assistant who ensures everything is handled smoothly.

2. Closing Files:

- It's important to close files after you're done working with them to release system resources.



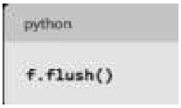
python

```python
f.close()
```

- Not closing files can lead to resource leaks and other issues.

3. Buffering and Flushing:

- When writing to a file, data might be stored in an output buffer until the file is closed or explicitly flushed.
- The `flush()` method forces the data waiting in the buffer to be immediately written into the file without waiting for the file to be closed.



python

```python
f.flush()
```

Access Specifiers in Files:

Access Mode for Text Files	Access Mode for Binary Files	Access Mode for CSV Files	Description	File Pointer Position
r	rb	r	Read mode. Opens a file for reading. If the file does not exist, open() raises a FileNotFoundError.	Beginning of File
r+	rb+		It opens the file for both reading and writing. If the file does not exist, open() raises a FileNotFoundError.	Beginning of File
w	wb	w	Write mode. It opens the file for writing only. If the file exists, the content of the file will be removed. If the file does not exist, it is created.	Beginning of File
w+	wb+		The w+ mode opens the file for both writing and reading. Like w, if the file exists, then the content of the file will be removed. If the file does not exist, it is created.	Beginning of File
a+	ab+		The a+ mode opens the file for both appending and reading. In this, the new content is added after the existing content. If the file does not exist, it is created.	End of File
a	ab	a	The a mode opens the file for appending. In this, the new content is added after the existing content. If the file does not exist, it creates a new file.	End of File

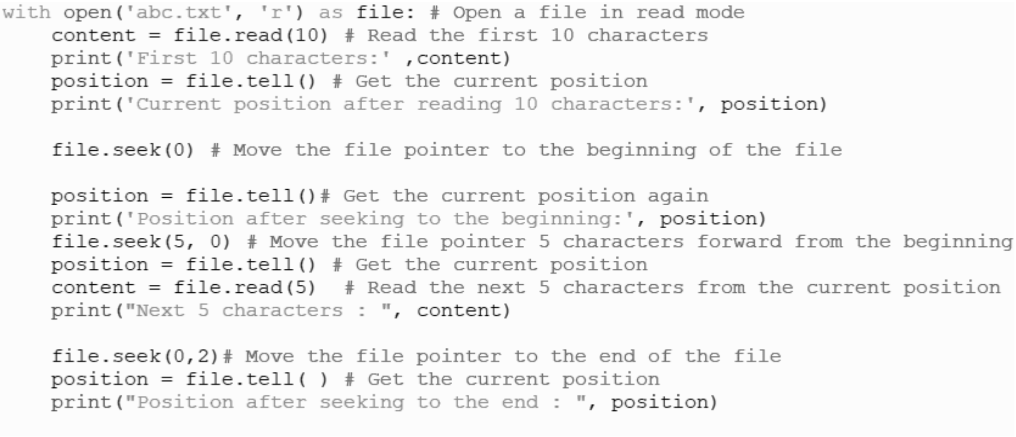
The default mode for file opening is “r” read mode. If we didn’t specify mode during the opening of the file then it will automatically open the file in read mode.

File Object Methods (seek( ) & tell( ))

Method	Prototype	Description
seek( )	**Syntax:** **<file**_object>.seek(<offset>,<whence>) offset: Number of bytes to move the file pointer. whence: Position reference: ● 0: Beginning of the file (default). ● 1: Current file position. ● 2: End of the file.	● It is used to change the position of the **File Handle** to a given specific position. ● A file handle is like a cursor, which defines from where the data has to be read or written in the file. **Note:** In the case of a text file we can use only ‘**0**’ as a reference point.
tell( )	**Syntax: <file_object>.tell( )** i.e. position=f.tell( ) where f is the file handle.	● returns the current position of the file object. ● It takes no parameters ● It returns an integer value.

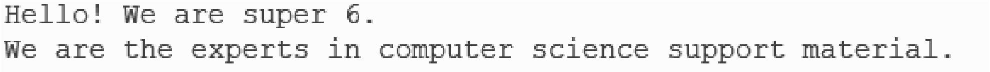
Example:

Python Code:



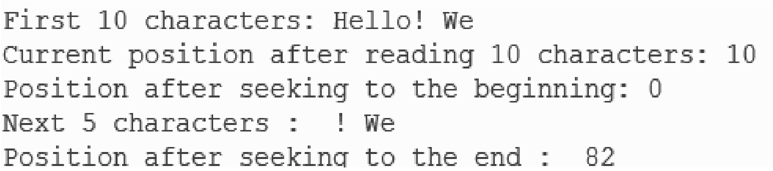
```python
with open('abc.txt', 'r') as file: # Open a file in read mode
    content = file.read(10) # Read the first 10 characters
    print('First 10 characters:' ,content)
    position = file.tell() # Get the current position
    print('Current position after reading 10 characters:', position)

    file.seek(0) # Move the file pointer to the beginning of the file

    position = file.tell()# Get the current position again
    print('Position after seeking to the beginning:', position)
    file.seek(5, 0) # Move the file pointer 5 characters forward from the beginning
    position = file.tell() # Get the current position
    content = file.read(5)  # Read the next 5 characters from the current position
    print("Next 5 characters : ", content)

    file.seek(0,2)# Move the file pointer to the end of the file
    position = file.tell( ) # Get the current position
    print("Position after seeking to the end : ", position)
```

Content of Text File:



```
Hello! We are super 6.
We are the experts in computer science support material.
```

Output after executing above code:



```
First 10 characters: Hello! We
Current position after reading 10 characters: 10
Position after seeking to the beginning: 0
Next 5 characters :  ! We
Position after seeking to the end :  82
```

Text Files:

- It stores information in the form of ASCII or Unicode characters
- Each line of text is terminated with a special character called EOL (End of Line), which is the new line character ('\n') in Python by default.
- The file extension will be .txt

Working with Text Files:

1. Reading data from a file.
2. Writing data into a file.

Reading data from files

There are three ways to read data from text files:

1. read function i.e. read( )
2. readline function i.e. readline( )
3. readlines function i.e. readlines( )

read( ): It is used in text files to read a specified number of data bytes from the file. It returns the result in the form of a string.

Syntax: file_object.read( )

file_pointer.read(n): It will read the maximum number of n bytes/characters from the file.

f.read(7) # It will read 7 bytes/characters from the position of the file pointer.

file_pointer.read( ): It will read the entire content of the file.

f.read( ) # It will read all the data of the file from the position of the file pointer.

readline( ): It will read one complete line in one go from the file. It returns the data in the form of a string.

Syntax: file_object.readline( )

file_pointer.readline( ): It will read the entire line.

f.readline( ) #it will read one complete line in one go.

file_pointer.readline(n): It will read the first 'n' bytes from the file.

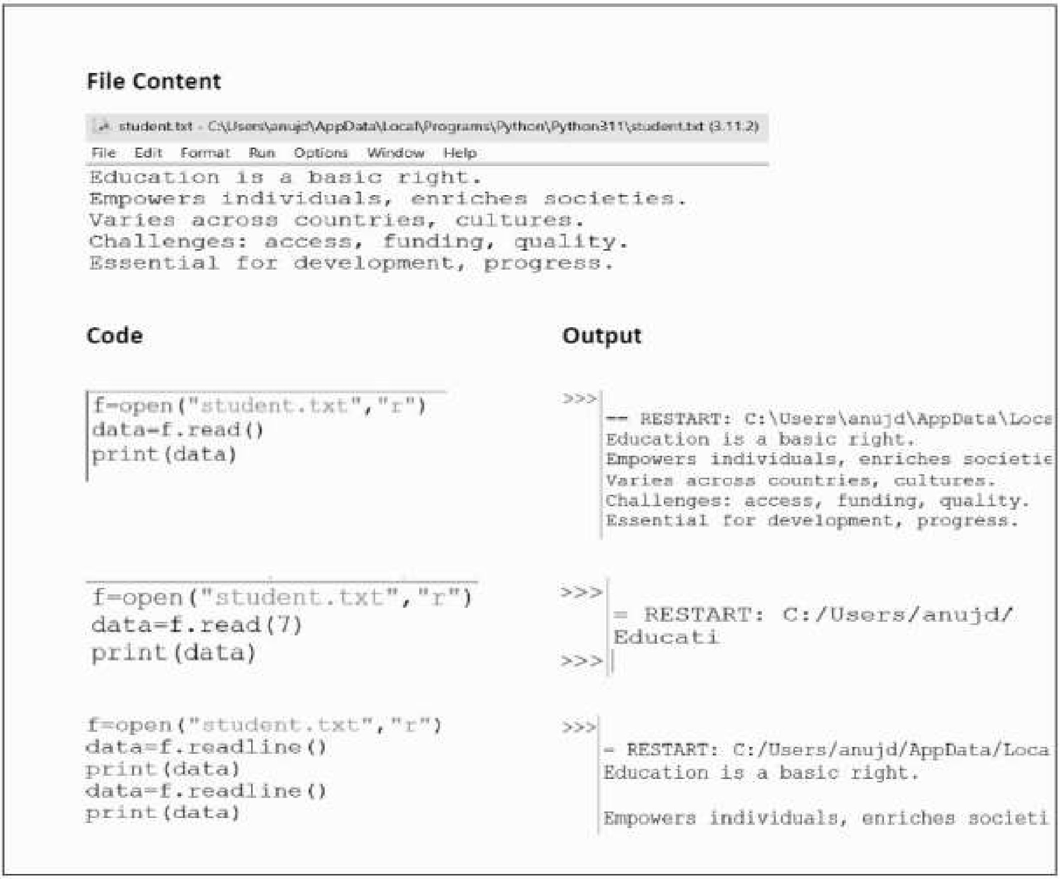
f.readline(5) #it will read the first 5 characters/bytes from the file.

readlines( ): It will return all the lines of the file as the elements of the list. I.e. the 1st line of the file will be the first element of the list and so on.

Syntax: file_object.readlines( )

file_pointer.readlines( ): It will read all the lines of the file as the elements of the list.

f.readlines( ) #it will read all the lines of the file as the elements of the list.



File Content

student.txt - C:\Users\anujd\AppData\Local\Programs\Python\Python311\student.txt (3.11.2)

```
Education is a basic right.
Empowers individuals, enriches societies.
Varies across countries, cultures.
Challenges: access, funding, quality.
Essential for development, progress.
```

Code

Output

```
f=open("student.txt","r")
data=f.read()
print(data)
```

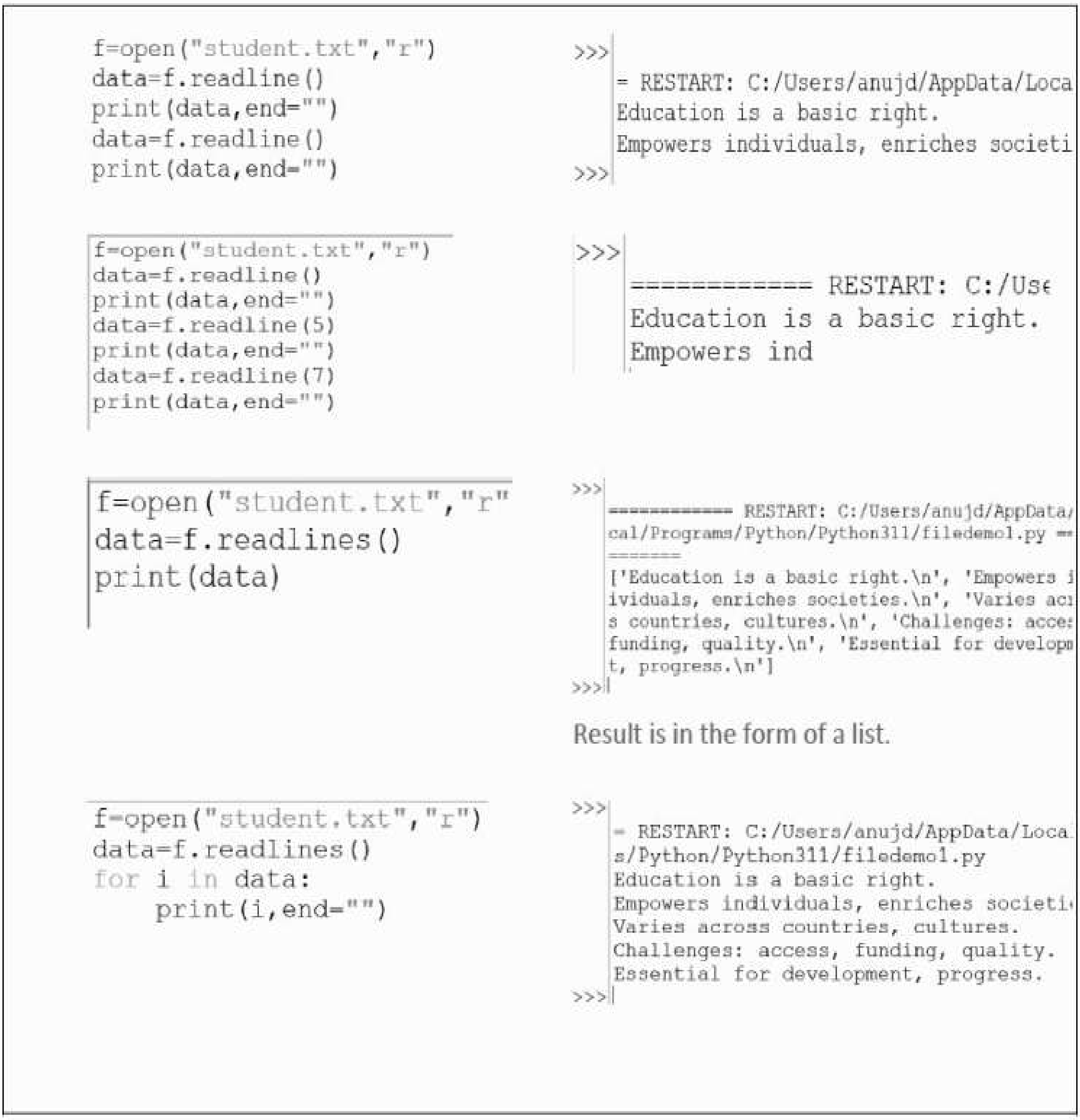
```
>>>
-- RESTART: C:\Users\anujd\AppData\Loca
Education is a basic right.
Empowers individuals, enriches societie
Varies across countries, cultures.
Challenges: access, funding, quality.
Essential for development, progress.
```

```
f=open("student.txt","r")
data=f.read(7)
print(data)
```

```
>>>
= RESTART: C:/Users/anujd/
Educati
>>>
```

```
f=open("student.txt","r")
data=f.readline()
print(data)
data=f.readline()
print(data)
```

```
>>>
= RESTART: C:/Users/anujd/AppData/Loca
Education is a basic right.

Empowers individuals, enriches societi
```



```
f=open("student.txt","r")
data=f.readline()
print(data,end="")
data=f.readline()
print(data,end="")
```

```
>>>
= RESTART: C:/Users/anujd/AppData/Loca
Education is a basic right.
Empowers individuals, enriches societi
>>>
```

```
f=open("student.txt","r")
data=f.readline()
print(data,end="")
data=f.readline(5)
print(data,end="")
data=f.readline(7)
print(data,end="")
```

```
>>>
============ RESTART: C:/Use
Education is a basic right.
Empowers ind
```

```
f=open("student.txt","r"
data=f.readlines()
print(data)
```

```
>>>
============ RESTART: C:/Users/anujd/AppData/
cal/Programs/Python/Python311/filedemo1.py ==
=======
['Education is a basic right.\n', 'Empowers i
ividuals, enriches societies.\n', 'Varies acr
s countries, cultures.\n', 'Challenges: acces
funding, quality.\n', 'Essential for developm
t, progress.\n']
>>>
```

Result is in the form of a list.

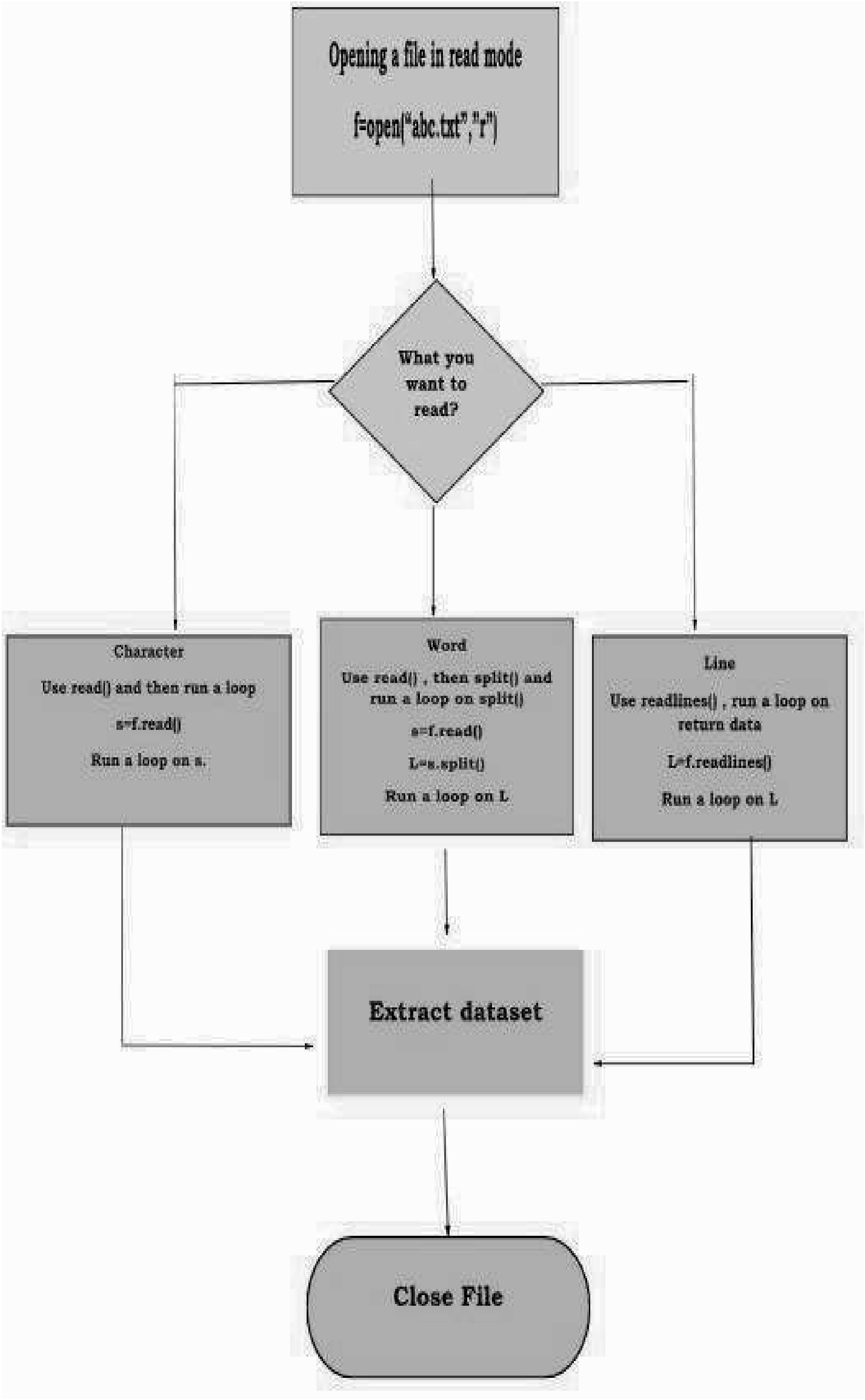
```
f=open("student.txt","r")
data=f.readlines()
for i in data:
    print(i,end="")
```

```
>>>
= RESTART: C:/Users/anujd/AppData/Loca
s/Python/Python311/filedemo1.py
Education is a basic right.
Empowers individuals, enriches societi
Varies across countries, cultures.
Challenges: access, funding, quality.
Essential for development, progress.
>>>
```

Tips on writing text file code in exams:

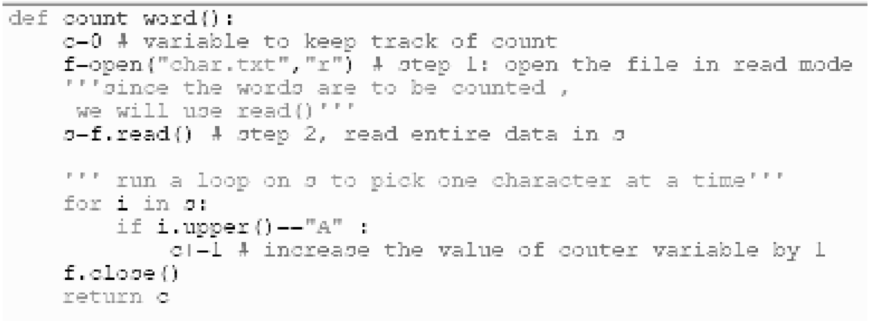
- Let the file name be "abc.txt".
- Read the question carefully and find out what has to be read from the file.

Follow the below flow chart to write the code.


Opening a file in read mode
f=open("abc.txt","r")
What you want to read?
Character
Use read() and then run a loop
s=f.read()
Run a loop on s.
Word
Use read() , then split() and run a loop on split()
s=f.read()
L=s.split()
Run a loop on L
Line
Use readlines() , run a loop on return data
L=f.readlines()
Run a loop on L
Extract dataset
Close File

Example 1:

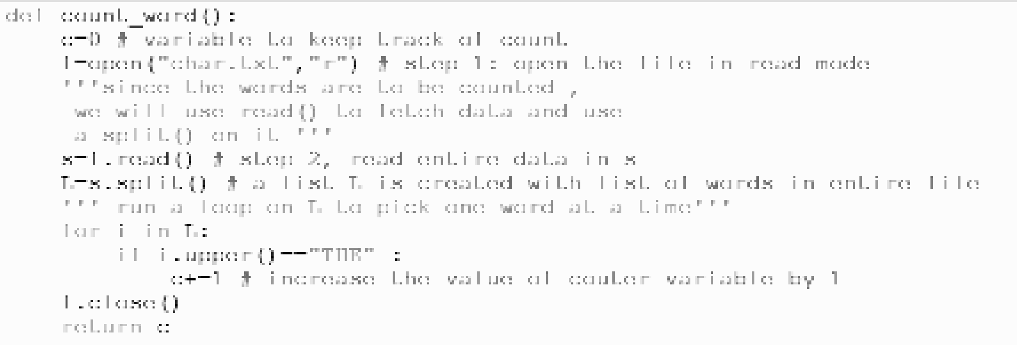
Write a function count_char() that reads a file named “char.txt” counts the number of times character “a” or “A” appears in it.



```
def count_word():
    c=0 # variable to keep track of count
    f=open("char.txt","r") # step 1: open the file in read mode
    '''since the words are to be counted ,
     we will use read()'''
    s=f.read() # step 2, read entire data in s

    ''' run a loop on s to pick one character at a time'''
    for i in s:
        if i.upper()=="A" :
            c+=1 # increase the value of couter variable by 1
    f.close()
    return c
```

Example 2:

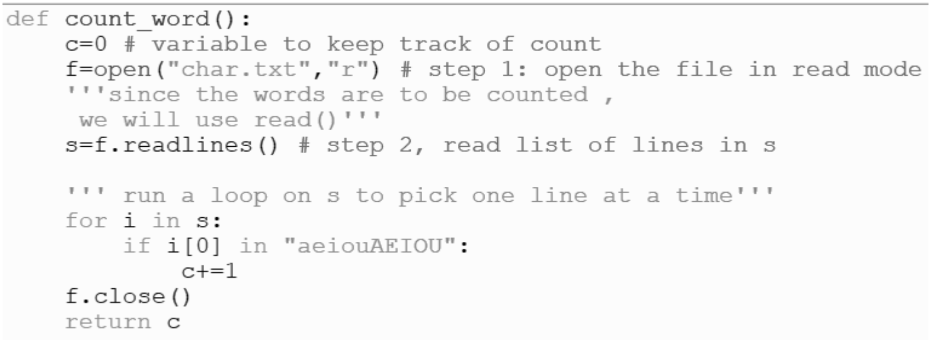
Write a function count_word() that reads a text file named “char.txt” and returns the number of times word “ the” exists.



```
def count_word():
    c=0 # variable to keep track of count
    f=open("char.txt","r") # step 1: open the file in read mode
    '''since the words are to be counted ,
     we will use read() to fetch data and use
     a split() on it '''
    s=f.read() # step 2, read entire data in s
    L=s.split() # a list L is created with list of words in entire file
    ''' run a loop on L to pick one word at a time'''
    for i in L:
        if i.upper()=="THE" :
            c+=1 # increase the value of couter variable by 1
    f.close()
    return c
```

Example 3:

Write a function count_line() that reads a text file named “char.txt” and returns the number of lines that start with a vowel.



```
def count_word():
    c=0 # variable to keep track of count
    f=open("char.txt","r") # step 1: open the file in read mode
    '''since the words are to be counted ,
     we will use read()'''
    s=f.readlines() # step 2, read list of lines in s

    ''' run a loop on s to pick one line at a time'''
    for i in s:
        if i[0] in "aeiouAEIOU":
            c+=1
    f.close()
    return c
```

Writing data into Files

1. **write( ):** It takes a string as an input and writes it into the file.

   Syntax: file_object.write(string)

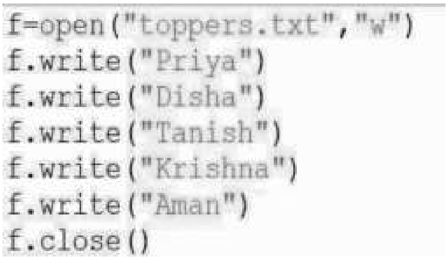
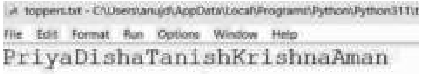
   i.e. f.write(“Hello World”)

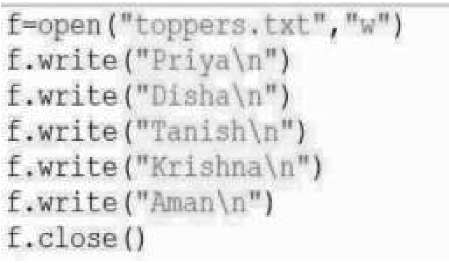
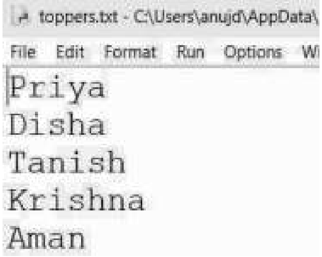
2. **writelines( ):** It is used to write multiple lines as a list of strings into the file. In this, each element of c in the list will be treated as a separate line in the file.

   Syntax: file_object.writelines(list of strings)

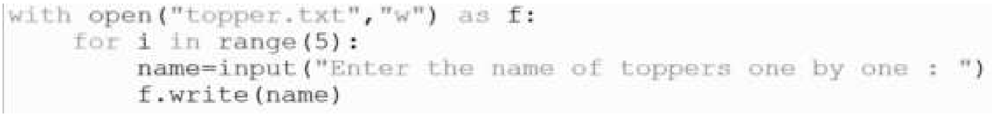
   i.e. data=[“I am a student of DOE”, “I studies in class 12th”]
   f.writelines(data)

Code	**Output**


f=open("toppers.txt","w")
f.write("Priya")
f.write("Disha")
f.write("Tanish")
f.write("Krishna")
f.write("Aman")
f.close()
	Content in “Topper.txt” file
	
	 toppers.txt - C:\Users\anujd\AppData\Local\Programs\Python\Python311\t
	File Edit Format Run Options Window Help
	PriyaDishaTanishKrishnaAman
	But, we want these names in separate lines.

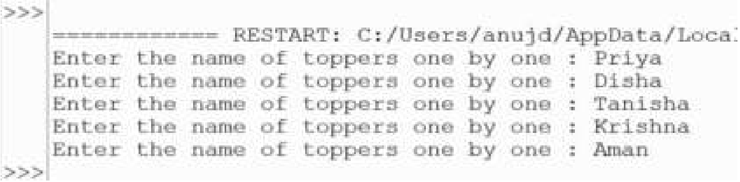

f=open("toppers.txt","w")
f.write("Priya\n")
f.write("Disha\n")
f.write("Tanish\n")
f.write("Krishna\n")
f.write("Aman\n")
f.close()
	
	 toppers.txt - C:\Users\anujd\AppData\
	File Edit Format Run Options Wi
	Priya
	Disha
	Tanish
	Krishna
	Aman

Code:



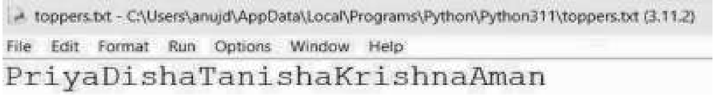
```
with open("topper.txt","w") as f:
    for i in range(5):
        name=input("Enter the name of toppers one by one : ")
        f.write(name)
```

Output:



```
>>>
    ============= RESTART: C:/Users/anujd/AppData/Loca
    Enter the name of toppers one by one : Priya
    Enter the name of toppers one by one : Disha
    Enter the name of toppers one by one : Tanisha
    Enter the name of toppers one by one : Krishna
    Enter the name of toppers one by one : Aman
>>>
```

Content of "Topper.txt" File:



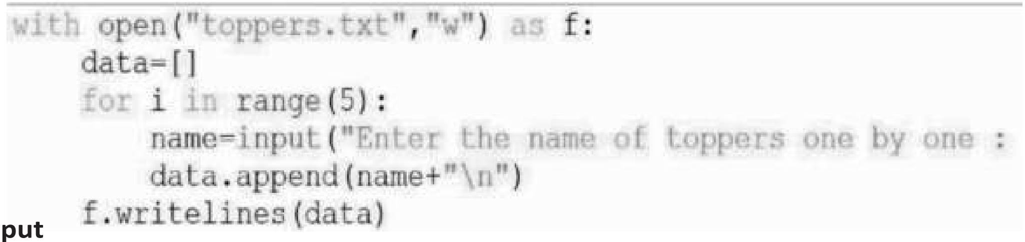
toppers.txt - C:\Users\anujd\AppData\Local\Programs\Python\Python311\toppers.txt (3.11.2)
File Edit Format Run Options Window Help

```
PriyaDishaTanishaKrishnaAman
```

Question: Write a program in python with reference to above program, the content of the text files should be in different lines.

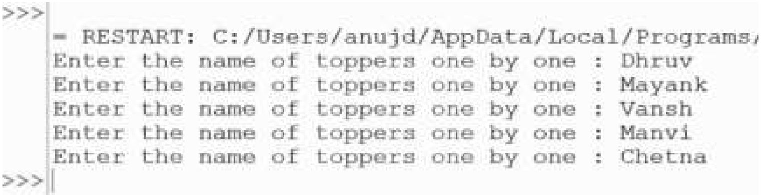
I.e.
Priya
Disha
Tanisha
Krishna
Aman

Code



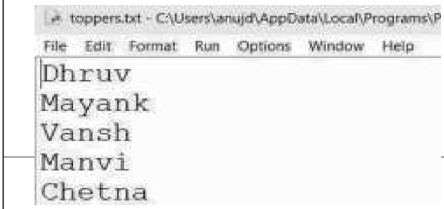
```
with open("toppers.txt","w") as f:
    data=[]
    for i in range(5):
        name=input("Enter the name of toppers one by one :
        data.append(name+"\n")
    f.writelines(data)
```

Output



```
>>>
    = RESTART: C:/Users/anujd/AppData/Local/Programs/
    Enter the name of toppers one by one : Dhruv
    Enter the name of toppers one by one : Mayank
    Enter the name of toppers one by one : Vansh
    Enter the name of toppers one by one : Manvi
    Enter the name of toppers one by one : Chetna
>>>
```

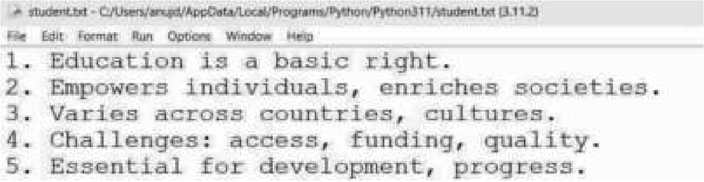
Content of “**Toppers.txt**” file:


toppers.txt - C:\Users\anujd\AppData\Local\Programs\P
File Edit Format Run Options Window Help

```
Dhruv
Mayank
Vansh
Manvi
Chetna
```

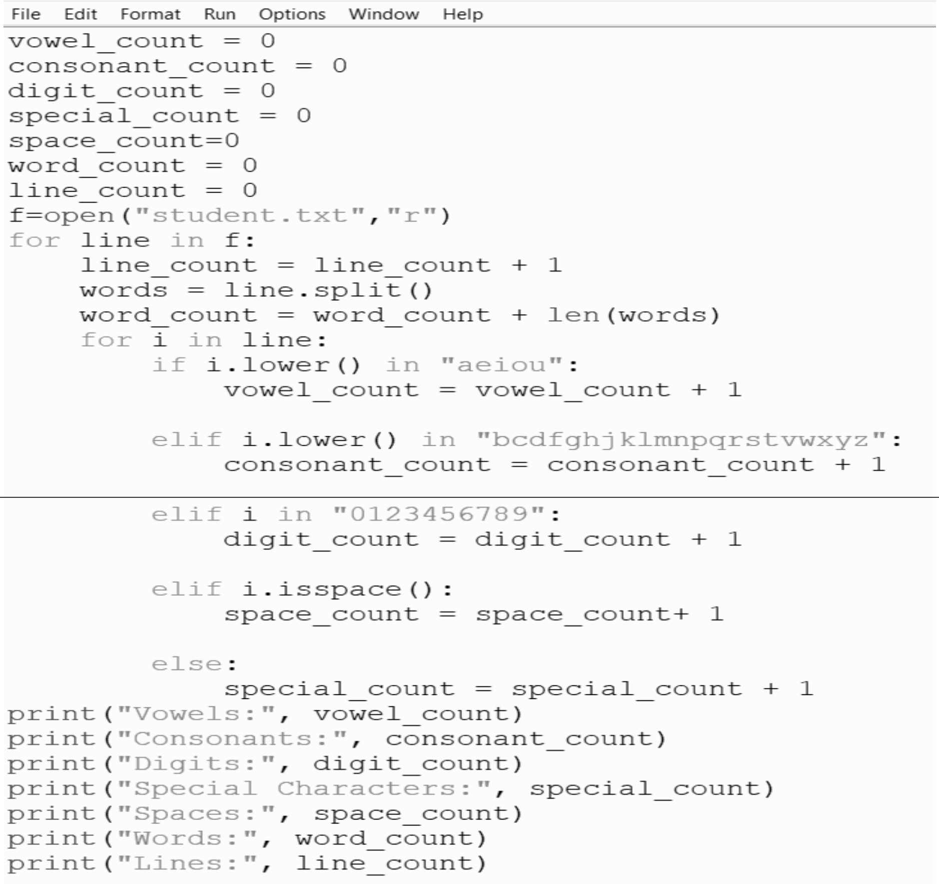
Write a program in python to count vowels, consonants, digits, spaces, special characters, spaces, words and lines from a text file named “student.txt”.

Content of File:


student.txt - C:/Users/anujd/AppData/Local/Programs/Python/Python311/student.txt (3.11.2)
File Edit Format Run Options Window Help

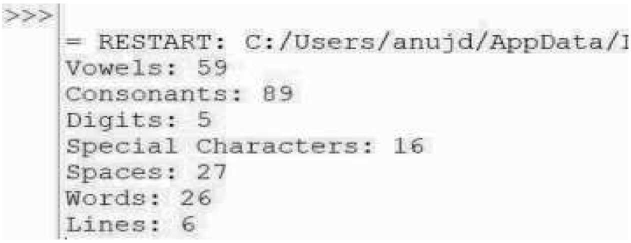
```
1. Education is a basic right.
2. Empowers individuals, enriches societies.
3. Varies across countries, cultures.
4. Challenges: access, funding, quality.
5. Essential for development, progress.
```

Code



```
vowel_count = 0
consonant_count = 0
digit_count = 0
special_count = 0
space_count=0
word_count = 0
line_count = 0
f=open("student.txt","r")
for line in f:
    line_count = line_count + 1
    words = line.split()
    word_count = word_count + len(words)
    for i in line:
        if i.lower() in "aeiou":
            vowel_count = vowel_count + 1

        elif i.lower() in "bcdfghjklmnpqrstvwxyz":
            consonant_count = consonant_count + 1
        elif i in "0123456789":
            digit_count = digit_count + 1

        elif i.isspace():
            space_count = space_count+ 1

        else:
            special_count = special_count + 1
print("Vowels:", vowel_count)
print("Consonants:", consonant_count)
print("Digits:", digit_count)
print("Special Characters:", special_count)
print("Spaces:", space_count)
print("Words:", word_count)
print("Lines:", line_count)
```

Output:



```
>>>
= RESTART: C:/Users/anujd/AppData/I
Vowels: 59
Consonants: 89
Digits: 5
Special Characters: 16
Spaces: 27
Words: 26
Lines: 6
```

Lab Exercise:

1. Define a function SGcounter() that counts and displays the number of S and G present in a text file ‘A.txt”

   e.g., SAGAR JOON IS GOING TO MARKET.
   It will display S:2 G:2

2. Write a function in Python that counts the number of “is”, “am” or “are” words present in a text file “HELLO.TXT”. If the “HELLO.TXT” contents are as follows: *Here are two sentences that contain "is," "am," or "are":*

   “She is studying for her final exams.
   We are planning a trip to the mountains next weekend.”
   The output of the function should be: Count of is/am/are in file: 2

3. Write a method in Python to read lines from a text file HELLO.TXT to find and display the occurrence of the word “hello”.

4. Write a user-defined function named Count() that will read the contents of a text file named “India.txt” and count the number of lines which start with either “I” or “T”.

   E.g. In the following paragraph, 2 lines are starting with “I” or “T”:
   “The Indian economy is one of the largest and fastest-growing in the world,
   characterized by a diverse range of industries including agriculture, manufacturing,
   services, and information technology. It boasts a sizable consumer base and a
   dynamic entrepreneurial spirit. However, it also faces challenges such as income
   inequality, poverty, and infrastructure gaps, which the government continually
   addresses through policy reforms and initiatives to foster sustainable growth .”

5. Write a method in Python to read lines from a text file AZAD.TXT and display those lines, which start with the alphabet ‘T’.

Binary Files:

1. Binary files are made up of non-human readable characters and symbols, which require specific programs to access their contents.
2. In this translation is not required because data is stored in byte form.
3. Faster than text files.
4. **pickle** module is used for working with binary files import pickle
5. The file extension will be **.dat**
6. There is no delimiter to end the file.

Working in Binary files:

Pickle module: The pickle module is used in binary files for the load( ) and dump( ) methods which are used for reading and writing into binary files respectively.

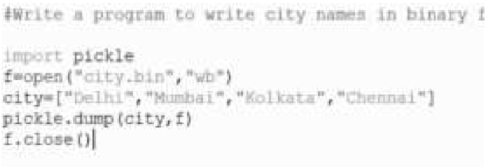
Pickling: It is the process of converting a Python object into a byte stream. Pickling is done at the time of writing into a binary file.

Unpickling: It is the process of converting a byte stream into a Python object. Unpickling is done at the time of reading from a binary file.

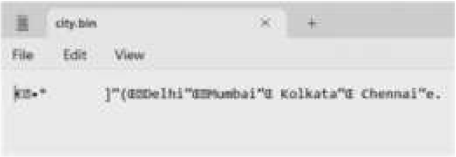
dump( ): It is used to write data into a binary file.

Syntax: identifier = pickle.dump(data , file_pointer)

Example: a= “My name is Anuj” **pickle.dump(a,f)**   #here ‘a’ contains data and ‘f’ is a file pointer.

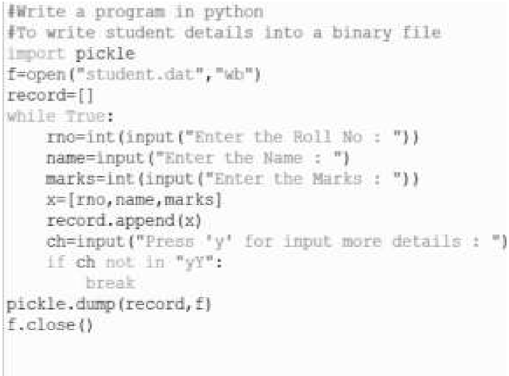


```
#Write a program to write city names in binary f

import pickle
f=open("city.bin","wb")
city=["Delhi","Mumbai","Kolkata","Chennai"]
pickle.dump(city,f)
f.close()
```



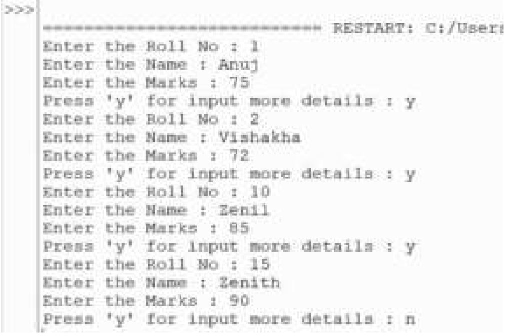
```
KE•"        ]"(Delhi"Mumbai" Kolkata" Chennai"e.
```

Program



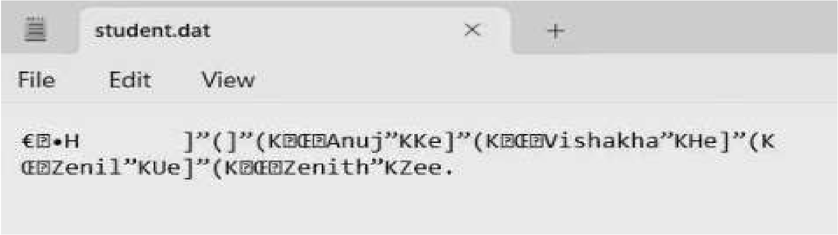
```
#Write a program in python
#To write student details into a binary file
import pickle
f=open("student.dat","wb")
record=[]
while True:
    rno=int(input("Enter the Roll No : "))
    name=input("Enter the Name : ")
    marks=int(input("Enter the Marks : "))
    x=[rno,name,marks]
    record.append(x)
    ch=input("Press 'y' for input more details : ")
    if ch not in "yY":
        break
pickle.dump(record,f)
f.close()
```

Input



```
>>>
============================== RESTART: C:/User
Enter the Roll No : 1
Enter the Name : Anuj
Enter the Marks : 75
Press 'y' for input more details : y
Enter the Roll No : 2
Enter the Name : Vishakha
Enter the Marks : 72
Press 'y' for input more details : y
Enter the Roll No : 10
Enter the Name : Zenil
Enter the Marks : 85
Press 'y' for input more details : y
Enter the Roll No : 15
Enter the Name : Zenith
Enter the Marks : 90
Press 'y' for input more details : n
```

File Content

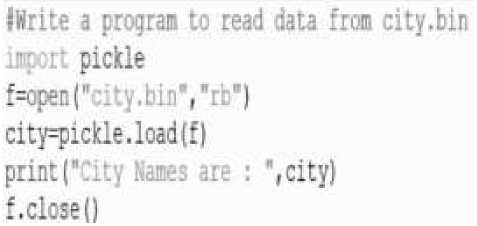


```
€⍰•H       ]”(]”(K⍰Œ⍰Anuj”KKe]”(K⍰Œ⍰Vishakha”KHe]”(K
Œ⍰Zenil”KUe]”(K⍰Œ⍰Zenith”KZee.
```

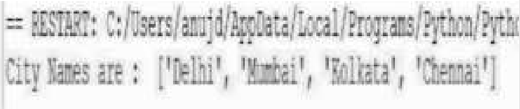
load( ): it is used to read data from binary file.

Syntax: identifier = pickle.load(file_pointer)

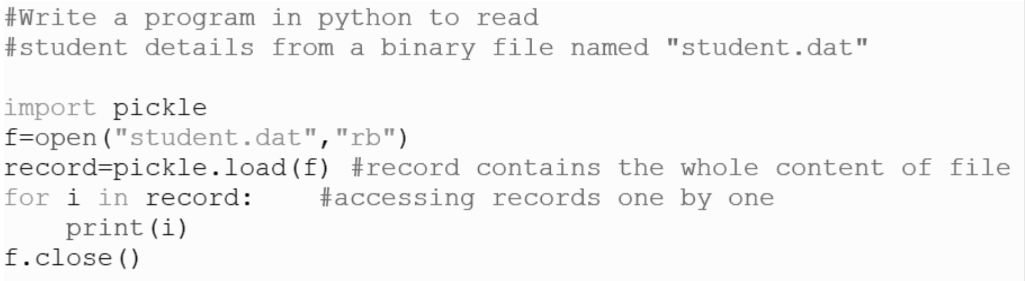
Example: data = pickle.load(f) #Here ‘data’ is an identifier and ‘f’ is a file pointer.



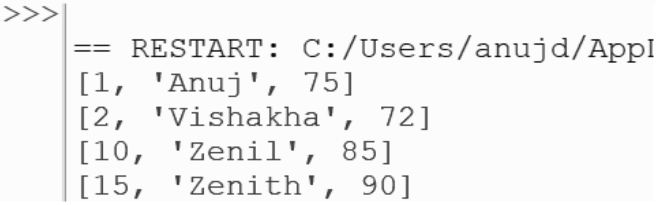
```
#Write a program to read data from city.bin
import pickle
f=open("city.bin","rb")
city=pickle.load(f)
print("City Names are : ",city)
f.close()
```



```
== RESTART: C:/Users/anujd/AppData/Local/Programs/Python/Pyth
City Names are :  ['Delhi', 'Mumbai', 'Kolkata', 'Chennai']
```



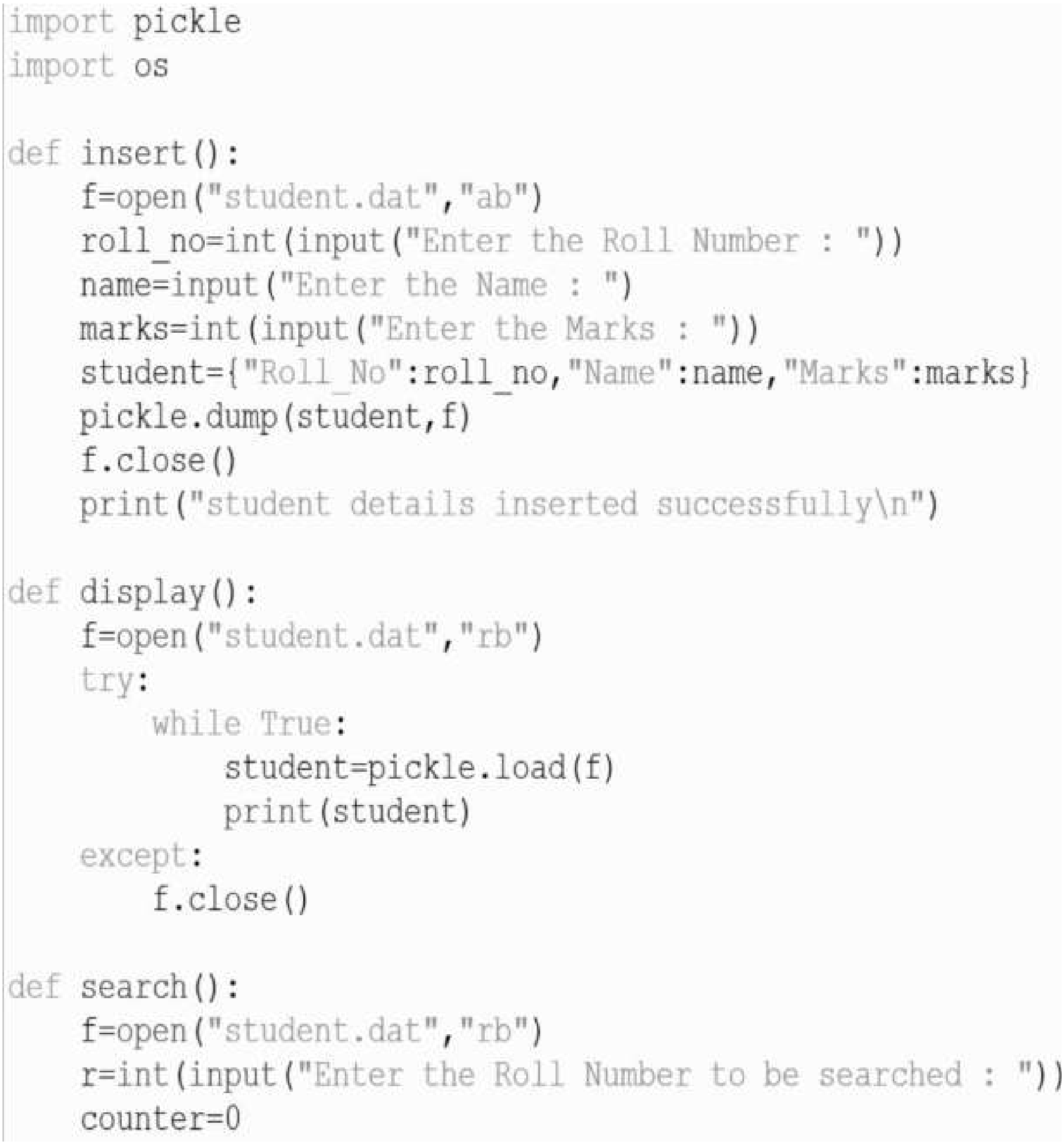
```
#Write a program in python to read
#student details from a binary file named "student.dat"

import pickle
f=open("student.dat","rb")
record=pickle.load(f) #record contains the whole content of file
for i in record:    #accessing records one by one
    print(i)
f.close()
```



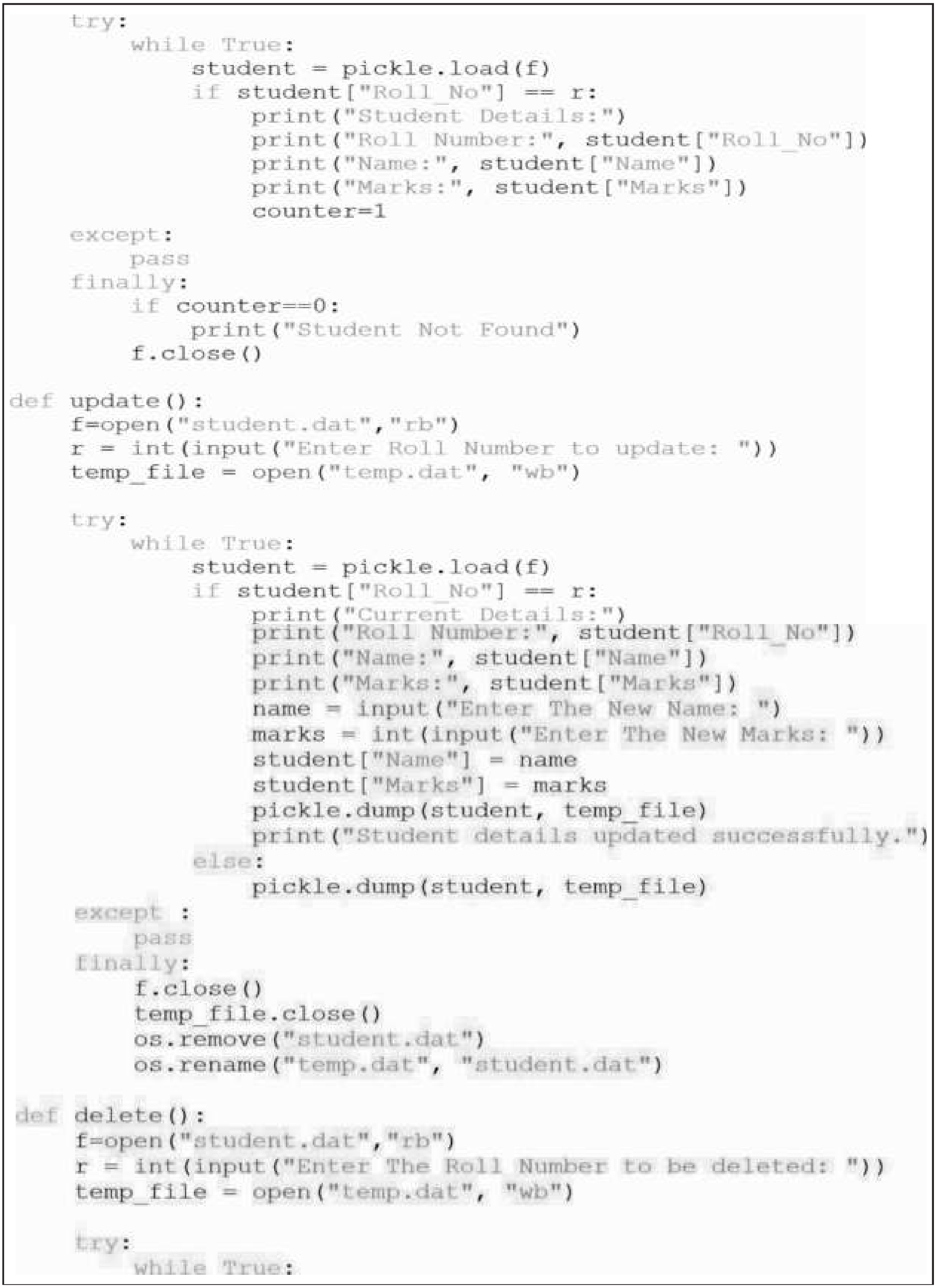
```
>>>
    == RESTART: C:/Users/anujd/AppI
    [1, 'Anuj', 75]
    [2, 'Vishakha', 72]
    [10, 'Zenil', 85]
    [15, 'Zenith', 90]
```

Question: Write a menu based program in python which contain student details in binary file and should have following facilities:

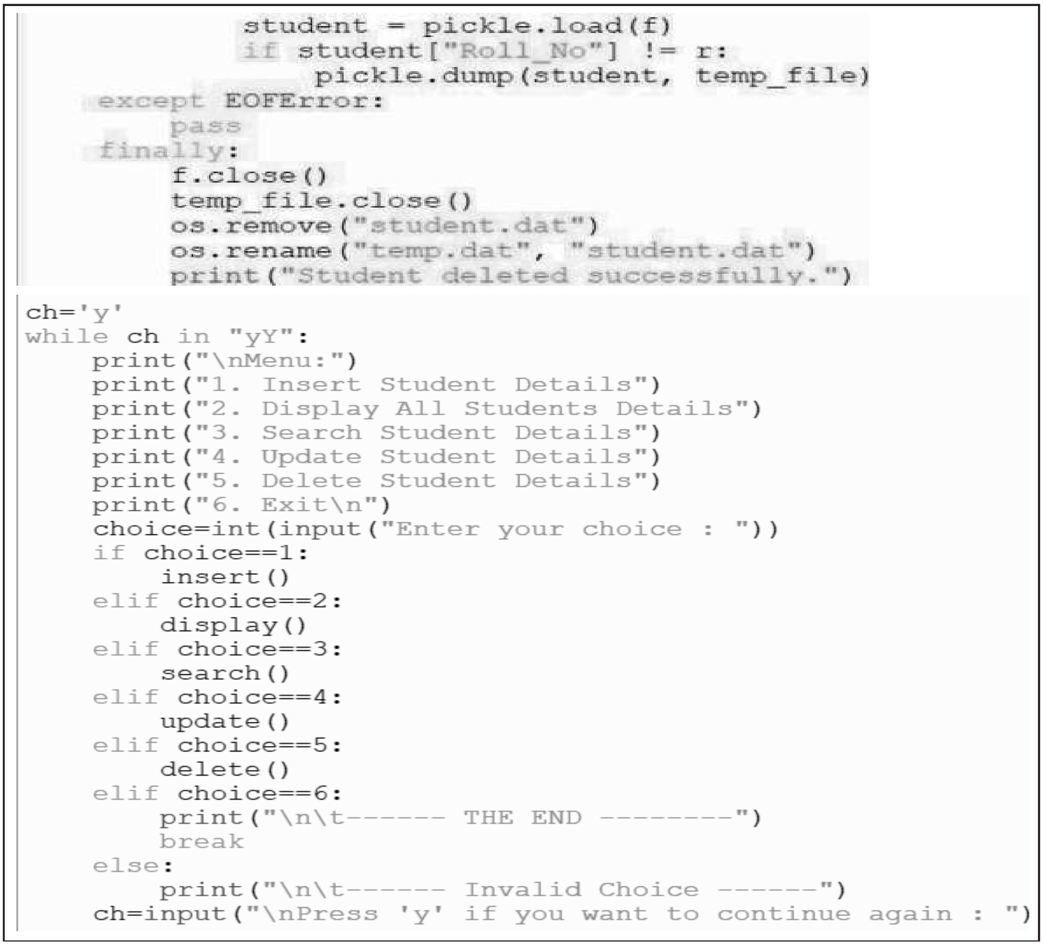
1. Writing student details.
2. Display all students' details
3. Search particular student details
4. Update any student details
5. Delete any student detail
6. Exit



```
import pickle
import os

def insert():
    f=open("student.dat","ab")
    roll_no=int(input("Enter the Roll Number : "))
    name=input("Enter the Name : ")
    marks=int(input("Enter the Marks : "))
    student={"Roll_No":roll_no,"Name":name,"Marks":marks}
    pickle.dump(student,f)
    f.close()
    print("student details inserted successfully\n")

def display():
    f=open("student.dat","rb")
    try:
        while True:
            student=pickle.load(f)
            print(student)
    except:
        f.close()

def search():
    f=open("student.dat","rb")
    r=int(input("Enter the Roll Number to be searched : "))
    counter=0
```



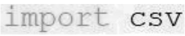
```
    try:
        while True:
            student = pickle.load(f)
            if student["Roll_No"] == r:
                print("Student Details:")
                print("Roll Number:", student["Roll_No"])
                print("Name:", student["Name"])
                print("Marks:", student["Marks"])
                counter=1
    except:
        pass
    finally:
        if counter==0:
            print("Student Not Found")
        f.close()

def update():
    f=open("student.dat","rb")
    r = int(input("Enter Roll Number to update: "))
    temp_file = open("temp.dat", "wb")

    try:
        while True:
            student = pickle.load(f)
            if student["Roll_No"] == r:
                print("Current Details:")
                print("Roll Number:", student["Roll_No"])
                print("Name:", student["Name"])
                print("Marks:", student["Marks"])
                name = input("Enter The New Name: ")
                marks = int(input("Enter The New Marks: "))
                student["Name"] = name
                student["Marks"] = marks
                pickle.dump(student, temp_file)
                print("Student details updated successfully.")
            else:
                pickle.dump(student, temp_file)
    except :
        pass
    finally:
        f.close()
        temp_file.close()
        os.remove("student.dat")
        os.rename("temp.dat", "student.dat")

def delete():
    f=open("student.dat","rb")
    r = int(input("Enter The Roll Number to be deleted: "))
    temp_file = open("temp.dat", "wb")

    try:
        while True:
```



```
            student = pickle.load(f)
            if student["Roll_No"] != r:
                pickle.dump(student, temp_file)
    except EOFError:
        pass
    finally:
        f.close()
        temp_file.close()
        os.remove("student.dat")
        os.rename("temp.dat", "student.dat")
        print("Student deleted successfully.")
ch='y'
while ch in "yY":
    print("\nMenu:")
    print("1. Insert Student Details")
    print("2. Display All Students Details")
    print("3. Search Student Details")
    print("4. Update Student Details")
    print("5. Delete Student Details")
    print("6. Exit\n")
    choice=int(input("Enter your choice : "))
    if choice==1:
        insert()
    elif choice==2:
        display()
    elif choice==3:
        search()
    elif choice==4:
        update()
    elif choice==5:
        delete()
    elif choice==6:
        print("\n\t------ THE END --------")
        break
    else:
        print("\n\t------ Invalid Choice ------")
    ch=input("\nPress 'y' if you want to continue again : ")
```

Comma Separated Value (CSV) Files:

1. It is a plain text file which stores data in a tabular format, where each row represents a record and each column represents a field and fields are separated by commas in CSV files.
2. CSV module is used for working with CSV files:



```
import csv
```

3. The file extension for the CSV file will be .csv
4. CSV module provides two main classes for working with CSV files:
   a. reader
   b. writer
5. CSV reader object can be used to iterate through the rows of the CSV file.
6. CSV writer object that can be used to write row(s) to the CSV file.

a. writerow(): This method is used to write a single row to the CSV file.

b. writerows(): This method is used to write multiple rows to the CSV file.

7. We can read CSV file data in an Excel file also.

i. **Reader Function:** For reading data from a CSV file we require **csv.reader( )** function.

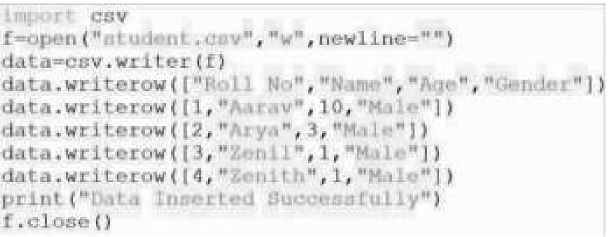
ii. **Writer Function:** For writing data to the CSV file we take a file object as input and write the data into the file.

a. **writerow( )**: It writes a single row of data to the CSV file.

b. **writerows( )**: It writes a list of rows of data to the CSV file.

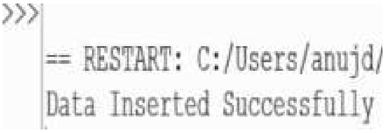
WRITING INTO CSV FILES

Code:



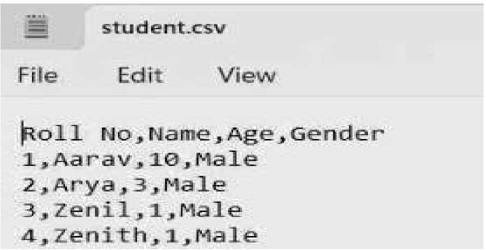
```
import csv
f=open("student.csv","w",newline="")
data=csv.writer(f)
data.writerow(["Roll No","Name","Age","Gender"])
data.writerow([1,"Aarav",10,"Male"])
data.writerow([2,"Arya",3,"Male"])
data.writerow([3,"Zenil",1,"Male"])
data.writerow([4,"Zenith",1,"Male"])
print("Data Inserted Successfully")
f.close()
```

Output in IDLE:



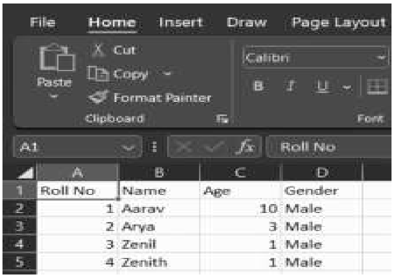
```
>>>
== RESTART: C:/Users/anujd/
Data Inserted Successfully
```

Result in Notepad:



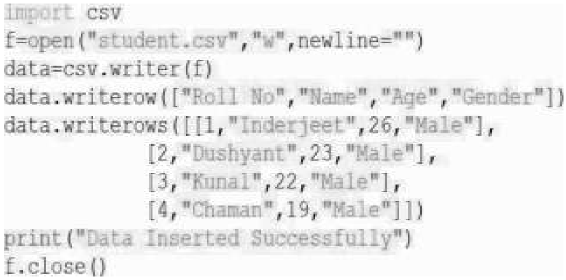
```
Roll No,Name,Age,Gender
1,Aarav,10,Male
2,Arya,3,Male
3,Zenil,1,Male
4,Zenith,1,Male
```

Result in Excel:



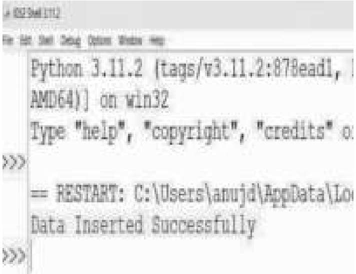
	A	B	C	D
1	Roll No	Name	Age	Gender
2	1	Aarav	10	Male
3	2	Arya	3	Male
4	3	Zenil	1	Male
5	4	Zenith	1	Male

Code:



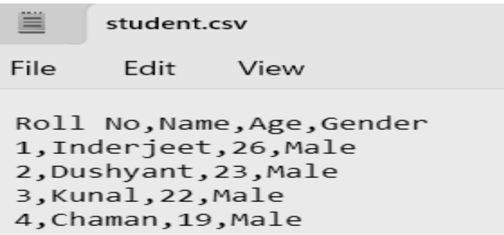
```
import csv
f=open("student.csv","w",newline="")
data=csv.writer(f)
data.writerow(["Roll No","Name","Age","Gender"])
data.writerows([[1,"Inderjeet",26,"Male"],
                [2,"Dushyant",23,"Male"],
                [3,"Kunal",22,"Male"],
                [4,"Chaman",19,"Male"]])
print("Data Inserted Successfully")
f.close()
```

Output in IDLE:



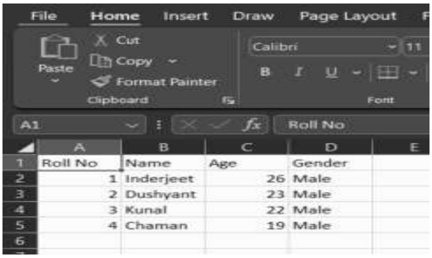
```
Python 3.11.2 (tags/v3.11.2:878ead1, 
AMD64)] on win32
Type "help", "copyright", "credits" o
>>>
== RESTART: C:\Users\anujd\AppData\Lo
Data Inserted Successfully
>>>
```

Result in Notepad:



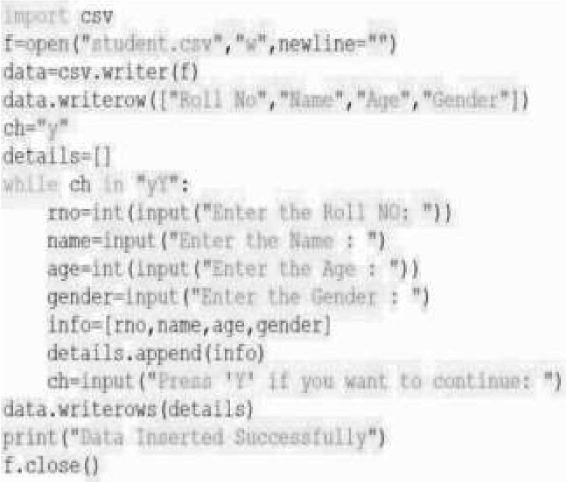
```
Roll No,Name,Age,Gender
1,Inderjeet,26,Male
2,Dushyant,23,Male
3,Kunal,22,Male
4,Chaman,19,Male
```

Result in Excel:



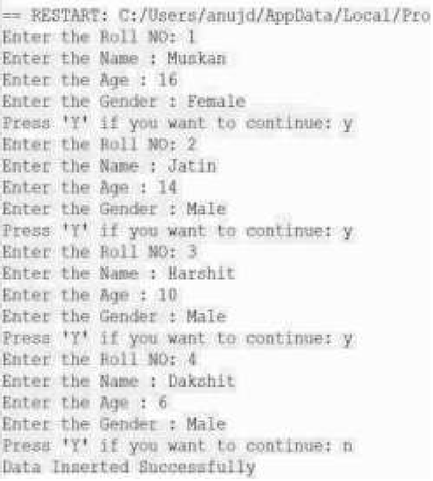
	A	B	C	D	E
1	Roll No	Name	Age	Gender	
2	1	Inderjeet	26	Male	
3	2	Dushyant	23	Male	
4	3	Kunal	22	Male	
5	4	Chaman	19	Male	
6					

Code:



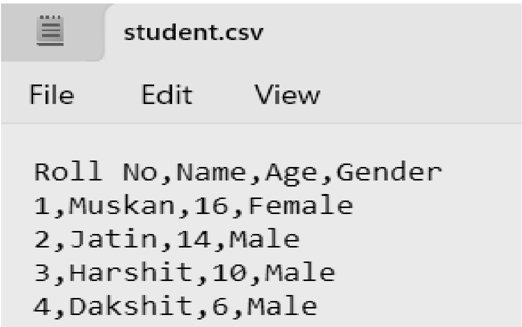
```
import csv
f=open("student.csv","w",newline="")
data=csv.writer(f)
data.writerow(["Roll No","Name","Age","Gender"])
ch="y"
details=[]
while ch in "yY":
    rno=int(input("Enter the Roll NO: "))
    name=input("Enter the Name : ")
    age=int(input("Enter the Age : "))
    gender=input("Enter the Gender : ")
    info=[rno,name,age,gender]
    details.append(info)
    ch=input("Press 'Y' if you want to continue: ")
data.writerows(details)
print("Data Inserted Successfully")
f.close()
```

Output in IDLE:



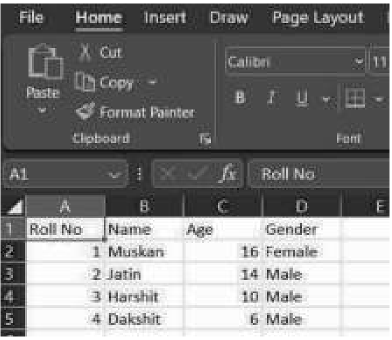
```
== RESTART: C:/Users/anujd/AppData/Local/Pro
Enter the Roll NO: 1
Enter the Name : Muskan
Enter the Age : 16
Enter the Gender : Female
Press 'Y' if you want to continue: y
Enter the Roll NO: 2
Enter the Name : Jatin
Enter the Age : 14
Enter the Gender : Male
Press 'Y' if you want to continue: y
Enter the Roll NO: 3
Enter the Name : Harshit
Enter the Age : 10
Enter the Gender : Male
Press 'Y' if you want to continue: y
Enter the Roll NO: 4
Enter the Name : Dakshit
Enter the Age : 6
Enter the Gender : Male
Press 'Y' if you want to continue: n
Data Inserted Successfully
```

Result in Notepad:



```
Roll No,Name,Age,Gender
1,Muskan,16,Female
2,Jatin,14,Male
3,Harshit,10,Male
4,Dakshit,6,Male
```

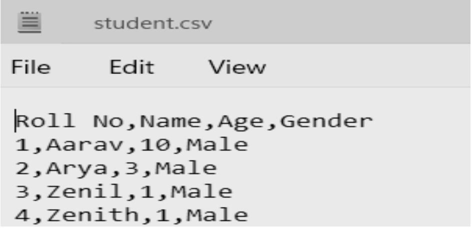
Result in Excel:



	A	B	C	D	E
1	Roll No	Name	Age	Gender	
2	1	Muskan	16	Female	
3	2	Jatin	14	Male	
4	3	Harshit	10	Male	
5	4	Dakshit	6	Male	

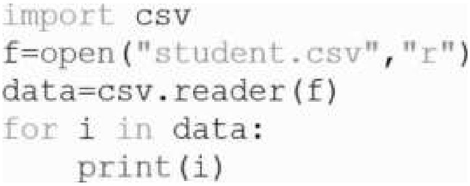
READING FROM CSV FILES

File Content:



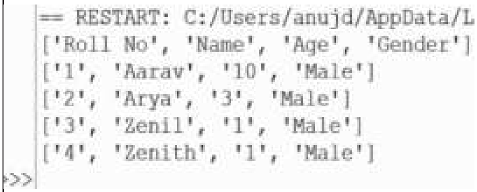
```
Roll No,Name,Age,Gender
1,Aarav,10,Male
2,Arya,3,Male
3,Zenil,1,Male
4,Zenith,1,Male
```

Code: To read all records of csv file



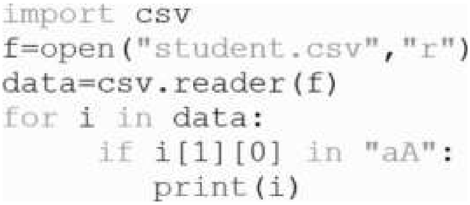
```
import csv
f=open("student.csv","r")
data=csv.reader(f)
for i in data:
    print(i)
```

Output:



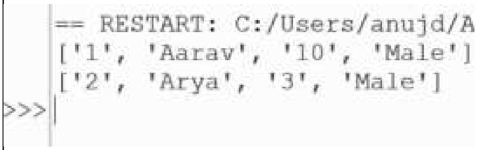
```
== RESTART: C:/Users/anujd/AppData/L
['Roll No', 'Name', 'Age', 'Gender']
['1', 'Aarav', '10', 'Male']
['2', 'Arya', '3', 'Male']
['3', 'Zenil', '1', 'Male']
['4', 'Zenith', '1', 'Male']
>>>
```

Code: To display all the records in which name starts with 'A'



```
import csv
f=open("student.csv","r")
data=csv.reader(f)
for i in data:
    if i[1][0] in "aA":
        print(i)
```

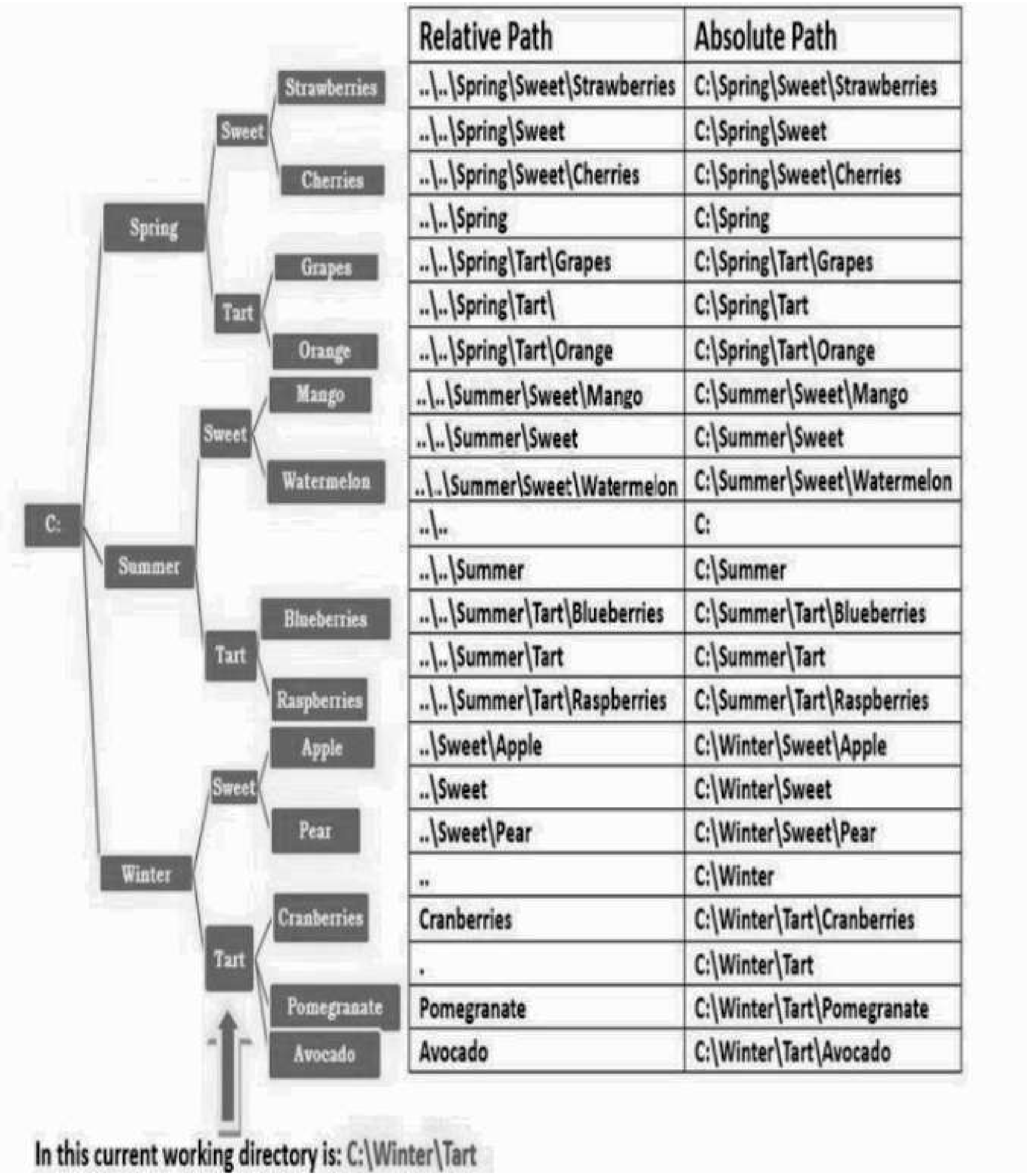
Output:



```
== RESTART: C:/Users/anujd/A
['1', 'Aarav', '10', 'Male']
['2', 'Arya', '3', 'Male']
>>>
```

ABSOLUTE AND RELATIVE PATH:

1. An absolute path is a path that starts from the root directory of the file system or we can say that it describes how to access a given file or directory, from the starting of the file system.
2. A relative path is a path that is relative to the current working directory or we can say that is interpreted from the perspective of the current working directory.



Relative Path	Absolute Path
..\..\Spring\Sweet\Strawberries	C:\Spring\Sweet\Strawberries
..\..\Spring\Sweet	C:\Spring\Sweet
..\..\Spring\Sweet\Cherries	C:\Spring\Sweet\Cherries
..\..\Spring	C:\Spring
..\..\Spring\Tart\Grapes	C:\Spring\Tart\Grapes
..\..\Spring\Tart\	C:\Spring\Tart
..\..\Spring\Tart\Orange	C:\Spring\Tart\Orange
..\..\Summer\Sweet\Mango	C:\Summer\Sweet\Mango
..\..\Summer\Sweet	C:\Summer\Sweet
..\..\Summer\Sweet\Watermelon	C:\Summer\Sweet\Watermelon
..\..	C:
..\..\Summer	C:\Summer
..\..\Summer\Tart\Blueberries	C:\Summer\Tart\Blueberries
..\..\Summer\Tart	C:\Summer\Tart
..\..\Summer\Tart\Raspberries	C:\Summer\Tart\Raspberries
..\Sweet\Apple	C:\Winter\Sweet\Apple
..\Sweet	C:\Winter\Sweet
..\Sweet\Pear	C:\Winter\Sweet\Pear
..	C:\Winter
Cranberries	C:\Winter\Tart\Cranberries
.	C:\Winter\Tart
Pomegranate	C:\Winter\Tart\Pomegranate
Avocado	C:\Winter\Tart\Avocado

In this current working directory is: C:\Winter\Tart

Assignment

1. What is the purpose of file handling in Python?

a. To store data permanently

b. To perform mathematical operations on files

c. To create graphical user interfaces

d. To handle network connections

2. Which function is used to open a file in Python?

a. open() b. read() c. write() d. close()

3. Which of the following modes in the file opening statement results or generates an error if the file does not exist?

a. a+ b. r+ c. w+ d. None of the above

4. What is the default mode in which a file is opened in Python?

a. r b. w c. a d. None of the above

5. What are the three modes in which a file can be opened in Python?

a. r, w, and a b. r+, w+, and a+ c. rb, wb, and ab d. All of the above

6. Which of the following statements is False

a. When you open a file for reading, if the file does not exist, an error occurs.

b. When you open a file for writing, if the file does not exist, a new file is created.

c. When you open a file for writing, if the file exists, the content of the existing file is appended with the new content.

d. None of the above.

7. Which of the following commands is used to open the file “STUDENT.DAT” for only writing in binary format?

a. F= open("STUDENT.DAT",'wb')

b. F= open("STUDENT.DAT",'w')

c. F= open("STUDENT.DAT",'wb+')

d. F= open("STUDENT.DAT",'w+')

8. What is the purpose of the 'with' statement in file handling?

a. It is used for creating new files.

b. It automatically closes the file when done.

c. It is used for reading binary files.

d. It is used for appending data to files.

9. Which of the following options is not correct?

a. if we try to read a text file that does not exist, an error occurs.

b. if we try to read a text file that does not exist, the file gets created.

c. if we try to write on a text file that does not exist, no error occurs.
d. if we try to write on a text file that does not exist, the file gets Created.

10. A text file readme.txt is opened in Python. What type of data is stored in f?

```
>>> file=open('readme.txt')
>>> f=file.readlines()
```

a. String b. Tuple c. List d. None of the above

11. Find P and Q from the options while performing object serialization

```
>>> import pickle
>>> spc=open("yoyo.dat", "wb")
>>> x=500
>>> pickle.dump(P,Q)
```

a. x,spc b. spc, x c. 'yoyo.dat',500 d. 'yoyo.dat','500'

12. What is NOT a valid difference between the write() and writelines() methods in Python?
a. The write() method writes a single string to a file, while the writelines() method writes a list of strings to a file.
b. The write() method does not append a newline character to the end of the string, while the writelines() method does.
c. The write() method moves the file pointer to the next position, while the writelines() method does not.
d. All of the above are valid.

13. What happens if you try to open a non-existent file in Python using 'r' mode?
a. It raises a FileNotFoundError
b. It creates an empty file
c. It raises an IOError
d. It opens a file in read mode

14. Which of the following options can be used to read the first line of a text file Myfile.txt?
a. myfile = open('Myfile.txt'); myfile.read()
b. myfile = open('Myfile.txt','r'); myfile.read(n)
c. myfile = open('Myfile.txt'); myfile.readline()
d. myfile = open('Myfile.txt'); myfile.readlines()

15. Assume that the position of the file pointer is at the beginning of 3rd line in a text file. Which of the following options can be used to read all the remaining lines?
a. myfile.read() b. myfile.read(n) c. myfile.readline() d. myfile.readlines()

16. A text file student.txt is stored in the storage device. Identify the correct option out of the following options to open the file in read mode.

i. myfile = open('student.txt','rb')
ii. myfile = open('student.txt','w')
iii. myfile = open('student.txt','r')
iv. myfile = open('student.txt')

a. only i   b. both i and iv   c. both iii and iv   d. both i and iii

17. Which of the following statements is incorrect in the context of binary files?

a. Information is stored in the same format in which the information is held in memory.
b. No character translation takes place
c. Every line ends with a new line character
d. pickle module is used for reading and writing

18. Which of the following commands is used to write the list L into the binary file, STUDENT.DAT, where f is the file handler?

a. pickle.write(L,f)   b. pickle.write(f, L)   c. pickle.dump(L,f)   d. f=pickle.dump(L)

19. Which of the following commands is used to read each record from the binary file STUDENT.DAT, where f is the file handler?

a) r = pickle.load(f)
b) pickle.read(r,f)
c) r= pickle.read(f)
d) pickle.load(r,f)

20. Which of the following statements is true?

a) pickling creates an object from a sequence of bytes
b) pickling is used for object serialization
c) pickling is used for object deserialization
d) pickling is used to manage all types of files in Python

21. The syntax of the seek function in Python is myfile.seek(offset, reference_point) where myfile is the file object. What is the default value of reference_point?

a) 0   b) 1   c) 2   d) 3

22. Which of the following characters acts as the default delimiter in a CSV file?

a) (colon) :   b) (hyphen) -   c) (comma) ,   d) (vertical line) |

23. The syntax for opening a Student.csv file in write mode is myfile = open("Student.csv","w",newline=''). What is the importance of newline=''?

a) A newline gets added to the file

b) An empty string gets appended to the first line.

c) The empty string gets appended to all lines.

d) EOL translation is suppressed

24. What is the correct expansion of CSV files?

a) Comma Separable Values

b) Comma Separated Values

c) Comma Split Values

d) Comma Separation Values

25. Which of the following is not a function/method of csv module in Python?

a) read() b) reader() c) writer() d) writerow()

26. Which of the following statements opens a binary file record.bin in write mode and writes data from a list lst1 = [1,2,3,4] on the binary file?

a)
```
with open('record.bin','wb') as myfile:
        pickle.dump(lst1,myfile)
```

b)
```
with open('record.bin','wb') as myfile:
        pickle.dump(myfile,lst1)
```

c)
```
with open('record.bin','wb+') as myfile:
        pickle.dump(myfile,lst1)
```

d)
```
with open('record.bin','ab') as myfile:
        pickle.dump(myfile,lst1)
```

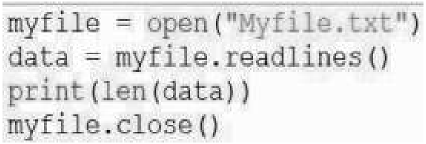
27. Which of the following functions changes the position of the file pointer?

a) flush() b) tell() c) seek() d) offset()

28. Suppose the content of 'Myfile.txt' is:

Twinkle twinkle little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky

What will be the output of the following code?



```
myfile = open("Myfile.txt")
data = myfile.readlines()
print(len(data))
myfile.close()
```

a) 3 b) 4 c) 5 d) 6

29. Which of the following statement(s) are correct regarding the file access modes?

a) ‘r+’ opens a file for both reading and writing. The file object points to its beginning.

b) ‘w+’ opens a file for both writing and reading. Adds at the end of the existing file if it exists and creates a new one if it does not exist.

c) ‘wb’ opens a file for reading and writing in binary format. Overwrites the file if it exists and creates a new one if it does not exist.

d) ‘a’ opens a file for appending. The file pointer is at the start of the file if the file exists.

30. What is the significance of the tell() method?

a) tells the path of the file

b) tells the current position of the file pointer within the file

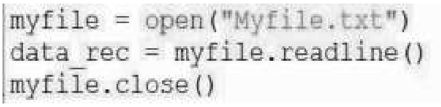
c) tells the end position within the file

d) checks the existence of a file at the desired location

31. Assume the content of the text file, 'student.txt' is:

Arjun Kumar
Ismail Khan
Joseph B
Hanika Kiran

What will be the data type of data_rec?



```
myfile = open("Myfile.txt")
data_rec = myfile.readline()
myfile.close()
```

a) string   b) list   c) tuple   d) dictionary

32. The syntax of seek() is: file_object.seek(offset [, reference_point]) What is the default value of reference_point?

a) 0   b) 1   c) 2   d) 3

33 What is the difference between a text file and a binary file?

a. Text files contain characters, while binary files contain arbitrary data.

b. Text files are human-readable, while binary files are not.

c. Text files are encoded using a character set, while binary files are not.

d. d. All of the above

34 What is the difference between the read() and readline() methods in Python?

a. The read() method reads all of the data from a file, while the readline() method reads only one line of data at a time.

b. The read() method returns a string, while the readline() method returns a list of strings.

c. The read() method does not move the file pointer, while the readline() method moves the file pointer to the next line.

d. All of the above

35 Which method is used to delete a file in Python?

a. os.delete() b. os.remove()   c. file.delete()   d. file.remove()

36 The correct syntax of the read() function from text files is:

a. file_object.read()

b. file_object(read)

c. read(file_object)

d. file_object().read

37 If the file pointer is at the beginning of the file and you use seek(5, 0), where will the file pointer be?

a. At the end of the file.

b. At the current position.

c. 5 bytes from the beginning.

d. 5 bytes from the end

38 Identify the missing part in the code to write the list object in the file

```
 >>> import pickle
>>> x=[1,3,5,7]
>>> f=open('w.dat','wb')
 >>> pickle._____(x,f)
>>> f.close()
```

a. write() b. writeline() c. load() d. dump()

39 A text file myfile0.txt has two lines of text, what will be stored in the variable 'ctr' when the following code is executed?

```
 >>> ctr=0
 >>> spc=open("myfile0.txt")
 >>> while(spc.readline()):
              ctr += 1
```

a. 0 b. 1 c. 2 d. 3

40 Write a statement to send the file pointer position 10 bytes forward from the current location of the file, consider fp as a file object.

a. fp.seek(10) b.fp.seek(10,1) c. fp.tell(10) d. fp.seek(1,10)

41 Myfile.txt is a file stored in the working directory. The total number of characters including spaces is 106 and there are 5 lines. What will be the output of len(file.readlines()) after the file was opened as file=open('Myfile.txt')

a). 4 b). 106 c). 5 d). 1

42 How many lines does the file myfile00.txt have after the code is executed?

```
 >>> mylist=['India', '\nis', '\nmy', '\ncountry', '\nand', '\nI', '\nam', '\na', '\nproud',
        '\ncitizen']
```

```
>>> spc=open("myfile00.txt","w")
 >>> spc.writelines(mylist)
```

a). 2 b). 10 c). 9 d). 1

43 Which character is used as the path separator in Python on the Windows system?
a). / b). \ c). : d). ,

44 What is the difference between 'rb' and 'r' when opening a file in Python?

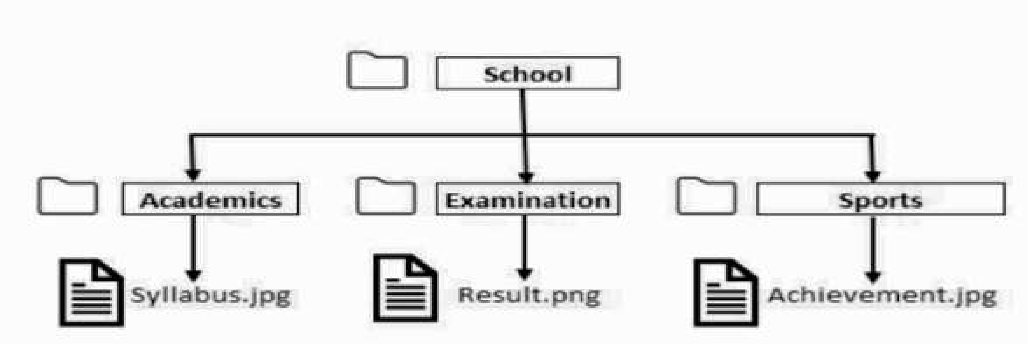
a. 'rb' opens the file in binary mode, 'r' in text mode

b. 'rb' reads the file backwards, 'r' reads it forward

c. 'rb' is for reading, 'r' is for writing

d. 'rb' and 'r' are the same

46 Consider the following directory structure.



Suppose the root directory (School) and present working directory are the same. What will be the absolute path of the file Syllabus.jpg?

a) School/syllabus.jpg

b) School/Academics/syllabus.jpg

c) School/Academics/../syllabus.jpg

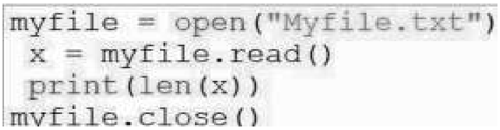
d) School/Examination/syllabus.jpg

Short Answer Questions

1. Suppose the content of **'Myfile.txt'** is:

Honesty is the best policy.

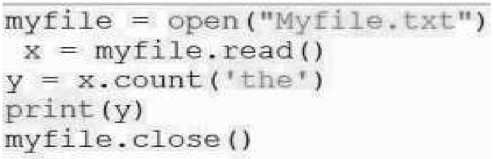
What will be the output of the following code?



```
myfile = open("Myfile.txt")
 x = myfile.read()
 print(len(x))
myfile.close()
```

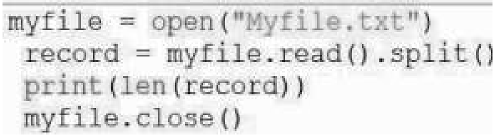
2 Suppose the content of **'Myfile.txt'** is:
Culture is the widening of the mind and the spirit.

What will be the output of the following code?



```
myfile = open("Myfile.txt")
 x = myfile.read()
y = x.count('the')
print(y)
myfile.close()
```

3 Suppose the content of **'Myfile.txt'** is: *Humpty Dumpty sat on a wall*
Humpty Dumpty had a great fall

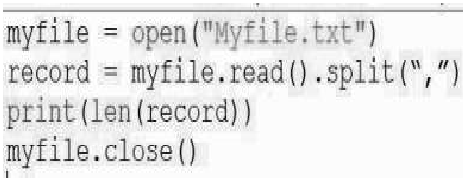


```
myfile = open("Myfile.txt")
 record = myfile.read().split()
 print(len(record))
 myfile.close()
```

What will be the output of the following code?

4 Suppose the content of **'Myfile.txt'** is

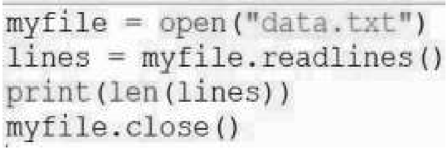
Twinkle, twinkle, little star,
How I wonder what you are! Up
above the world so high, Like a
diamond in the sky.
What will be the output of the following code?



```
myfile = open("Myfile.txt")
record = myfile.read().split(“,”)
print(len(record))
myfile.close()
```

5 Suppose the content of **'data.txt'** is:

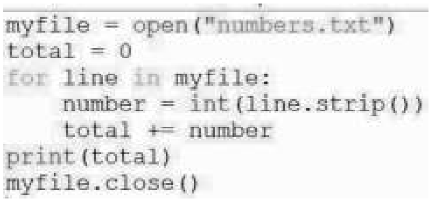
Python is a powerful programming language. It is
used widely for various applications.
What will be the output of the following code?



```
myfile = open("data.txt")
lines = myfile.readlines()
print(len(lines))
myfile.close()
```

6 Suppose the content of **'numbers.txt'** is:
10
20
30
40

What will be the output of the following code?



```
myfile = open("numbers.txt")
total = 0
for line in myfile:
    number = int(line.strip())
    total += number
print(total)
myfile.close()
```

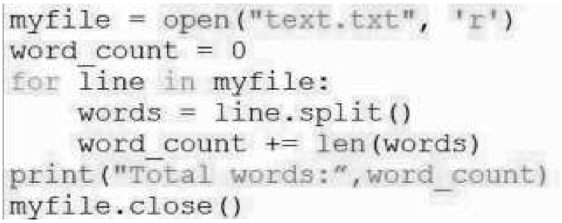
●

7. Consider a file named **'text.txt'** with the following content:

Python is a powerful programming language.

It is widely used for various applications.

Write a Python script to count the total number of words in the file.



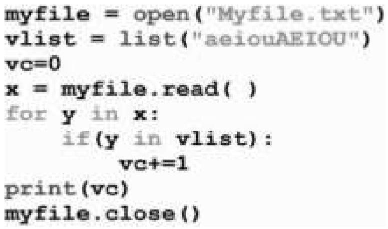
```
myfile = open("text.txt", 'r')
word_count = 0
for line in myfile:
    words = line.split()
    word_count += len(words)
print("Total words:”,word_count)
myfile.close()
```

What will be the output if **'text.txt'** has the above content?

8 Suppose the content of **'Myfile.txt'** is

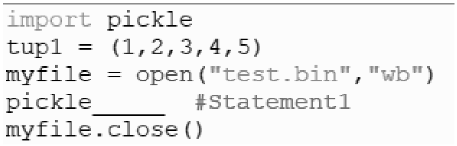
Ek Bharat Shreshtha Bharat

What will be the output of the following code?



```
myfile = open("Myfile.txt")
vlist = list("aeiouAEIOU")
vc=0
x = myfile.read( )
for y in x:
    if(y in vlist):
        vc+=1
print(vc)
myfile.close()
```

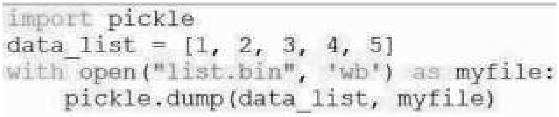
9. Raghav is trying to write a tuple tup1 = (1,2,3,4,5) on a binary file “**test.bin**”. Consider the following code written by him.



```
import pickle
tup1 = (1,2,3,4,5)
myfile = open("test.bin","wb")
pickle______  #Statement1
myfile.close()
```

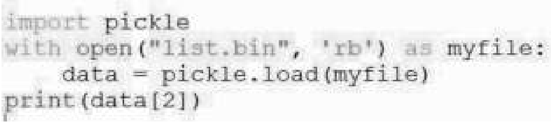
Write the missing code in Statement 1

10. Given the following code to write a list to a binary file:



```
import pickle
data_list = [1, 2, 3, 4, 5]
with open("list.bin", 'wb') as myfile:
    pickle.dump(data_list, myfile)
```

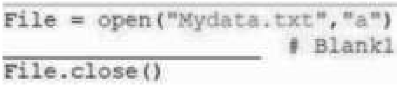
The following code to read from the binary file



```
import pickle
with open("list.bin", 'rb') as myfile:
    data = pickle.load(myfile)
print(data[2])
```

What will be the output of the print statement?

11. Observe the following code and answer the questions that follow:

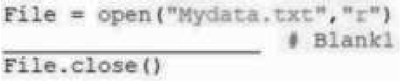


```
File = open("Mydata.txt","a")
_____________________ # Blank1
File.close()
```

i. What type (Text/Binary) of file is Mydata?

ii. Fill Blank 1 with a statement to write “ABC” in the file “Mydata”.

12. Observe the following code and answer the questions that follow:



```
File = open("Mydata.txt","r")
_____________________ # Blank1
File.close()
```

i. What type (Text/Binary) of file is "Mydata"?

ii. Fill the Blank 1 with a statement to read the entire content of the file "Mydata" into a variable content.

ASSERTION AND REASONING based questions.
Mark the correct choice as

(a) Both A and R are true and R is the correct explanation for A
(b) Both A and R are true and R is not the correct explanation for A
(c) A is True but R is False
(d) A is false but R is True

1. **Assertion (A):** CSV (Comma Separated Values) is a file format for data storage which looks like a text file.
Reason (R): The information is organized with one record on each line and each field is separated by comma.

2 **Assertion (A):** A binary file is a computer-readable file that contains binary data.
Reason (R): Binary files can store data in formats such as images, audio, and executables.

3. **Assertion (A):** Text files are easier to read and edit using simple text editors.
Reason (R): Text files store data in a human-readable format with plain text.

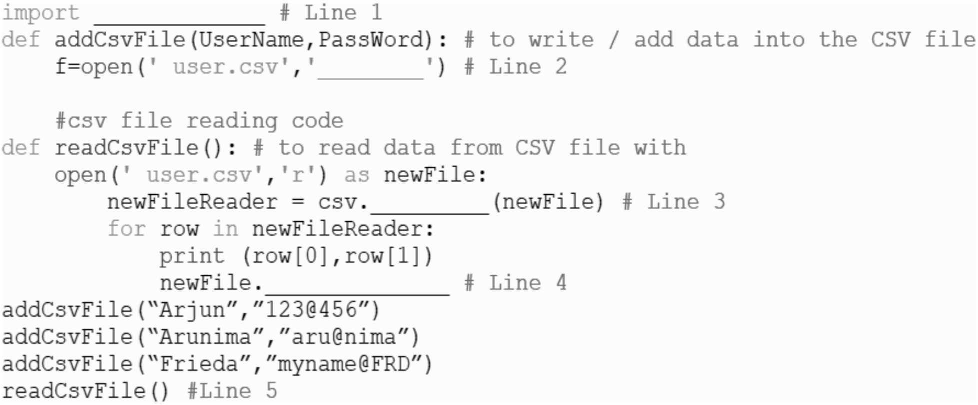
4. **Assertion (A):** Binary files are more prone to corruption than text files.
Reason (R): Binary files are usually larger in size and contain more complex data.

5. **Assertion (A):** Opening a file in read mode ('r') will create a new file if it does not exist.
Reason (R): The read mode is used for reading the content of an existing file.

6. **Assertion (A):** Opening a file in append mode ('a') will allow adding new data without modifying the existing content.
Reason (R): The append mode places the file cursor at the beginning of the file.

LONG ANSWER TYPE QUESTIONS

1. Mr. Manish Kumar is writing a program to create a CSV file “user.csv” which will contain user name and password for some entries. He has written the following code. As a programmer, help him to successfully execute the given task.

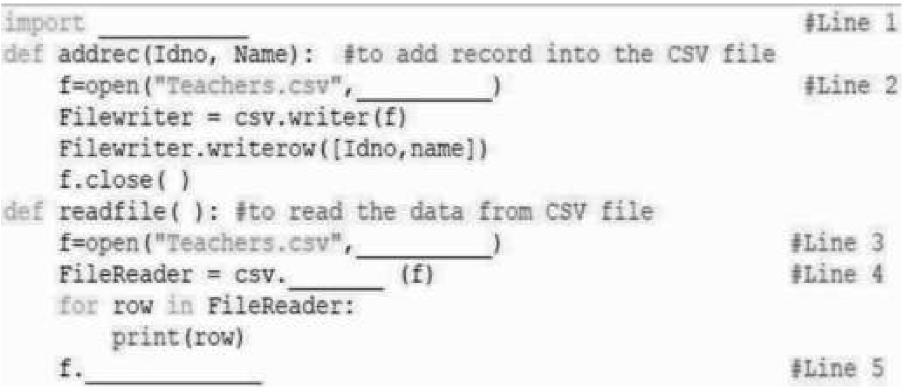


```
import _____________ # Line 1
def addCsvFile(UserName,PassWord): # to write / add data into the CSV file
    f=open(' user.csv','________') # Line 2

    #csv file reading code
def readCsvFile(): # to read data from CSV file with
    open(' user.csv','r') as newFile:
        newFileReader = csv._________(newFile) # Line 3
        for row in newFileReader:
            print (row[0],row[1])
            newFile.______________ # Line 4
addCsvFile(“Arjun”,”123@456”)
addCsvFile(“Arunima”,”aru@nima”)
addCsvFile(“Frieda”,”myname@FRD”)
readCsvFile() #Line 5
```

a) Name the module he should import in Line 1.
b) In which mode, Manish should open the file to add data into the file
c) Fill in the blank in Line 3 to read the data from a csv file.
d) Fill in the blank in Line 4 to close the file.

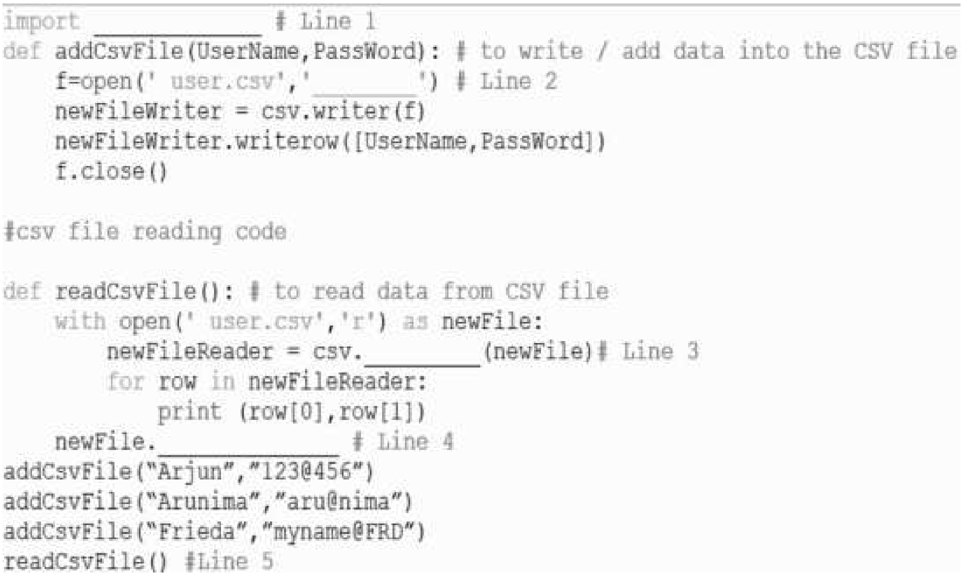
2. Vaishali of class 12 is writing a program in Python for her project work to create a CSV file “Teachers.csv” which will contain information for every teacher’s identification Number , Name for some entries. She has written the following code. However, she is unable to figure out the correct statements in few lines of code, hence she has left them blank. Help her write the statements correctly for the

missing parts in the code.



```
import ___________                                          #Line 1
def addrec(Idno, Name):  #to add record into the CSV file
    f=open("Teachers.csv",_________)                        #Line 2
    Filewriter = csv.writer(f)
    Filewriter.writerow([Idno,name])
    f.close( )
def readfile( ): #to read the data from CSV file
    f=open("Teachers.csv",_________)                       #Line 3
    FileReader = csv._______  (f)                          #Line 4
    for row in FileReader:
        print(row)
    f.____________                                         #Line 5
```

3. Sachin of class 12 is writing a program to create a CSV file “user.csv” which will contain user name and password for some entries. He has written the following code. As a programmer, help him to successfully execute the given task.

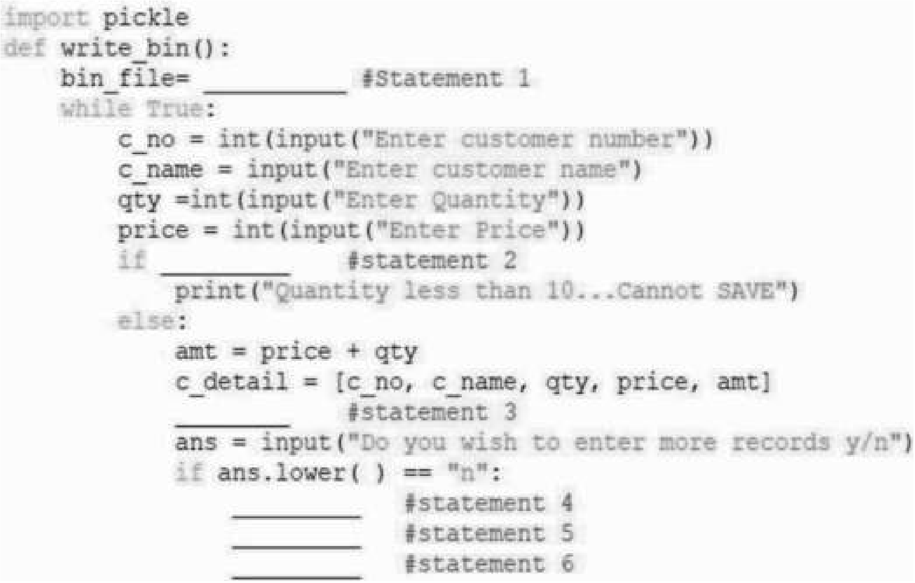


```
import ____________ # Line 1
def addCsvFile(UserName,PassWord): # to write / add data into the CSV file
    f=open(' user.csv','_______') # Line 2
    newFileWriter = csv.writer(f)
    newFileWriter.writerow([UserName,PassWord])
    f.close()

#csv file reading code

def readCsvFile(): # to read data from CSV file
    with open(' user.csv','r') as newFile:
        newFileReader = csv.________(newFile)# Line 3
        for row in newFileReader:
            print (row[0],row[1])
    newFile.____________  # Line 4
addCsvFile(“Arjun”,”123@456”)
addCsvFile(“Arunima”,”aru@nima”)
addCsvFile(“Frieda”,”myname@FRD”)
readCsvFile() #Line 5
```

a) Name the module he should import in Line 1.
b) In which mode, Sachin should open the file to add data into the file in line 2.
c) Fill in the blank in Line 3 to read the data from a csv file.
d) Fill in the blank in Line 4 to close the file.
e) Write the output he will obtain while executing Line 5.

4 Shanvika is a programmer, who has recently been given a task to write a user defined function named write_bin() to create a binary file called Cust_file.dat containing customer information customer number (c_no), name (c_name), quantity (qty), price (price) and amount (amt) of each customer.

The function accepts customer number, name, quantity and price. Thereafter, it displays the message Quantity less than 10 Cannot SAVE', if quantity entered is less than 10. Otherwise the function calculates amount as price quantity and then writes the record in the form of a list into the binary file.



```
import pickle
def write_bin():
    bin_file= __________ #Statement 1
    while True:
        c_no = int(input("Enter customer number"))
        c_name = input("Enter customer name")
        qty =int(input("Enter Quantity"))
        price = int(input("Enter Price"))
        if ________     #statement 2
            print("Quantity less than 10...Cannot SAVE")
        else:
            amt = price + qty
            c_detail = [c_no, c_name, qty, price, amt]
            ________    #statement 3
            ans = input("Do you wish to enter more records y/n")
            if ans.lower( ) == "n":
                _________   #statement 4
                _________   #statement 5
                _________   #statement 6
```

(i) Write the correct statement to open a file 'Cust_file.dat' for writing the data of the customer.

(ii) Which statement should Shanvika fill in Statement 2 to check whether quantity is less than 10.

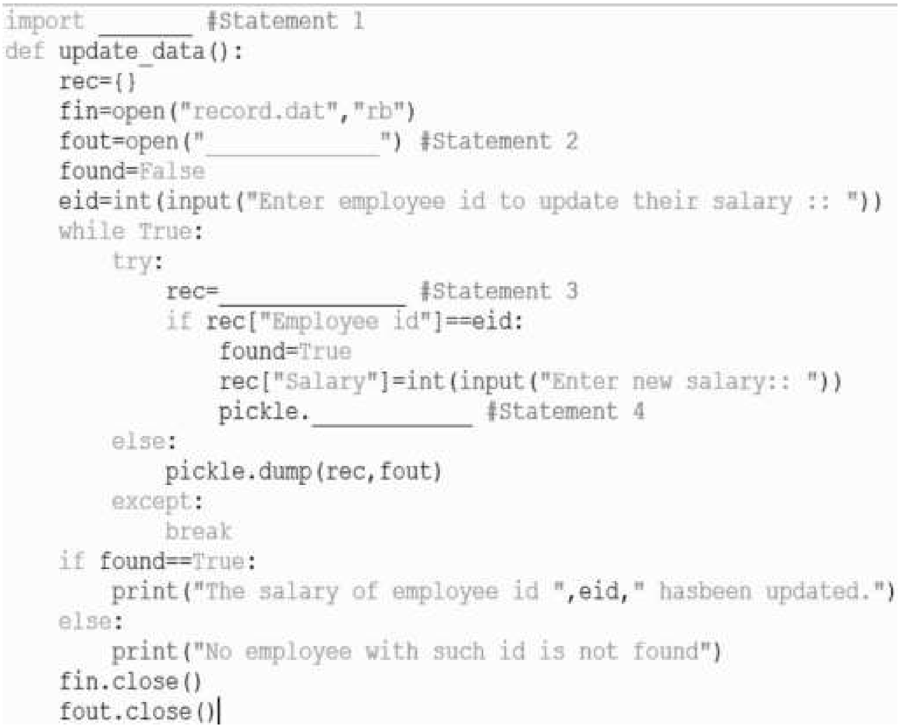
(iii) Which statement should Shanvika fill in Statement 3 to write data to the binary file and in Statement 4 to stop further processing if the user does not wish to enter more records.

(iv) What should Shanvika fill in Statement 5 to close the binary file named Cust_file.dat and in Statement 6 to call a function to write data in binary file?

5. Divisha is a Python programmer. She has written a code and created a binary file record.dat with employe eid, ename and salary. The file contains 10 records. She

now has to update a record based on the employee id entered by the user and update the salary. The updated record is then to be written in the file temp.dat. The records which are not to be updated also have to be written to the file temp.dat. If the employee id is not found, an appropriate message should to be displayed. As a Python expert, help her to complete the following code based on the requirement given above:

(i) Which module should be imported in the program? (Statement 1)



```
import _______ #Statement 1
def update_data():
    rec={}
    fin=open("record.dat","rb")
    fout=open("____________") #Statement 2
    found=False
    eid=int(input("Enter employee id to update their salary :: "))
    while True:
        try:
            rec=_____________ #Statement 3
            if rec["Employee id"]==eid:
                found=True
                rec["Salary"]=int(input("Enter new salary:: "))
                pickle.___________ #Statement 4
        else:
            pickle.dump(rec,fout)
        except:
            break
    if found==True:
        print("The salary of employee id ",eid," hasbeen updated.")
    else:
        print("No employee with such id is not found")
    fin.close()
    fout.close()
```

(ii) Write the correct statement required to open a temporary file named temp.dat. (Statement 2)

(iii) Which statement should Divisha fill in Statement 3 to read the data from the binary file, record.dat and in Statement 4 to write the updated data in the file, temp.dat?

Lab exercise

1. Write the definition of a function Count_Line() in Python, which should read each line of a text file "SHIVAJI.TXT" and count total number of lines present in text file. For example, if the content of the file "SHIVAJI.TXT" is as follows:

Shivaji was born in the family of Bhonsle.
He was devoted to his mother Jijabai.
India at that time was under Muslim rule.

The function should read the file content and display the output as follows:

Total number of lines: 3

2 Write a function in Python that counts the number of "Me" or "My" words present in a text file "STORY.TXT". If the "STORY.TXT" contents are as follows: *My first book was Me and My Family. It gave me chance to be Known to the world.*
Output : Number of words ending with a digit are 4

3 Write a function AMCount() in Python, which should read each character of a text file STORY.TXT, should count and display the occurrence of alphabets A and M (including small cases a and m too).
Example: If the file content is as follows:

Updated information As simplified by official websites.

The EUCount() function should display the output as:
A or a:4
M or m :2

4 Write the definition of a Python function named LongLines ( ) which reads the contents of a text file named 'LINES.TXT' and displays those lines from the file which have at least 10 words in it. For example, if the content of 'LINES.TXT' is as follows:

Once upon a time, there was a woodcutter He lived
in a little house in a beautiful, green wood.

One day, he was merrily chopping some wood.

He saw a little girl skipping through the woods, whistling happily. The girl
was followed by a big gray wolf.

Then the function should display output as :

He lived in a little house in a beautiful, green wood.
He saw a little girl skipping through the woods, whistling happily.

5 Write a function count Words (in Python to count the words ending with a digit in a text file "Details.txt".
Example:
If the file content is as follows:
On seat2 VIP1 will sit and
On seat1 VVIP2 will be sitting
Output will be: 4

6. Write a method/function DISPLAYWORDS() in python to read lines from a text file STORY.TXT, and display those words, which are less than 4 characters.

7 A binary file "STUDENT.DAT" has structure (admission_number, Name, Percentage).

Write a function countrec() in Python that would read contents of the file “STUDENT.DAT” and display the details of those students whose percentage is above 75. Also display number of students scoring above 75%.

8 A binary file “Book.dat” has structure [BookNo, Book_Name, Author, Price].

i. Write a user defined function CreateFile() to input data for a record and add to Book.dat .

ii. Write a function CountRec(Author) in Python which accepts the Author name as parameter and count and return number of books by the given Author are stored in the binary file “Book.dat”

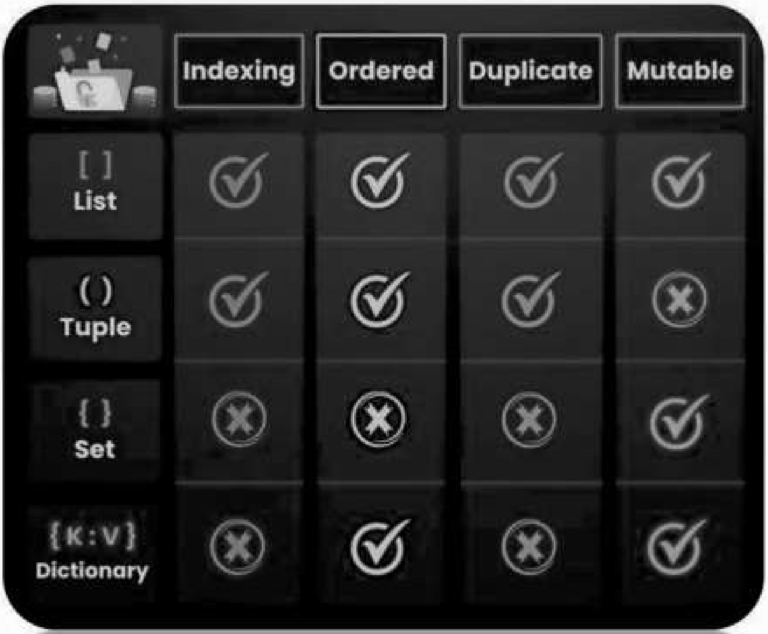
9. A binary file "PLANTS.dat" has structure (ID, NAME, PRICE).

Write the definition of a function WRITEREC () in Python, to input data for records from the user and write them to the file PLANTS.dat.

Write the definition of a function SHOWHIGH () in Python, which reads the records of PLANTS. dat and displays those records for which the PRICE is more than 500. 10 A binary file "PATIENTS.dat" has structure (PID, NAME, DISEASE).

Write the definition of a function countrec () in Python that would read contents of the file "PATIENTS.dat" and display the details of those patients who have the DISEASE as 'COVID-19'. The function should also display the total number of such patients whose DISEASE is 'COVID-19'.

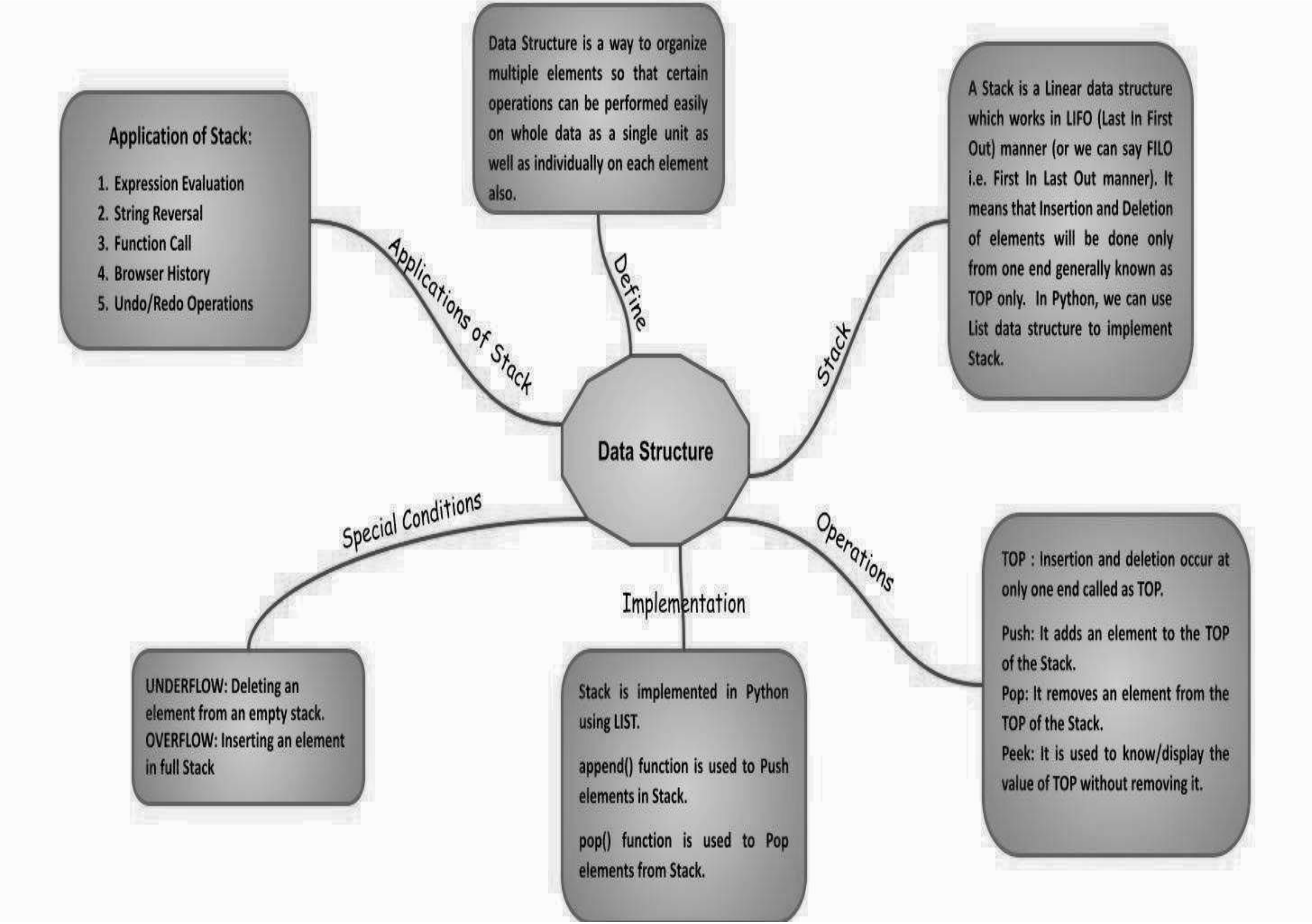
DATA STRUCTURES



	Indexing	Ordered	Duplicate	Mutable
[ ] List	✓	✓	✓	✓
( ) Tuple	✓	✓	✓	✗
{ } Set	✗	✗	✗	✓
{K:V} Dictionary	✗	✓	✗	✓

Topic to be covered:

- **Stack**
- **Operation On Stack (Push, Pop, Peek, Display)**
- **Implementation Of Stack Using List**
- **Applications Of Stack**



Data Structure is a way to organize multiple elements so that certain operations can be performed easily on whole data as a single unit as well as individually on each element also.

A Stack is a Linear data structure which works in LIFO (Last In First Out) manner (or we can say FILO i.e. First In Last Out manner). It means that Insertion and Deletion of elements will be done only from one end generally known as TOP only. In Python, we can use List data structure to implement Stack.

TOP : Insertion and deletion occur at only one end called as TOP.

Push: It adds an element to the TOP of the Stack.

Pop: It removes an element from the TOP of the Stack.

Peek: It is used to know/display the value of TOP without removing it.

Stack is implemented in Python using LIST.

append() function is used to Push elements in Stack.

pop() function is used to Pop elements from Stack.

UNDERFLOW: Deleting an element from an empty stack.
OVERFLOW: Inserting an element in full Stack

Application of Stack:

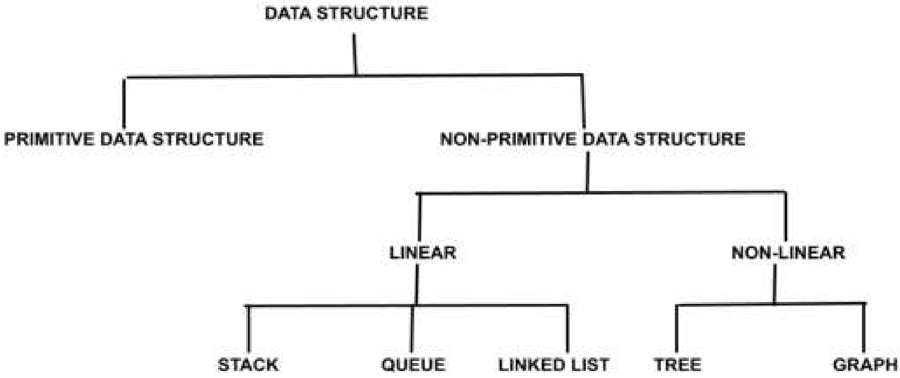
1. Expression Evaluation
2. String Reversal
3. Function Call
4. Browser History
5. Undo/Redo Operations

DATA STRUCTURE

Data structure can be defined as a set of rules and operations to organize and store data efficiently. We can also say that it is a way to store data in a structured way. We can apply different operations like reversal, slicing, counting etc. of different data structures. Hence, Data Structure is a way to organize multiple elements so that certain operations can be performed easily on whole data as a single unit as well as individually on each element.

In Python, Users are allowed to create their own Data Structures which enable them to define the functionality of created data structures. Examples of User-defined data structures in Python are Stack, Queue, Tree, Linked List etc. There are some built-in data structures also available in Python like List, Tuple, Dictionary and Set.

Types of Data Structure:



Primitive Data Structures

Primitive Data Structures contain simplified data values and are directly operated by machine-level instructions. For example, integer, real, character etc. are primitive data structures.

Non-Primitive Data Structures

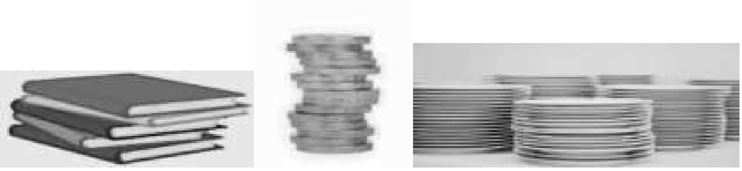
Non-Primitive Data Structures are derived from the primitive data structures. There are two types of non – Primitive Data Structures:

Linear data structures are single-level data structures having their elements in a sequence like a stack, queue and linked list.

Non-linear data structures are multilevel data structures like tree and graph.

STACK:

A Stack is a Linear data structure which works in a LIFO (Last In First Out) manner (or we can say FILO i.e. First In Last Out manner). It means that Insertion and Deletion of elements will be done only from one end generally known as TOP only. In Python, we can use a List data structure to implement Stack.



Application of Stack:

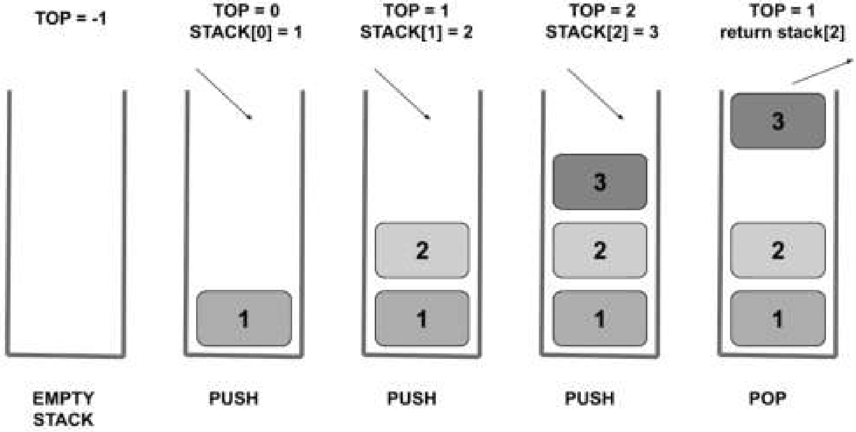
1. Expression Evaluation
2. String Reversal
3. Function Call
4. Browser History
5. Undo/Redo Operations

Operations of Stack:

The Stack supports the following operations:

1. **Push:** It adds an element to the TOP of the Stack.
2. **Pop:** It removes an element from the TOP of the Stack.
3. **Peek:** It is used to know/display the value of TOP without removing it.
4. **isEmpty:** It is used to check whether the Stack is empty.

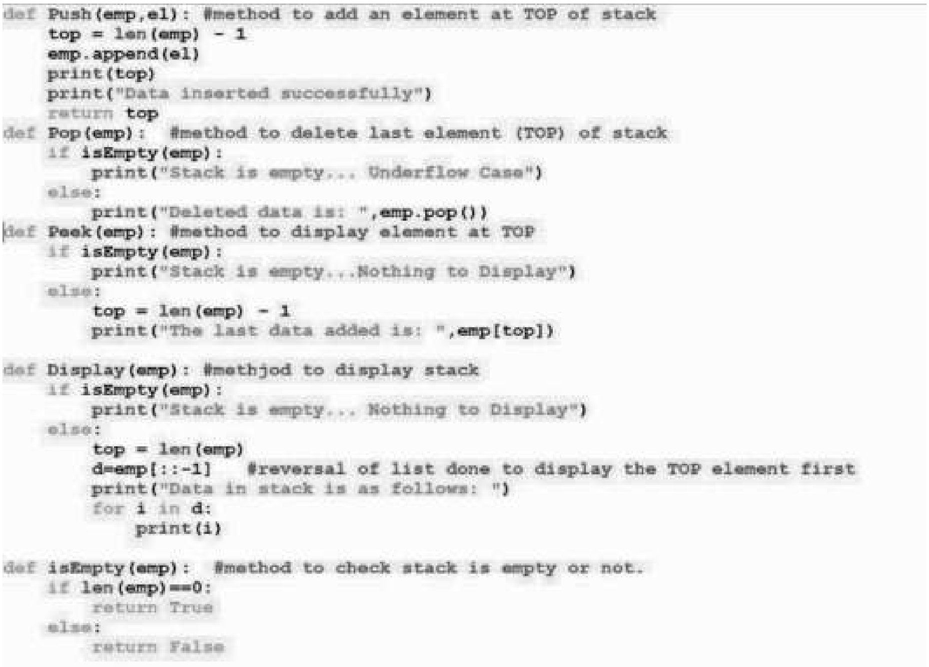
OVERFLOW: It refers to the condition in which we try to PUSH an item in a Stack which is already FULL. **UNDELOW:** It refers to the condition in which we are trying to POP an item from an empty Stack.



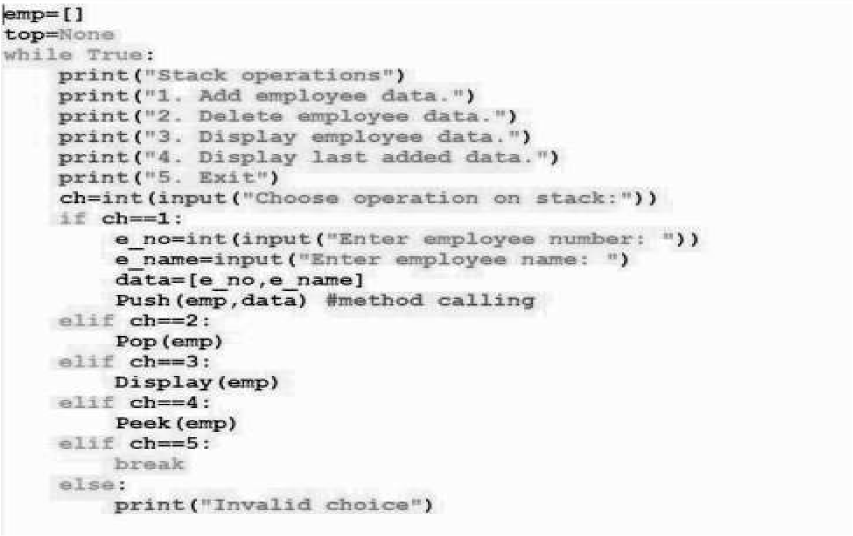
Implementation of Stack:

In Python, the List data structure is used to implement Stack. For the PUSH operation, we use **the append()** method of List while for the POP operation, we use the **pop()** method of List.

PROGRAM: To illustrate the basic operations of Stack:



```
def Push(emp,el): #method to add an element at TOP of stack
    top = len(emp) - 1
    emp.append(el)
    print(top)
    print("Data inserted successfully")
    return top
def Pop(emp):  #method to delete last element (TOP) of stack
    if isEmpty(emp):
        print("Stack is empty... Underflow Case")
    else:
        print("Deleted data is: ",emp.pop())
def Peek(emp): #method to display element at TOP
    if isEmpty(emp):
        print("Stack is empty...Nothing to Display")
    else:
        top = len(emp) - 1
        print("The last data added is: ",emp[top])

def Display(emp): #methjod to display stack
    if isEmpty(emp):
        print("Stack is empty... Nothing to Display")
    else:
        top = len(emp)
        d=emp[::-1]   #reversal of list done to display the TOP element first
        print("Data in stack is as follows: ")
        for i in d:
            print(i)

def isEmpty(emp):  #method to check stack is empty or not.
    if len(emp)==0:
        return True
    else:
        return False
```



```
emp=[]
top=None
while True:
     print("Stack operations")
     print("1. Add employee data.")
     print("2. Delete employee data.")
     print("3. Display employee data.")
     print("4. Display last added data.")
     print("5. Exit")
     ch=int(input("Choose operation on stack:"))
     if ch==1:
         e_no=int(input("Enter employee number: "))
         e_name=input("Enter employee name: ")
         data=[e_no,e_name]
         Push(emp,data) #method calling
     elif ch==2:
         Pop(emp)
     elif ch==3:
         Display(emp)
     elif ch==4:
         Peek(emp)
     elif ch==5:
         break
     else:
         print("Invalid choice")
```

2. A list contains the following record of customers: [CBSE EXAM 2022-23]

[Customer_name, Room_type]

Write the following user-defined functions to perform given operations on the stack named "Hotel":

i. Push_Cust() - Push customer's names of those customers who are staying in the 'Delux' Room Type.

ii. Pop_Cust() - To Pop the names of Customers from the stack and display them. Also, display "Underflow" when there are no customers in the stack.

For example: If the list with customer details is as follows:

["Siddarth", "Delux"]

[“Rahul”, ”Standard”]

[“Jerry”, “Delux”]

The stack should contain:

Jerry

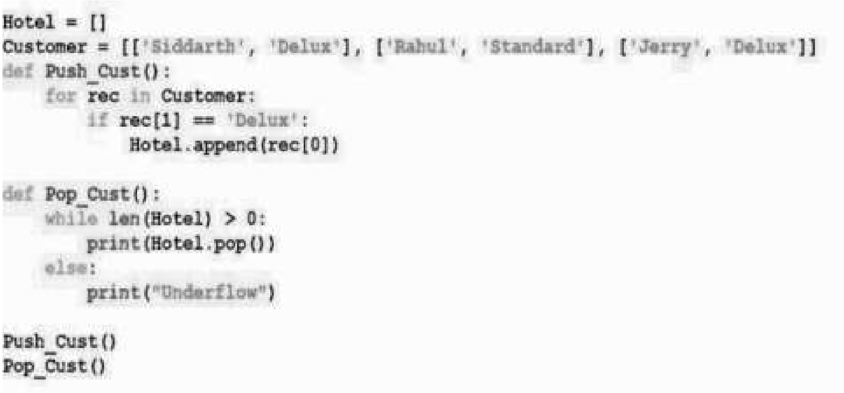
Siddharth

The output should be:

Jerry

Siddharth

Underflow



```
Hotel = []
Customer = [['Siddarth', 'Delux'], ['Rahul', 'Standard'], ['Jerry', 'Delux']]
def Push_Cust():
    for rec in Customer:
        if rec[1] == 'Delux':
            Hotel.append(rec[0])

def Pop_Cust():
    while len(Hotel) > 0:
        print(Hotel.pop())
    else:
        print("Underflow")

Push_Cust()
Pop_Cust()
```

Assignment

PART A

1 Sanya wants to remove an element from the empty stack. Which of the following terms is related to this?

(a) Empty Stack (b) Overflow (c) Underflow (d) Clear Stack

2 In Stack, all operations take place at___________.

a. Top

b. Front

c. Rear

d. Any

3 Insertion and Deletion operations of the Stack are known as ___________ respectively.

a. Insertion and Deletion

b. Push and Pop

c. Pop and Push

d. Enqueue and Dequeue

4 When a Stack is empty, the **TOP** will reset to:

a. None

b. 1

c. -1

d. None of the above.

5 Which of the following is not an inherent application of Stack?

a. Reversing a String.

b. Evaluation of postfix expression.

c. Implementation of recursion.

d. Job Scheduling.

6 Data structures are:

a. Network structures.

b. Group of data.

c. Different types of data

d. Different operations on data

PART B

ASSERTION & REASON:

(A): Both A and R are true and R is the correct explanation for A.

(B): Both A and R are true and R is not the correct explanation for A.

(C): A is true but R is false.

(D): A is false but R is true.

1. **ASSERTION (A):** Using append(), many elements can be added at a time.
   REASON(R): For adding more than one element, the extend() method can be used.
2. **ASSERTION (A):** A data structure is a named group of data types.
   REASON(R): A data structure has well-defined operations, behaviour and properties.
3. **ASSERTION (A):** LIFO is a technique to access data from queues.
   REASON(R): LIFO stands for Last In First Out.
4. **ASSERTION (A):** A Stack is a Linear Data Structure that stores the elements in LIFO order.
   REASON(R): A new element is added at one end and the element is removed from that end only.
5. **ASSERTION (A):** A situation occurs when one tries to delete an element from an empty stack.
   REASON(R): This situation is called an Overflow.
6. **ASSERTION (A):** A stack is a LIFO structure.
   REASON (R): Any new element pushed into the stack always gets positioned at the index after the last existing element in the stack.

PART C

1 Write a function in Python POPSTACK (L) where L is a stack implemented by a list of numbers. The function returns the value deleted from the stack.

2

A list contains the following record of a customer:

[Customer_name, Phone_number, City]

Write the following user-defined functions to perform given operations on the stack named 'status':

(i) Push_element() - Push an object containing the name and Phone number of customers who live in Goa to the stack

(ii) Pop_element() - Pop the objects from the stack and display them. Also, display "Stack Empty" when there are no elements in the stack.

For example: If the lists of customer details are:

["Gurdas", "99999999999","Goa"]

["Julee", "8888888888","Mumbai"]

["Murugan","77777777777","Cochin"]

["Ashmit", "1010101010","Goa"]

the stack should contain

["Ashmit","1010101010"]

["Gurdas","9999999999"]

The output should be:

["Ashmit","1010101010"]

["Gurdas","9999999999"] Stack Empty

3

Write a function in Python, Push(SItem) where SItem is a dictionary containing the details of stationery items– {Sname:price}.

The function should push the names of those items in the stack that have a price greater than 75. Also, display the count of elements pushed into the stack. For example: If the dictionary contains the following data:

Ditem={"Pen":106,"Pencil":59,"Notebook":80,"Eraser":25}

The stack should contain: Notebook

Pen

Output should be: 'Stack has 2 items'.

4. Write a function, Push (Vehicle) where Vehicle is a dictionary containing details of vehicles – {Car_Name: Maker}. The function should push the name of a car manufactured by 'TATA' (including all the possible cases like Tata, TaTa, etc.) to the stack. For example:

If the dictionary contains the following data:

Vehicle = {'Santro': 'Hyundai', 'Nexon': 'TATA', 'Safari' : 'Tata'}

The stack should contain:

Safari

Nexon [CBSE EXAM 2022-23]

5. A list, NList, contains the following records as list elements:

[City, Country, distance from Delhi]

Each of these records is nested together to form a nested list. Write the following user-defined functions in Python to perform the specified operations on the stack named travel.

i. Push_element(NList): It takes the nested list as an argument and pushes a list object containing the name of the city and country, which are not in India and the distance is less than 3500 km from Delhi.

ii. **Pop_element():** It pops the objects from the stack and displays them. Also, the function should display "Stack Empty" when there are no elements in the stack.

For example: If the nested list contains the following data:

NList=[["Newyork","USA",11734], ["Naypyidaw", "Myanmar",3219], ["Dubai","UAE",2194], ["London","England",6693], ["Gangtok", "India",1580], ["Columbo", "Sri Lanka", 3405]]

The stack should contain:

["Naypyidaw", "Myanmar",3219], ["Dubai","UAE",2194], ["Columbo", "Sri Lanka", 3405]

The output should be:

["Columbo", "Sri Lanka", 3405]
["Dubai","UAE",2194]
["Naypyidaw", "Myanmar",3219]
Stack Empty [CBSE SQP 2023]

6. Write a function in Python, Push(SItem) where, SItem is a dictionary containing the details of stationary items—{Sname:price}. The function should Push the names of those items in the stack that have a price greater than 75. Also, display the count of elements pushed into the stack.

For example:

If the dictionary contains the following data:

SItem: {'Pen': 106, 'Pencil': 59, 'Notebook': 80, 'Eraser':25}

The stack should contain:

Notebook
Pen

The output should be: The count of elements in the stack is 2. [CBSE SQP 2022]

LAB EXERCISE

1 Create a stack that stores dictionaries as elements. Each dictionary represents a person's information (name, age, city).

- Implement a `push_dict` method to push a dictionary onto the stack of persons above age 20.
- Implement a `pop_dict` method to pop the top dictionary from the stack.

2 Create a stack to manage student records. Each student record is represented as a dictionary containing attributes like student ID, name, and GPA.

- Implement a `push_student` method to push student records onto the stack with GPA above 60.
- Implement a `pop_student` method to pop the top student record from the stack.

3 Assume a nested dictionary. Each dictionary can contain other dictionaries as values.the format of the dictionary is as follows: {1:{‘a’:’one,’b’:’two},2:{‘x’:10},3:{‘y’:100,’z’:200}....} Create a stack that stores dictionaries as elements.

- Implement a `push_nested_dict` method to push the values of a given nested dictionary onto the stack.
- Implement a `pop_nested_dict` method to pop the top element from the stack.

for example : after implementing push_nested_dict() ,

the stack becomes: {‘y’:100,’z’:200}

{‘x’:10}

{‘a’:’one,’b’:’two}

4 Write the definition of a function POP_PUSH (LPop, LPush, N) in Python. The function should Pop out the last N elements of the list LPop and Push them into the list LPush.
For example:
If the contents of the list LPop are [10, 15, 20, 30] and the value of N is 2, then the function should create the list LPush as [30, 20] And the list LPop should now contain [10, 15]
NOTE: If the value of N is more than the number of elements present in LPop, then display the message "Pop not possible".

5 A list contains the following records of customers:
[Customer_name, Room Type]
Write the following user-defined functions to perform given operations on the stack

named 'Hotel':

- **PushCust()** - To Push customers' names of those customers who are staying in the 'Delux' Room Type.
- **PopCust()** - To Pop the names of customers from the stack and display them. Also, display "Underflow" when there are no customers in the stack.

For example:

If the lists with customer details are as follows:

[["Siddarth", "Delux"] , ["Rahul", "Standard"], ["Jerry", "Delux"]]

The stack should contain:

Jerry
Siddharth

The output should be:

Jerry
Siddharth
Underflow

6 Write a function in Python, Push (Vehicle) where Vehicle is a dictionary containing details of vehicles (Car_Name: Maker). The function should push the name of the car manufactured by TATA (including all the possible cases like Tata, TaTa, etc.) to the stack. 3

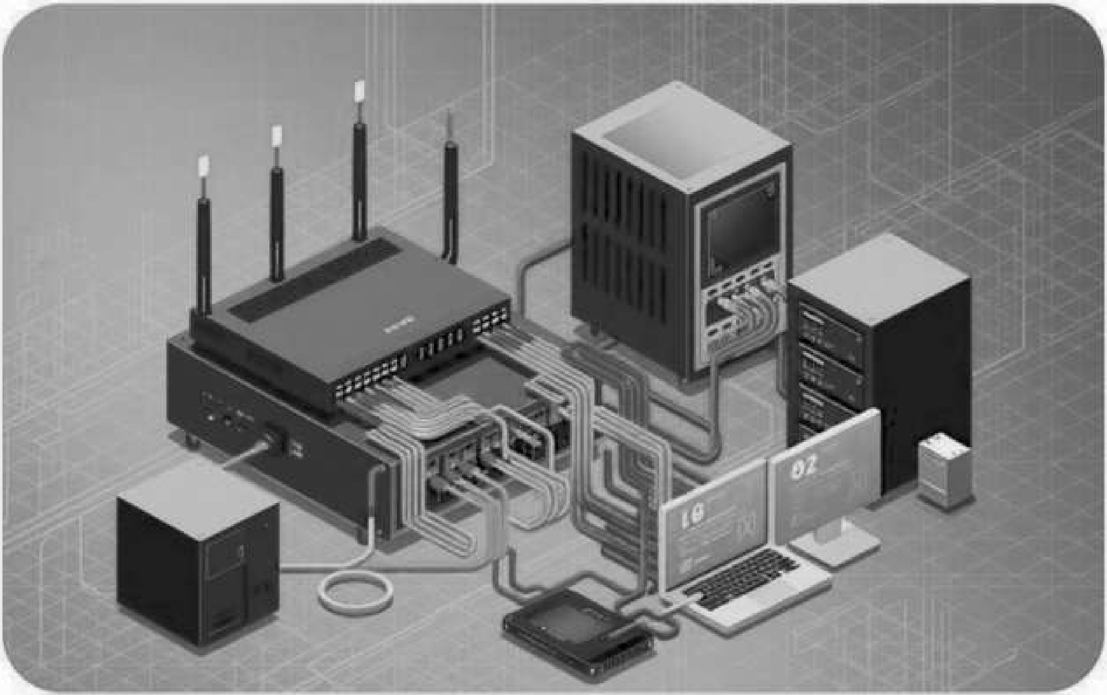
For example:

If the dictionary contains the following data: Vehicle=("Santro": "Hyundai", "Nexon": "TATA", "Safari": "Tata"}

The stack should contain

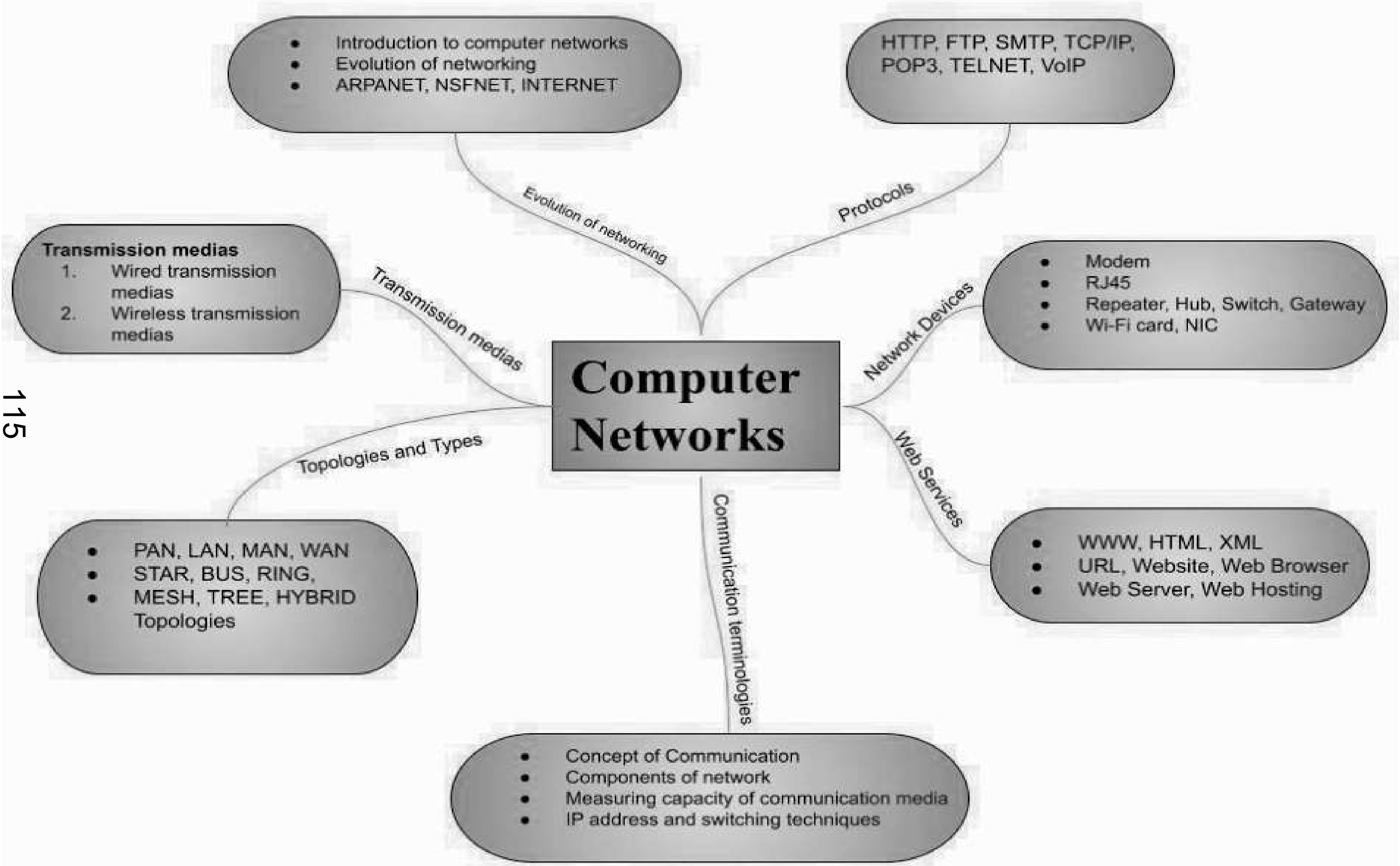
Safari
Nexon

COMPUTER NETWORKS



Topics to be covered:

- Evolution of networking
- Data communication terminologies
- Transmission Media
- Network Devices
- Network Topologies
- Network Types
- Network Protocols


Computer Networks
Evolution of networking
Introduction to computer networks
Evolution of networking
ARPANET, NSFNET, INTERNET
Protocols
HTTP, FTP, SMTP, TCP/IP, POP3, TELNET, VoIP
Transmission medias
Transmission medias
1. Wired transmission medias
2. Wireless transmission medias
Network Devices
Modem
RJ45
Repeater, Hub, Switch, Gateway
Wi-Fi card, NIC
Topologies and Types
PAN, LAN, MAN, WAN
STAR, BUS, RING,
MESH, TREE, HYBRID Topologies
Web Services
WWW, HTML, XML
URL, Website, Web Browser
Web Server, Web Hosting
Communication terminologies
Concept of Communication
Components of network
Measuring capacity of communication media
IP address and switching techniques

Computer Network

ABBREVATIONS:

ARPANET: *Advanced Research Projects Agency Network.*

CDMA: *Code Division Multiple Access*
GSM: *Global System for Mobile communication.*

HTML: *Hyper Text Mark-up Language.*

HTTP: *Hyper Text Transfer Protocol*

IP: *Internet Protocol*

TCP/IP: *Transmission Control Protocol / Internet Protocol*

ISP: *Internet Service Provider*

LAN: *Local Area Network.*

MAN: *Metropolitan Area Network.*

WAN: *Wide Area Network.*

PAN: *Personal Area Network*

CAN: *Campus Area Network*

USB: *Universal Serial Bus*

bps: *Bits per Second.*

Bps: *Bytes per Second.*

Mbps: *Megabits per second*

Gbps: *Gigabits per second*

MODEM: *Modulator Demodulator*

IMEI: *International Mobile equipment Identity.*

www: *World Wide Web.*

MAC: *Media Access Control*

NIC: *Network Interface Card*

P2P: *Peer to Peer Network.*

SMTP: *Simple Mail Transfer Protocol*

VoIP: *Voice over Internet Protocol*

VPN: *Virtual Private Network.*

TP: *Unshielded Twisted Pair*

STP: *Shielded Twisted Pair*

RJ: *Registered Jack*

Wi-Fi: *Wireless Fidelity*

IAAS: *Infrastructure As A Service*

PAAS: *Platform As A Service*

SAAS: *Software As A Service*

DAAS: *Desktop As A Service*

CSMA/CD: *Carrier Sense Multiple Access/ Collision Detection*

CSMA/CA: *Carrier Sense Multiple Access/ Collision Avoidance*

DNS: *Domain Name System*

URL: *Uniform Resource Locator*

FTP: *File Transfer Protocol*

SIM: *Subscriber Identity Module*

HTML: *HyperText Markup Language*

XML: *Extensible Markup Language*

HTTPs: *HyperText Transfer Protocol secure*

GPRS: *General Packet Radio Service*

SMS: *Short Message Service*

TELNET: *Teletype Network*

GPRS: *General Packet Radio Service*

POP: *Post Office Protocol*

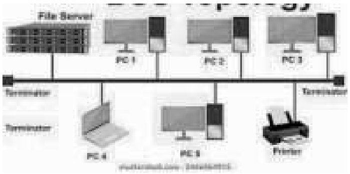
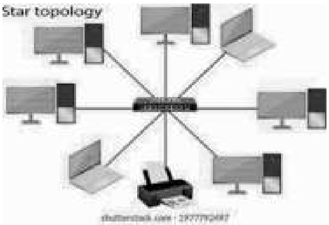
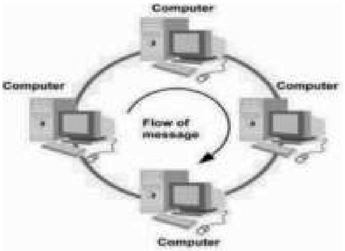
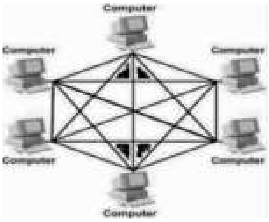
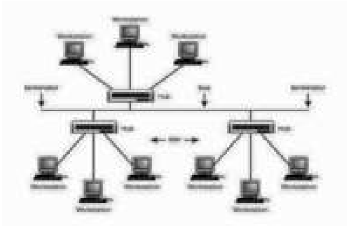
AM: *Amplitude Modulation.*

FDM: *Frequency Division Multiplexing*

TDM: *Time Division Multiplexing*

VoLTE: *Voice over Long-Term Evolution.*

Topologies in Computer Network

Topology	Description	Advantages	Disadvantages	Diagram
Bus Topology	Devices are connected to a single central cable (the bus).	Easy to install and cost-effective.	If the bus cable fails, the whole network stops working.	 File Server PC 1 PC 2 PC 3 Terminator Terminator Terminator PC 4 PC 5 Printer
Star Topology	Devices are connected to a central hub or switch.	Easy to manage and expand. If one device fails, others remain unaffected.	If the hub fails, the entire network stops working.	
Ring Topology	Devices are connected in a circular fashion, with each device connected to two others.	Can be faster for data transfer in small networks.	If the ring is broken, the entire network fails.	 Computer Computer Computer Flow of message Computer
Mesh Topology	Every device is connected directly to every other device.	Very reliable and fault-tolerant. Many paths for data to travel.	Expensive and complex to set up and maintain.	 Computer Computer Computer Computer Computer Computer
Tree Topology	A hybrid of star and bus topologies, with devices arranged in a hierarchical tree structure.	Scalable and can cover large areas.	Can be affected by failures in the root or main branches.	

Transmission media refers to the physical path through which data travels from one device to another in a network. It can be wired or wireless. Here's a simple breakdown of the different types of transmission media:

Transmission Media	Description	Types	Advantages	Disadvantages
Guided Media (Wired)	Data is transmitted through physical cables.	Twisted Pair Cable- Coaxial Cable- Fiber Optic Cable	Stable and secure- High speed with fiber optics- Easy to manage	Expensive installation- Limited mobility- Vulnerable to damage
Unguided Media (Wireless)	Data is transmitted through air, using radio waves or other signals.	Radio Waves- Microwaves- Infrared- Satellite	Flexible and convenient- Can cover large distances- No physical cables needed	Less secure- Interference from weather, buildings, etc.- Can have limited range or slower speeds

Guided Media and Unguided Media:

Guided Media (Wired)

Type	Description	Advantages	Disadvantages
Twisted Pair Cable	Pairs of insulated copper wires twisted together.	Inexpensive- Easy to install- Widely used for LANs	Prone to interference- Limited bandwidth and distance
Coaxial Cable	A single copper wire surrounded by insulation, shielding, and a protective outer layer.	More resistant to interference- Higher bandwidth than twisted pair	Heavier and more expensive than twisted pair- Bulky installation
Fiber Optic Cable	Transmits data using light signals through glass or plastic fibers.	Very high bandwidth- Fast transmission- Long-distance communication	Expensive installation- Fragile and difficult to handle

Unguided Media (Wireless)

Type	Description	Advantages	Disadvantages
Radio Waves	Uses electromagnetic waves to carry data through the air.	Covers large areas- Easy to set up- Works well for mobile devices	Prone to interference- Security concerns- Limited bandwidth
Microwaves	Uses high-frequency radio waves for long-distance communication, often with relay stations.	High-speed transmission- Can cover long distances	Requires clear line-of-sight- Affected by weather conditions
Infrared	Uses light waves to transmit data over short distances.	Simple and inexpensive- Secure (cannot pass through walls)	Short range- Requires line-of-sight- Can be affected by physical obstructions
Satellite	Uses satellites to transmit data to and from the earth.	Can cover vast areas- Available in remote locations	Expensive- High latency- Affected by weather and atmospheric conditions

TYPES OF PROTOCOL

Protocol	Description	Use	Advantages
HTTP (HyperText Transfer Protocol)	Protocol used for transferring web pages on the internet.	Web browsing (used by browsers like Chrome, Firefox).	Simple and widely used for web access. Works seamlessly across different systems.
FTP (File Transfer Protocol)	Used for transferring files between computers over a network.	File sharing and downloading/uploading files.	Supports large file transfers. Allows for both uploading and downloading.
PPP (Point-to-Point Protocol)	A data link layer protocol used to establish a direct connection between two nodes.	Dial-up internet connections, VPNs, and other point-to-point communications.	Provides error detection. Can be used over different physical media.

SMTP (Simple Mail Transfer Protocol)	Protocol for sending emails from one server to another.	Email sending (used by email servers).	Reliable for sending messages. Supports both text and multimedia content.
TCP/IP (Transmission Control Protocol/Internet Protocol)	A set of protocols that defines the standards for data transmission over the internet.	Basic communication on the internet, web browsing, email, file transfer, etc.	Robust and scalable. Widely used as the internet communication standard.
POP3 (Post Office Protocol 3)	Used to retrieve emails from a mail server to a local client.	Email retrieval (download emails to your device).	Easy to use and set up. Offline access to downloaded emails.
HTTPS (HyperText Transfer Protocol Secure)	A secure version of HTTP that encrypts data for secure communication.	Secure web browsing (banking, online shopping).	Provides encryption for security. Ensures data privacy and integrity.
TELNET	A protocol used for remote access to another computer over a network.	Remote login to devices like routers or servers.	Simple and quick access to remote systems. Allows control over remote devices.
VoIP (Voice over Internet Protocol)	A technology that allows voice communication over the internet.	Voice calls over the internet (Skype, WhatsApp).	Cost-effective for long-distance calls. Can support video calls and conferencing.

Device	**Description**	**Function**
Modem	A device that converts digital signals into analog signals and vice versa.	Connects a home or office network to the internet via telephone or cable lines.
Ethernet Card	A hardware component (also called a NIC - Network Interface Card).	Enables a computer to connect to a network using an Ethernet cable.

RJ45	A physical connector with 8 pins, used in Ethernet networking.	Connects devices like computers, routers, and switches using Ethernet cables.
Repeater	A device that regenerates and amplifies weak network signals.	Extends the range of a network by boosting the signal strength.
Hub	A basic network device that connects multiple devices in a local network.	Broadcasts data to all connected devices, without intelligence to direct data.
Switch	An advanced version of a hub that can direct data to specific devices.	Efficiently forwards data only to the intended device in the network.
Router	A device that connects different networks and directs data between them.	Connects a local network to the internet and manages data traffic between devices and networks.
Gateway	A device that acts as a bridge between different types of networks.	Translates data between networks with different protocols (e.g., LAN to the internet).
WiFi Card	A hardware component (wireless network adapter).	Allows a computer or device to connect to a wireless network (Wi-Fi).

Types of Network:

Type of Network	Full Form	Description	Coverage Area	Examples
PAN	Personal Area Network	A small network designed for personal use, connecting devices like phones, laptops, and printers.	A few meters (around 10m)	Bluetooth connection between phone and headphones, personal hotspot.
LAN	Local Area Network	A network within a small area like a home, office, or school, connecting multiple devices.	A building or campus	Office networks, school computer labs.

MAN	Metropolitan Area Network	A larger network that spans a city or town, connecting multiple LANs.	A city or metropolitan area	Cable TV networks, citywide Wi-Fi networks.
WAN	Wide Area Network	A very large network connecting multiple LANs and MANs over large distances, like countries or continents.	Worldwide	The internet, multinational company networks.

Data Communication Terminologies:

Term	**Description**
Concept of Communication	The process of transferring data, information, or messages from one device to another.
Components of Data Communication	
Sender	The device or entity that initiates the message (e.g., a computer or phone).
Receiver	The device or entity that receives the message.
Message	The information (data, text, image, audio, or video) being transmitted.
Communication Media	The medium through which the message travels (e.g., wired or wireless).
Protocols	A set of rules that define how data is transmitted and received (e.g., HTTP, TCP/IP).
Measuring Capacity of Communication Media	
Bandwidth	The maximum amount of data that can be transmitted over a communication channel, measured in bits per second (bps).
Data Transfer Rate	The actual speed at which data is transmitted, considering factors like bandwidth and network conditions.

IP Address	A unique address that identifies a device on a network, enabling communication between devices. IP addresses can be **IPv4** (e.g., 192.168.1.1) or **IPv6** (e.g., 2001:0db8::1).

Switching techniques:

Switching Technique	Description	How It Works	Examples	Advantages	Disadvantages
Circuit Switching	A dedicated communication path is established between sender and receiver.	A connection is set up before data transfer starts. The path remains reserved for the session.	Traditional telephone networks.	Reliable and consistent connection. No data delays once the circuit is established.	Inefficient for low traffic (path remains reserved even if no data is sent). Time-consuming setup.
Packet Switching	Data is divided into small packets, sent independently, and reassembled at the destination.	Packets take the best available routes. They may arrive out of order and are reassembled at the receiver.	Internet, email, file downloads.	Efficient use of network resources. Handles large volumes of data effectively.	Packets may get delayed or lost. Requires complex protocols for reassembly.

COMPARISON:

Circuit Switching	Packet Switching (Datagram Type)	Packet Switching (Virtual Circuit Type)
Dedicated path	No Dedicated path	No Dedicated path
Path is established for entire conversation	Route is established for each packet	Route is established for entire conversation
Call setup delay	Packet transmission delay	Call setup delay as well as packet transmission delay
Overload may block call setup	Overload increases packet delay	Overload may block call setup and increases packet delay

Feature	Simplex	Half-Duplex	Full-Duplex
Direction of Data	One direction only	Both directions, one at a time	Both directions, simultaneously
Communication	Unidirectional	Bidirectional (alternate)	Bidirectional (simultaneous)
Example	Keyboard to computer	Walkie-talkie	Mobile phone call
Speed	Slow (one-way only)	Moderate (takes turns)	Fast (both sides communicate at once)
Cost/Complexity	Simple and low cost	Moderate complexity	More complex and costly

For scenario-based question from computer network

1. **Install server where maximum computers are placed.**
2. **Use Fiber Optics Cable for high speed data transfer through wired(Guided) media.**
3. **Use Microwave communication for high speed data transfer through wireless(unguided) media.**
4. **Satellite Communication for Global Coverage through wireless media.**
5. **Unshielded Twisted Pair is best when cost is the main factor for wired transmission media.**
6. **Radio waves are best when cost is the main factor for wireless transmission media.**
7. **Use switch/hub to connect all computers within the premises.**
8. **VoIP (Voice over Internet Protocol) will be best suitable for online interaction.**
9. **Telnet Protocol is used for remote login.**
10. **Use Firewall for security enhancement.**
11. **Repeater will be placed between the blocks where distance is more than 100m as per the layout.**

Multiple Choice Questions:

1) Which network topology is characterized by a central node that connects all devices on the network?
   a) Bus
   b) Star
   c) Ring
   d) Mesh
2) In a bus topology, what happens if there is a break in the main communication line?
   a) The entire network becomes inoperative
   b) Only the affected segment becomes inoperative
   c) Data packets are automatically rerouted through an alternative path
   d) The network speed decreases but remains functional
3) Which network topology provides fault tolerance and redundancy due to multiple interconnections between devices?
   a) Bus
   b) Star
   c) Ring
   d) Mesh
4) In a ring topology, what prevents data packets from endlessly circulating the loop?
   a) Bridges
   b) Routers
   c) Token passing
   d) Firewalls
5) What is the main disadvantage of a mesh topology?
   a) High installation cost
   b) Difficult to set up
   c) Limited scalability
   d) Susceptible to data collisions
6) Which network topology is commonly used in Ethernet networks and provides centralized management?
   a) Bus
   b) Star
   c) Ring
   d) Mesh
7) In a star topology, what happens if the central hub fails?
   a) The entire network becomes inoperative
   b) Only the affected segment becomes inoperative
   c) Data packets are automatically rerouted through an alternative path
   d) The network speed decreases but remains functional
8) Which network topology is best suited for small networks and has a simple and cost- effective design?
   a) Bus
   b) Star

c) Ring

d) Mesh

9) What network topology is typically used in token ring networks to regulate data transmission?

a) Bus

b) Star

c) Ring

d) Mesh

10) Which transmission medium uses electrical signals to transmit data over short distances inside a computer or between devices in a local network?

a) Coaxial cable

b) Fiber-optic cable

c) Twisted-pair cable

d) Infrared transmission

11) What type of transmission medium is suitable for underwater communications and long- distance networking?

a) Coaxial cable

b) Fiber-optic cable

c) Twisted-pair cable

d) Infrared transmission

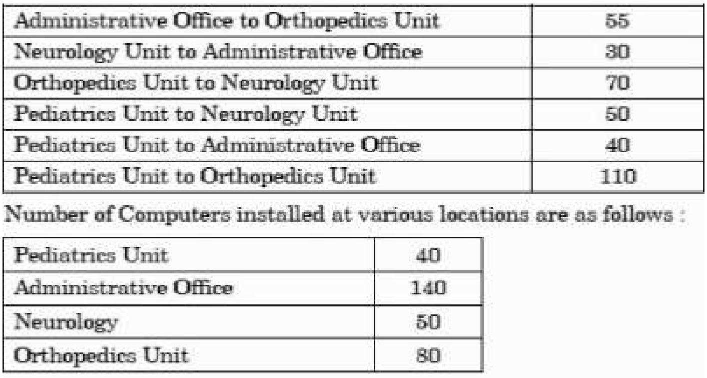
Short Answer Type Questions

1. What is topology? Explain various types of topologies.
2. What is the most commonly used network topology in modern Ethernet LANs?
3. What advantage does the mesh topology offer in terms of reliability?
4. Which topology is commonly used in wireless networks?
5. What advantage does the tree topology have over the bus topology?
6. What are the different types of transmission media?
7. What are the advantages and drawbacks of twisted pair cables?
8. What are the main drawbacks of coaxial cables?
9. What advantage do fiber optic cables have over coaxial cables?
10. Name some examples of devices that use infrared signals.
11. Which type of transmission media offer more security- guided or unguided? Why?
12. Which transmission media is commonly used to transmit data over short and long distances?
13. What are the main factors that influence the choice of transmission media in a network?
14. Explain LAN, MAN, WAN, PAN.
15. What is Virtual Private Network (VPN)?
16. What is a domain name service?

Case Study based questions

Ques: 1. Ayurveda Training Educational Institute is setting up its centre in Hyderabad with four specialized departments for Orthopedics, Neurology and Pediatrics along with an administrative office in separate buildings. The physical distances between these department buildings and the number of computers to be installed in these departments and administrative offices are given as follows. You, as a network expert, have to answer the queries as raised by them in (i) to (iv). Shortest distances between various locations in meters:

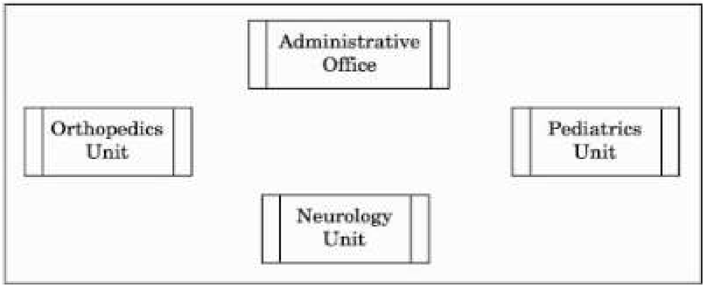
1. Suggest the most suitable location to install the main server of this institution to get efficient connectivity.



Administrative Office to Orthopedics Unit	55
Neurology Unit to Administrative Office	30
Orthopedics Unit to Neurology Unit	70
Pediatrics Unit to Neurology Unit	50
Pediatrics Unit to Administrative Office	40
Pediatrics Unit to Orthopedics Unit	110

Number of Computers installed at various locations are as follows :

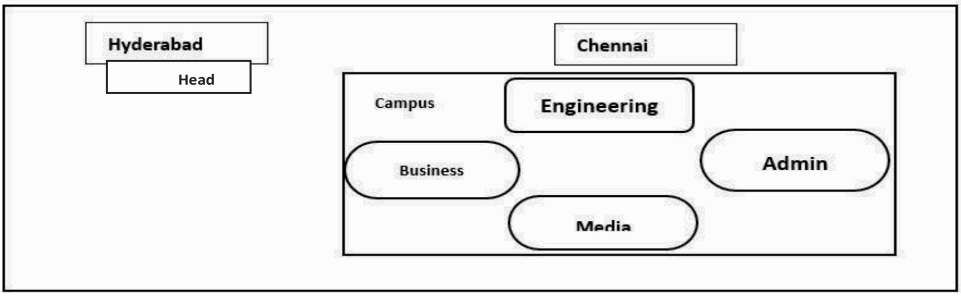
Pediatrics Unit	40
Administrative Office	140
Neurology	50
Orthopedics Unit	80



2. Suggest the best cable layout for effective network connectivity of the building having a server with all the other buildings.
3. Suggest the devices to be installed in each of these buildings for connecting computers installed within the building out of the following: Gateway, Modem, Switch
4. Suggest the topology of the network and network cable for efficiently connecting each computer installed in each of the buildings out of the following:
   Topologies: Bus Topology, Star Topology

   Network Cable: Single Pair Telephone Cable, Coaxial Cable, Ethernet Cable

Ques: 2 M/S Adco Informatics Services is an educational service organization. It is planning to setup its India campus in Chennai with its head office in Hyderabad. The Chennai campus has 4 buildings- ADMIN, MEDIA, ENGINEERING, and BUSINESS.



Block to block distances (in Meters)

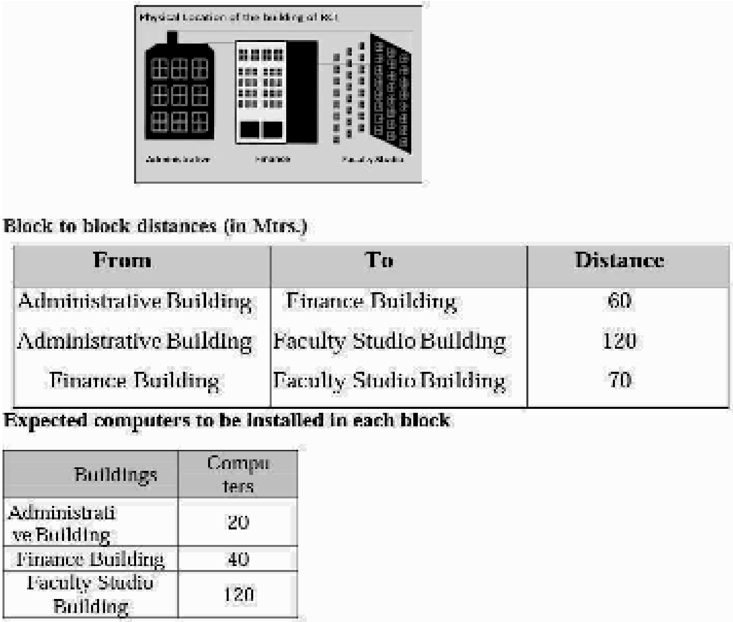
From	To	Distance
Admin	Engineering	45 m
Admin	Business	80 m
Engineering	Business	25 m
Admin	Media	60 m
Engineering	Media	60 m
Business	Media	75 m
Hyderabad Head Office	Chennai Campus	692 Km

The number of Computers in each of the blocks/Centre is as follows:

Admin	140
Engineering	70
Business	35
Media	20
Hyderabad	30

1. Suggest and draw the cable layout to efficiently connect various blocks of buildings within the Chennai campus for connecting the devices.
2. Which network device will be used to connect computers in each block to form a local area network?
3. Which block, in Chennai Campus should be made the server? Justify.
4. Which fast and effective wireless transmission medium should preferably be used to connect the head office at Hyderabad with the campus in Chennai?
5. Is there a requirement for a repeater in the given cable layout? Why/Why not?

Ques: 3 Rocksons Communications International (RCI) is an online corporate training provider company for IT-related courses. The company is setting up their new campus in Bengaluru. You as a network expert have to study the physical locations of various blocks and the number of computers to be installed. In the planning phase, provide the best possible answers for the queries (i) to (v) raised by them.


Physical Location of the building of RCI
Administrative
Finance
Faculty Studio

Block to block distances (in Mtrs.)

From	To	Distance
Administrative Building	Finance Building	60
Administrative Building	Faculty Studio Building	120
Finance Building	Faculty Studio Building	70

Expected computers to be installed in each block

Buildings	Computers
Administrative Building	20
Finance Building	40
Faculty Studio Building	120

i. Suggest the most appropriate block, where RCI should plan to install the server.

ii. Suggest the most appropriate block-to-block cable layout to connect all three blocks for efficient communication.

iii. Which type of a network out of the following is formed by connecting the computers of these three blocks? (LAN, MAN, WAN, PAN)

iv. Which wireless channel out of the following should be opted by RCI to connect to students from all over the world? (Infrared, Microwave, Satellite)

v. What is the satellite communication?

Ques: 4 Ayurveda Training Educational Institute is setting up its centre in Hyderabad with four specialized departments for Orthopedics, Neurology and Pediatrics along with an administrative office in separate buildings. The physical distances between these department buildings and the number of computers to be installed in these departments and administrative offices are given as follows. You, as a network expert, have to answer the queries raised by them.

Shortest distances between various locations in metres :

Administrative Office to Orthopedics Unit	76
Neurology unit to Administrative Office	25
Orthopedics Unit to Neurology Unit	50
Pediatrics Unit to Neurology Unit	40
Pediatrics Unit to Administrative Office	65
Pediatrics Unit to Orthopedics Unit	150

The number of Computers installed at various locations is as follows:

Pediatrics Unit	45
Administrative Office	190
Neurology	60
Orthopedics Unit	85

i. Suggest the most suitable location to install the main server of this institution to get efficient connectivity.

ii. Suggest the best cable layout for effective network connectivity of the building having a server with all the other buildings.

iii. Suggest the devices to be installed in each of these buildings for connecting computers installed within the building out of the following:

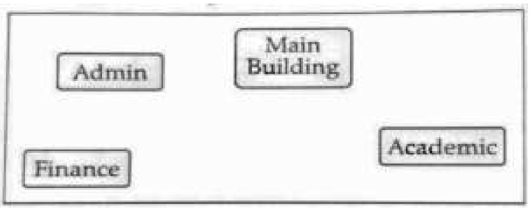
(1) Gateway (2) Modem (3) Switch

iv. Suggest the topology of the network and network cable for efficiently connecting each computer installed in each of the buildings out of the following:

Topologies: Bus Topology, Star Topology

Network Cable: Single Pair Telephone Cable, Coaxial Cable, Ethernet Cable.

Ques: 5 Sanskar University of Uttarakhand is setting up a secured network for its campus at Nainital for operating day-to-day office and web-based activities. They are planning to have network connectivity between four buildings. Answer the questions after going through the building positions on the campus and other details given below:



The distance between various buildings of the university is given as follows:

Building 1	**Building 2**	**Distance (in mtrs.)**
Main	Admin	50
Main	Finance	100
Main	Academics	70
Admin	Finance	50
Finance	Academics	70
Admin	Academics	40

Number of Computers:

Building	**No. of Computers**
Main	200
Admin	100
Finance	75
Academics	70

As an expert, you are required to give the best possible solutions for the given queries of the university administration:

a. Suggest the cable layout for the connection between the various buildings.

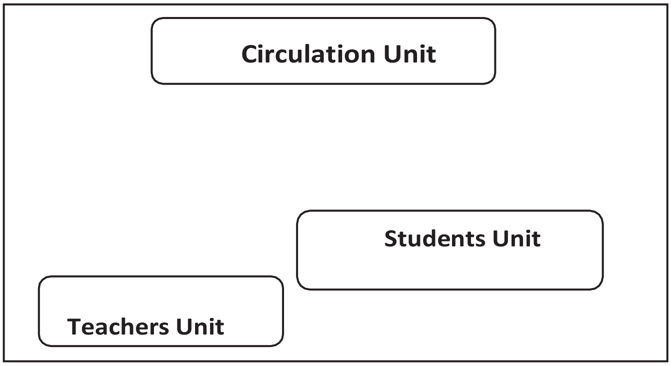
b. Suggest the most suitable building to house the server of the network of the university.

c. Suggest the placement of the following devices with justification: Switch/Hub Repeater

d. Suggest the technology out of the following for setting-up very fast internet connectivity among buildings of the university

Optical Fibre Coaxial Cable Ethernet Cable

Ques: 6 A school library is connecting computers in its units in a LAN. The library has 3 units as shown in the diagram below:



The three units are providing the following services:

Teachers Unit: For the access of the Library Books by teachers

Students Unit: For access to the Library Books by students

Circulation Unit: For issue and return of the books for teachers and students.

Centre to Centre distances between the 3 units are as follows:

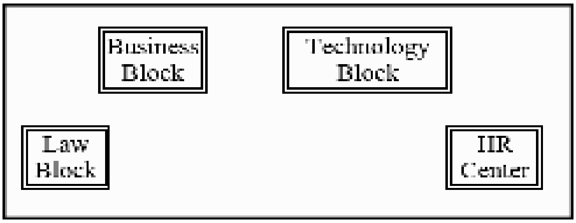
- Circulation Unit to Teachers Unit – 30 meters
- Circulation Unit to Students Unit – 25 meters
- Teachers Unit to Students Unit – 15 meters

The number of computers in each of the units is as follows:

- Circulation Unit 20
- Teachers Unit 15
- Students Unit 15

1. Suggest the most suitable place to install the server of this library with a suitable reason.
2. Suggest an ideal layout for connecting these units for wired connectivity.
3. Which device will you suggest to be installed and where should it be placed to provide Internet connectivity to all the units?
4. Suggest the type of the most efficient and economical wired medium for connecting all the computers in the network.
5. The university is planning to connect the library with the school Principal's computer which is in his office at a distance of 50 meters. Which type of network of LAN, MAN or WAN will be used for the network? Justify your answer.

Ques: 7 MyPace University is setting up its academic blocks at Naya Raipur and is planning to set up a network. The University has 3 academic blocks and one Human Resource Centre as shown in the diagram below:



Centre to Centre distances between various blocks/centres are as follows:

Law Block to Business Block	40 m
Law Block to Technology Block	80 m
Law Block to HR Block	105 m
Business Block to Technology Block	30 m
Business Block to HR Centre	35 m
Technology Block to HR Centre	15 m

The number of computers in each of the blocks/Centre is as follows:

Law Block: 15

Technology Block: 40

HR Centre: 115

Business Block: 25

a. Suggest the most suitable place (i.e., Block/Centre) to install the server of this university with a suitable reason.

b. Suggest an ideal layout for connecting these blocks/centres for wired connectivity.

c. Which device will you suggest to be placed/installed in each of these blocks/centres to efficiently connect all the computers within these blocks/centres?

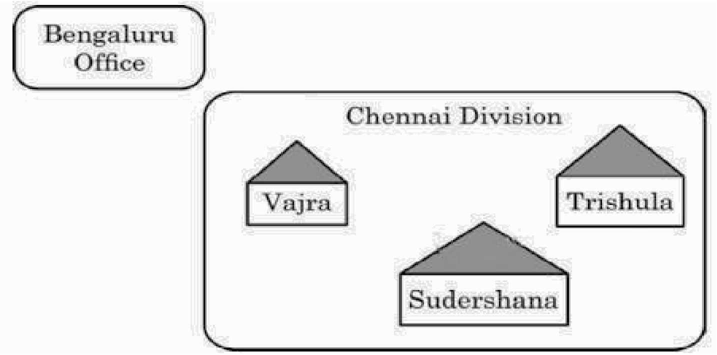
d. Suggest the placement of a Repeater in the network with justification.

e. The university is planning to connect its admission office in Delhi, which is more than 1250 Km from the university. Which type of network of LAN, MAN or WAN will be formed? Justify your answer.

Exercise Case Study Based Questions

Ques 1. Infotainment Ltd. Is an event management company with its prime office located in Bengaluru. The company is planning to open its new division at three different locations in Chennai named Vajra, Trishula, and Sudershana.

You, as a networking expert need to suggest solutions to the questions in part (a) to (e), keeping in mind the distances and other given parameters.



Distances between various locations:

Vajra to Trishula	350 m
Trishula to Sudershana	415 m
Sudershana to Vajra	300 m
Bengaluru Office to Chennai	2000 km

Number of computers installed at various locations:

Vajra	120
Sudershana	75
Trishul	60
Bengaluru Office	215

a. Suggest and draw the cable layout to efficiently connect various locations in the Chennai division for connecting digital devices.

b. Which block in the Chennai division should host the server? Justify your answer.

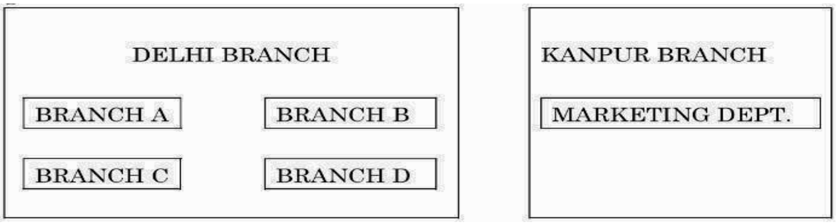
c. Which fast and effective wired transmission medium should be used to connect the prime office at Bengaluru with the Chennai division?

d. Which network device will be used to connect the digital devices within each location of the Chennai division so that they may communicate with each other?

e. A considerable amount of data loss is noticed between different locations of the Chennai division, which are connected to the network. Suggest a networking device that should be installed to refresh the data and reduce the data loss during transmission to and from different locations of the Chennai division.

Ques 2. Quackers Inc., an IT-based firm, located in Delhi is planning to setup a network for its four branches within a city with its Marketing department in Kanpur. As a network professional, give solutions to questions (i) to (v), after going through the branch locations and other details which are given below:

Distance between various branches is as follows:



Branch	Distance
Branch A to Branch B	40 m
Branch A to Branch C	80 m
Branch A to Branch D	65 m
Branch B to Branch C	30 m
Branch B to Branch D	35 m
Branch C to Branch D	15 m
Delhi Branch to Kanpur	300 km

Number of computers in each of the branches:

Branch	Number of Computers
Branch A	15
Branch B	25
Branch C	40
Branch D	115

i.Suggest the most suitable place to install the server for the Delhi branch with a suitable reason.

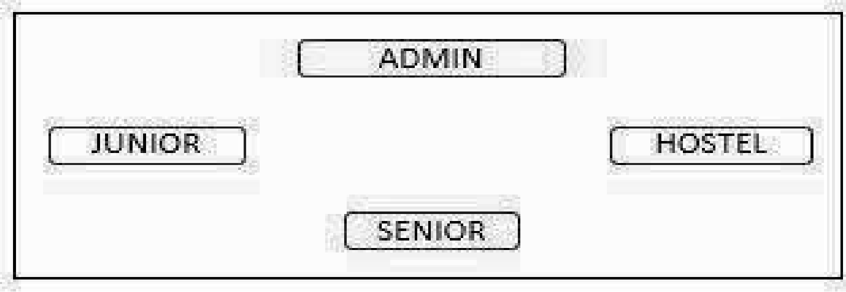
ii. Suggest an ideal layout for connecting all these branches within Delhi.

iii. Which device will you suggest that should be placed in each of these branches to efficiently connect all the computers within these branches?

iv. Delhi firm is planning to connect to its Marketing department in Kanpur which is approximately 300 km away. Which type of network of LAN, MAN, or WAN will be formed? Justify your answer.

v.Suggest a protocol that shall be needed to provide help for transferring files between the Delhi and Kanpur branches.

Ques 3. The Indian Heights School in Mussoorie is starting up the network between its different wings. There are four buildings named SENIOR, JUNIOR, ADMIN, and HOSTEL as shown below.



The distance between various buildings is as follows:

ADMIN to JUNIOR	150 m
HOSTEL TO ADMIN	50 m
SENIOR TO JUNIOR	220 m
SENIOR TO ADMIN	200 m
HOSTEL TO JUNIOR	210 m
SENIOR TO HOSTEL	205 m

a. Suggest a suitable layout of connections between the buildings. Also, give the name of the topology used.

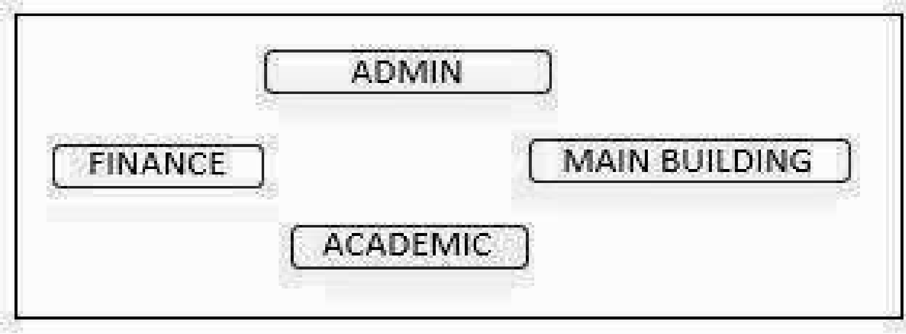
b. Suggest the most suitable place (i.e. building) to house the server of this school with a suitable reason.

c. Suggest the placement of the following devices with justification:
- Repeater
- Hub/Switch

d. The organization also has an enquiry office in another city about 65 km away in Roorkee. Suggest the suitable transmission media to interconnect to the school and inquiry office out of the following:
- Fiber Optic Cable
- Microwave
- Radio-wave

Also, give the type of this network.

Ques 4. The Aakar University of Madhya Pradesh is setting up a secured network for its campus in Himachal Pradesh for operating its day-to-day office and web-based activities. They are planning to have network connectivity between four buildings. Answer the question (a) to (d) after going through the building positions on the campus and other details which are given below:



The distances between various buildings of the university are given as:

Building 1	Building 2	Distance (in meters)
Main	Admin	50
Main	Finance	100
Main	Academic	70
Admin	Finance	50
Finance	Academic	70
Admin	Academic	60

Number of Computers:

Building	No. of Computers
Main	150
Admin	75
Finance	50
Academic	60

As a network expert, you are required to give the best possible solutions for the given queries of the university administration:

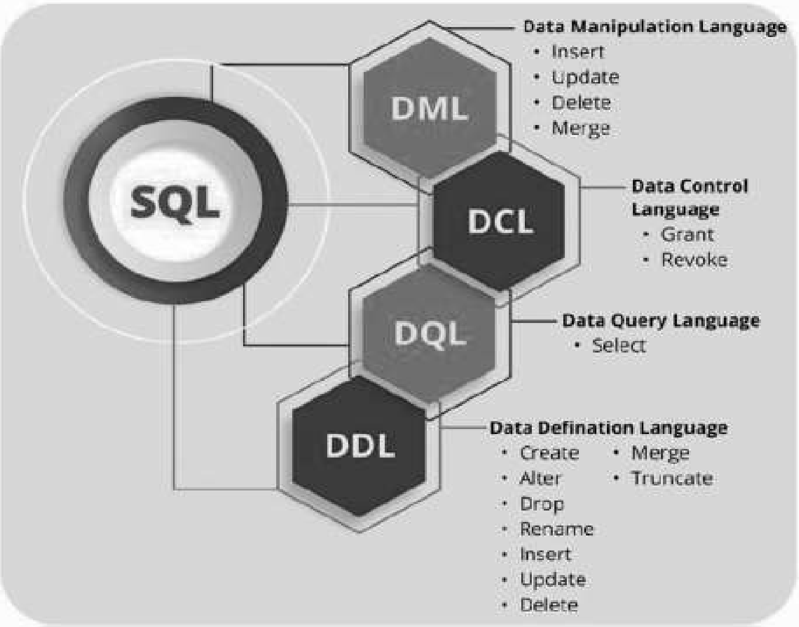
a. Suggest the cable layout for the connections between the various buildings.

b. Suggest the most suitable building to house the server of the network of the university.

c. Suggest the placement of the following with justification:
- Switch/Hub
- Repeater

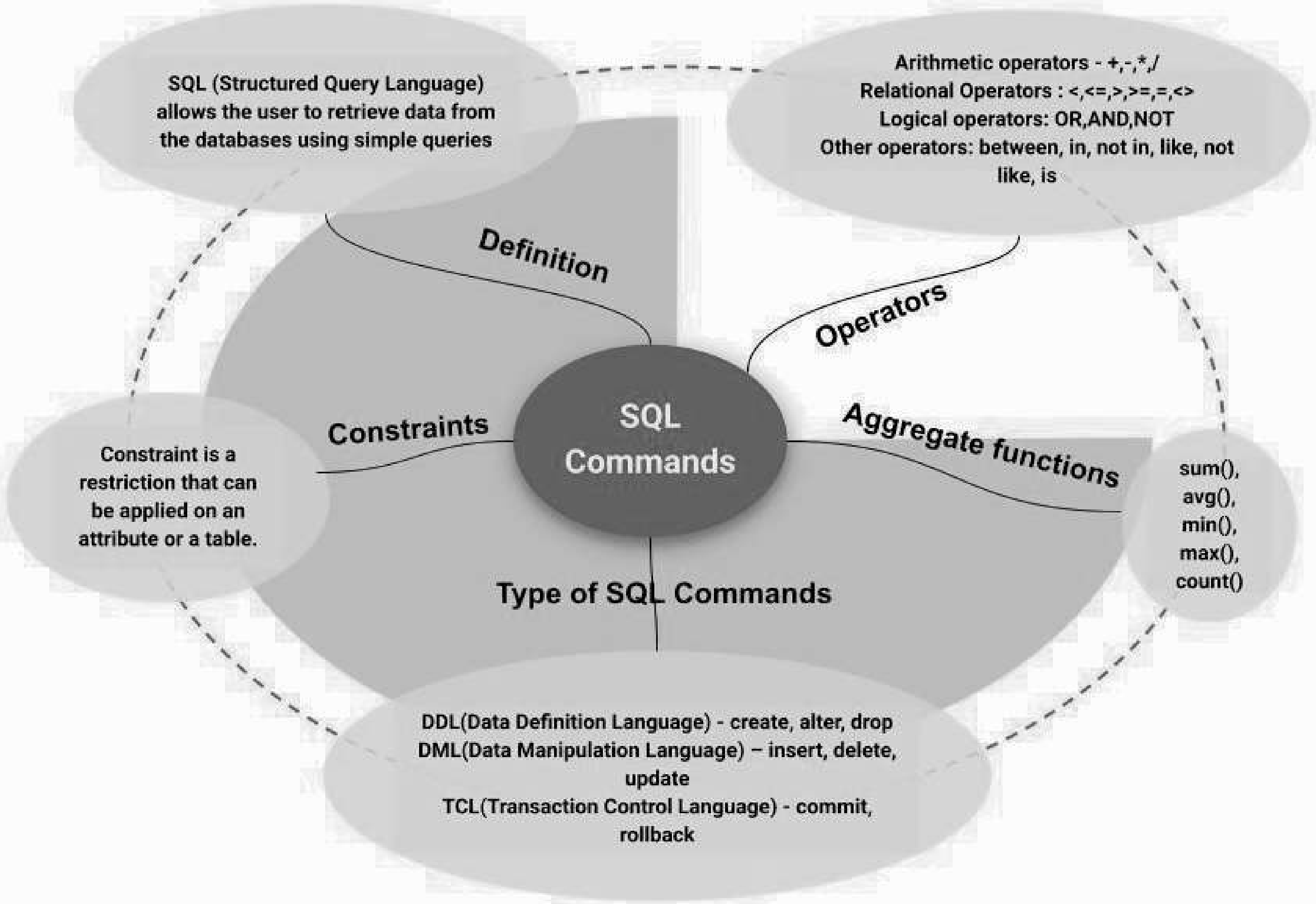
d. Suggest the communication media out of the following for setting-up very fast Internet connectivity among buildings of the university
- Optical Fiber
- Coaxial Cable
- Ethernet Cable

DATABASE MANAGEMENT SYSTEMS



Topics to be covered

- ✓ **Database concepts**
- ✓ **Relational data model**
- ✓ **Structured Query Language**
- ✓ **Interface of Python with an SQL database**



SQL Commands

Definition

SQL (Structured Query Language) allows the user to retrieve data from the databases using simple queries

Operators

Arithmetic operators - +,-,*,/
Relational Operators : <,<=,>,>=,=,<>
Logical operators: OR,AND,NOT
Other operators: between, in, not in, like, not like, is

Aggregate functions

sum(),
avg(),
min(),
max(),
count()

Type of SQL Commands

DDL(Data Definition Language) - create, alter, drop
DML(Data Manipulation Language) – insert, delete, update
TCL(Transaction Control Language) - commit, rollback

Constraints

Constraint is a restriction that can be applied on an attribute or a table.

UNIT-III

DATABASE MANAGEMENT

DATABASE CONCEPT:

INTRODUCTION OF DATABASE:

Database is a word which is composed of two words: Data and Base. **Data** means raw facts and figures and **base** is the place or location where data is being stored.

Or we can say that a Database is a collection of **interrelated data or records** in an organized form. So, it can be easily accessed, managed and updated**. Interrelated data** means that the data is related or connected with respect to the given attribute/column.

The database uses various fields to manage and store large amounts of information in an organized and structured format.

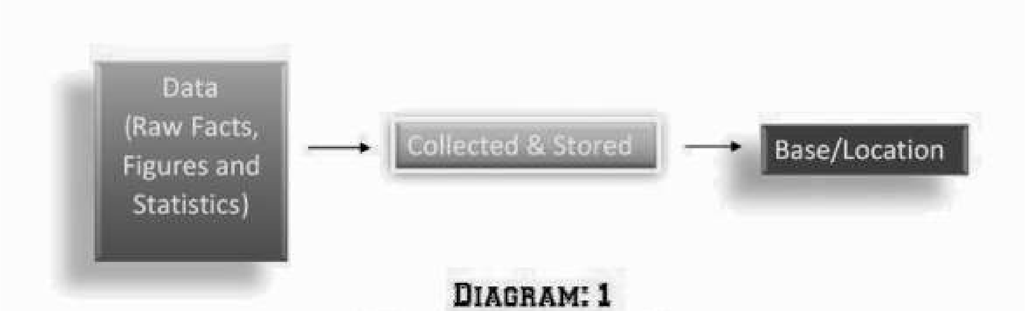


DIAGRAM: 1
HOW DATABASE WORKS

Introduction of Database Management System (DBMS):

DBMS is a shortened name used for Database Management System. Now as we are aware of databases, we need to understand what DBMS actually is.

DBMS is a software system which is used to manage databases. DBMS acts as an interface between a user and a database which enables the user to create, insert, retrieve, update and delete the data.

Need of Database:

1. **Centralized Storage**: Storage of data in a single location or central database.
2. **Data Integrity**: Enforces data integrity rules which ensures that the information stored is accurate, valid and consistent.
3. **Data Security**: Control access to sensitive data and protect data from unauthorized access.
4. **Data Retrieval**: Authorized Users/Applications can access and retrieve the information as per their need.
5. **Efficient Data Retrieval**: Database helps users to retrieve data efficiently.

DBMS Model:

DBMS refers to the architecture/approach for how data is stored, organized and manipulated in a database. There are several types of DBMS Models.

1. **Relational Model:**
   Data is organized in tables with rows and columns.
2. **Hierarchical Model:**
   Data is organized in a Tree-like structure to the parent-child relationship.
3. **Network Model**:
   Similar to the hierarchical model. It uses pointers to navigate through data.
4. **Object-Oriented Model:**
   Data is represented as an object. This model uses object-oriented databases.

Now as per your CBSE syllabus, we will discuss the Relational Data Model in detail.

DBMS1 - QUESTION RELATED TO THE DATABASE CONCEPT

- What is a database and how is it different from a database management system?
- What do you mean by raw facts and figures in a database?
- Why do we need a database?
- In which DBMS model data is organized into rows and columns?

Relational Data Model:

The relation Data Model was proposed by E.F. Codd in 1970.

In simple words, we can say that a Relational data model is a model which uses **relation** to organize **data**. Here, **Relation means table** and table is composed of **rows** and **columns**.

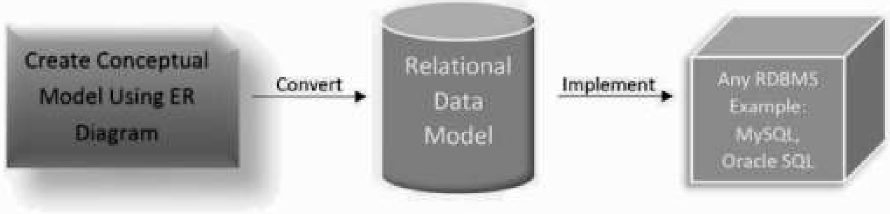

Create Conceptual Model Using ER Diagram
Convert
Relational Data Model
Implement
Any RDBMS Example: MySQL, Oracle SQL

DIAGRAM: 2

ABOVE DIAGRAM SHOWS THAT HOW TO IMPLEMENT RELATIONAL DATA MODEL

After creating a conceptual data model using an ER (Entity Relationship) Diagram we need to convert it into a Relational Data Model so that we can implement it using any RDBMS language like MySQL, Oracle SQL etc.

Before we proceed further, let's discuss some aspects of the Conceptual Data Model

Conceptual Data Model is used to capture the meaning of data from the viewpoint of the user and try to represent it using Data Model tools like ER diagrams and Object Oriented Diagram

Relation/Table:

Relation is also known as table and it is a collection of related data and information. The relation is composed of rows and columns.

Row/Tuple/Record:

Row represents the horizontal form of Table/Relation. A row is also known as a tuple/record.

Column/Attributes:

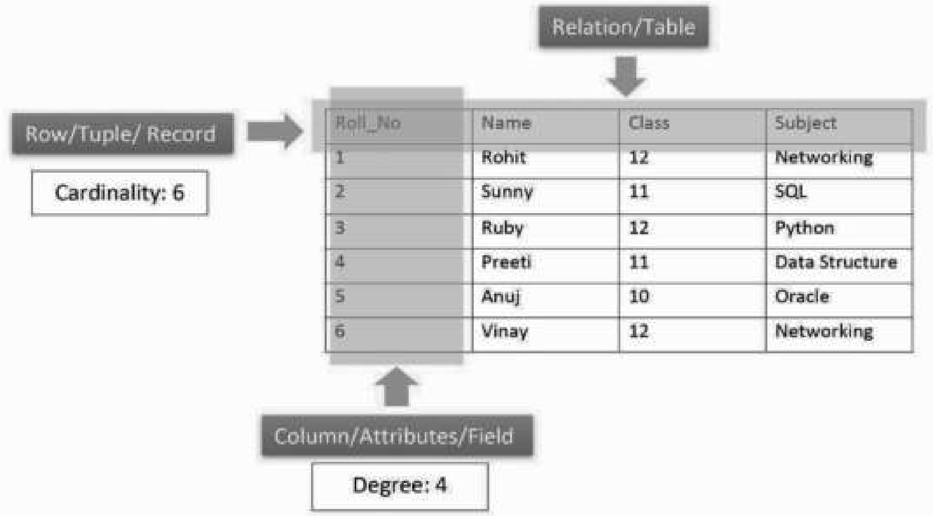
Columns represent the vertical form of the Table/Relation. The column is also known as an attribute.

Cardinality:

The total number of rows/records/tuples in a relation is called Cardinality.

Degree:

The total number of columns/attributes in a relation is called the Degree.



Roll_No	Name	Class	Subject
1	Rohit	12	Networking
2	Sunny	11	SQL
3	Ruby	12	Python
4	Preeti	11	Data Structure
5	Anuj	10	Oracle
6	Vinay	12	Networking

DIAGRAM: 3

RELATION/TABLE COMPOSED OF ROW/RECORD AND COLUMN/ATTRIBUTES

Domain:

Domain is a **set of possible values** or **range of valid values** or a **set of all unique values** that an attribute/column can hold.

A Domain of a database is a set of atomic values (which can’t further be distributed) of a particular attribute/column.

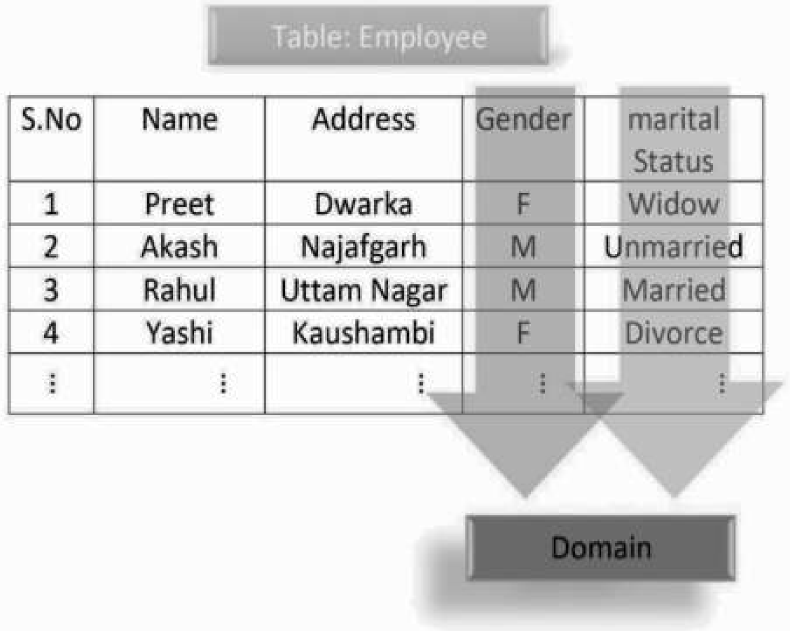
For Example:

In the table Employee, Attribute/column ‘gender’ may have only ‘M’, ‘F’ and ‘T’ domains. Only these values are valid for that column.

The domain of ‘S_No’ contains a set of all possible roll numbers.

Domain of ‘Martial_Status’ contains a set of all possible values like ‘Married’, ‘Widow’, ‘Unmarried’ or ‘Divorce’.

In the below diagram 4, table Employee contains ‘S_No’, ‘Name’, ‘Address’, ‘Gender’ and ‘Marital_status’. Two domains are showing ‘gender’ and ‘marital_status’ which contain a set of possible values that an attribute can hold. Gender can only hold three possible values and marital status can only hold four possible values.



S.No	Name	Address	Gender	marital Status
1	Preet	Dwarka	F	Widow
2	Akash	Najafgarh	M	Unmarried
3	Rahul	Uttam Nagar	M	Married
4	Yashi	Kaushambi	F	Divorce
⋮	⋮	⋮	⋮	⋮

DIAGRAM: 4

DOMAIN IN A RELATION

DATATYPES IN SQL:

Before discussing commands in detail we need to learn about the datatype of column/attribute:

We need to assign a data type when we are declaring any column/attribute. Every column required a name and datatype. Data type is used to declare the type of data that will be stored in a particular column. There are lots of data types available in SQL. We will discuss some important data types here.

Commonly used datatype in SQL:

1. Numeric Type:

- INT: Integer type
- FLOAT: Floating-point number
- DECIMAL or NUMERIC: Fixed-point number

2. Character String Type:

- CHAR(n): Fixed-length character string with a maximum length of n
- VARCHAR(n): Variable-length character string with a maximum length of n
- TEXT Type : Variable-length character string with no specified maximum length

3. Date and Time Type:

- DATE : for date only
- TIME: for time only
- DATETIME or TIMESTAMP: for date and time combined

4. Other Data type:

- NULL: to represent a missing/unknown/empty value
- ENUM: An enumeration type for a set of predefined values

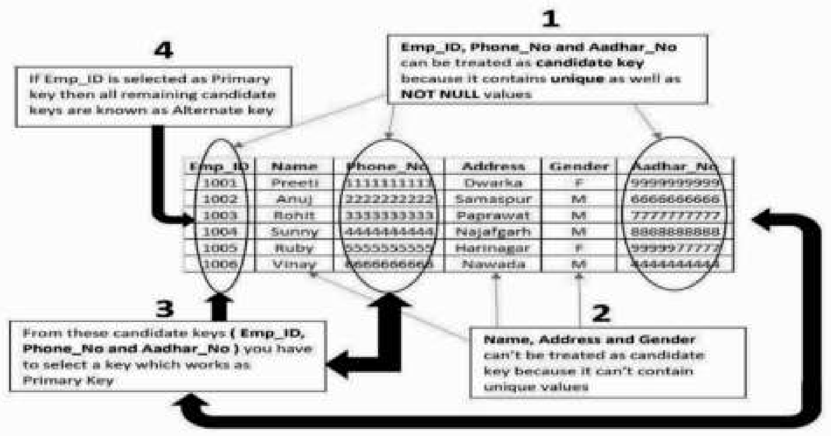
Now let's learn datatype in detail as per your syllabus

NUMERIC DATATYPE	INT/INTEGER	The signed range is from -2147483648 to 2147483647. The unsigned range is from 0 to 4294967295
	SMALLINT(SIZE)	The signed range is from -32768 to 32767. the unsigned range is from 0 to 65535
	TINYINT(SIZE)	The signed range is from -128 to 127. the unsigned range is from 0 to 255
	MEDIUMINT(SIZE)	The signed range is from -8388608 to 8388607. the unsigned range is from 0 to 16777215

	BIGINT(SIZE)	The signed range is from -9223372036854775808 to 9223372036854775807. the unsigned range is from 0 to 18446744073709551615
	FLOAT(SZ,D)	SZ is size and d is no. of digits after decimal
STRING DATATYPE	CHAR(SIZE)	char is a fixed-sized string as per the size defined in parenthesis. size of char datatype in the range of 0 to 255. by default size of the char is 1
	VARCHAR(SIZE)	varchar stands for a variable length character string. range of varchar is 0 to 65535
DATE AND TIME DATATYPE	DATE	As the name suggests it is used to store the date in any attribute supported format : YYYY-MM-DD
	TIME	Used to store time in any attribute supported format : HH:MM:SS

Keys:

In the database, the key applies some kind of constraint/restriction on the table (depending on the type of key applied) or we can say that keys are used to uniquely identify a tuple in a table/relation. To extract any particular row/record from a table, we need a key attribute which contains unique values.



1 **Emp_ID, Phone_No and Aadhar_No** can be treated as **candidate key** because it contains **unique** as well as **NOT NULL** values

2 **Name, Address and Gender** can't be treated as candidate key because it can't contain unique values

3 From these candidate keys **(Emp_ID, Phone_No and Aadhar_No)** you have to select a key which works as Primary Key

4 If Emp_ID is selected as Primary key then all remaining candidate keys are known as Alternate key

Emp_ID	Name	Phone_No	Address	Gender	Aadhar_No
1001	Preeti	1111111111	Dwarka	F	9999999999
1002	Anuj	2222222222	Samaspur	M	6666666666
1003	Rohit	3333333333	Paprawat	M	7777777777
1004	Sunny	4444444444	Najafgarh	M	8888888888
1005	Ruby	5555555555	Harinagar	F	9999977777
1006	Vinay	6666666666	Nawada	M	4444444444

DIAGRAM: 5

KEYS IN TABLE (PLEASE READ IN ORDER OF NUMBER)

Primary Key:

Primary Key is a unique identifier which identifies a unique record in a particular table. It must contain unique values for each record and the Primary key attribute/column/field can’t be NULL. A table can have only ONE primary key.

Note: A Primary key must be a candidate key but not all candidate keys are Primary key.

Candidate Key:

Candidate key is a set of one or more columns that could be used as primary key and from the set of these candidate keys, one column is selected as primary key.

From Diagram 5, Candidate keys can have more than one attribute like Emp_ID, Name, and Address etc.

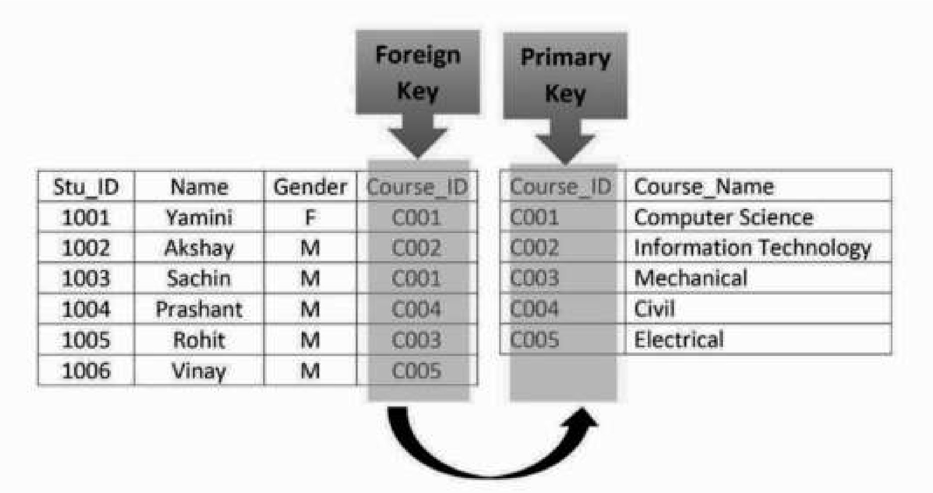
Alternate Key:

After selecting the primary key from the candidate key, the remaining keys (which are also eligible for the primary key) are called Alternate Key.

Foreign Key:

A Foreign key is a column or group of columns in a table that provides a link between two tables.

Let’s see how foreign keys work on the table.



Stu_ID	Name	Gender	Course_ID
1001	Yamini	F	C001
1002	Akshay	M	C002
1003	Sachin	M	C001
1004	Prashant	M	C004
1005	Rohit	M	C003
1006	Vinay	M	C005

Course_ID	Course_Name
C001	Computer Science
C002	Information Technology
C003	Mechanical
C004	Civil
C005	Electrical

DIAGRAM: 6

HOW FOREIGN KEY WORKS BETWEEN TWO TABLES

It is an attribute whose values can be derived from the primary key of some other table.

It ensures **Referential Integrity**. Referential Integrity is a protocol that ensures that the relationship between record/row is valid and can’t change in related data.

DBMS2 - QUESTION-RELATED TO RELATIONAL DATA MODEL

1 Who proposed the Relational Model?

2 What is a conceptual data model?

3 What do you understand from ER Model and what is the use of ER Model?

4 What do you mean by tuple and attribute in relation?

5 What is cardinality in relation?

6 What is a degree in relation?

7 What is Domain in relation?

8 What is a datatype in relation? Write some commonly used datatype in SQL.

9 What is the difference between char and varchar in SQL?

10 What is the role of keys in relation?

11 Write about the following keys :

1. Primary Key
2. Candidate Key
3. Alternate Key
4. Foreign Key

12 What do you understand by SQL? And writes its operation?

13 Which term is used in SQL to filter and analyse data quickly?

14 Write SQL Command :

1. Write SQL Command to create the database with the name class12.
2. Write SQL Command to show the database of the table named class12.
3. Write SQL Command to select a database named class12.
4. Write SQL Command to show a list of available tables in SQL.
5. Write SQL Command to create a table with the name student12 with the following attribute as follows:

student_id -	integer
student_name-	char-size(30)
age-	integer
phone-	integer
address	varchar(50)

6. Write SQL Command to view the structure of table name student12
7. Write SQL syntax to insert into to table named student12.
8. Write SQL Command to view data/content of table named student12.
9. Write SQL Command to delete data as well as structure of table named student12.

15 What is Constraints? And Writes its types.

16 What is DDL Command? Give its example.

17 What is DML Command? Give its example.

18 What is the process of assigning a nick name to a table or column in SQL?

19 What is the difference between drop and delete in SQL?

20 Which clause is used to remove duplicate values from the table?

21 Which clause is used to filter the results of the SELECT statement by specifying one or more condition

22 Which string type has a fixed-length character string?

23 Which key is used to ensure that it must contain unique values for each record and can’t have a NULL value?

24 Which key is eligible for the primary key?

25 Which clause is used to sort the result set in ascending order

26 Which operator is used to search for a specific pattern in a column?

27 Key constraints ensure that each column of the table contains a unique value for each row.

28 Which key constraints ensure that a column in a table does not contain any NULL values?

29 Which DDL command is used to modify a table.

30 Which clause is used to filter the result set of groups?

ed Query Language:

Introduction:

SQL is shortened for Structured Query Language. And it is pronounced as ‘Sequel’. SQL is used to manage databases. SQL was developed in 1970 in IBM Laboratory and it became a standard of the ANSI (American National Standard Institute) in 1986.

SQL is a query language, not a database system. You are required to install DBMS software in your system to perform SQL Language operations with help to query Example – Oracle, MySQL, MongoDB, PostgreSQL, SQL Server, DB2 etc.

SQL is mainly used for maintaining the data in relational database management systems. SQL provides interaction between the user and the database via a set of standard commands.

In simple words, SQL is a language that helps users to communicate with databases. SQL is not a case-sensitive language means you can type your query in small or capital letters.

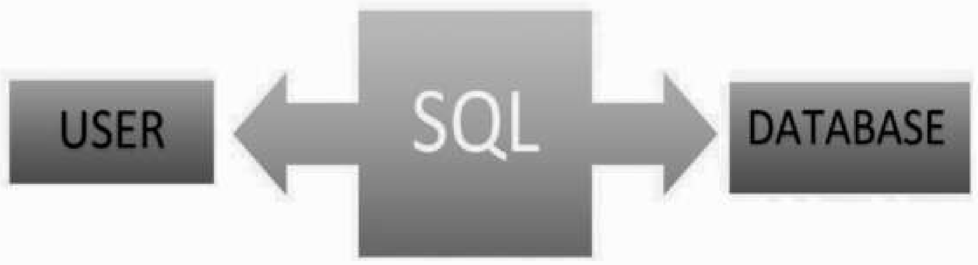


DIAGRAM: 7

HOW SQL PROVIDE INTERACTION BETWEEN USER AND DATABASE

You may write comments in SQL using “--” (Double hyphen)

Users may get information from a database file from the required query. A query is a request in the form of an SQL command to retrieve information with some condition. You will see lots of queries performing different types of operations. SQL is a query language (Query-based language) which works on structured data (data in structured form).

Now let’s discuss all the SQL Commands in a categorized way.

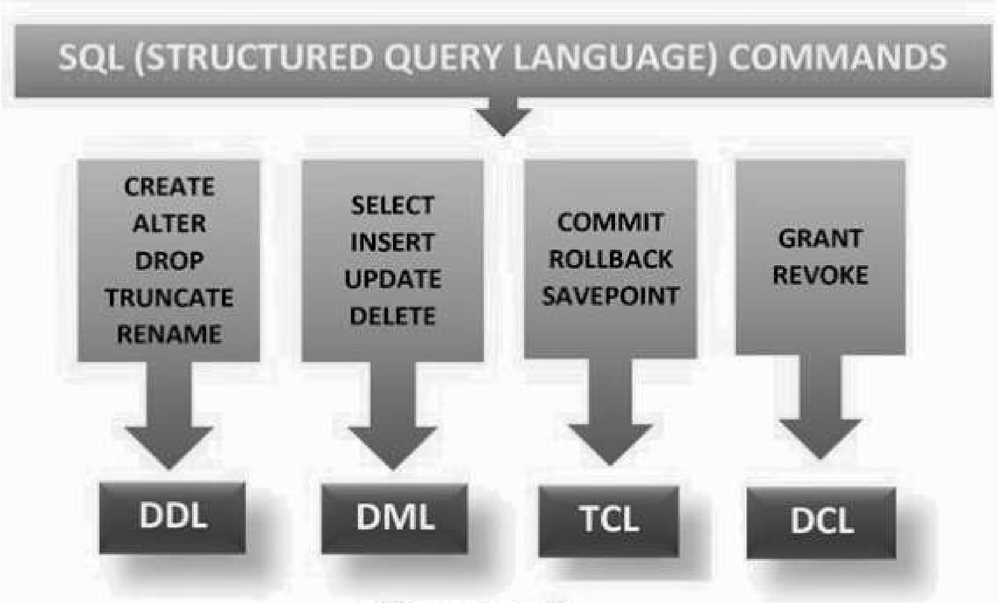


DIAGRAM: 8

SQL COMMANDS CAN BE CLASSIFIED INTO FOLLOWING CATEGORY

SQL perform the following operation:

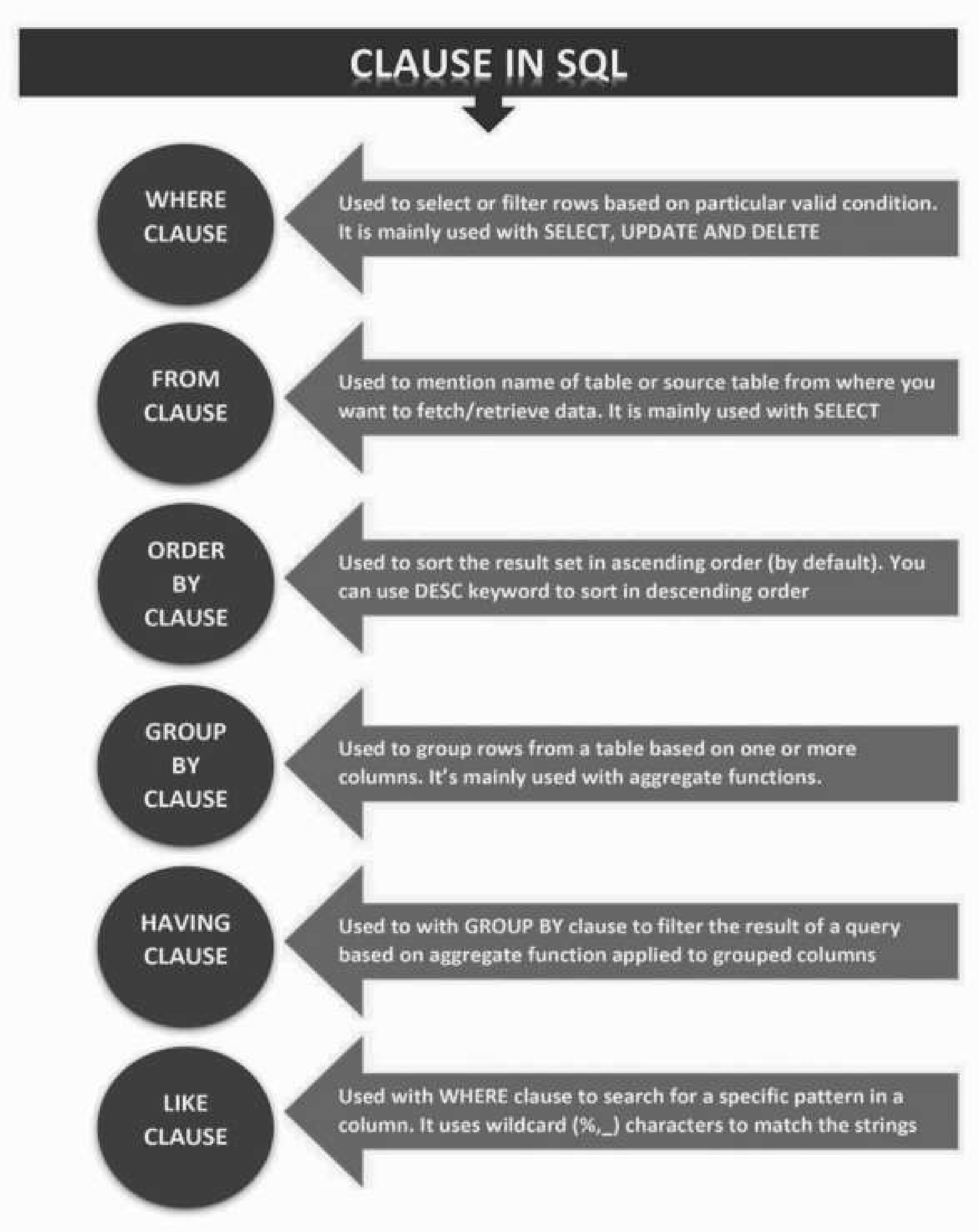
- Create a database
- Create a table
- Create view
- Insert data
- Update data
- Delete data
- Execute Query
- Set Permission or Constraints in the table

SQL Commands:

SQL commands are a predefined set of commands which are already defined in SQL. Commands are combinations of keywords and statements you want to execute. Keywords are reserved words that have special meaning for SQL, you don't need to define them as they are already defined in SQL. All you need to use these keywords with your particular statements.

CLAUSE IN SQL:

Clause: Clause are built-in functions which are used to deal with data inside the table that help SQL to filter and analyze data quickly.



CLAUSE IN SQL

WHERE CLAUSE — Used to select or filter rows based on particular valid condition. It is mainly used with SELECT, UPDATE AND DELETE

FROM CLAUSE — Used to mention name of table or source table from where you want to fetch/retrieve data. It is mainly used with SELECT

ORDER BY CLAUSE — Used to sort the result set in ascending order (by default). You can use DESC keyword to sort in descending order

GROUP BY CLAUSE — Used to group rows from a table based on one or more columns. It's mainly used with aggregate functions.

HAVING CLAUSE — Used to with GROUP BY clause to filter the result of a query based on aggregate function applied to grouped columns

LIKE CLAUSE — Used with WHERE clause to search for a specific pattern in a column. It uses wildcard (%,_) characters to match the strings

Any SQL statement is composed of two or more clauses.

These clauses are used with your SQL statements to filter commands you may learn in detail in further sections. Some mainly used clauses are discussed below.

NOTE: All SQL statements are terminated with (;) semicolon
SQL statements are not case sensitive, which means SQL treats both upper and lowercase commands as the same.

Constraints:

Constraints in SQL are sets of rules that are applied to the data in a relation/table. Constraints are used to ensure the accuracy and reliability of the data. Constraints can be at column level or table level. Column-level constraints apply to a column, and table-level constraints apply to the whole table. After applying constraints to the table, if any violation happens, then the data can't be inserted or the action can't be completed.

Types of constraints:

1. **Unique:** This constraint ensures that all values present or inserted in a column are different.
2. **Not Null:** This constraint ensures that no null values are present or inserted in a column.
3. **Primary key:** The primary key applies both unique and not null constraints to the column. It uniquely identifies a unique tuple/row in a relation/table.
4. **Foreign Key:** unique, not null and primary key constraint applies to a single table whereas the foreign key constraint applies to two tables.

For example: we have two tables' - student and awards, as follows:

Table: Student

ID	Name	Age	City
1	Amit	15	Delhi
2	Madhu	14	Gurugram
3	Manoj	15	Noida
4	Asif	15	Faridabad

Table: Awards

ID	Award	Sport
1	Gold	Badminton
2	Silver	Tennis
1	Silver	Hockey
4	Bronze	Badminton

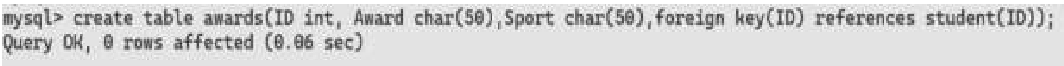
Here we will establish foreign key constraints on the column **ID** of the **Student table** and the column **ID** of the **Awards table.** Here we will consider **the Student table** as **the parent table** and the **Awards table** as the **child table.** The following rules must be followed:

(a) Column **ID** of the Student table must be its primary key.
(b) Column **ID** of the Awards table may or may not be the primary key of the Awards table.

Let's assume that our parent table **Student** has already been created. Now we will create child table **awards** and apply **foreign key** constraints on it. (Note: Foreign key relation can only be applied on child table)

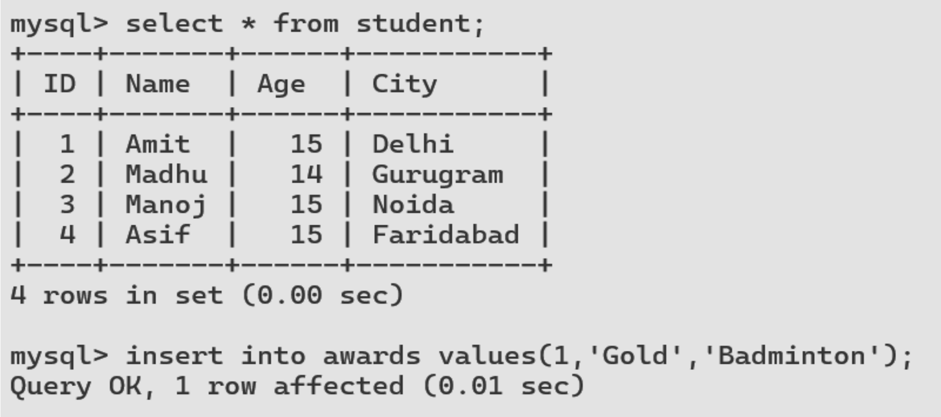
SQL Query

create table <child table name> (<column name 1> <data type>, <column name 2> <data type>,..., foreign key(<column name>) references <parent table name>(<column name>));



```
mysql> create table awards(ID int, Award char(50),Sport char(50),foreign key(ID) references student(ID));
Query OK, 0 rows affected (0.06 sec)
```

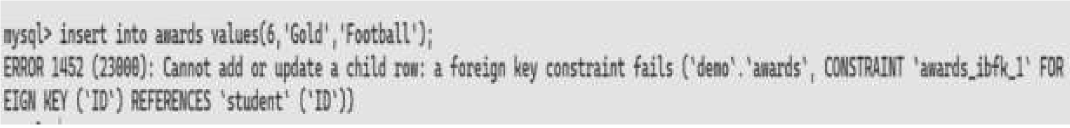
- ➢ **Foreign key constraint ensures that only that data can be inserted in column ID of the Awards table which is present in column ID of the Student table.**



```
mysql> select * from student;
+----+-------+------+-----------+
| ID | Name  | Age  | City      |
+----+-------+------+-----------+
|  1 | Amit  |   15 | Delhi     |
|  2 | Madhu |   14 | Gurugram  |
|  3 | Manoj |   15 | Noida     |
|  4 | Asif  |   15 | Faridabad |
+----+-------+------+-----------+
4 rows in set (0.00 sec)

mysql> insert into awards values(1,'Gold','Badminton');
Query OK, 1 row affected (0.01 sec)
```

Here, As ID ‘1’ is already present in the parent table ‘student’, So, ID ‘1’ can be inserted in the child table ‘awards’.

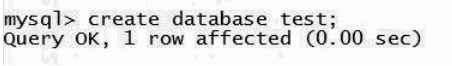
Now, we will try to insert ID ‘6’ in the child table ‘awards’ which is not present in the parent table ‘student’.

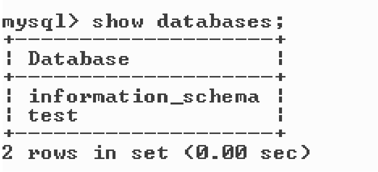


```
mysql> insert into awards values(6,'Gold','Football');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`demo`.`awards`, CONSTRAINT `awards_ibfk_1` FOR
EIGN KEY (`ID`) REFERENCES `student` (`ID`))
```

Here we can see that when we tried to insert value 6 in column ID of child table awards, it showed an error. This is how a foreign key works.

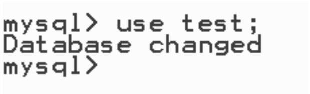
A few basic SQL Queries:

Create database: To create a new database, the create database command is used.
Query: create database <database name>;
	
	`mysql> create database test;`
	`Query OK, 1 row affected (0.00 sec)`

Show databases: To view the list of available/created databases in SQL, the ***show databases*** command is used.
Query: show databases;
	
	`mysql> show databases;`
	`+--------------------+`
	`¦ Database           ¦`
	`+--------------------+`
	`¦ information_schema ¦`
	`¦ test               ¦`
	`+--------------------+`
	`2 rows in set (0.00 sec)`

Use database: To select a database among available databases, ***use*** command is used.

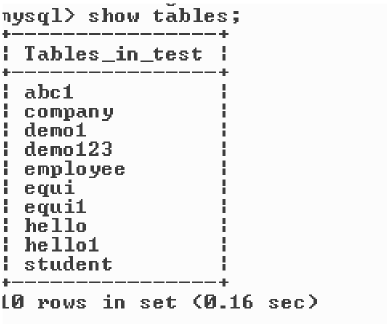
Query: use <database name>;



```
mysql> use test;
Database changed
mysql>
```

Show tables: To view the list of available/created databases in SQL, the ***show tables*** command is used.

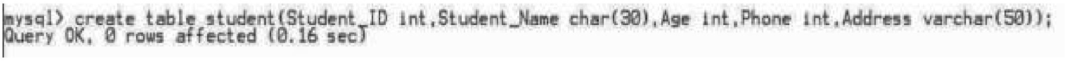
Query: show tables;



```
nysql> show tables;
+-----------------+
| Tables_in_test  |
+-----------------+
| abc1            |
| company         |
| demo1           |
| demo123         |
| employee        |
| equi            |
| equi1           |
| hello           |
| hello1          |
| student         |
+-----------------+
10 rows in set (0.16 sec)
```

Create table: To create a new table in the selected database. **For example**, if I want to create a table **Student** with the following attributes and data types:

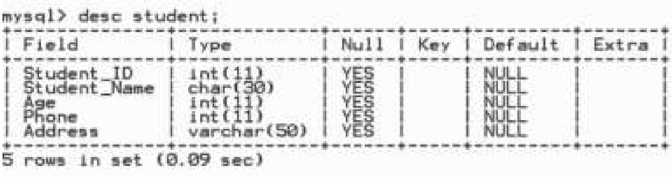
Name of attribute	**Data Type**	**SQL Query**
Student_ID	Int	create table <table name>(<attribute name> <data type> (size), <attribute name> <data type> (size) …);
Student_Name	char(30)	
Age	Int	
Phone	Int	
Address	varchar(50)	



```
mysql> create table student(Student_ID int,Student_Name char(30),Age int,Phone int,Address varchar(50));
Query OK, 0 rows affected (0.16 sec)
```

Describe table: To view the structure of the table (like attributes and its data types, keys, constraints, and default values), the **desc** command is used.

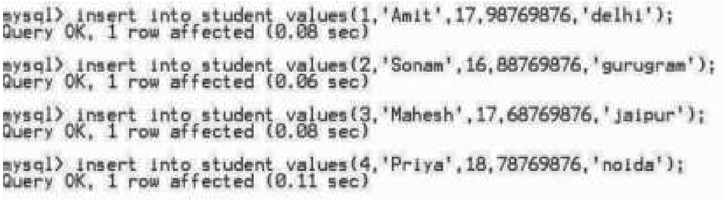
Query: desc <table name>;



```
mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Student_ID   | int(11)     | YES  |     | NULL    |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.09 sec)
```

Insert command: To insert data in a table, an insert command is used (one row at a time). Here in this example, data of 4 students are inserted in the table student.

Query: insert into <table name> values (<value1>, <value2> , <value3> …);

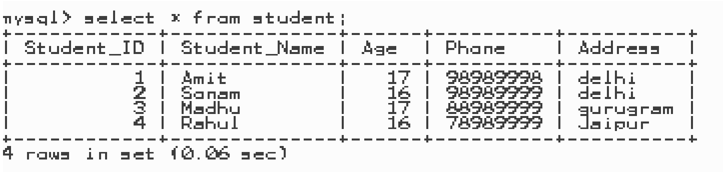


```
mysql> insert into student values(1,'Amit',17,98769876,'delhi');
Query OK, 1 row affected (0.08 sec)

mysql> insert into student values(2,'Sonam',16,88769876,'gurugram');
Query OK, 1 row affected (0.06 sec)

mysql> insert into student values(3,'Mahesh',17,68769876,'jaipur');
Query OK, 1 row affected (0.08 sec)

mysql> insert into student values(4,'Priya',18,78769876,'noida');
Query OK, 1 row affected (0.11 sec)
```

Select command: To show the data of a table, select command is used. Let's show the data of 4 students in the student table that was inserted in the previous command.

Query: select * from <table name>;

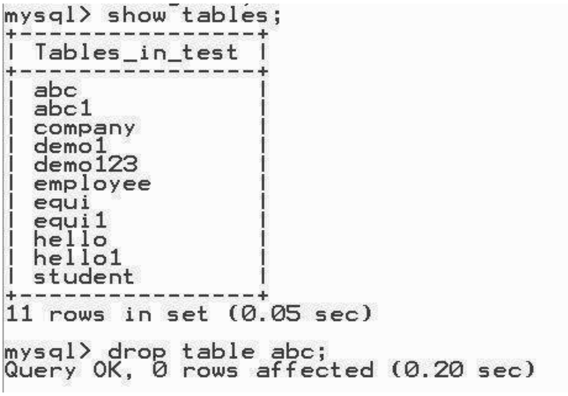
Here * means all columns



```
mysql> select * from student;
+------------+--------------+------+----------+----------+
| Student_ID | Student_Name | Age  | Phone    | Address  |
+------------+--------------+------+----------+----------+
|          1 | Amit         |   17 | 98989998 | delhi    |
|          2 | Sonam        |   16 | 98989999 | delhi    |
|          3 | Madhu        |   17 | 88989999 | gurugram |
|          4 | Rahul        |   16 | 78989999 | Jaipur   |
+------------+--------------+------+----------+----------+
4 rows in set (0.06 sec)
```

Drop table command: To delete data as well as the structure of a table, drop command is used.

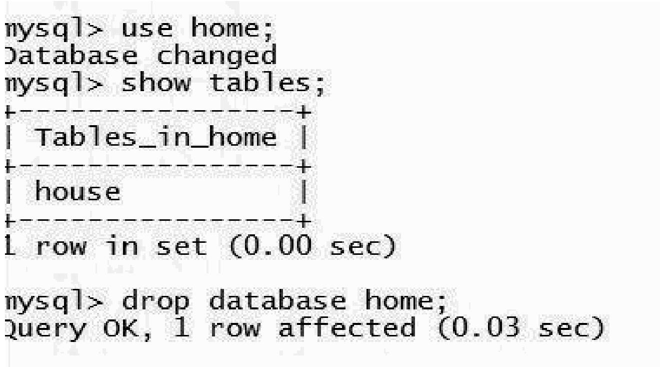
Query: drop table <table name>



```
mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| abc            |
| abc1           |
| company        |
| demo1          |
| demo123        |
| employee       |
| equi           |
| equi1          |
| hello          |
| hello1         |
| student        |
+----------------+
11 rows in set (0.05 sec)

mysql> drop table abc;
Query OK, 0 rows affected (0.20 sec)
```

Drop database command: To delete a database along with all the tables present in the database, drop command is used.

Query: drop database <database name>



```
mysql> use home;
Database changed
mysql> show tables;
+-----------------+
| Tables_in_home |
+-----------------+
| house          |
+-----------------+
1 row in set (0.00 sec)

mysql> drop database home;
Query OK, 1 row affected (0.03 sec)
```

DDL (Data Definition Language) Commands:

These commands are used to make any changes in the structure of the table/database. These commands don't change the data of the table.

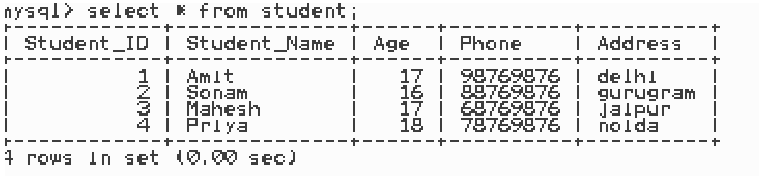
Example: create table, alter table, drop table, create database, create view etc.

We have already covered a few DDL Commands like create database, create table, drop database, and drop table. A few more DDL commands like alter table will be discussed now.

Alter Table: This is a DDL command and it is used to modify a table. This command can be used to add, delete, or modify columns, add or drop constraints etc.

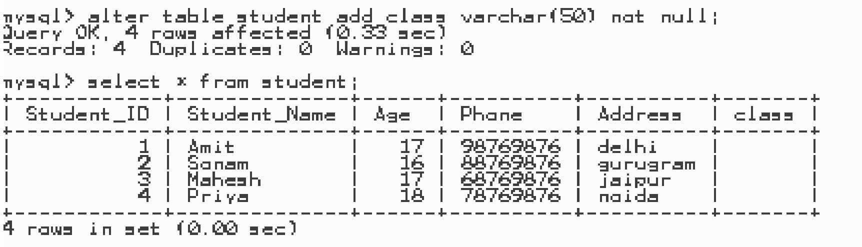
Query: alter table <table name> [alter option]

(a) Add a column to the table: We have a table student which was created in the previous section.



```
mysql> select * from student;
+------------+--------------+-----+----------+----------+
| Student_ID | Student_Name | Age | Phone    | Address  |
+------------+--------------+-----+----------+----------+
|          1 | Amit         |  17 | 98769876 | delhi    |
|          2 | Sonam        |  16 | 88769876 | gurugram |
|          3 | Mahesh       |  17 | 68769876 | jaipur   |
|          4 | Priya        |  18 | 78769876 | noida    |
+------------+--------------+-----+----------+----------+
4 rows in set (0.00 sec)
```

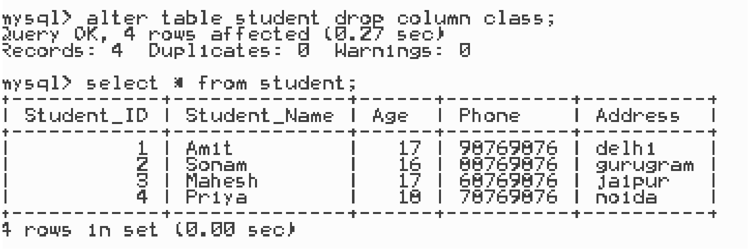
Query: alter table <table name> add <column name><data type> [constraint];
Example: Let's add a column 'class' with data type varchar and size 50 and nulls are not allowed.



```
mysql> alter table student add class varchar(50) not null;
Query OK, 4 rows affected (0.33 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from student;
+------------+--------------+-----+----------+----------+-------+
| Student_ID | Student_Name | Age | Phone    | Address  | class |
+------------+--------------+-----+----------+----------+-------+
|          1 | Amit         |  17 | 98769876 | delhi    |       |
|          2 | Sonam        |  16 | 88769876 | gurugram |       |
|          3 | Mahesh       |  17 | 68769876 | jaipur   |       |
|          4 | Priya        |  18 | 78769876 | noida    |       |
+------------+--------------+-----+----------+----------+-------+
4 rows in set (0.00 sec)
```

(b) Drop a column from the table: Let's delete the column 'class' from the table 'student' which we added in the previous section.

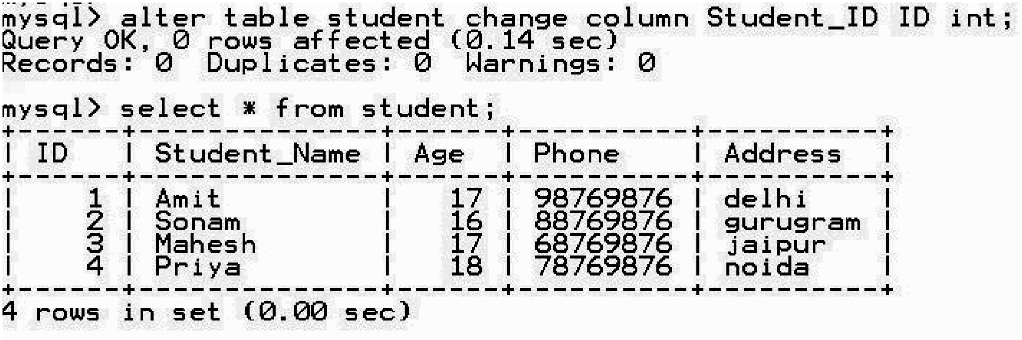
Command: alter table <table name> drop column<column name>;



```
mysql> alter table student drop column class;
Query OK, 4 rows affected (0.27 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from student;
+------------+--------------+------+----------+----------+
| Student_ID | Student_Name | Age  | Phone    | Address  |
+------------+--------------+------+----------+----------+
|          1 | Amit         |   17 | 98769876 | delhi    |
|          2 | Sonam        |   16 | 88769876 | gurugram |
|          3 | Mahesh       |   17 | 68769876 | jaipur   |
|          4 | Priya        |   18 | 78769876 | noida    |
+------------+--------------+------+----------+----------+
4 rows in set (0.00 sec)
```

(c) Modifying column of a table: We have different ways to modify a table like column name, data type, default value, size, order of columns, and constraints.

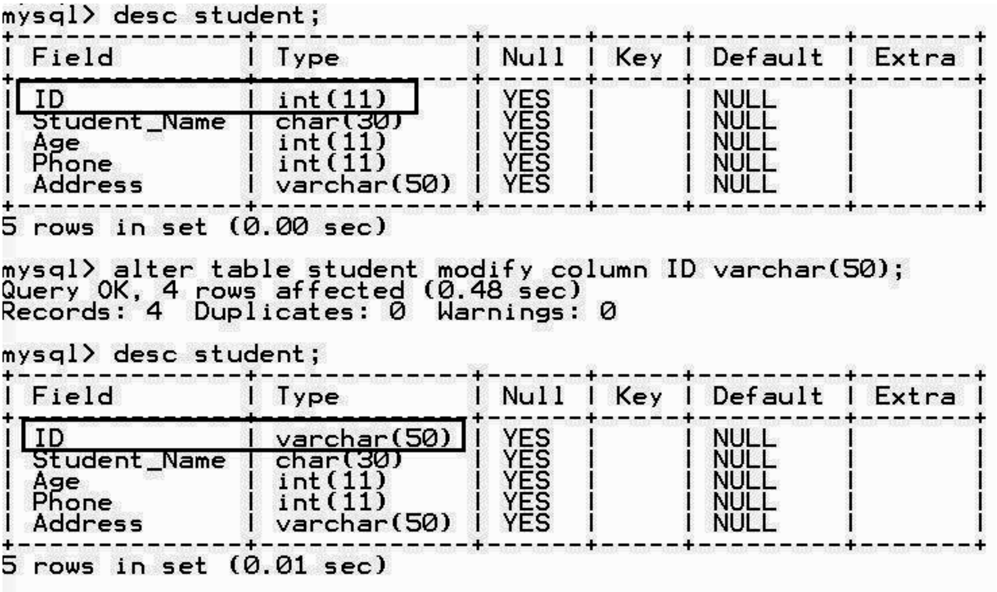
(i) Changing column name: We can change the column name of a table using the alter command.
Example: In table student, Let's change column name Student_ID to ID.
Query: alter table <table name> change column <old column name> <new column name> <data type>



```
mysql> alter table student change column Student_ID ID int;
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from student;
+------+--------------+------+----------+----------+
| ID   | Student_Name | Age  | Phone    | Address  |
+------+--------------+------+----------+----------+
|    1 | Amit         |   17 | 98769876 | delhi    |
|    2 | Sonam        |   16 | 88769876 | gurugram |
|    3 | Mahesh       |   17 | 68769876 | jaipur   |
|    4 | Priya        |   18 | 78769876 | noida    |
+------+--------------+------+----------+----------+
4 rows in set (0.00 sec)
```

(ii) Changing column data type: We can change the column data type from varchar to char or int to varchar etc. of a table using the alter command.
Example: in table student, Let's change the datatype of column 'ID' from int to varchar.

Query: alter table <table name> modify column <column name> <new data type> <size>

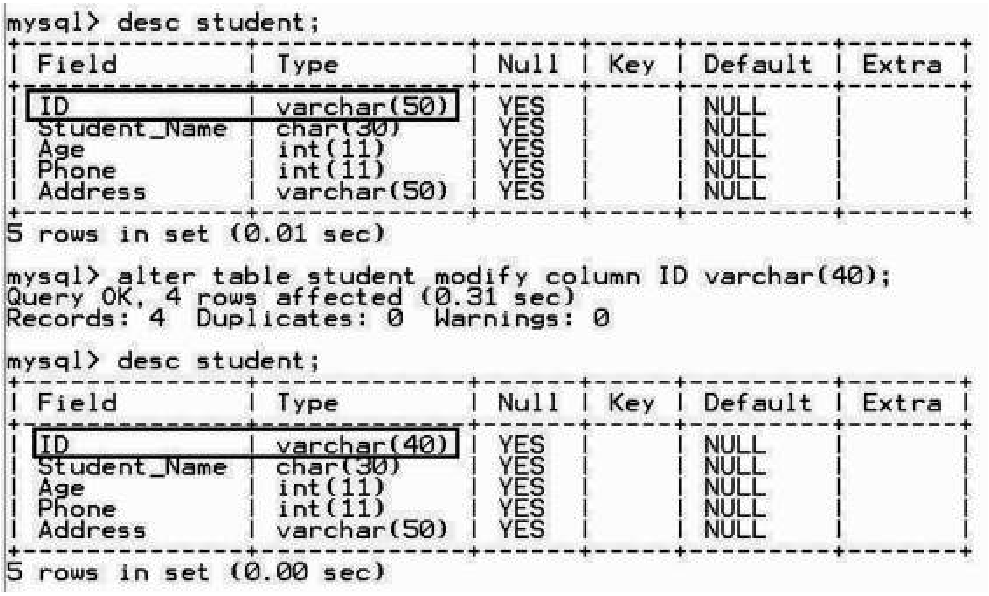


```
mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | int(11)     | YES  |     | NULL    |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> alter table student modify column ID varchar(50);
Query OK, 4 rows affected (0.48 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | varchar(50) | YES  |     | NULL    |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

(iii) Changing the maximum size of the data in a column: We can change the maximum size of the data in a column of a table using the alter command.

Example: In table student, Let's change the size of column ID from varchar(50) to varchar(40)

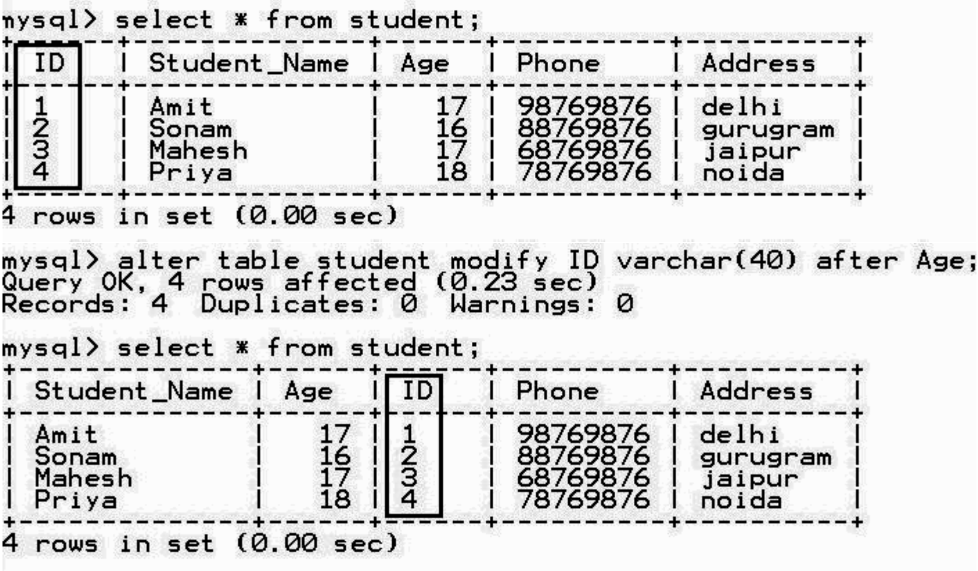
Query: alter table <table name> modify column <column name> <data type with size>



```
mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | varchar(50) | YES  |     | NULL    |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

mysql> alter table student modify column ID varchar(40);
Query OK, 4 rows affected (0.31 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | varchar(40) | YES  |     | NULL    |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

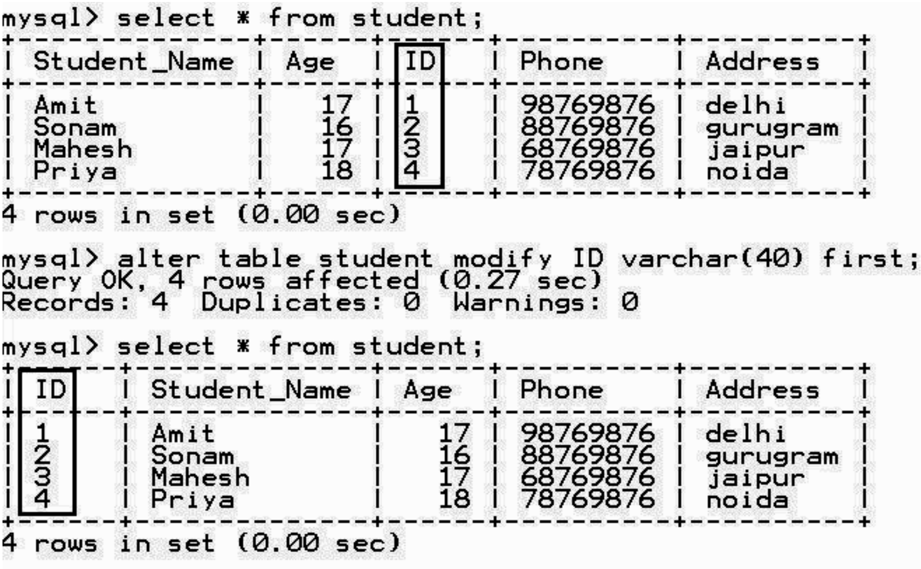
(iv) Changing the order of the column: We can change the order of the column of a table using the alter command. For example, in table student, we are going to place column ID after column Age.

Query: alter table <table name> modify <column name> <data type with size> [first|after <column name>]



```
nysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | delhi    |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
+----+--------------+------+----------+----------+
4 rows in set (0.00 sec)

mysql> alter table student modify ID varchar(40) after Age;
Query OK, 4 rows affected (0.23 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from student;
+--------------+------+----+----------+----------+
| Student_Name | Age  | ID | Phone    | Address  |
+--------------+------+----+----------+----------+
| Amit         |   17 | 1  | 98769876 | delhi    |
| Sonam        |   16 | 2  | 88769876 | gurugram |
| Mahesh       |   17 | 3  | 68769876 | jaipur   |
| Priya        |   18 | 4  | 78769876 | noida    |
+--------------+------+----+----------+----------+
4 rows in set (0.00 sec)
```

Now we are going to put column ID back to the first position.

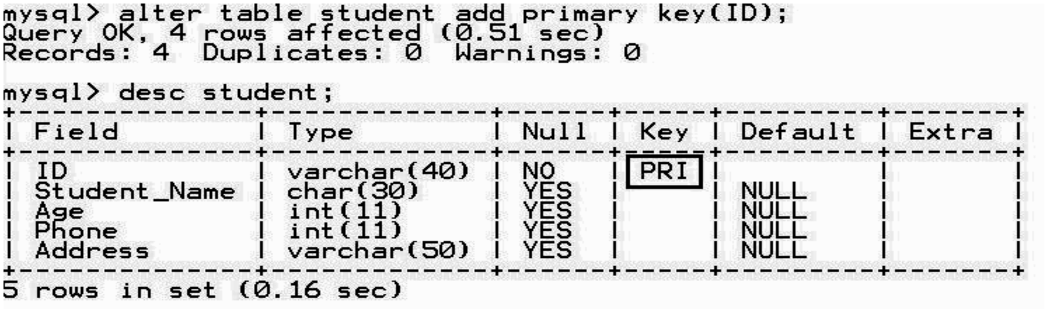


```
mysql> select * from student;
+--------------+------+----+----------+----------+
| Student_Name | Age  | ID | Phone    | Address  |
+--------------+------+----+----------+----------+
| Amit         |   17 | 1  | 98769876 | delhi    |
| Sonam        |   16 | 2  | 88769876 | gurugram |
| Mahesh       |   17 | 3  | 68769876 | jaipur   |
| Priya        |   18 | 4  | 78769876 | noida    |
+--------------+------+----+----------+----------+
4 rows in set (0.00 sec)

mysql> alter table student modify ID varchar(40) first;
Query OK, 4 rows affected (0.27 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | delhi    |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
+----+--------------+------+----------+----------+
4 rows in set (0.00 sec)
```

(v) Add/drop constraints/column: We can add/drop constraints in a table using the alter command.

- **Adding primary key**: Let's add the primary key at column 'ID' using the alter command.

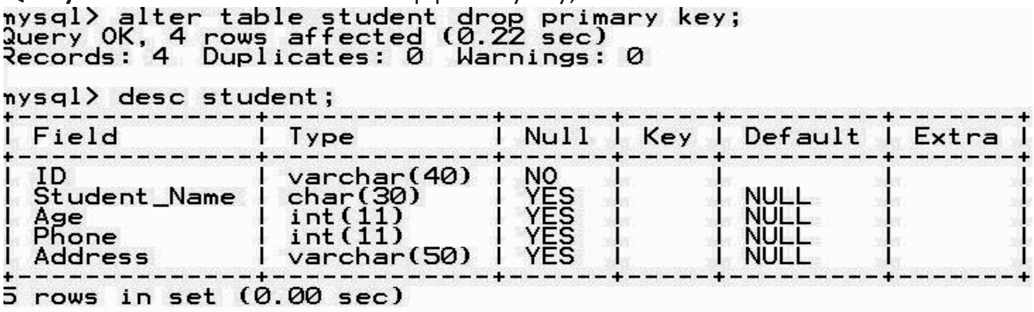
  Query: alter table <table name> add primary key(<column name>);



```
mysql> alter table student add primary key(ID);
Query OK, 4 rows affected (0.51 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | varchar(40) | NO   | PRI |         |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.16 sec)
```

- **Dropping primary key:** Let's remove the primary key at column ID which was added in the previous section.

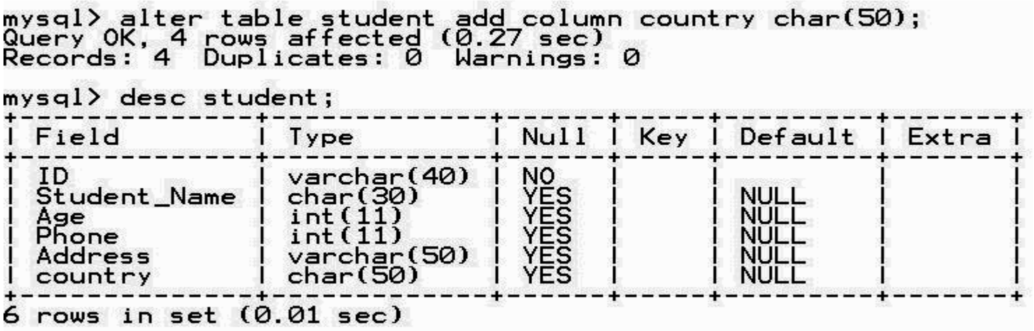
  Query: alter table <table name> drop primary key;



```
nysql> alter table student drop primary key;
Query OK, 4 rows affected (0.22 sec)
Records: 4  Duplicates: 0  Warnings: 0

nysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | varchar(40) | NO   |     |         |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

- **Adding a new column:** Let's add a column 'country' with data type char of size 50 to the table 'student' using the alter command.

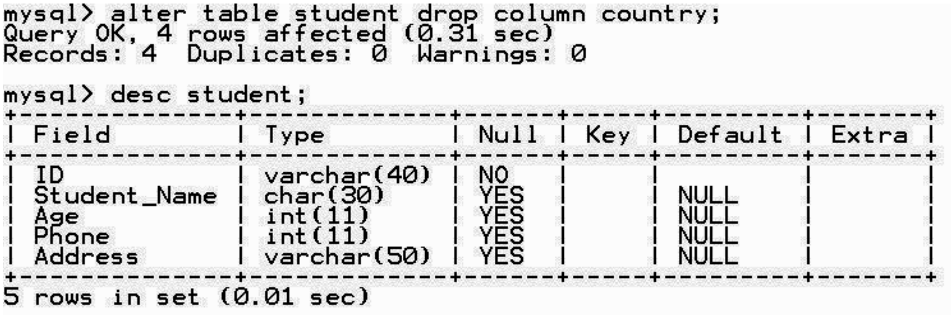
  Query: alter table <table name> add column <column name> <data type with size>;



```
mysql> alter table student add column country char(50);
Query OK, 4 rows affected (0.27 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | varchar(40) | NO   |     |         |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
| country      | char(50)    | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

- **Dropping a column:** Let's remove a column 'country' which was added in the last section using the alter command.

  Command: alter table <table name> drop column <column name>;



```
mysql> alter table student drop column country;
Query OK, 4 rows affected (0.31 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| ID           | varchar(40) | NO   |     |         |       |
| Student_Name | char(30)    | YES  |     | NULL    |       |
| Age          | int(11)     | YES  |     | NULL    |       |
| Phone        | int(11)     | YES  |     | NULL    |       |
| Address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

DML (Data Manipulation Language) Commands:

These commands are used to make any changes in the data of the table.

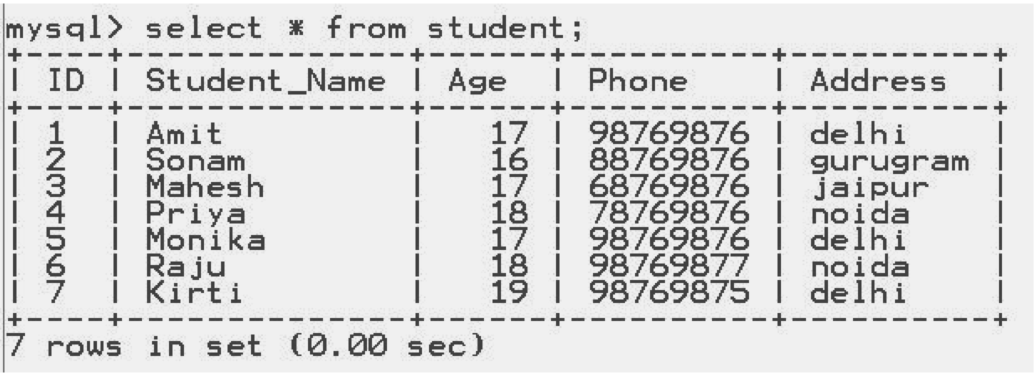
DML commands: insert, delete, update, select etc.

We have already covered a few DML Commands like insert and select. Now we will discuss delete and update commands.

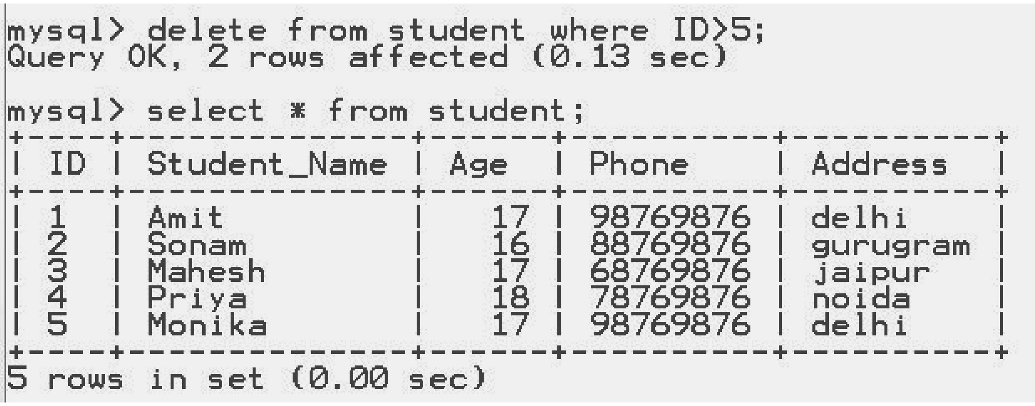
1. **Delete command:** The delete command is used to delete data from the table. **Where clause is used to give condition in a SQL query.** All those tuples which satisfy the condition will be deleted from the table.

   Command: delete from <table name> where <condition>;

   Now, let's delete the data of all those students from the student table whose ID is greater than 5.



```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | delhi    |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 6  | Raju         |   18 | 98769877 | noida    |
| 7  | Kirti        |   19 | 98769875 | delhi    |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)
```

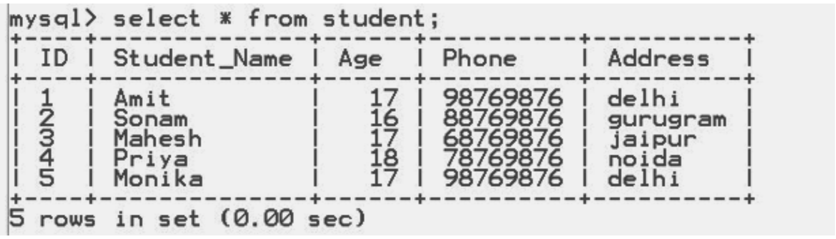


```
mysql> delete from student where ID>5;
Query OK, 2 rows affected (0.13 sec)

mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | delhi    |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
+----+--------------+------+----------+----------+
5 rows in set (0.00 sec)
```

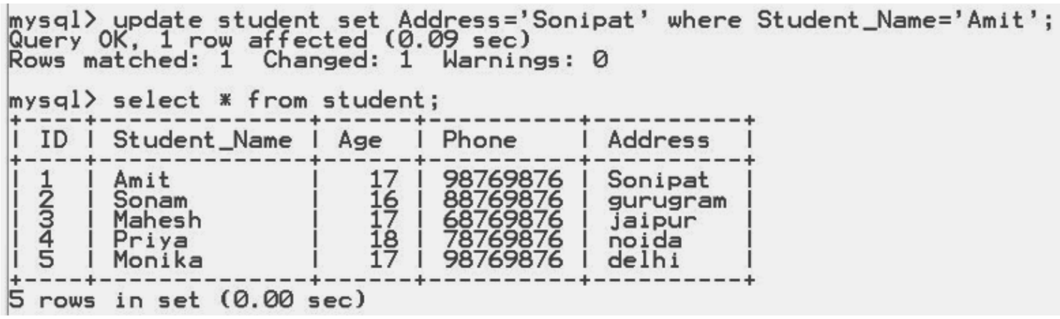
2. **Update command:** The update command is used to update data from the table. **Where clause is used to give condition in a SQL query.** All those tuples which satisfy the condition will be updated in the table.

   Command: update <table name> set <column name>=<new data> where <condition>;

   Now, let's update the Address from 'Delhi' to 'Sonipat' of that student whose name is 'Amit'.



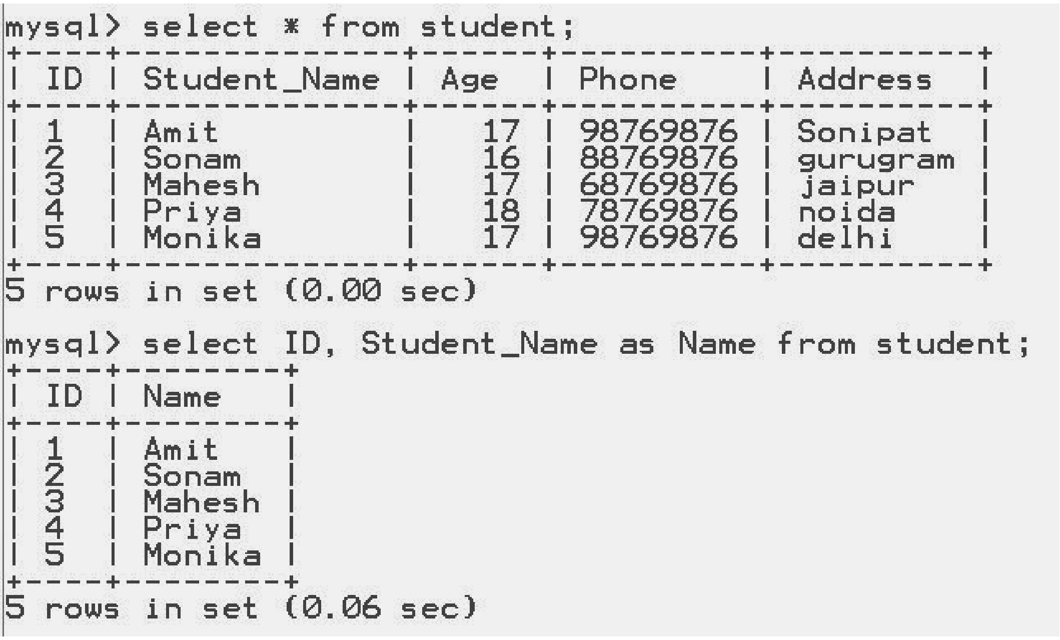
```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | delhi    |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
+----+--------------+------+----------+----------+
5 rows in set (0.00 sec)
```



```
mysql> update student set Address='Sonipat' where Student_Name='Amit';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
+----+--------------+------+----------+----------+
5 rows in set (0.00 sec)
```

Aliasing:

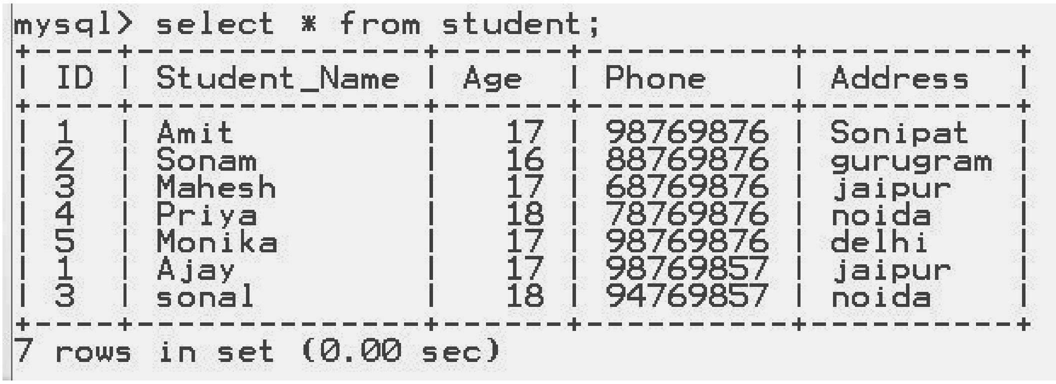
Aliasing in SQL is the process of assigning a nick name or a temporary name to a table or column. We create aliases to make queries more readable and easier to use. The alias can be created using the '**as'** keyword. Creating aliases doesn't change the name of any table or column permanently.



```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
+----+--------------+------+----------+----------+
5 rows in set (0.00 sec)

mysql> select ID, Student_Name as Name from student;
+----+--------+
| ID | Name   |
+----+--------+
| 1  | Amit   |
| 2  | Sonam  |
| 3  | Mahesh |
| 4  | Priya  |
| 5  | Monika |
+----+--------+
5 rows in set (0.06 sec)
```

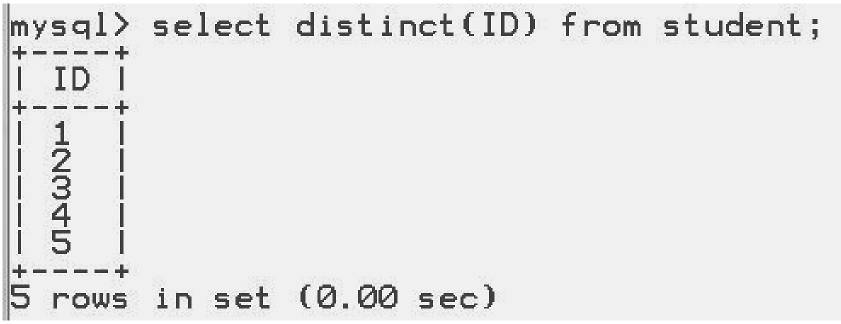
Distinct clause:

The distinct clause is used to display unique values by neglecting all the duplicate values. Distinct clause can be used for more than one column.



```
mysql> select * from student;
+----+---------------+------+----------+----------+
| ID | Student_Name  | Age  | Phone    | Address  |
+----+---------------+------+----------+----------+
| 1  | Amit          |   17 | 98769876 | Sonipat  |
| 2  | Sonam         |   16 | 88769876 | gurugram |
| 3  | Mahesh        |   17 | 68769876 | jaipur   |
| 4  | Priya         |   18 | 78769876 | noida    |
| 5  | Monika        |   17 | 98769876 | delhi    |
| 1  | Ajay          |   17 | 98769857 | jaipur   |
| 3  | sonal         |   18 | 94769857 | noida    |
+----+---------------+------+----------+----------+
7 rows in set (0.00 sec)
```

Here in the ‘student’ table, two duplicate IDs 1 and 3 are present. Now using distinct clauses we can get unique values.



```
mysql> select distinct(ID) from student;
+----+
| ID |
+----+
| 1  |
| 2  |
| 3  |
| 4  |
| 5  |
+----+
5 rows in set (0.00 sec)
```

As we can see all duplicate IDs are removed but it is temporary. Duplicate values are not removed and are still present in the table.

Where clause: The WHERE clause in SQL is used to filter the results of a SELECT statement by specifying one or more conditions. All those tuples which meet the condition will be included in the final result.

The **WHERE clause** is a very powerful technique to select particular rows from a table. It can be used to filter by the values in a column, by the values in multiple columns, or by the outcome of any calculation.

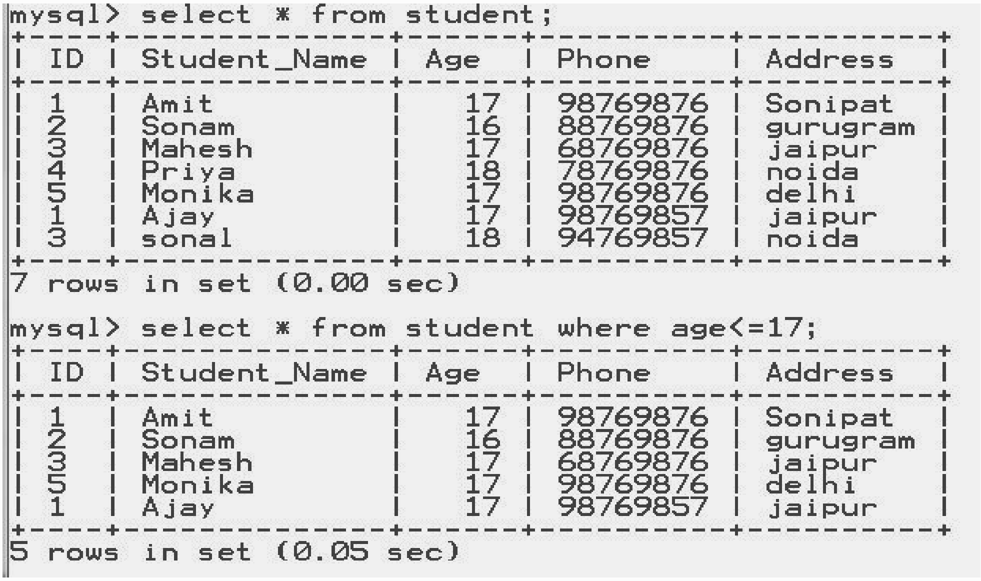
Uses of where clause:

- Where clause can be used with the select statement to filter the result.
- Where clause can be used with an update statement to update the data of the table that matches with the condition.
- Where clause can be used with a delete statement to delete the rows of the table that match with the condition.

Query: where <condition>;

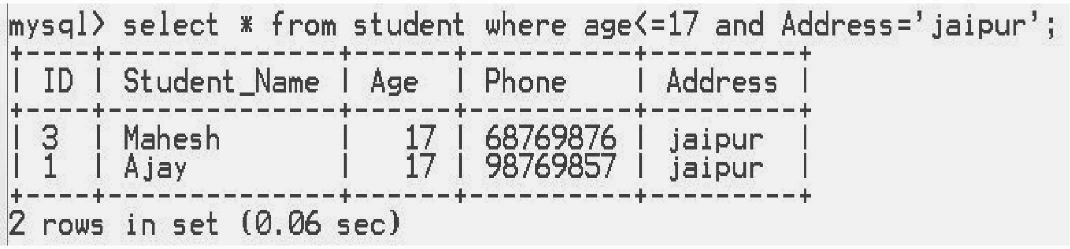
Examples:

1. To filter the result based on only one condition:



```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | sonal        |   18 | 94769857 | noida    |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)

mysql> select * from student where age<=17;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
+----+--------------+------+----------+----------+
5 rows in set (0.05 sec)
```

2. To filter the result based on multiple conditions:



```
mysql> select * from student where age<=17 and Address='jaipur';
+----+--------------+------+----------+---------+
| ID | Student_Name | Age  | Phone    | Address |
+----+--------------+------+----------+---------+
| 3  | Mahesh       |   17 | 68769876 | jaipur  |
| 1  | Ajay         |   17 | 98769857 | jaipur  |
+----+--------------+------+----------+---------+
2 rows in set (0.06 sec)
```

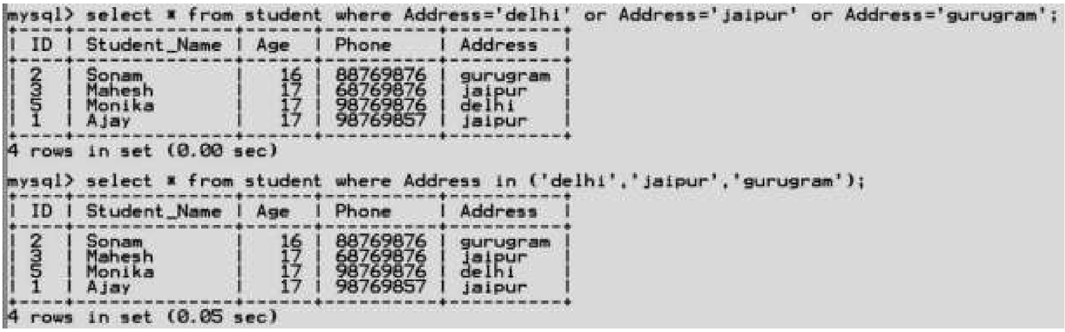
In clause and not in clause:

in clause and **not in clause** in SQL is used to filter the rows in output based on a list of values.

Query for in clause: where <column name> in (item1, item2,…);

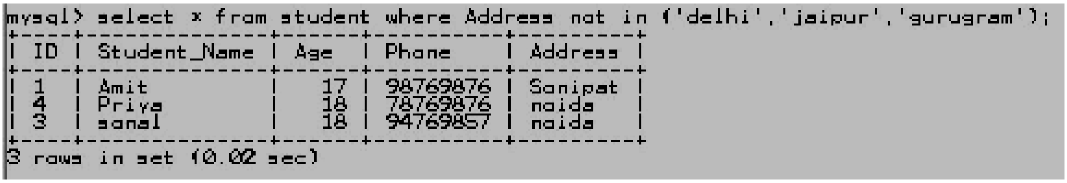
Query for not in clause: where <column name> not in (item1, item2,…);

Example of in clause: if we want to find the data of those students who live in either 'Delhi' or 'Jaipur' or 'Gurugram'. Now to solve this problem, we have two ways. Either we write multiple comparisons using **or keyword** or we can use **in clause**. Now you will see that using in clause for comparing with a list of items is an easy option.



```
mysql> select * from student where Address='delhi' or Address='jaipur' or Address='gurugram';
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
+----+--------------+------+----------+----------+
4 rows in set (0.00 sec)

mysql> select * from student where Address in ('delhi','jaipur','gurugram');
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
+----+--------------+------+----------+----------+
4 rows in set (0.05 sec)
```

Example of not in clause: if we want to find the data of those students who don't live in 'Delhi' or 'Jaipur' or 'Gurugram'.

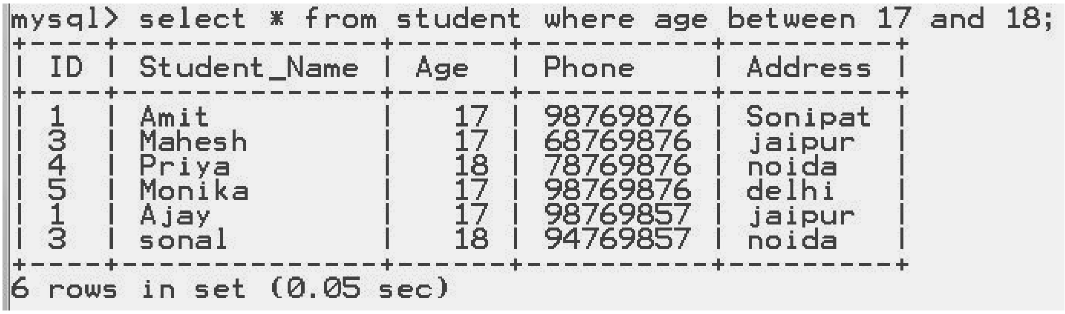


```
mysql> select * from student where Address not in ('delhi','jaipur','gurugram');
+----+--------------+------+----------+---------+
| ID | Student_Name | Age  | Phone    | Address |
+----+--------------+------+----------+---------+
| 1  | Amit         |   17 | 98769876 | Sonipat |
| 4  | Priya        |   18 | 78769876 | noida   |
| 3  | sonal        |   18 | 94769857 | noida   |
+----+--------------+------+----------+---------+
3 rows in set (0.02 sec)
```

Between Clause: It is used to filter the rows in output based on the range of values.

Query: where <column name> between <starting value> and <ending value>;

Note: The final result of the between clause filters the rows of the table based on the range of values including starting and ending values.

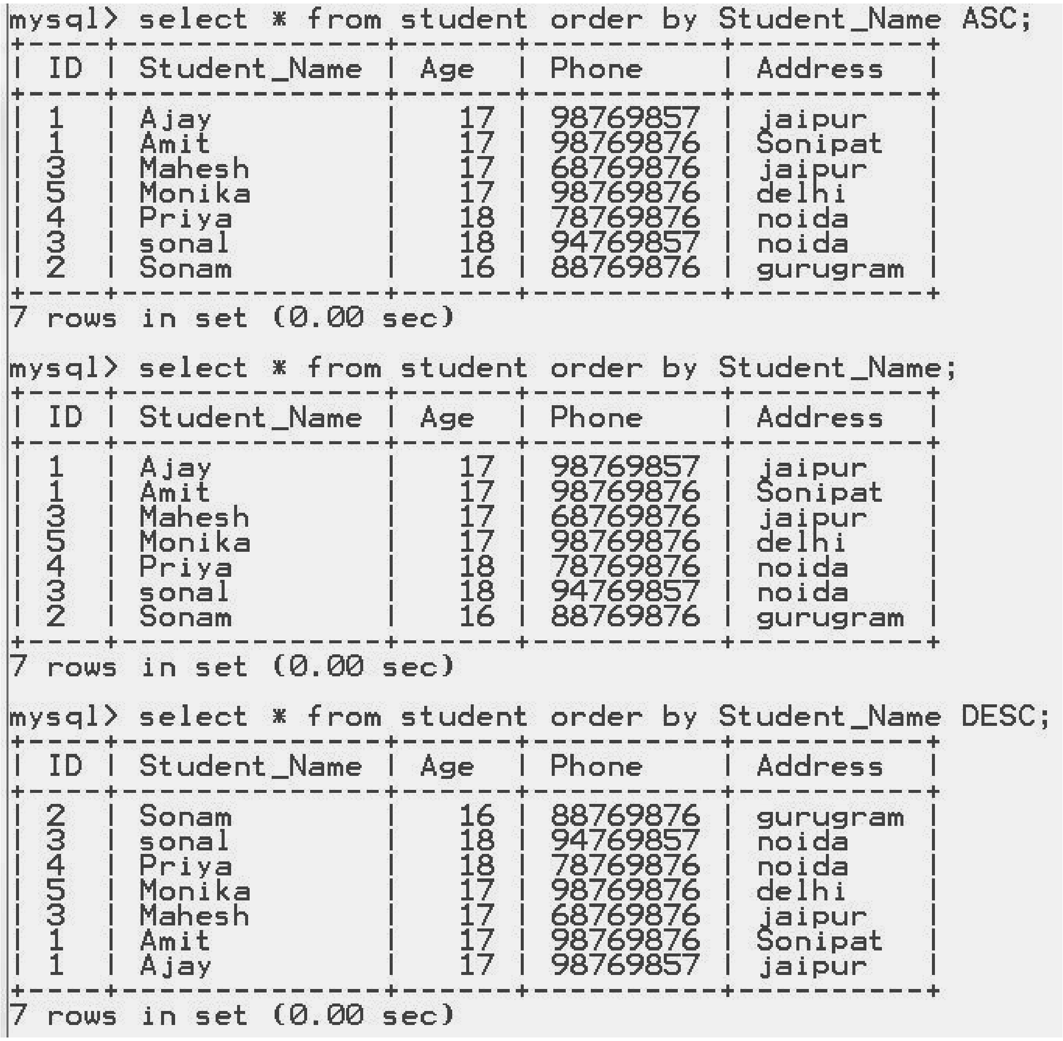


```
mysql> select * from student where age between 17 and 18;
+----+--------------+------+----------+---------+
| ID | Student_Name | Age  | Phone    | Address |
+----+--------------+------+----------+---------+
| 1  | Amit         |   17 | 98769876 | Sonipat |
| 3  | Mahesh       |   17 | 68769876 | jaipur  |
| 4  | Priya        |   18 | 78769876 | noida   |
| 5  | Monika       |   17 | 98769876 | delhi   |
| 1  | Ajay         |   17 | 98769857 | jaipur  |
| 3  | sonal        |   18 | 94769857 | noida   |
+----+--------------+------+----------+---------+
6 rows in set (0.05 sec)
```

Order by Clause: It is used to sort the output of the select statement in ascending or descending order.

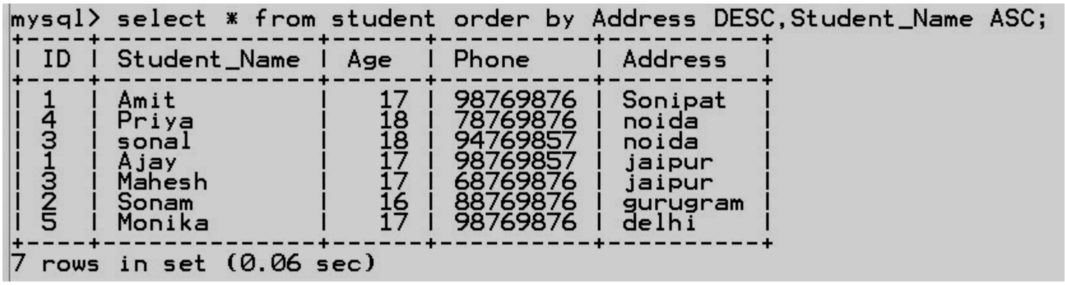
Query: order by <column name> [ASC|DESC];

Note: If not mentioned, by default it will sort the output in ascending order. So, if you want to sort the data in ascending order, you need not mention the order of sorting as it is the default mode of sorting.



```
mysql> select * from student order by Student_Name ASC;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 4  | Priya        |   18 | 78769876 | noida    |
| 3  | sonal        |   18 | 94769857 | noida    |
| 2  | Sonam        |   16 | 88769876 | gurugram |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)

mysql> select * from student order by Student_Name;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 4  | Priya        |   18 | 78769876 | noida    |
| 3  | sonal        |   18 | 94769857 | noida    |
| 2  | Sonam        |   16 | 88769876 | gurugram |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)

mysql> select * from student order by Student_Name DESC;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | sonal        |   18 | 94769857 | noida    |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)
```

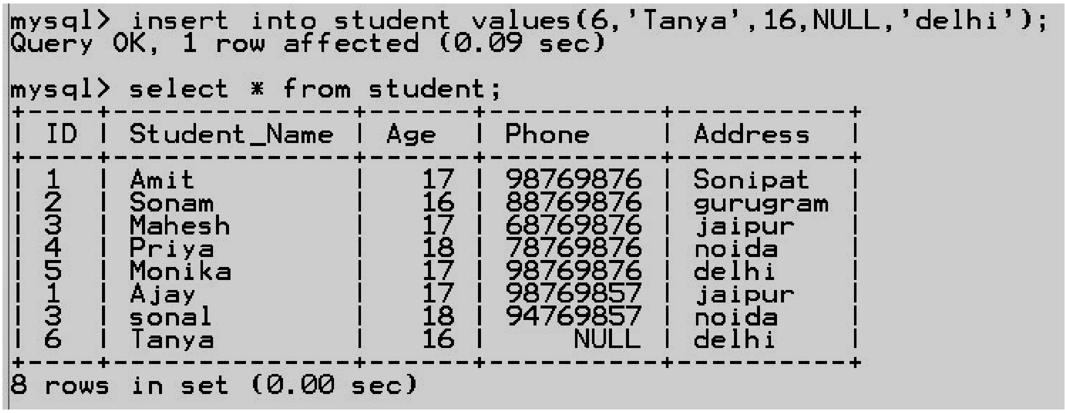
We can sort multiple columns together in ASC and DESC order.



```
mysql> select * from student order by Address DESC,Student_Name ASC;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 4  | Priya        |   18 | 78769876 | noida    |
| 3  | sonal        |   18 | 94769857 | noida    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 5  | Monika       |   17 | 98769876 | delhi    |
+----+--------------+------+----------+----------+
7 rows in set (0.06 sec)
```

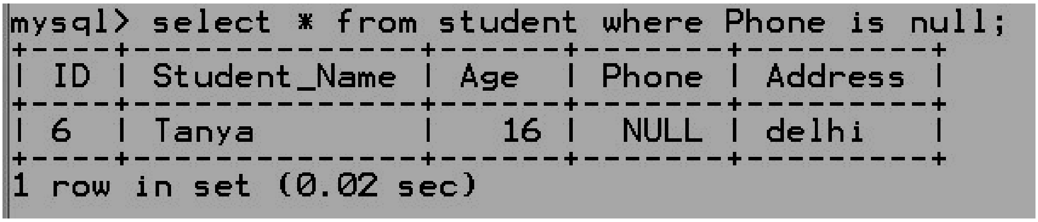
NULL: In SQL, null is a special value which means the absence of a value or a field doesn't have a value. Null doesn't mean zero. Null also doesn't mean empty string. Null is a kind of placeholder of that value which is not present or not known.

Example: If the phone number of a student is not known at present, we can store NULL instead of zero or make it empty.



```
mysql> insert into student values(6,'Tanya',16,NULL,'delhi');
Query OK, 1 row affected (0.09 sec)

mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
8 rows in set (0.00 sec)
```

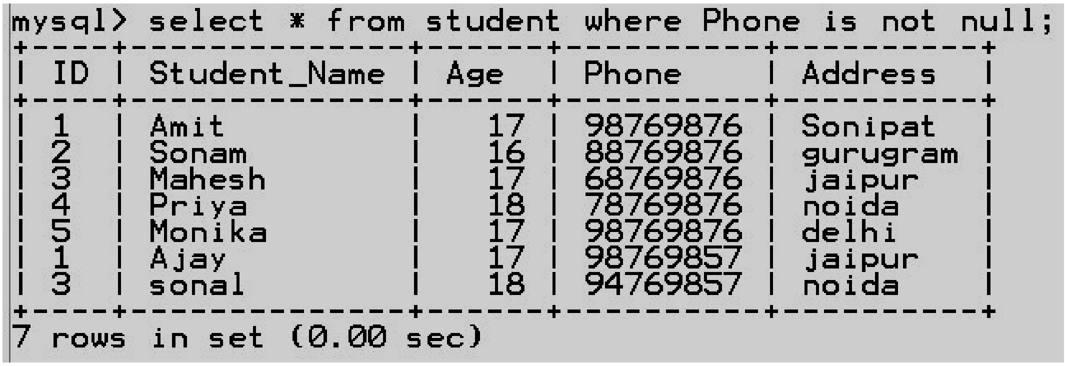
Example: To find out the names of students whose phone numbers are NULL.



```
mysql> select * from student where Phone is null;
+----+--------------+------+-------+---------+
| ID | Student_Name | Age  | Phone | Address |
+----+--------------+------+-------+---------+
| 6  | Tanya        |   16 |  NULL | delhi   |
+----+--------------+------+-------+---------+
1 row in set (0.02 sec)
```

Note: is keyword is used to compare values of a column with **NULL**.

Example: To find out the names of students whose phone numbers are not NULL.

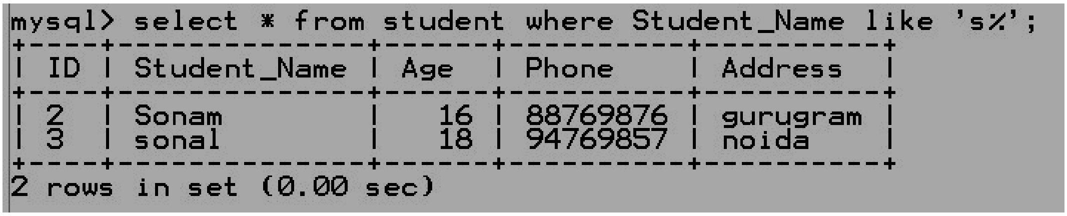


```
mysql> select * from student where Phone is not null;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | sonal        |   18 | 94769857 | noida    |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)
```

Like operator: Like operator is used to match a pattern. The like operator is used with the **where** clause. Like operator has 2 wildcards:

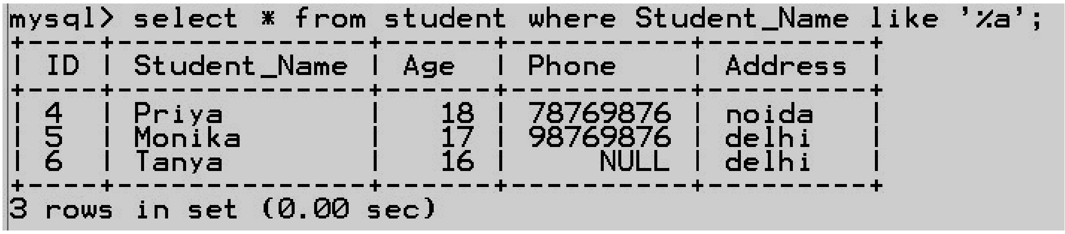
1. **_ (underscore):** It is used to match one character.
2. **% (percentage sign)**: It is used to match zero or more characters.

Example 1: To match a string that starts with 's', its pattern will be 's%'. As we don't know how many characters are there after 's', so '%' sign is used after 's'.



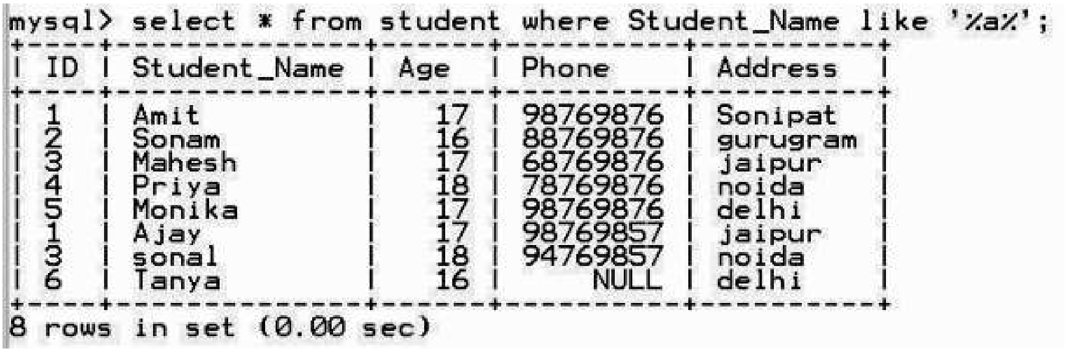
```
mysql> select * from student where Student_Name like 's%';
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | sonal        |   18 | 94769857 | noida    |
+----+--------------+------+----------+----------+
2 rows in set (0.00 sec)
```

Example 2: To match a string that ends with 'a', its pattern will be '%a'. As we don't know how many characters are there before 'a', so '%' sign is used before 'a'.



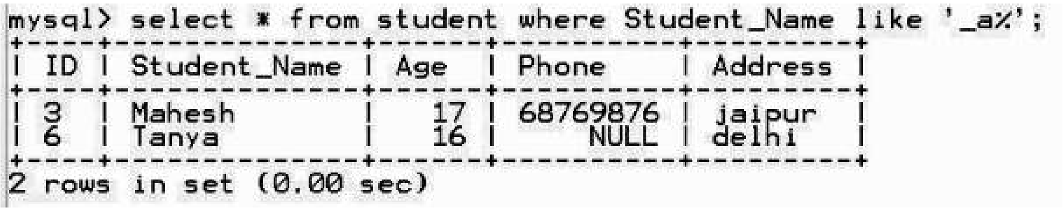
```
mysql> select * from student where Student_Name like '%a';
+----+--------------+------+----------+---------+
| ID | Student_Name | Age  | Phone    | Address |
+----+--------------+------+----------+---------+
| 4  | Priya        |   18 | 78769876 | noida   |
| 5  | Monika       |   17 | 98769876 | delhi   |
| 6  | Tanya        |   16 |     NULL | delhi   |
+----+--------------+------+----------+---------+
3 rows in set (0.00 sec)
```

Example 3: To match a string that contains 'a', its pattern will be '%a%'. As we don't know how many characters are there before or after 'a', so '%' sign is used before and after 'a'.



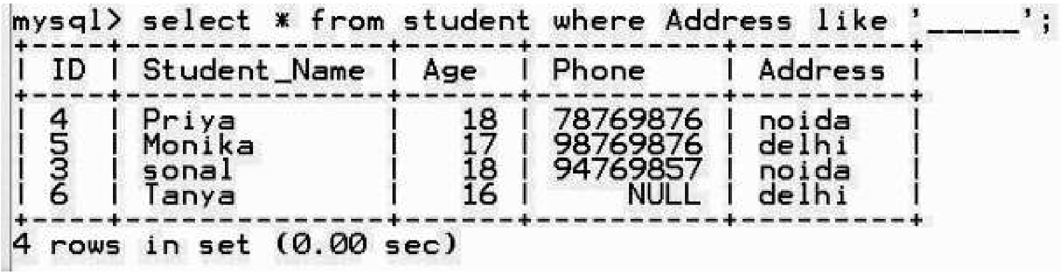
```
mysql> select * from student where Student_Name like '%a%';
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
8 rows in set (0.00 sec)
```

Example 4: To match a string that has a letter 'a' at the second position, its pattern will be '_a%'. As we know there must be exactly one character before 'a' and we don't know how many characters are thereafter 'a', so the '_' sign is used before 'a' and the '%' sign is used after 'a'.



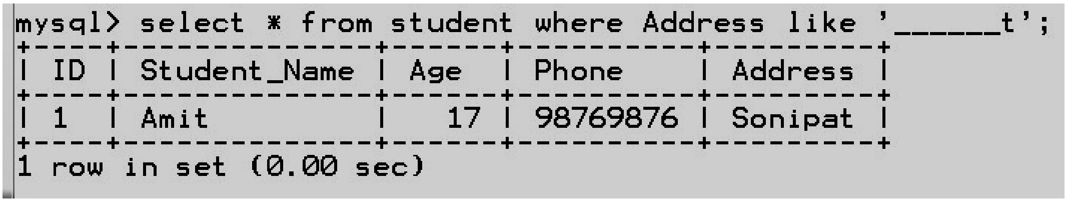
```
mysql> select * from student where Student_Name like '_a%';
+----+--------------+------+----------+---------+
| ID | Student_Name | Age  | Phone    | Address |
+----+--------------+------+----------+---------+
| 3  | Mahesh       |   17 | 68769876 | jaipur  |
| 6  | Tanya        |   16 |     NULL | delhi   |
+----+--------------+------+----------+---------+
2 rows in set (0.00 sec)
```

Example 5: To match a string that has exactly 5 characters, its pattern will be ‘_ _ _ _ _’. As we know there must be exactly 5 characters, so the ‘_’ sign is used 5 times.



```
mysql> select * from student where Address like '_____';
+----+--------------+------+----------+---------+
| ID | Student_Name | Age  | Phone    | Address |
+----+--------------+------+----------+---------+
| 4  | Priya        |   18 | 78769876 | noida   |
| 5  | Monika       |   17 | 98769876 | delhi   |
| 3  | sonal        |   18 | 94769857 | noida   |
| 6  | Tanya        |   16 |     NULL | delhi   |
+----+--------------+------+----------+---------+
4 rows in set (0.00 sec)
```

Example 6: To match a string that has exactly 7 characters and ends with ‘t’, its pattern will be ‘_ _ _ _ _ _ t’. As we know there must be exactly 7 characters, so the ‘_’ sign is used 6 times before ‘t’.



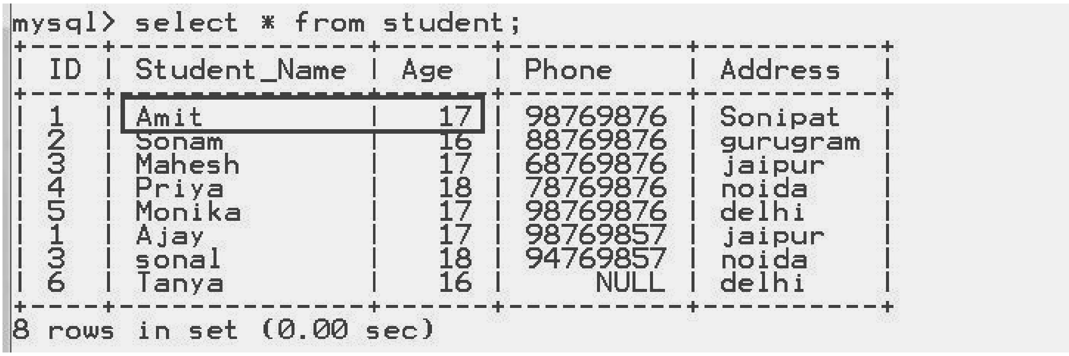
```
mysql> select * from student where Address like '______t';
+----+--------------+------+----------+---------+
| ID | Student_Name | Age  | Phone    | Address |
+----+--------------+------+----------+---------+
| 1  | Amit         |   17 | 98769876 | Sonipat |
+----+--------------+------+----------+---------+
1 row in set (0.00 sec)
```

Update Command:

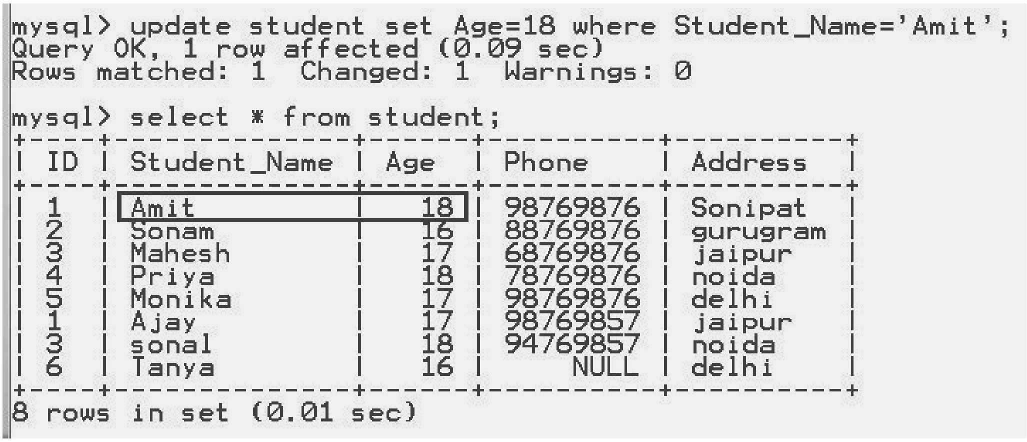
It is used to update the existing data in a table.

Command: update <table name> set <column name> = <new data> where <condition>;

Example 1: Let’s update the Age to 18 of that student whose name is Amit.

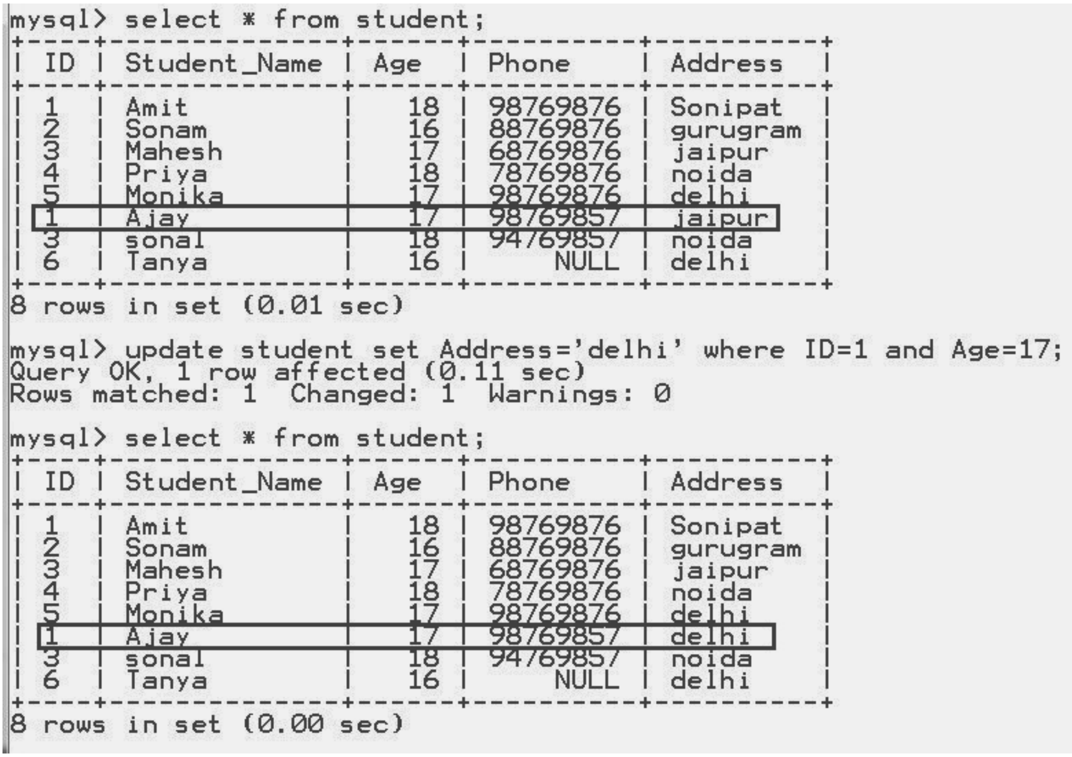


```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   17 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
8 rows in set (0.00 sec)
```



```
mysql> update student set Age=18 where Student_Name='Amit';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
8 rows in set (0.01 sec)
```

Example 2: Let's update the city to 'Delhi' of that student whose ID is 1 and Age is 17.

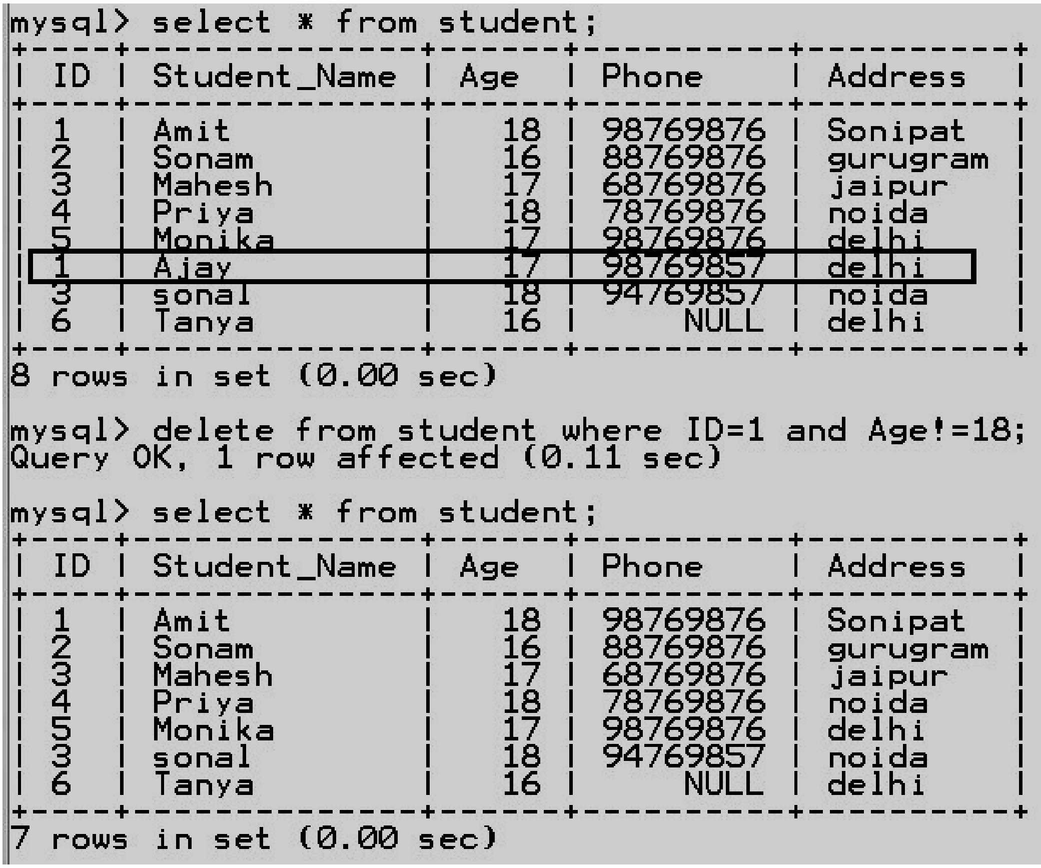


```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | jaipur   |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
8 rows in set (0.01 sec)

mysql> update student set Address='delhi' where ID=1 and Age=17;
Query OK, 1 row affected (0.11 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | delhi    |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
8 rows in set (0.00 sec)
```

Delete Command: It is used to delete the existing rows in a table that matches the condition.

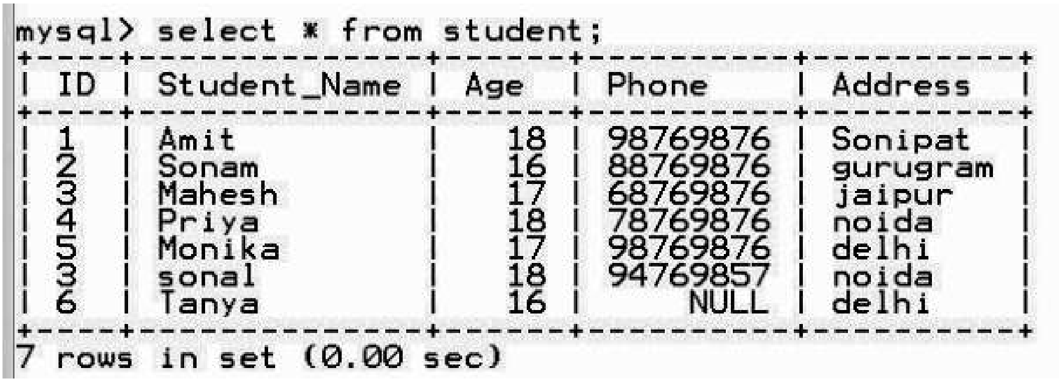
Query: delete from <table name> where <condition>;

Example 1: Let's delete the data of those students whose ID is 1 but whose age is not 18.

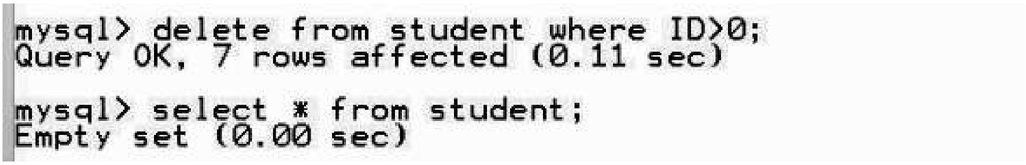


```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 1  | Ajay         |   17 | 98769857 | delhi    |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
8 rows in set (0.00 sec)

mysql> delete from student where ID=1 and Age!=18;
Query OK, 1 row affected (0.11 sec)

mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)
```

Example 2: Let's delete all data from the student table. To do this, we need to give a condition that matches all the records. As IDs are greater than 0, so let's delete all those records where ID is greater than 0. Note: The same can be done using the truncate command. ***truncate student;***



```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | gurugram |
| 3  | Mahesh       |   17 | 68769876 | jaipur   |
| 4  | Priya        |   18 | 78769876 | noida    |
| 5  | Monika       |   17 | 98769876 | delhi    |
| 3  | sonal        |   18 | 94769857 | noida    |
| 6  | Tanya        |   16 |     NULL | delhi    |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)
```

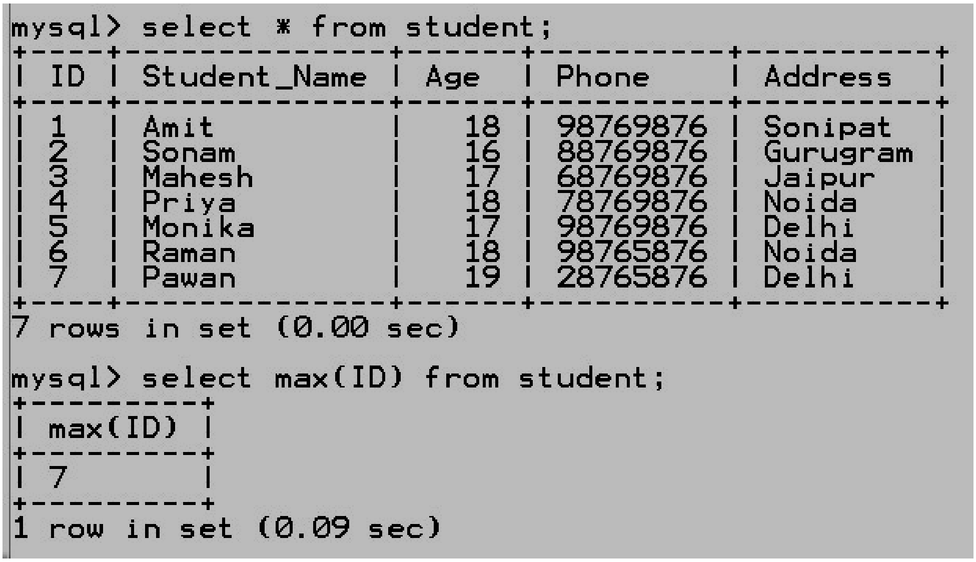


```
mysql> delete from student where ID>0;
Query OK, 7 rows affected (0.11 sec)

mysql> select * from student;
Empty set (0.00 sec)
```

Aggregate Functions:

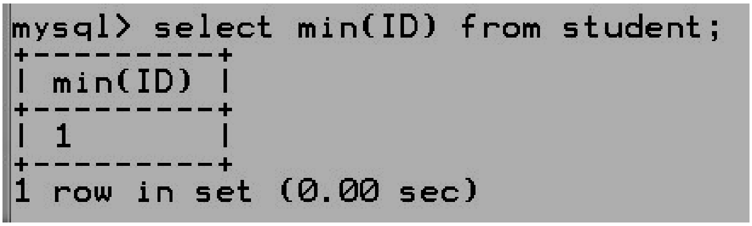
Aggregate functions are those functions that operate on a list of values and return a single-digit value or we can summarize the data using aggregate functions.

1. **Max**: it is used to find out the maximum value from a column.



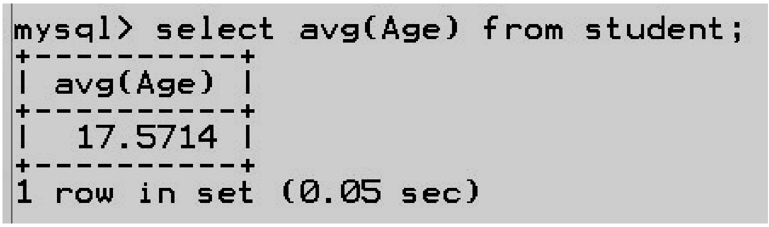
```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | Gurugram |
| 3  | Mahesh       |   17 | 68769876 | Jaipur   |
| 4  | Priya        |   18 | 78769876 | Noida    |
| 5  | Monika       |   17 | 98769876 | Delhi    |
| 6  | Raman        |   18 | 98765876 | Noida    |
| 7  | Pawan        |   19 | 28765876 | Delhi    |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)

mysql> select max(ID) from student;
+---------+
| max(ID) |
+---------+
| 7       |
+---------+
1 row in set (0.09 sec)
```

2. **Min**: it is used to find out the minimum value from a column.



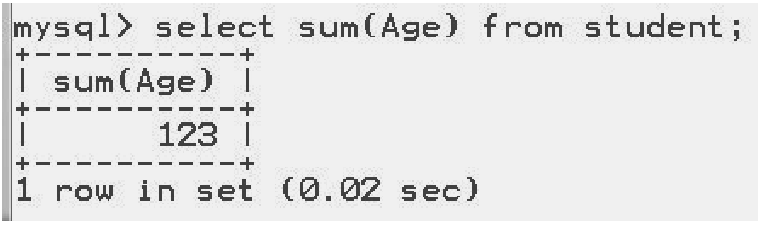
```
mysql> select min(ID) from student;
+---------+
| min(ID) |
+---------+
| 1       |
+---------+
1 row in set (0.00 sec)
```

3. **Avg**: it is used to find out the average value from a column.



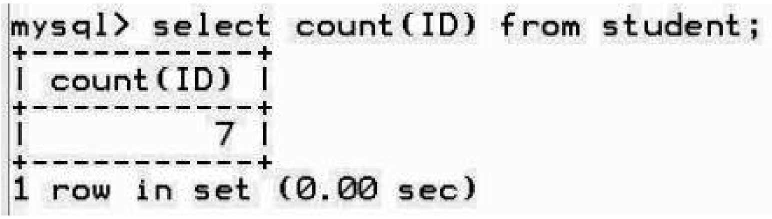
```
mysql> select avg(Age) from student;
+----------+
| avg(Age) |
+----------+
|  17.5714 |
+----------+
1 row in set (0.05 sec)
```

4. **Sum**: it is used to find out the sum of all values of a column.



```
mysql> select sum(Age) from student;
+----------+
| sum(Age) |
+----------+
|      123 |
+----------+
1 row in set (0.02 sec)
```

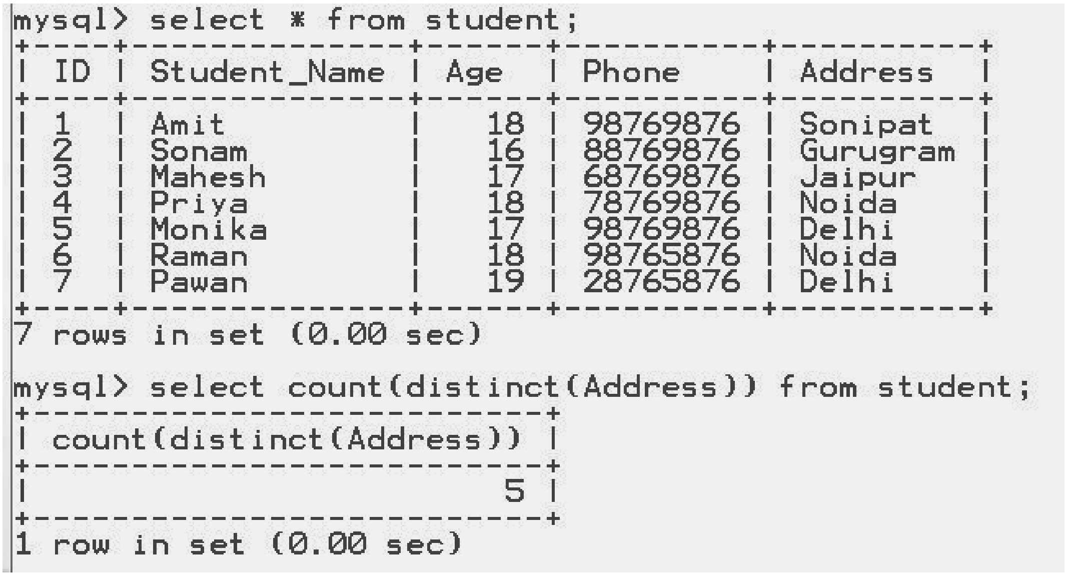
5. **Count**: it is used to count the number of values in a column.



```
mysql> select count(ID) from student;
+-----------+
| count(ID) |
+-----------+
|         7 |
+-----------+
1 row in set (0.00 sec)
```

<u>Note</u>: Distinct keyword can be used with aggregate functions to find out the max, min, sum, avg, and count of unique values.

Example: Let's find out the total number of cities from where students came to study. Here more than one student is from the same city. So, we need to use distinct keyword along with the count function.



```
mysql> select * from student;
+----+---------------+------+----------+----------+
| ID | Student_Name  | Age  | Phone    | Address  |
+----+---------------+------+----------+----------+
| 1  | Amit          |   18 | 98769876 | Sonipat  |
| 2  | Sonam         |   16 | 88769876 | Gurugram |
| 3  | Mahesh        |   17 | 68769876 | Jaipur   |
| 4  | Priya         |   18 | 78769876 | Noida    |
| 5  | Monika        |   17 | 98769876 | Delhi    |
| 6  | Raman         |   18 | 98765876 | Noida    |
| 7  | Pawan         |   19 | 28765876 | Delhi    |
+----+---------------+------+----------+----------+
7 rows in set (0.00 sec)

mysql> select count(distinct(Address)) from student;
+--------------------------+
| count(distinct(Address)) |
+--------------------------+
|                        5 |
+--------------------------+
1 row in set (0.00 sec)
```

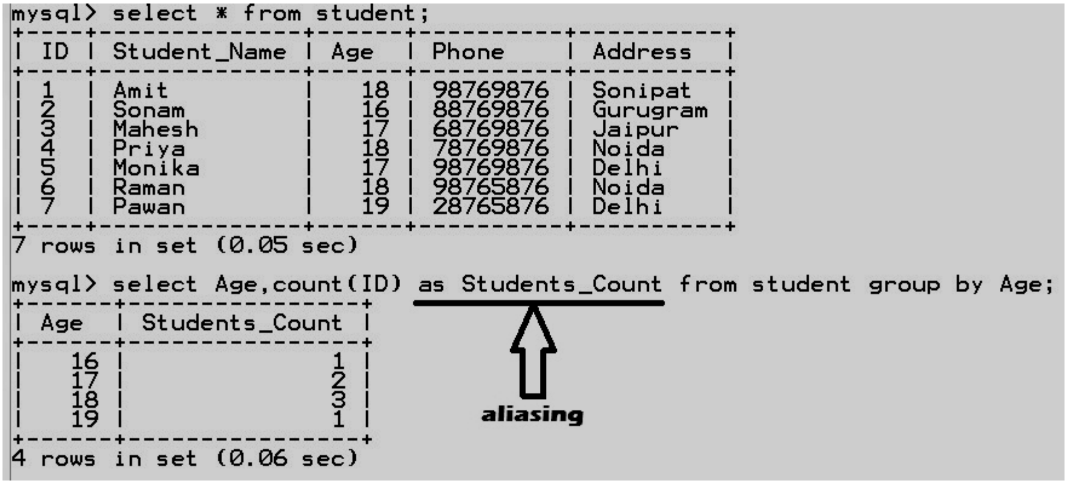
Group by clause:

The GROUP BY clause is used to group rows that have the same values into summary rows. Group by clause is often used with aggregate functions like MAX(), MIN(), SUM(), AVG() and COUNT() to group the result by one or more columns.

- It can be used with or without where clause in the select statement.
- It is applied only to numeric values.
- It can’t be applied with distinct keyword.

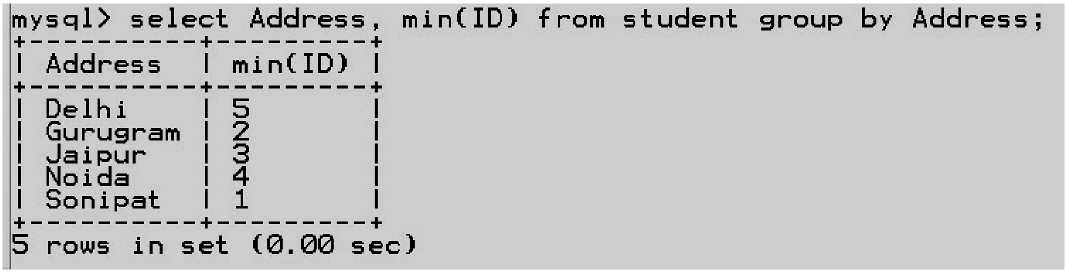
Query: group by <column name>

Example 1: Let’s count the number of students of having same age in the student table.



```
mysql> select * from student;
+----+---------------+------+----------+----------+
| ID | Student_Name  | Age  | Phone    | Address  |
+----+---------------+------+----------+----------+
| 1  | Amit          |   18 | 98769876 | Sonipat  |
| 2  | Sonam         |   16 | 88769876 | Gurugram |
| 3  | Mahesh        |   17 | 68769876 | Jaipur   |
| 4  | Priya         |   18 | 78769876 | Noida    |
| 5  | Monika        |   17 | 98769876 | Delhi    |
| 6  | Raman         |   18 | 98765876 | Noida    |
| 7  | Pawan         |   19 | 28765876 | Delhi    |
+----+---------------+------+----------+----------+
7 rows in set (0.05 sec)

mysql> select Age,count(ID) as Students_Count from student group by Age;
+------+----------------+
| Age  | Students_Count |
+------+----------------+
|   16 |              1 |
|   17 |              2 |
|   18 |              3 |
|   19 |              1 |
+------+----------------+
4 rows in set (0.06 sec)
```

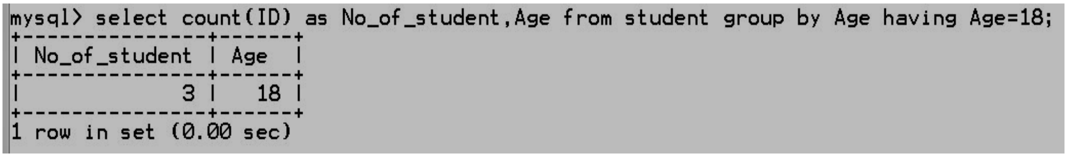
Example 2: Let’s city-wise find out the minimum value of ID.



```
mysql> select Address, min(ID) from student group by Address;
+----------+---------+
| Address  | min(ID) |
+----------+---------+
| Delhi    | 5       |
| Gurugram | 2       |
| Jaipur   | 3       |
| Noida    | 4       |
| Sonipat  | 1       |
+----------+---------+
5 rows in set (0.00 sec)
```

Having clause: It is used to filter the result set of group by clause in the select statement.

Note: To filter the result set of the group by clause, only **having clause** can be used whereas for all other queries where clause is used.



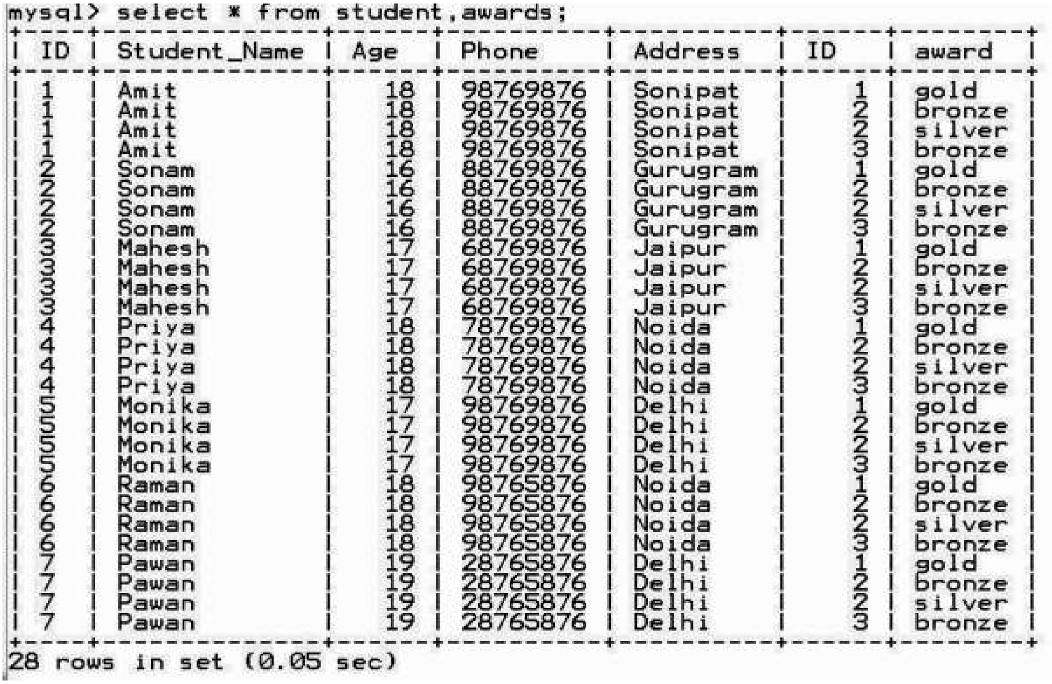
```
mysql> select count(ID) as No_of_student,Age from student group by Age having Age=18;
+---------------+------+
| No_of_student | Age  |
+---------------+------+
|             3 |   18 |
+---------------+------+
1 row in set (0.00 sec)
```

Joins:

Joins are used to combine rows from multiple tables.

Types of joins:

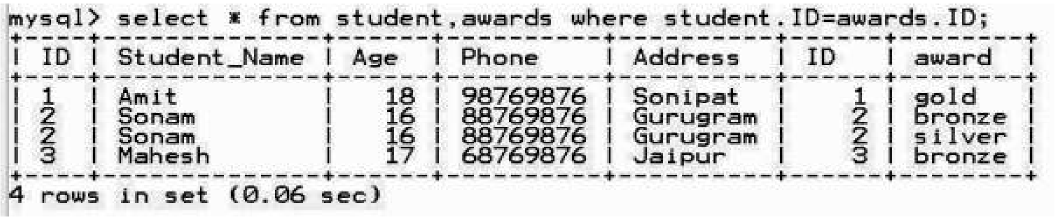
1. **Cartesian product (cross join):** It gives all possible combinations from more than one table. It combines every row from one table with every row from another table. Suppose we have 5 rows in the first table and 4 rows in the second table then the total number of rows in the Cartesian product of these two tables will be 20 rows.
   Cardinality of the final table of Cartesian product = cardinality of the first table * cardinality of the second table.
   Example: we have two tables' 'student' and 'awards'. Let's apply the Cartesian product to these two tables. (Refer page no. 180 for student and awards table)



```
mysql> select * from student,awards;
+----+--------------+-----+----------+----------+----+--------+
| ID | Student_Name | Age | Phone    | Address  | ID | award  |
+----+--------------+-----+----------+----------+----+--------+
| 1  | Amit         |  18 | 98769876 | Sonipat  |  1 | gold   |
| 1  | Amit         |  18 | 98769876 | Sonipat  |  2 | bronze |
| 1  | Amit         |  18 | 98769876 | Sonipat  |  2 | silver |
| 1  | Amit         |  18 | 98769876 | Sonipat  |  3 | bronze |
| 2  | Sonam        |  16 | 88769876 | Gurugram |  1 | gold   |
| 2  | Sonam        |  16 | 88769876 | Gurugram |  2 | bronze |
| 2  | Sonam        |  16 | 88769876 | Gurugram |  2 | silver |
| 2  | Sonam        |  16 | 88769876 | Gurugram |  3 | bronze |
| 3  | Mahesh       |  17 | 68769876 | Jaipur   |  1 | gold   |
| 3  | Mahesh       |  17 | 68769876 | Jaipur   |  2 | bronze |
| 3  | Mahesh       |  17 | 68769876 | Jaipur   |  2 | silver |
| 3  | Mahesh       |  17 | 68769876 | Jaipur   |  3 | bronze |
| 4  | Priya        |  18 | 78769876 | Noida    |  1 | gold   |
| 4  | Priya        |  18 | 78769876 | Noida    |  2 | bronze |
| 4  | Priya        |  18 | 78769876 | Noida    |  2 | silver |
| 4  | Priya        |  18 | 78769876 | Noida    |  3 | bronze |
| 5  | Monika       |  17 | 98769876 | Delhi    |  1 | gold   |
| 5  | Monika       |  17 | 98769876 | Delhi    |  2 | bronze |
| 5  | Monika       |  17 | 98769876 | Delhi    |  2 | silver |
| 5  | Monika       |  17 | 98769876 | Delhi    |  3 | bronze |
| 6  | Raman        |  18 | 98765876 | Noida    |  1 | gold   |
| 6  | Raman        |  18 | 98765876 | Noida    |  2 | bronze |
| 6  | Raman        |  18 | 98765876 | Noida    |  2 | silver |
| 6  | Raman        |  18 | 98765876 | Noida    |  3 | bronze |
| 7  | Pawan        |  19 | 28765876 | Delhi    |  1 | gold   |
| 7  | Pawan        |  19 | 28765876 | Delhi    |  2 | bronze |
| 7  | Pawan        |  19 | 28765876 | Delhi    |  2 | silver |
| 7  | Pawan        |  19 | 28765876 | Delhi    |  3 | bronze |
+----+--------------+-----+----------+----------+----+--------+
28 rows in set (0.05 sec)
```

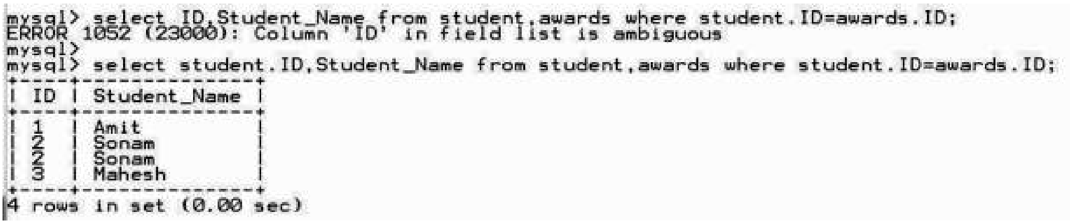
2. **Equi join**: It joins the tables based on one common column. However, the final result will consist of a common column from both tables.

Example: we have two tables - student and awards. Let's apply equi join on these two tables.



```
mysql> select * from student,awards where student.ID=awards.ID;
+----+--------------+-----+----------+----------+----+--------+
| ID | Student_Name | Age | Phone    | Address  | ID | award  |
+----+--------------+-----+----------+----------+----+--------+
| 1  | Amit         |  18 | 98769876 | Sonipat  |  1 | gold   |
| 2  | Sonam        |  16 | 88769876 | Gurugram |  2 | bronze |
| 2  | Sonam        |  16 | 88769876 | Gurugram |  2 | silver |
| 3  | Mahesh       |  17 | 68769876 | Jaipur   |  3 | bronze |
+----+--------------+-----+----------+----------+----+--------+
4 rows in set (0.06 sec)
```

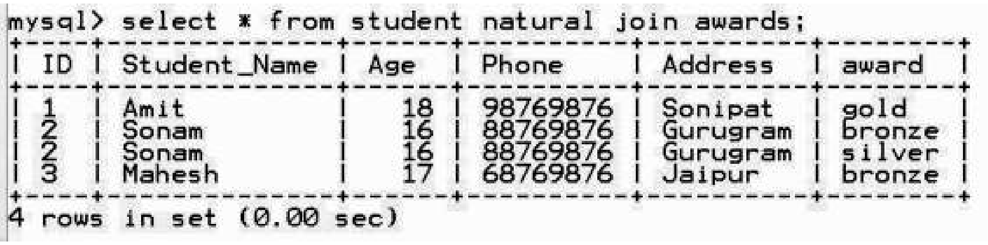
Here both the tables have common column IDs. So, to avoid ambiguity (confusion), we need to mention the table name before the column name.



```
mysql> select ID,Student_Name from student,awards where student.ID=awards.ID;
ERROR 1052 (23000): Column 'ID' in field list is ambiguous
mysql>
mysql> select student.ID,Student_Name from student,awards where student.ID=awards.ID;
+----+--------------+
| ID | Student_Name |
+----+--------------+
| 1  | Amit         |
| 2  | Sonam        |
| 2  | Sonam        |
| 3  | Mahesh       |
+----+--------------+
4 rows in set (0.00 sec)
```

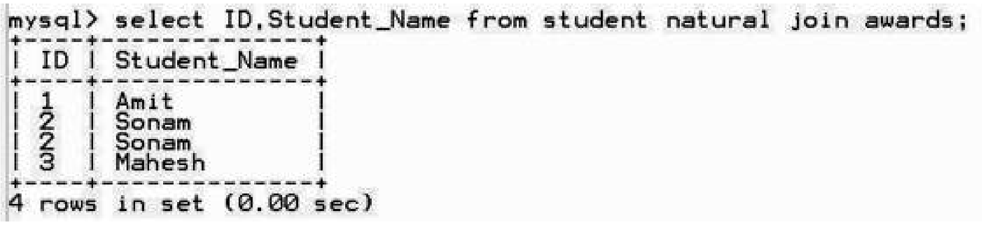
3. **Natural Join:** It joins the tables based on one common column. However, the final result will consist of a common column only once.

Example: we have two tables' - students and awards. Let's apply natural join on these two tables.



```
mysql> select * from student natural join awards;
+----+--------------+------+----------+----------+--------+
| ID | Student_Name | Age  | Phone    | Address  | award  |
+----+--------------+------+----------+----------+--------+
| 1  | Amit         |   18 | 98769876 | Sonipat  | gold   |
| 2  | Sonam        |   16 | 88769876 | Gurugram | bronze |
| 2  | Sonam        |   16 | 88769876 | Gurugram | silver |
| 3  | Mahesh       |   17 | 68769876 | Jaipur   | bronze |
+----+--------------+------+----------+----------+--------+
4 rows in set (0.00 sec)
```

Here both the tables have common column IDs. But there is no ambiguity arises on the name of the common column.



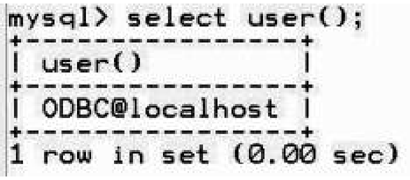
```
mysql> select ID,Student_Name from student natural join awards;
+----+--------------+
| ID | Student_Name |
+----+--------------+
| 1  | Amit         |
| 2  | Sonam        |
| 2  | Sonam        |
| 3  | Mahesh       |
+----+--------------+
4 rows in set (0.00 sec)
```

Python and MySQL Interface

In real-life scenarios, users interact with applications whereas data is stored in the database. We have already developed Python applications and database tables. Now, we will connect the Python application and MySQL database.

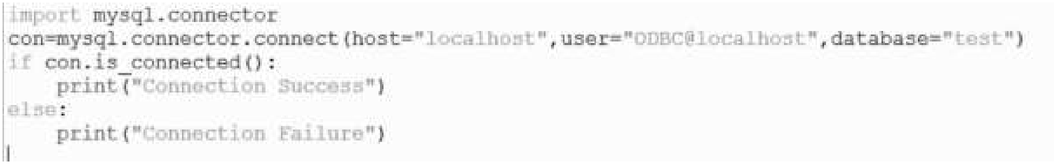
Steps for creating Python and MySQL connectivity applications:

1. **import mysql.connector**
2. **Create a connection**
   <connection object> =mysql.connector.connect (host="localhost", user=<username>, passwd= <password>, database =<database name>)
   Username = To know the user name, run the **select user();** command on your MySQL.



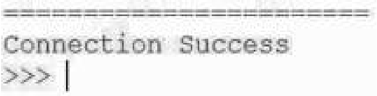
```
mysql> select user();
+----------------+
| user()         |
+----------------+
| ODBC@localhost |
+----------------+
1 row in set (0.00 sec)
```

Password = It should be the same as we set during MySQL installation.
Example: Let's create a connection of Python with MySQL and test it.



```
import mysql.connector
con=mysql.connector.connect(host="localhost",user="ODBC@localhost",database="test")
if con.is_connected():
    print("Connection Success")
else:
    print("Connection Failure")
```

Output:



```
=======================
Connection Success
>>> |
```

Here in the above example, I had not set any database password, so, the passwd option is not mentioned in the connection string. **is_connected()** function checks whether the connection string can connect Python with MySQL or not. If the connection string can make a connection with Python and MySQL, is_connected() returns True otherwise False.

3. **Create a cursor**

   Cursor: It is a pointer or iterator which points towards the resultset of the SQL query. Whenever a SQL query runs, It gives the entire result set in one go. We may not require the entire resultset at once. So, a cursor is created and data from the entire resultset will be fetched row by row as per our requirement.

   Syntax: <cursor object> = <connection object>.cursor()

4. **Execute query**

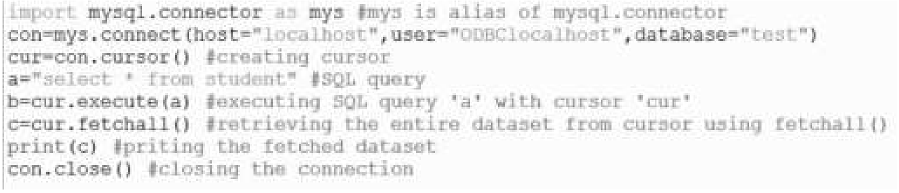
   Syntax: <cursor object>. execute(<SQL query string>)

5. **Extract data from the result set**

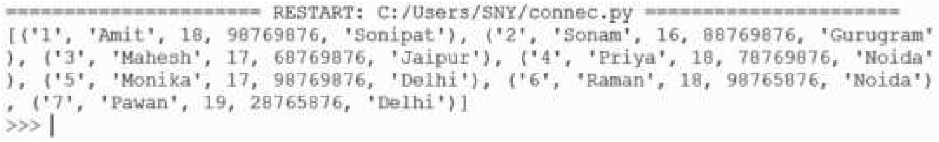
   As we know, Data from the database is retrieved using the **select** query. After running the select query, we get the resultset. Now to fetch the data from the resultset, the following functions are used:

   a) **fetchall():** It returns all the records from the resultset. Each record will be in the form of a tuple whereas the entire resultset will be in the form of a list.

      Syntax: <variable name>=<cursor>.fetchall()



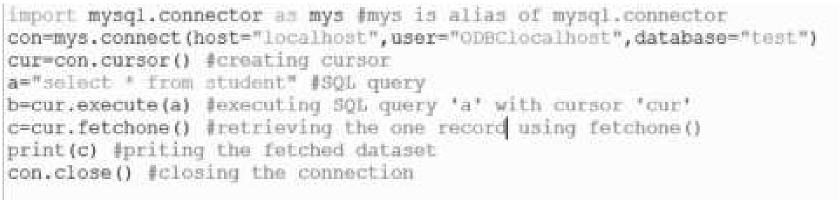
```
import mysql.connector as mys #mys is alias of mysql.connector
con=mys.connect(host="localhost",user="ODBClocalhost",database="test")
cur=con.cursor() #creating cursor
a="select * from student" #SQL query
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
c=cur.fetchall() #retrieving the entire dataset from cursor using fetchall()
print(c) #priting the fetched dataset
con.close() #closing the connection
```



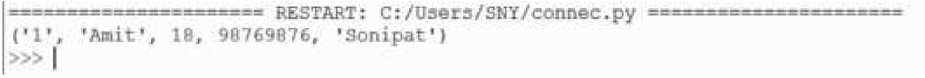
```
======================= RESTART: C:/Users/SNY/connec.py =======================
[('1', 'Amit', 18, 98769876, 'Sonipat'), ('2', 'Sonam', 16, 88769876, 'Gurugram'
), ('3', 'Mahesh', 17, 68769876, 'Jaipur'), ('4', 'Priya', 18, 78769876, 'Noida'
), ('5', 'Monika', 17, 98769876, 'Delhi'), ('6', 'Raman', 18, 98765876, 'Noida')
, ('7', 'Pawan', 19, 28765876, 'Delhi')]
>>> |
```

   b) **fetchone():** It returns one row from the resultset in the form of a tuple. It returns None if no more records are there. To get multiple rows, we need to run fetchone() multiple times.

      Syntax: <variable name>=<cursor>.fetchone()



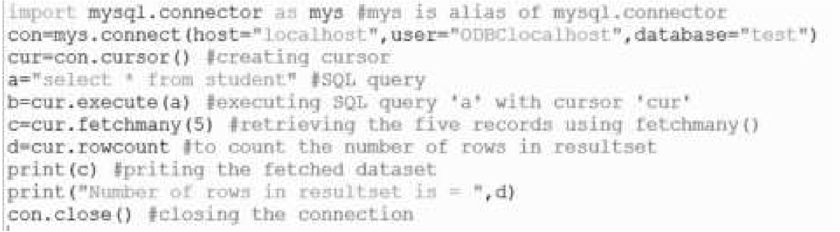
```
import mysql.connector as mys #mys is alias of mysql.connector
con=mys.connect(host="localhost",user="ODBClocalhost",database="test")
cur=con.cursor() #creating cursor
a="select * from student" #SQL query
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
c=cur.fetchone() #retrieving the one record using fetchone()
print(c) #priting the fetched dataset
con.close() #closing the connection
```



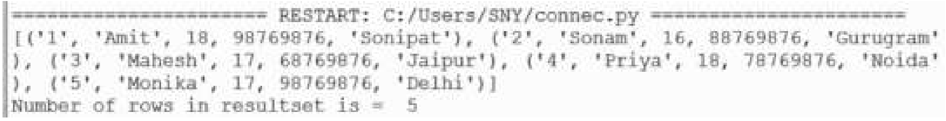
```
======================= RESTART: C:/Users/SNY/connec.py =======================
('1', 'Amit', 18, 98769876, 'Sonipat')
>>> |
```

c) fetchmany(): It returns n number of records from the resultset in the form of a list where each record is in the form of a tuple. It returns an empty tuple if no more records are there.

Syntax: <variable name>=<cursor>.fetchmany(n)



```
import mysql.connector as mys #mys is alias of mysql.connector
con=mys.connect(host="localhost",user="ODBClocalhost",database="test")
cur=con.cursor() #creating cursor
a="select * from student" #SQL query
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
c=cur.fetchmany(5) #retrieving the five records using fetchmany()
d=cur.rowcount #to count the number of rows in resultset
print(c) #priting the fetched dataset
print("Number of rows in resultset is = ",d)
con.close() #closing the connection
```



```
======================= RESTART: C:/Users/SNY/connec.py =======================
[('1', 'Amit', 18, 98769876, 'Sonipat'), ('2', 'Sonam', 16, 88769876, 'Gurugram'
), ('3', 'Mahesh', 17, 68769876, 'Jaipur'), ('4', 'Priya', 18, 78769876, 'Noida'
), ('5', 'Monika', 17, 98769876, 'Delhi')]
Number of rows in resultset is =  5
```

d) rowcount: It is the cursor's property to count the number of rows in the resultset.

Syntax: <variable name>=<cursor>.rowcount

Example: Refer to the example of fetchmany()

6. **Close the connection**

After doing all the processing, the connection should be closed.

Syntax: <connection object>.close()

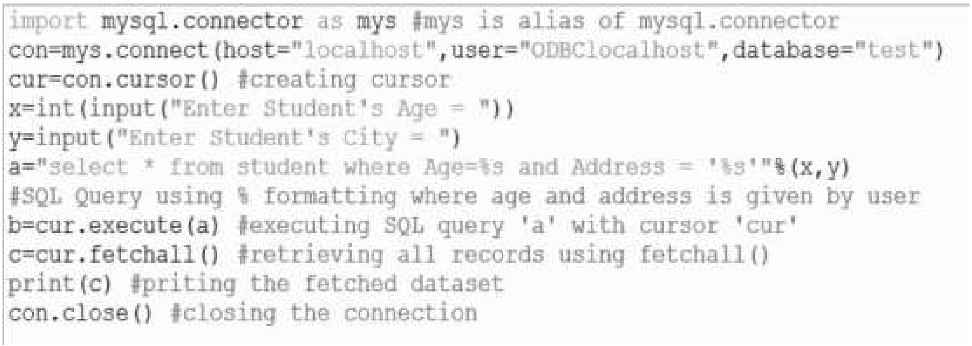
Example: Refer to the example of fetchmany()

Format specifier: we need a format specifier to write SQL queries based on user input. To do this we have two ways:

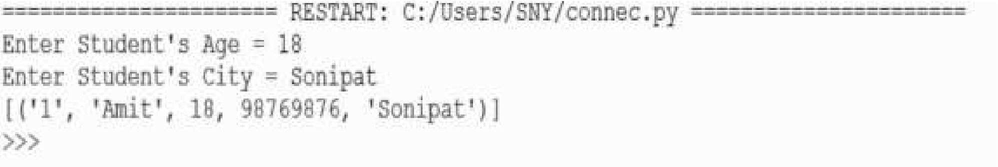
1. **Using % formatting:**

Whenever we need to complete an SQL query based on user input, we write a placeholder '%s' on that place.

Example: Let's fetch all the data of the student table where age is 'x' and city is 'y'. (Here, the value of x and y will be given by the user during run time).

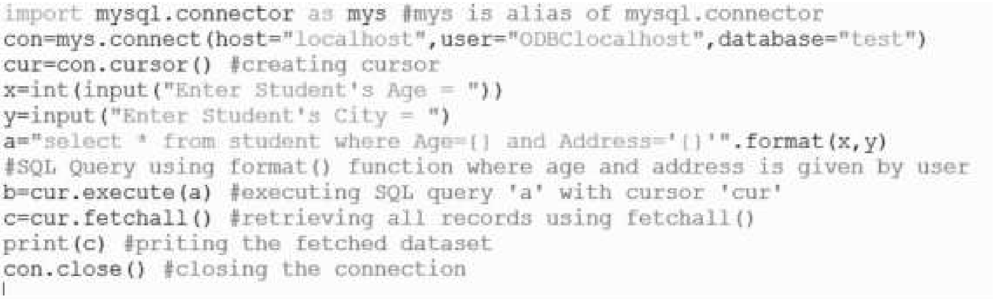


```
import mysql.connector as mys #mys is alias of mysql.connector
con=mys.connect(host="localhost",user="ODBClocalhost",database="test")
cur=con.cursor() #creating cursor
x=int(input("Enter Student's Age = "))
y=input("Enter Student's City = ")
a="select * from student where Age=%s and Address = '%s'"%(x,y)
#SQL Query using % formatting where age and address is given by user
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
c=cur.fetchall() #retrieving all records using fetchall()
print(c) #priting the fetched dataset
con.close() #closing the connection
```

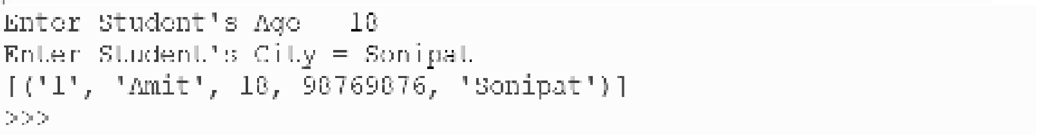


```
====================== RESTART: C:/Users/SNY/connec.py ======================
Enter Student's Age = 18
Enter Student's City = Sonipat
[('1', 'Amit', 18, 98769876, 'Sonipat')]
>>>
```

2. **Using format() function:**
   Whenever we need to complete an SQL query based on user input, we write a placeholder {} on that place. If we need to complete the SQL query based on multiple user inputs, we write placeholder {}, {}, {} and so on at those places and pass user-defined values in the format function sequentially. Here 1st values passed in the format function will be passed to 1st {}, 2nd values passed to 2nd {}, 3rd value passed to 3rd {} and so on.
   Example: Let's fetch all the data of the student table where age is 'x' and city is 'y'. (Here, the value of x and y will be given by the user during run time).



```
import mysql.connector as mys #mys is alias of mysql.connector
con=mys.connect(host="localhost",user="ODBClocalhost",database="test")
cur=con.cursor() #creating cursor
x=int(input("Enter Student's Age = "))
y=input("Enter Student's City = ")
a="select * from student where Age={} and Address='{}'".format(x,y)
#SQL Query using format() function where age and address is given by user
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
c=cur.fetchall() #retrieving all records using fetchall()
print(c) #priting the fetched dataset
con.close() #closing the connection
```



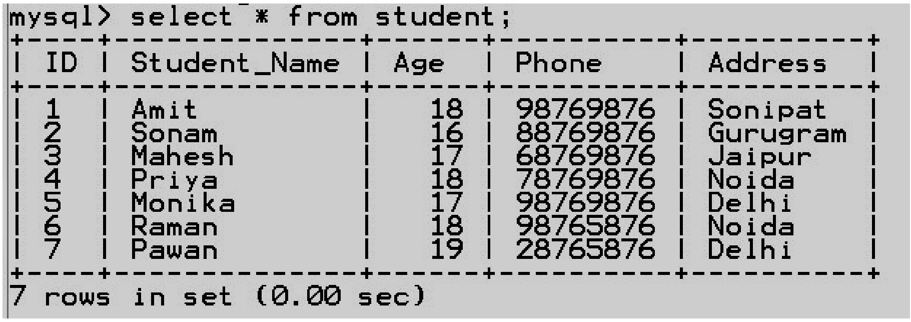
```
Enter Student's Age   18
Enter Student's City = Sonipat
[('1', 'Amit', 18, 98769876, 'Sonipat')]
>>>
```

Commit: Whenever we perform update, delete or insert query, the commit() function must be run before closing the connection.

Syntax: <connection object>.commit()

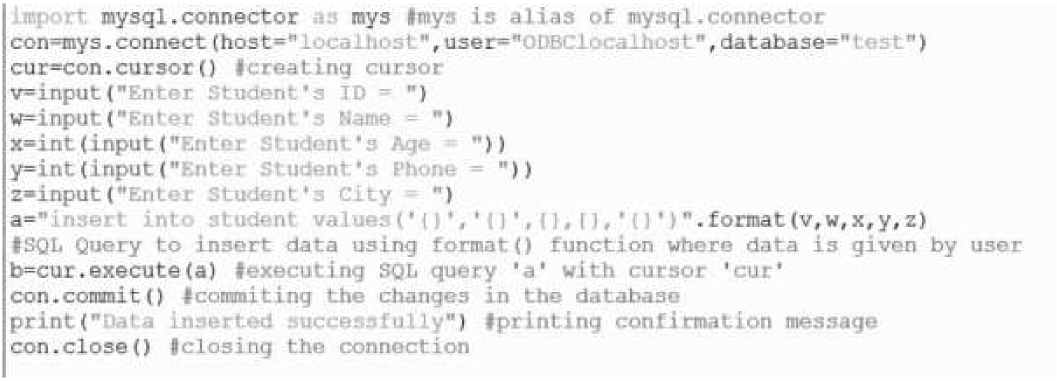
Example 1: Insert data in the student table which will be given by the user during run time.

Data in the student table before insertion



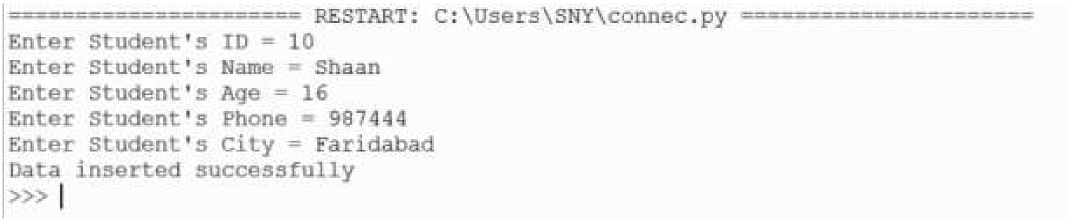
```
mysql> select * from student;
+----+--------------+------+----------+----------+
| ID | Student_Name | Age  | Phone    | Address  |
+----+--------------+------+----------+----------+
| 1  | Amit         |   18 | 98769876 | Sonipat  |
| 2  | Sonam        |   16 | 88769876 | Gurugram |
| 3  | Mahesh       |   17 | 68769876 | Jaipur   |
| 4  | Priya        |   18 | 78769876 | Noida    |
| 5  | Monika       |   17 | 98769876 | Delhi    |
| 6  | Raman        |   18 | 98765876 | Noida    |
| 7  | Pawan        |   19 | 28765876 | Delhi    |
+----+--------------+------+----------+----------+
7 rows in set (0.00 sec)
```

Python code:



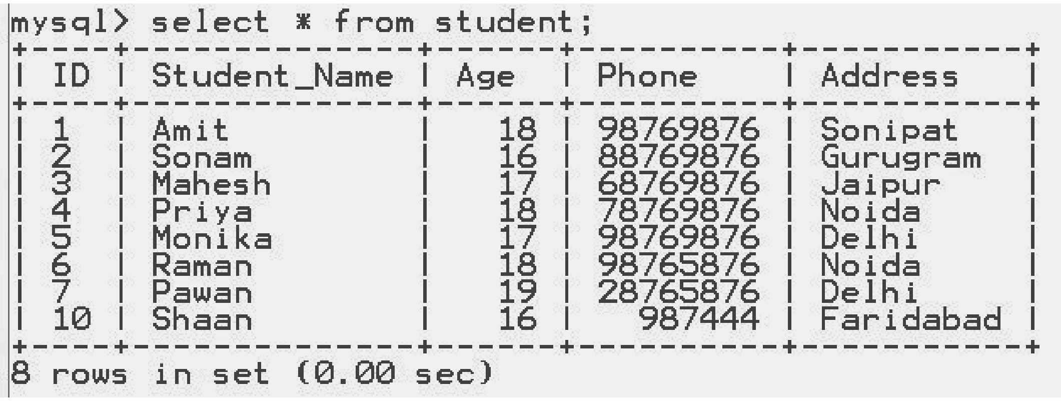
```
import mysql.connector as mys #mys is alias of mysql.connector
con=mys.connect(host="localhost",user="ODBClocalhost",database="test")
cur=con.cursor() #creating cursor
v=input("Enter Student's ID = ")
w=input("Enter Student's Name = ")
x=int(input("Enter Student's Age = "))
y=int(input("Enter Student's Phone = "))
z=input("Enter Student's City = ")
a="insert into student values('{}','{}',{},{},'{}')".format(v,w,x,y,z)
#SQL Query to insert data using format() function where data is given by user
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
con.commit() #commiting the changes in the database
print("Data inserted successfully") #printing confirmation message
con.close() #closing the connection
```

Output:



```
====================== RESTART: C:\Users\SNY\connec.py ======================
Enter Student's ID = 10
Enter Student's Name = Shaan
Enter Student's Age = 16
Enter Student's Phone = 987444
Enter Student's City = Faridabad
Data inserted successfully
>>> |
```

Data in the student table after insertion

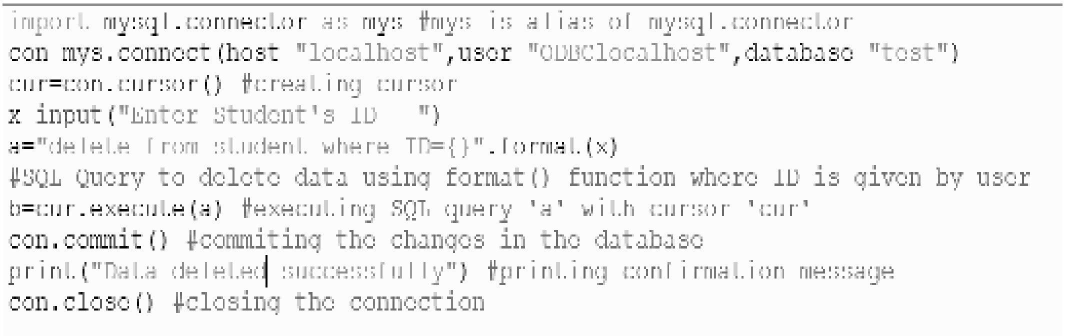


```
mysql> select * from student;
+----+---------------+------+----------+-----------+
| ID | Student_Name  | Age  | Phone    | Address   |
+----+---------------+------+----------+-----------+
| 1  | Amit          |   18 | 98769876 | Sonipat   |
| 2  | Sonam         |   16 | 88769876 | Gurugram  |
| 3  | Mahesh        |   17 | 68769876 | Jaipur    |
| 4  | Priya         |   18 | 78769876 | Noida     |
| 5  | Monika        |   17 | 98769876 | Delhi     |
| 6  | Raman         |   18 | 98765876 | Noida     |
| 7  | Pawan         |   19 | 28765876 | Delhi     |
| 10 | Shaan         |   16 |   987444 | Faridabad |
+----+---------------+------+----------+-----------+
8 rows in set (0.00 sec)
```

Example 2: Delete data of those students whose ID is given by the user during run time in the student table.

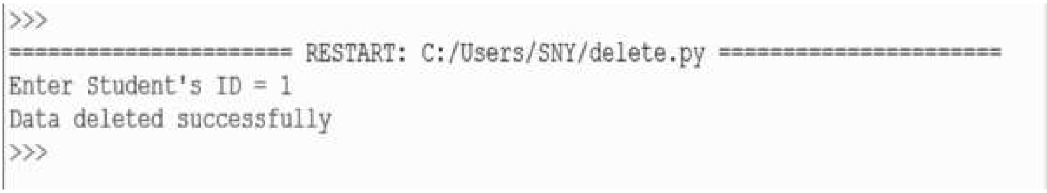
In the previous example, after inserting a record in the student table, we have 8 records in the student table. (Kindly refer previous example for the current state of the student table).

Python code:



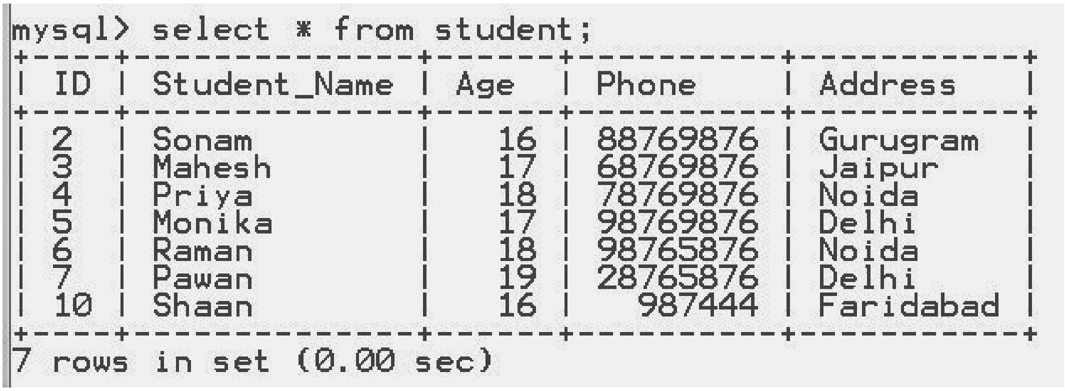
```
import mysql.connector as mys #mys is alias of mysql.connector
con mys.connect(host "localhost",user "ODBClocalhost",database "test")
cur=con.cursor() #creating cursor
x input("Enter Student's ID   ")
a="delete from student where ID={}".format(x)
#SQL Query to delete data using format() function where ID is given by user
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
con.commit() #commiting the changes in the database
print("Data deleted successfully") #printing confirmation message
con.close() #closing the connection
```

Output:



```
>>>
====================== RESTART: C:/Users/SNY/delete.py ======================
Enter Student's ID = 1
Data deleted successfully
>>>
```

Data in the student table after deletion

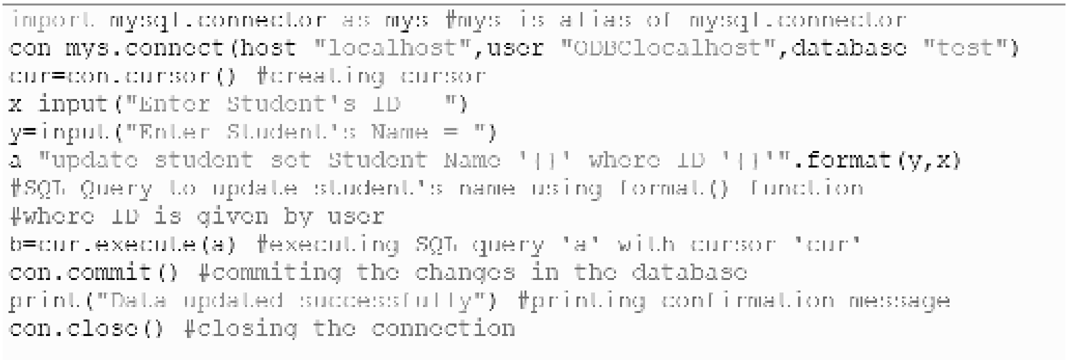


```
mysql> select * from student;
+----+--------------+------+----------+-----------+
| ID | Student_Name | Age  | Phone    | Address   |
+----+--------------+------+----------+-----------+
| 2  | Sonam        |   16 | 88769876 | Gurugram  |
| 3  | Mahesh       |   17 | 68769876 | Jaipur    |
| 4  | Priya        |   18 | 78769876 | Noida     |
| 5  | Monika       |   17 | 98769876 | Delhi     |
| 6  | Raman        |   18 | 98765876 | Noida     |
| 7  | Pawan        |   19 | 28765876 | Delhi     |
| 10 | Shaan        |   16 |   987444 | Faridabad |
+----+--------------+------+----------+-----------+
7 rows in set (0.00 sec)
```

Example 3: Update the name of those students whose name and ID are given by the user during run time in the student table.

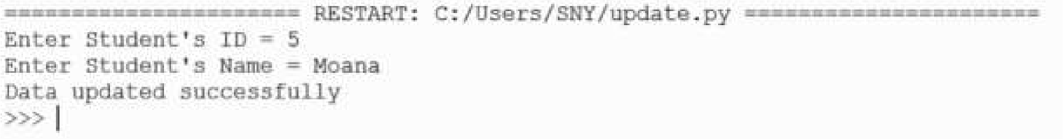
In the previous example, after deleting a record in the student table, we have 7 records in the student table. (Kindly refer previous example for the current state of the student table).

Python code:



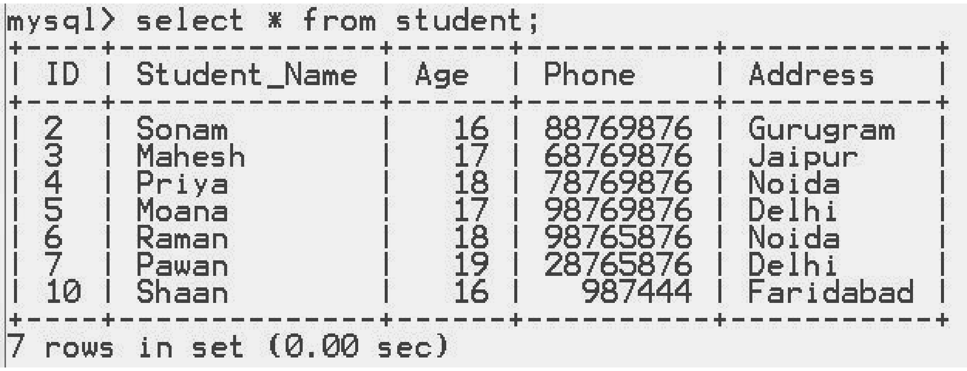
```
import mysql.connector as mys #mys is alias of mysql.connector
con=mys.connect(host="localhost",user="ODBClocalhost",database="test")
cur=con.cursor() #creating cursor
x=input("Enter Student's ID = ")
y=input("Enter Student's Name = ")
a="update student set Student_Name='{}' where ID='{}'".format(y,x)
#SQL Query to update student's name using format() function
#where ID is given by user
b=cur.execute(a) #executing SQL query 'a' with cursor 'cur'
con.commit() #commiting the changes in the database
print("Data updated successfully") #printing confirmation message
con.close() #closing the connection
```

Output:



```
======================== RESTART: C:/Users/SNY/update.py ========================
Enter Student's ID = 5
Enter Student's Name = Moana
Data updated successfully
>>> |
```

Data in student table after updation:



```
mysql> select * from student;
+----+--------------+------+----------+-----------+
| ID | Student_Name | Age  | Phone    | Address   |
+----+--------------+------+----------+-----------+
| 2  | Sonam        |   16 | 88769876 | Gurugram  |
| 3  | Mahesh       |   17 | 68769876 | Jaipur    |
| 4  | Priya        |   18 | 78769876 | Noida     |
| 5  | Moana        |   17 | 98769876 | Delhi     |
| 6  | Raman        |   18 | 98765876 | Noida     |
| 7  | Pawan        |   19 | 28765876 | Delhi     |
| 10 | Shaan        |   16 |   987444 | Faridabad |
+----+--------------+------+----------+-----------+
7 rows in set (0.00 sec)
```

DBMS3 - MULTIPLE CHOICE QUESTIONS (MCQ)

1. What is the use of DBMS?

a. Data Retrieval
b. Data Storage
c. Data Manipulation
d. All of the above

2. What is a relational data model?

a. Where data is organized in a tree-like structure with a parent-child relationship
b. Where the pointer is used to navigate through data
c. Where data is represented as objects.
d. Where data is organized into tables with rows and columns

3. What is a tuple in a relation?

a. row b. column c. field d. attribute

4. What is the attribute in a relation?

a. row **b.** column **c.** tuple **d.** key

5. What is the degree in a database?

a. Foreign key of the table
b. Primary key of the table
c. Number of columns in the table
d. Number of rows in the table

6. What is the do in a database?

a. The alternate key of a table
b. The primary key of a table
c. The set of all possible values in the column
d. The number of rows in a table

7. What is cardinality in a database?

a.Number of columns

b. Number of rows
c. Number of the primary keys
d. Number of attribute key

8. What is the role of a key in SQL?
a. Which is used to uniquely identify a record
b. To maintain data integrity
c. To enable efficient data retrieval
d. All of the above

9. Which key is used to uniquely identify a record in a table and doesn't allow NULL values?
a. Alternate key
b. Primary key
c. Super key
d. Foreign key

10. An attribute that could potentially be used as the primary key?
a. Alternate key
b. Primary key
c. Candidate key
d. Surrogate key

11. A candidate key that is not selected as the primary key can serve as a unique identifier.
a. Alternate key
b. Primary key
c. Candidate key
d. Surrogate key

12. What is the use of foreign keys?
a. To uniquely identify a record
b. To enforce data integrity
c. To define the data type of the column
d. To establish a link between two tables

13. A column or set of columns in a table that refers to the primary key of another table.
a. Alternate key
b. Primary key
c. Candidate key
d. Foreign key

14. Which of the following SQL commands is used to view a list of all databases?
a. select databases
b. show databases
c. view databases
d. project databases

15. Which of the following SQL commands is used to use/select a particular database?
a. use
b. selec
t
c. view
d. proje
ct

16. Which SQL commands are used to define and maintain the physical structure or schema of a table in a database like creating, altering and deleting database objects such as tables and constraints?
a. DDL
b. DML
c. DCL
d. TCL

17. Which SQL command is used to change or modify any attribute/column like the addition or deletion of columns in the database?

a. create

b. alter

c. delete

d. update

18. A user request to retrieve data or information from a database/table is called?

a. key

b. data

c. query

d. view

19. Which of the following is not a valid datatype in SQL?

a. Date

b. String

c. Decimal

d. Char

20. Which command is used to delete the database permanently?

a. create database

b. delete database

c. drop database

d. use database

21. Which command is used to display a list of all the tables in the current database?

a. display tables;

b. show tables;

c. view tables;

d. select all tables;

22. Which command is used in the where clause to search NULL values in a particular column?

a. IS NULL

b. IN NULL

c. NOT NULL

d. IS NOT NULL

23. Wildcard operators (%,_) are used with?

a. count

b. max

c. like

d. min

24. Which SQL function is used to determine the no. of row or non-null values?

a. min

b. max

c. count
d. sum

25. Which SQL function is used to count the entire number of rows in the database table?
a. count
b. count(*)
c. max
d. min

26. In a relational database, a key that establishes a link between two tables?
a. Primary key
b. Alternate key
c. Foreign key
d. Candidate key

27. A table containing data organized in the form of rows and columns in a relational data model?
a. Relation
b. View
c. Tuple
d. Database

28. Which SQL clause is used to filter the result of a SELECT statement in a Relational Database?
a. from
b. where
c. join
d. having

29. INSERT command is used in the database to perform
a. To retrieve a set of statement
b. To add a new row of data to a table
c. To change already existing data in a table
d. To delete any existing row in a table

30. How many candidate keys can a relation/table have in a relational database
a. One
b. Two
c. Multiple
d. None

31. What is the main difference between the candidate key and the primary key?
a. A candidate key can contain a NULL value but a primary key can’t contain a NULL value
b. A primary key can contain unique value but the candidate key doesn't need to have a unique value
c. The primary key is chosen by the designer from the available list of candidate keys.
d. No difference

32. Which DDL command is used to modify the structure of an existing table?

a. create table
b. modify table
c. alter table
d. update table

33. Which DDL command is used to remove a table along with its all content from a database?
a. DELETE TABLE
b. DROP TABLE
c. REMOVE TABLE
d. ERASE TABLE

34. Which constraint enforces data integrity by ensuring that a column always contains values?
a. NULL
b. NOT NULL
c. CHECK
d. DEFAULT

35. Which of the following SQL commands is used to change the sequence of attributes in a relation?
a. modify
b. change
c. column
d. sequence

36. Which SQL clause ensures the selection of unique data from an attribute?
a. different
b. distinct
c. not null
d. prime

37. What is the main difference between CHAR and VARCHAR datatypes in SQL?
a. CHAR is case-sensitive while VARCHAR is non case sensitive
b. CHAR stores variable length strings while VARCHAR stores fixed length string
c. CHAR stores fixed length strings while VARCHAR stores variable length string
d. CHAR is used for storing numeric data while VARCHAR is used for text data

38. What is the prime purpose of unique key constraints in a relational database?
a. To ensure that values in a column are NULL
b. To ensure that values in a column are unique
c. To define the relationship between tables
d. To specify a condition that must be met for data to be valid in a column

39. Which constraint key is used when you want to enforce uniqueness but allows NULL values in the database
a. PRIMARY KEY
b. UNIQUE KEY
c. NOT NULL
d. CHECK

40. Which of the following is not a valid DML command in SQL?

a.INSERT
b. UPDATE
c.ALTER
d. DELETE

41. Which keyword is used for table aliasing that involves giving a table short and alternative name to simplify query syntax?

a.from
b. as
c.where
d. on

42. Which SQL clause is used in the database table to eliminate duplicate rows from the query result?

a.group by
b. distinct
c.describe
d. duplicate

43. Which of the following clauses in SQL is most appropriate to use to select matching tuples in a specific range of values?

a.IN
b. LIKE
c.BETWEEN
d. IS

44. Which of the following SQL datatype allows NULL values by default?

a.INT
b. CHAR
c.VARCHAR
d. All of the above

45. Which of the following is NOT a required argument to the connect() function?

a.Hostname
b. Username
c.Password
d. Database name

46. Which method is used to execute a SQL query on a MySQL database?

a. execute()
b. query()
c. run()
d. None of the above

47. Which method is used to fetch n numbers of results from a SQL query?

a. fetchall()

b. fetchone()
c. fetchmany()
d. All of the above

48. _______ it is a pointer or iterator which points towards the resultset of the SQL query.
a.cursor
b. rset
c.temp
d. None of these

49. Which of the following is not a valid method to fetch records from a database in Python.
a. fetchmany()
b. fetchone()
c. fetchmulti()
d. fetchall()

50. To get all the records from the result set, you may use ___________.
a.cursor.fetchmany()
b. cursor.fetchall()
c.cursor.fetchone()
d. cursor.execute()

DBMS4 - SHORT ANSWER TYPE QUESTIONS:

.

1. What is SQL?
2. Differentiate between DML and DDL. Explain with the help of examples.
3. What are the different types of SQL data type?
4. What is the difference between a where statement and a having statement in SQL?
5. What is a primary key?
6. Differentiate between a database table and a database record.
7. What is Foreign Key? And how does it relate to a database?
8. What is the Domain in a database?
9. Explain the role of the clause in a query of SQL commands.
10. What is Aliasing?
11. Explain char, varchar and int datatype in SQL with examples.
12. Differentiate between candidate key and alternate key?
13. Differentiate between degree and cardinality of a table with the help of an example.
14. What are database keys? Explain all the database keys.
15. What are the constraints in SQL?
16. Explain the wildcards used with the like operator.
17. What are joins in SQL?
18. Differentiate between Natural Join and Equi Join.
19. Explain any two aggregate functions in SQL.
20. What is a cursor?

21. What is the role of the commit() function in SQL?

22. What is a Cartesian product? What will be the number of rows in the output table if we apply the Cartesian product on two tables T1 and T2 with 9 and 10 rows respectively?

23. Explain the alter table command with the help of an example.

24. Differentiate between the fetchall() function and the fetchmany() function in SQL.

25. In SQL, Define the aggregate function and write the name of the aggregate function which will display the cardinality of a table.

DBMS5 - Assertion-and-Reason Type

In the following questions, Assertion (A) and Reason (R).
Choose the correct choice as:

(a) Both Assertion (A) and Reason (R) are the true and Reason (R) is a correct explanation of Assertion (A).

(b) Both Assertion (A) and Reason (R) are the true but Reason (R) is not a correct explanation of Assertion (A).

(c) Assertion (A) is true and Reason (R) is false.

(d) Assertion (A) is false and Reason (R) is true.

1. **Assertion (A):** The COUNT function can work with distinct keyword.
   Reason (R): The DISTINCT keyword can only be used with the COUNT function.

2. **Assertion (A):** The HAVING clause can only be used with a GROUP BY statement.
   Reason (R): WHERE clause can be used in place of HAVING clause in GROUP BY statement.

3. **Assertion (A):** The LIKE operator is used for pattern matching in the WHERE clause.
   Reason (R): % and _ wildcard are used with the LIKE operator for making a pattern.

4. **Assertion (A):** The primary key is applied to a column of a table.
   Reason (R): NOT NULL constraint restricts NULL values in a column.

5. **Assertion (A):** SUM and COUNT are aggregate functions.
   Reason (R): Aggregate functions work on multiple tuples.

6. **Assertion (A):** Primary key constraints allow NULL values.
   Reason (R): The primary key constraints ensure that a column can contain unique Values for each row.

7. **Assertion (A):** Unique key constraint allows NULL values.
Reason (R): The unique key constraints ensure that a column can contain unique Values.

8. **Assertion (A):** The HAVING clause is used to filter aggregated data in SQL queries.
Reason (R): The HAVING clause is used to group rows with similar values in one or more columns into result sets.

9. **Assertion (A):** Inner Join retrieves rows that have matching values in both tables being joined.
Reason (R): Inner join excludes rows with no matching values

10. **Assertion (A):** Between operator is used to filter data within a specified range.
Reason (R): Where clause works exactly the same as between operator

DBMS6 - LONG ANSWER TYPE QUESTIONS:

.

1. Write a MySQL query to create the table 'Employee' with the following structure and constraint. Table: Employee

Column_Name	**DataType(size)**	**Constraint**
Emp_ID	Int(20)	Primary key
Emp_Name	char(100)	Not Null
Salary	int(20)	Not Null
Department	char(30)	
Age	int(15)	Not Null
Address	Varchar(200)	Unique

2. Write a MySQL query to create the table 'Student' and 'Activities' with the following structure and constraint.

Table: Student

Column_Name	**DataType(size)**	**Constraint**
Student_ID	varchar(20)	Primary key
Student_Name	char(80)	Not Null
Gender	char(20)	Not Null
Class	varchar(30)	
Age	int(20)	Not Null
Address	Varchar(150)	Unique
Phone	Int(15)	Not Null, Unique

Table: Activities

Column_Name	DataType(size)	Constraint
Student_ID	varchar(20)	Foreign key references to Student_ID of Employee table
Activity_Name	char(80)	Not Null
Position	char(30)	Not Null

3. Anmol maintains a database of medicines for his pharmacy using SQL to store the data. The structure of the table PHARMA for the purpose is as follows:

- Name of the table-PHARMA
- The attributes of PHARMA are as follows:

MID - numeric
MNAME - character of size 20
PRICE - numeric
UNITS - numeric
EXPIRY – date

Table: PHARMA

MID	MNAME	PRICE	UNITS	EXPIRY
M1	PARACETAMOL	12	120	2022-12-25
M2	CETIRIZINE	6	125	2022-10-12
M3	METFORMIN	14	150	2022-05-23
M4	VITAMIN B-6	12	120	2022-07-01
M5	VITAMIN D3	25	150	2022-06-30
M6	TELMISARTAN	22	115	2022-02-25

(a) Write the degree and cardinality of the table PHARMA.

(b) Identify the attribute best suited to be declared as a primary key.

(c) Anmol has received a new medicine to be added to his stock, but for which he does not know the number of UNITS. So he decides to add the medicine without its value for UNITS. The rest of the values are as follows:

MID	MNAME	PRICE	EXPIRY
M7	SUCRALFATE	17	2022-03-20

Write the SQL command that Anmol should execute to perform the required task.

(d) Anmol wants to change the name of the attribute UNITS to QUANTITY in the table PHARMA. Which of the following commands will he use for the purpose?

I. UPDATE
II. DROP TABLE
III. CREATE TABLE
IV. ALTER TABLE

(e) Now Anmol wants to increase the PRICE of all medicines.Which of the following command will he use for the purpose?

I. UPDATE SET
II. INCREASE BY
III. ALTER TABLE
IV. INSERT INTO

4. For the following SQL Table named PASSENGERS in a database TRAVEL:

TNO	**NAME**	**START**	**END**
T1	RAVI KUMAR	DELHI	MUMBAI
T2	NISHANT JAIN	DELHI	KOLKATA
T3	DEEPAK PRAKASH	MUMBAI	PUNE

A cursor named Cur is created in Python for a connection of a host which contains the database TRAVEL. Write the output for the execution of the following Python statements for the above SQL Table PASSENGERS:

```
Cur.execute('USE TRAVEL')
Cur.execute('SELECT * FROM PASSENGERS')
Recs=Cur.fetchall()
For R in Recs:
    Print(R[1])
```

5. Write SQL statements for the following queries (i) to (v) based on the relations CUSTOMER and TRANSACTION given below:

Table: CUSTOMER

ACNO	**NAME**	**GENDER**	**BALANCE**
C1	RISHABH	M	15000
C2	AAKASH	M	12500
C3	INDIRA	F	9750
C4	TUSHAR	M	14600
C5	ANKITA	F	22000

Table: TRANSACTION

ACNO	TDATE	AMOUNT	TYPE
C1	2020-07-21	1000	DEBIT
C5	2019-12-31	1500	CREDIT
C3	2020-01-01	2000	CREDIT

1. To display all information about the CUSTOMERS whose NAME starts with 'A'.
2. To display the NAME and BALANCE of Female CUSTOMERS (with GENDER as 'F') whose TRANSACTION Date (TDATE) is in the year 2019.
3. To display the total number of CUSTOMERS for each GENDER.
4. To display the CUSTOMER NAME and BALANCE in ascending order of GENDER.
5. To display CUSTOMER NAME and their respective INTEREST for all CUSTOMERS where INTEREST is calculated as 8% of BALANCE.

6. The IT Company XYZ has asked their IT manager Ms. Preeti to maintain the data of all the employees in two tables EMPLOYEE and DEPT. Ms. Preeti has created two tables EMPLOYEE and DEPT. She entered 6 rows in the 'EMPLOYEE' table and 5 rows in the 'DEPT' table.

Table: DEPT

D_CODE	D_NAME	CITY
D001	INFRASTRUCTURE	DELHI
D002	MARKETING	DELHI
D003	MEDIA	MUMBAI
D005	FINANCE	KOLKATA
D004	HUMAN RESOURCE	MUMBAI

Table: EMPLOYEE

E_NO	NAME	DOJ	DOB	GENDER	D_CODE	Salary
1001	Vinay	2013-09-02	1991-09-01	MALE	D001	250000
1002	Ruby	2012-12-11	1990-12-15	FEMALE	D003	270000
1003	Anuj	2013-02-03	1987-09-04	MALE	D005	240000
1007	Sunny	2014-01-17	1988-10-19	MALE	D004	250000
1004	Rohit	2012-12-09	1986-11-14	MALE	D001	270000
1005	Preeti	2013-11-18	1989-03-31	FEMALE	D002	NULL

Note: DOJ refers to the date of joining and DOB refers to the date of Birth of employees.
Based on the above data, answer the following questions:

1. Identify the column which can be considered as the primary key in the 'EMPLOYEE' table.
2. Identify the column which can be considered as primary key in the DEPT table
3. What is the degree and cardinality of the' EMPLOYEE' table?
4. What is the degree and cardinality of the' DEPT' table?
5. **Write SQL queries for the following:**

1) Insert two new rows in the Employee table with the following data:

1006	Rahul	2019-11-06	1992-01-04	MALE	D003	156000
1008	Sonam	2022-01-06	1991-04-06	FEMALE	D005	167000

2) To display E_NO, NAME, GENDER from the table EMPLOYEE in descending order of E_NO.
3) To display the NAME of all the 'FEMALE' employees from the table EMPLOYEE.
4) To display the E_NO and NAME of those employees from the table EMPLOYEE who are born between '1987- 01-01' and '1991-12-01'.
5) To display the NAME and CITY of those employees whose DEPARTMENT is either 'MEDIA' or 'FINANCE'.
6) To display the NAME of those employees whose name starts with the letter 'R'.
7) To display the 'NAME' of those employees whose name contains the letter 'n'.
8) To display the 'NAME' of those employees whose name has exactly 5 letters.
9) To display D_NAME and CITY from table DEPT where D_NAME ends with the letter 'G' and CITY is 'DELHI'.
10) To display the maximum SALARY of the 'EMPLOYEE' table.
11) To delete data of all those employees whose age is less than 25.
12) To update SALARY to 230000 of those employees whose 'E_NO' is 1004.
13) To change the sequence of the DOB column in the employee table and move it before the DOJ column.
14) To add a new column MOBILE int(20) before column SALARY in the employee table.
15) To set SALARY to 300000 of all those employees whose age is NULL.
16) To Increase the salary of all employees by 30000 in the EMPLOYEE table.
17) To display the average SALARY of the EMPLOYEE table.
18) To display the names of employees who have a SALARY of more than 200000 in ascending order of NAME.
19) To display department-wise average salary of employees.
20) To display the total number of departments in XYZ company.
21) To delete data of all the employees whose D_CODE is not 'D001'.
22) To display E_NO, NAME and SALARY of all those employees who don't live in 'DELHI'.
23) To change column name CITY to D_CITY in the DEPT table.
24) To delete the EMPLOYEE table.
25) To delete the 'D_NAME' column from the DEPT table.
26) To display name and city of all the employees.

7. A garment store is considering maintaining its inventory using SQL to store the data. as a database administrator, Mr.Rohit has decided that:
   - Name of the database – **STORE**
   - Name of the table – **GARMENT**
   - The attributes of GARMENT table are as follows:

- **GCODE** – numeric
- **DESCRIPTION** – character of size 50
- **PRICE** – numeric
- **FCODE** – varchar of size 10

Table: GARMENT

GCODE	**DESCRIPTION**	**PRICE**	**FCODE**
10023	JEANS	1150	F01
10001	SHIRT	750	F02
10044	SHORTS	600	F05
10005	TIE	400	F04
10002	JACKET	5000	F01
10022	SOCKS	150	NULL

1. Categorize the following commands as DDL or DML:
   ALTER, INSERT, UPDATE, CREATE, DROP, DELETE
2. Identify the attribute of the Garment table to be declared as the primary key.
3. Write the degree and cardinality of the table GARMENT
4. Write SQL query to create database STORE.
5. Write SQL query to display the list of available databases.
6. Write SQL query to use database STORE.
7. Write SQL query to display the list of available tables in database STORE.
8. Write SQL query to create table GARMENT with aforementioned attributes.
9. Obtain the output of the following SQL queries based on the data given in table GARMENT:
   (i) SELECT MAX(PRICE), MIN(PRICE) FROM GARMENT;
   (ii) SELECT GCODE, DESCRIPTION FROM GARMENT;
   (iii) SELECT FCODE,GCODE FROM GARMENT WHERE PRICE BETWEEN 500 AND 800;
   (iv) SELECT * FROM GARMENT WHERE DESCRIPTION NOT IN ('JEANS','TIE');
   (v) SELECT GCODE FROM GARMENT WHERE DESCRIPTION LIKE '%S%';
   (vi) SELECT GCODE,PRICE FROM GARMENT WHERE DESCRIPTION LIKE '_ _ _';
   (vii) SELECT DISTINCT FCODE FROM GARMENT;
   (viii) SELECT SUM(PRICE) FROM GARMENT;
   (ix) SELECT * FROM GARMENT WHERE DESCRIPTION LIKE '%T%' AND FCODE!='F02';
   (x) SELECT * FROM GARMENT ORDER BY PRICE DESC;
   (xi) SELECT PRICE*10 FROM GARMENT;
   (xii) SELECT COUNT(DISTINCT FCODE) FROM GARMENT;
   (xiii) SELECT * FROM GARMENT WHERE FCODE NOT IN ('F01','F02') AND PRICE<500;
   (xiv) SELECT GCODE, PRICE FROM GARMENT WHERE FCODE IS NULL;
   (xv) SELECT * FROM GARMENT WHERE PRICE >500 AND PRICE <1000;

8. Write the output of the following SQL queries based on table TRANSACTION given below:
Table: TRANSACTION

T_NO	M_NO	AMOUNT	CARD_TYPE	DATE	STATUS
1	11	5000	CREDIT	2019-10-11	SUCCESS
2	11	170	CREDIT	2019-10-14	FAILURE
3	13	800	DEBIT	2019-10-24	FAILURE
4	12	90	CREDIT	2019-11-10	SUCCESS
5	13	1400	DEBIT	2019-11-11	SUCCESS
6	11	500	DEBIT	2019-11-18	SUCCESS
7	13	1600	DEBIT	2019-11-27	FAILURE

Table: COMPANY

T_NO	QTY_ISSUED	COMPANY
1	15	SBI
3	50	ICICI
4	34	HDFC

(i) SELECT M_NO, MIN(AMOUNT) FROM TRANSACTION GROUP BY M_NO HAVING COUNT(*)>2;

(ii) SELECT T_NO,AMOUNT FROM TRANSACTION WHERE CARD_TYPE='CREDIT';

(iii) SELECT * FROM TRANSACTION WHERE M_NO>12 AND STATUS='FAILURE';

(iv) SELECT CARD_TYPE, SUM(AMOUNT) FROM TRANSACTION GROUP BY CARD_TYPE;

(v) SELECT T_NO, AMOUNT*10 AS PAYMENT FROM TRANSACTION WHERE STATUS IN ('SUCCESS');

(vi) SELECT COUNT(*) FROM TRANSACTION WHERE STATUS='SUCCESS';

(vii) SELECT T_NO, AMOUNT, CARD_TYPE FROM TRANSACTION WHERE T_NO>5;

(viii) SELECT DISTINCT M_NO FROM TRANSACTION;

(ix) SELECT T.T_NO, M_NO,CARD_TYPE, COMPANY FROM TRANSACTION T, COMPANY C WHERE T.T_NO=C.T_NO AND AMOUNT>1000;

(x) SELECT T.T_NO AS TRANS_NO FROM TRANSACTION T, COMPANY C WHERE T.T_NO=C.T_NO;

9. The code given below reads the records from the table employee and displays only those records of employee table who don't live in the city 'Delhi':

E_ID – varchar(50)
E_Name – char(50)
Salary – int(15)
City – char(20)

Note the following to establish connectivity between Python and MySQL:

- **Username** is root

- **Password** is 12345
- The table exists in a MySQL database named *company*.
- The table has four attributes (E_ID, E_Name, Salary, City).

Write the following statements to complete the code:
Statement 1- to import the desired library.
Statement 2- to create a cursor
Statement 3- to write a SQL query that fetches records of all those employees who don't live in Delhi.
Statement 4- to execute the query
Statement 5- to print all the records fetched in statement 4

```
Import _________ as mysql                                #Statement 1
def print():
    a=mysql.connect (host="localhost", user "root",passwd=12345, database= "company")
    b=____________                                        #Statement 2
    query = ________________                              #Statement 3
    c = ___________________                               #Statement 4
    for z in c:
        _________                                         #Statement 5
```

10. The code given below deletes the records from the table employee which contains the following record structure:

E_ID – varchar(50)

E_Name – char(50)

Salary – int(15)

City – char(20)

Note the following to establish connectivity between Python and MySQL:

- **Username** is root
- **Password** is 12345
- The table exists in a MySQL database named *company*.
- The table has four attributes (E_ID, E_Name, Salary, City).

Write the following statements to complete the code:
Statement 1- to import the desired library.
Statement 2- to connect to the database
Statement 3- to create a cursor
Statement 4- to write a query that deletes the record with E_ID='A101' and E_Name starts with the letter 'D'.
Statement 5- to remove the record permanently from the database.

```
Import _________ as mysql                              #Statement 1
def del():
    a=_________ (host="localhost", user "root",passwd=12345, database= "company")
                                                        #Statement 2
    b=____________                                      #Statement 3
    query = ________________                            #Statement 4
    b.execute(query)
    ___________________                                 #Statement 5
    print('Data deleted')
```

11. Write a Python program to delete all the tuples from the Student table whose age>14.
Note the following to establish connectivity between Python and MySQL:
Username is root
Password is 12345
The table exists in a MySQL database named *School*.
The table has five attributes (Stu_ID, Stu_Name, age, Address)

12. Write a Python program to insert 5 records in the Employee table. Take these 5 records as input from the user (One record at a time).
Note the following to establish connectivity between Python and MySQL:
Username is root
Password is 12345
The table exists in a MySQL database named *company*.
The table has five attributes (Emp_ID, Emp_Name, DOJ, Gender, Salary)

13. Write a Python program to update the name of an employee in the Employee table whose employee ID is ‘E1001’ (Take updated name as input from the user).
Note the following to establish connectivity between Python and MySQL:
Username is root
Password is 12345
The table exists in a MySQL database named *company*.
The table has five attributes (Emp_ID, Emp_Name, DOJ, Gender, Salary)

CLASS: XII SESSION: 2024-25

COMPUTER SCIENCE (083)

PRACTICE SET

Time allowed: 3 Hours ***Maximum Marks: 70***

General Instructions:

- Please check this question paper contains 35 questions.
- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A, consists of 18 questions (1 to 18). Each question carries 1 Mark.
- Section B, consists of 7 questions (19 to 25). Each question carries 2 Marks.
- Section C, consists of 5 questions (26 to 30). Each question carries 3 Marks.
- Section D, consists of 2 questions (31 to 32). Each question carries 4 Marks.
- Section E, consists of 3 questions (33 to 35). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.

QUES NO.	QUESTION	MARKS
	SECTION A	
1	State True or False: In MySQL, the GROUP BY clause is used to combine all such records of a table which have identical values in specified field(s).	1
2.	Which of the following statements is not valid for CSV files? a. CSV files are the text files. b. The newline argument suppresses blank rows in the file. c. By default, EOL takes place while writing data into the file. d. None of the above.	1
3.	What will be the value of **p** when **p=int(17/2+11.5)**? 20 b. 20.0 c. 28 d. 28.0	1
4.	________error occurs when an identifier used in the expression does not find its value during run time. a. ValueError b. TypeError c. NameError d. IdentifierError	1
5.	To work with MySQL, you need to enter a password, known as___________ a. Root password b. System Password c. MySQL Password d. Data Language Password	1

6.	Physical or logical arrangement of network refers to as: a. Routing b. Looping c. Topology d. Networking	1
7.	Write the output of the code given below: T= (12,72,34,76,21,89,4,83,11,2) RESULT = list(T [-4:]) print(RESULT) a. [4, 83, 11, 2] b. (89,4, 83, 11) c. (4,83,11,2) d. [89,4,83,11]	1
8.	Find the output of the following code: String = "Python Programming" String.isalnum() (a) True (b) False (c) TypeError (d) None of these	1
9.	What is output of following code segment T= min(1,6,0,-3,100) (a) 1 (b) -3 (c) 0 (d) None of above	1
10.	What is the purpose of the random module? a.) To generate random numbers for cryptography. b.) To generate random numbers for statistical analysis c.) To generate random numbers for various applications. d.) To generate truly random numbers for secure applications.	1
11.	The internet facility that facilitates remote login is: a. HTTP b. FTP c. TELNET d. LAN	1
12.	A____________ is a block of organized and reusable code. a. Parameter b. Argument c. Definition d. Function	1
13.	Stack is a linear data structure which implements? a. Last in First out b. Last in Last out c. First in First out d. None of the above	1
14.	Which of the following TCP/IP protocols is used for transferring files from one computer to another? a. FTP b. PPP c. SMTP d. HTTP	1
15.	Which of the following statement is correct for 'wb+' mode? a. It enables writing and reading operations.	1

b. It disables writing but enables reading operations.
c. It enables reading followed by writing operations.
d. None of the above

16. Which of the following function is used with the csv modules in Python to read the content of a csv file into an object? 1

a. readrow()   b. readrows()   c. reader()   d. load()

Q17 and 18 are ASSERTION (A) and REASONING (R) based questions. Mark the correct choice as

(a) Both A and R are true and R is the correct explanation for A
(b) Both A and R are true and R is not the correct explanation for A
(c) A is True but R is False.
(d) A is false but R is True

17. **Assertion (A):** The key concept of data structure is to manage the storage of data in the memory efficiently. 1
Reason (R): The data structure is a way to create a logical structure in the memory so that the storage of numerous data values could be managed using minimum space. It helps in a fast accessing of data during execution of a program.

18. **Assertion(A):** Every open file maintains a file-pointer and keeps track of its position after every operation. 1
Reason (R): Every read and write operation takes place at the current position of the file pointer.

SECTION B

19. Give two advantages and two disadvantages of star topology 2

OR

Define the following terms:
www, **web hosting**

20. What will be the flow of execution of the following code? 2

```
1  #function definition
2  def Increment(X):
3       X += 1
4  X = 3
5   print(X)
6   Increment(X)
7   print(X)
```

21. Explain the Scope of a variable in Python with example. 2

OR

Differentiate between actual parameter(s) and formal parameter(s) with a suitable example for each

22. Write a function Name(N) in Python, that accepts a list, N, of names and display the names starting with letter 'A'. For example, if the list is 2

N = ['Ashima', 'Neha', 'Rahul', 'Ishaan', 'Ashu', 'Kirti']

Then the output should be:

Ashima

Ashu

OR

Write definition of a method SumEvenOdd(K) to display the sum of Odd and Even values separately from the list K. For example, if list K is defined as:
K = [14,25,16,23,12,9,7,23]
Then the output should be:
Sum of Even numbers:42

```
Sum of Odd numbers:87
```

23. Categorize following commands into DDL and DML commands?

INSERT INTO, DROP TABLE, ALTER TABLE,UPDATE...SET

2

24. Differentiate between char(n) and varchar(n) data types with respect of databases.

2

25. Predict the Output for the following code:

```
x = [1,2,3,4,5,6,7,8,9]
a=[]
n=len(x)
for i in range(n):
   if x[i]%2 == 0:
      a.append(x[i]/2)
   else:
      a.append(x[i]*2)
print("New List is: ",a)
```

OR

Predict the output for the following code:

```
L1 = [10,20,30]
L2 = [110, 220, 330]
L3 = L1+L2
L4 = L3[0:4]
print(L4)
L4[0] = L4[0]*10
L4[2] = L4[1]*5
L4[1] = L4[2]
L4[3] = L4[3] - 10
print(L4)
```

2

SECTION C

26. Charu has to create a database named **MYEARTH** in MYSQL.
She now needs to create a table named **CITY** in the database to storethe records of various cities across the globe. The table **CITY** has the following structure:

Table: CITY

FIELD NAME	DATA TYPE	REMARKS
CityCode	CHAR(5)	Primary Key
CityName	VARCHAR(20)	
Size	Integer	
AvgTemp	Integer	
PollutionRate	Integer	
Population	Integer	

3

27. Write a function in Python with function call to count the number of lines in a text file 'Detail.txt' which is starting with alphabet 'T'.

OR

Write a function in Python with function call to read the text file "Data.txt" and count the number of times 'my' occurs in the file.

3

28.
a. What is meant by pickling and unpickling of files in Python?
b. Difference between r+ and w+ modes of text files?

1+1+1

c. Name the modules required for working on text files, binary files and csv files in Python.

29. 1. Expand the following: SMTP, XML

2. Out of the following, which is the fastest wired and wireless mediumof transmission?

Infrared, coaxial cable, optical fibre,microwave, Ethernet cable

3. How is packet switching different from circuit switching?

1+1+1

30. Write a program in Python to implement a stack for these book details: (bookno, bookname), i.e. Each node of the stack should contain two information about a book, book number and book name. Also provide the option for user if he/she wants to view all records of stack or only last added record.

3

SECTION D

31. Ashok is a Python programmer who has written a code and created a binary file 'emp.dat' with employee id, name and salary . The file contains 10 records. He wants to update a record based on the employee id entered by user and update the salary. The updated record is then to be written in the file 'temp.dat' along with the records which are not to be updated. An appropriate message is displayed if employee id is not found. As a programmer, help him to complete the following code:

4

```
import_______________                    #Statement 1
def update_data():
    rec={}
    f = open ("emp.dat", "rb")
    t = open ("_______________")
    found = false
        p_id = int(input("Enter employee id to update record : "))
    while True:
    try:
        rec = ______________________                #Statement 3
                                    _id:
            found= True

            pickle.___________________               #Statement 4
        else:
            pickle.dump(rec,t)
    except:
        break
if found == True:
    print("The salary is updated.")
else:
    print("Employee id not found!!")
f.close()
t.close()
```

1. Which module should be imported in the program? **(Statement 1)**

2. Write the correct statement required to open a temporary file named 'temp.dat' **(Statement 2)**

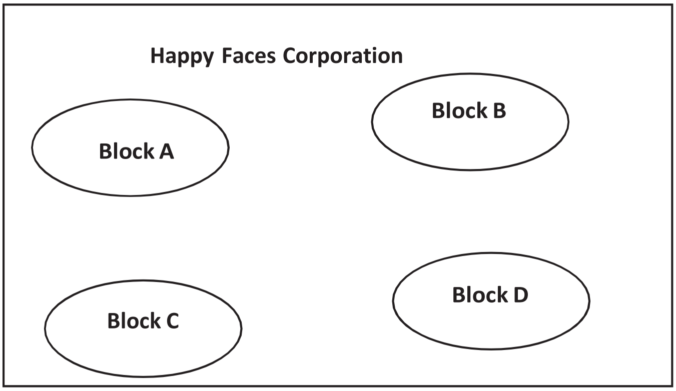
3. Which statement should Ashok fill in **Statement 3** to read the data from the binary file?

4. Write appropriate code to update data in the file. **(Statement 4)**

32. Write a program in Python to get student details (Rollno, Name and marks) for multiple students from the user and create a CSV file by writing all the student details in one go. Also read and display the data of CSV file. 4

SECTION E

33. Happy Faces Corporation has set up its new centre at Noida, Uttar Pradesh for its office and web-based activities. It has 4 blocks ofbuildings. 5



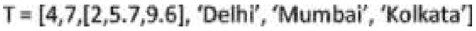
Distance between the various blocks is as follows:A to B 40 m

B to C	120m
C to D	100m
A to D	170m
B to D	150m
A to C	70m

Numbers of computers in each block
Block A -25
Block B - 50
Block C - 125
Block D - 10

a. Suggest and draw the cable layout to efficiently connectvarious blocks of buildings within the Noida centre for connecting the digital devices.
b. Which block will be best suitable to place the server and why?
c. Suggest the placement of the following device withjustification
    a. Repeater b. Hub/Switch
d. Which kind of network (PAN/LAN/WAN) will be formed if theNoida office is connected to its head office in Mumbai?
e. Which fast and very effective wireless transmission mediumshould preferably be used to connect the head office at Mumbai with the centre at Noida?

34. Read the following passage and answer the questions that follow: 5
Lists are the container which store the heterogeneous elements of any type as integer, character, floating that store in square brackets [].
Consider the following list:


T = [4,7,[2,5.7,9.6], 'Delhi', 'Mumbai', 'Kolkata']

1. To print Value 9.6, the correct python statement will be:
    a. L [2][2] b. L [2][-1] c. L [-4][-1] d. All of the above
2. The length of L is:

a. 6 b. 8 c. 18 d. None of these

3. The output for statement len(L[1]) will be:

   a. 1 b. 3 c. Exception d. Syntax Error

4. print(L[:2] will generate value:

   a. [4] b. [4,7] c. [4,7,[2,5.7,9.6]] d. Error

5. L.append[4,5,6] will generate output:

   a. Error

   b. Append a sublist [4,5,6] into L

   c. Append three elements in list 4, 5 and 6

   d. None of these

OR

Read the following passage and answer the questions that follows:

Lists are the container which store the heterogeneous elements of any type like integer, character or floating point that store in square brackets[].

Consider the following list for Python:

D = ["Raja", 4.87, 'Delhi', 'Kerala', [23, 7.56, 'Neha'], 98]

What will the following statements do as per above definition of D:

1. D [4][-3]
2. len(D[3])
3. d [2] = 54
4. print (D[0] + D[1])
5. print (D[1:5:3]

35. Ami has joined as teaching assistant. She is working with data files using Python. She is currently working on the following incomplete code: 5

```
Color= ['Red', 'Orange', 'Crimson', 'Magenta', 'Fuschia', 'Blue', 'Black']
def LenColors(________,_______):              #Statement 1
   with________________________:              #Statement 2
       for c in Color:
           ___________________________:        #Statement 3
           ______________________________      #Statement 4
Fname ='Colors.txt'
N = int(input("Enter a number (between 3 to 7): "))
______________________________                   #Statement 5
FILE = open(Fname)
print(File.read())
```

Help her complete the code as per instruction given below:

a. Complete **Statement 1** and **Statement 5** so that Statement 5 invokes the function LenColors() by passing arguments—the filename and the integer value n.

b. **Statement 2**: Open the file for writing using a with statement in the file object namely F.

c. Inside the loop, write code so that only those colors from the color list get written in separate lines on the file that have word length more than n(the second parameter received) **(Statement 3 and Statement 4)**

d. What will be the output produced, if the value of n is entered as 6?

OR

CLASS: XII SESSION: 2024-25

COMPUTER SCIENCE (083)

PRACTICE SET

Time allowed: 3 Hours ***Maximum Marks: 70***

General Instructions:

- Please check this question paper contains 35 questions.
- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A, consists of 18 questions (1 to 18). Each question carries 1 Mark.
- Section B, consists of 7 questions (19 to 25). Each question carries 2 Marks.
- Section C, consists of 5 questions (26 to 30). Each question carries 3 Marks.
- Section D, consists of 2 questions (31 to 32). Each question carries 4 Marks.
- Section E, consists of 3 questions (33 to 35). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.

Q No	Question	Marks
	Section A	
1	State True or False: CSV file is also known as delimited text file which can be opened in Notepad and Spreadsheet both.	1

2	The query to display all the tables stored in a database is:
	a. Display Tables;
	b. Show Tables;
	c. All Tables;
	d. Show tables in <database name>;
		1

3	Consider the following expression: True or not False and False and not True Which of the following will be correct output if the given expression is evaluated? True b. False c. None d. Null	1
4	______________ function splits the given string based on delimiter and store the result in the form of list. partition() b. split() c. divide() d. clear()	1
5	In a table there can be more than one attribute which contains unique values. These columns are known as ____________________ Primary Key b. Alternate Key c. Candidate Key	1
6	A device that amplifies a signal being transmitted on the network is known as:	1

	a. Modem b. Repeater c. Hub d. Switch	
7	Write the output of the code given below: dict = {"stname": "Ajay", "age": 17} dict['age'] = 27 dict['address'] = "Bikaner" print(dict)	1
8	Find the output of the following code number= [1,5,7,0,4] print(number[2:3]) (a) [5] (b) [7] (c) [7,0] (d) None of above	1
9	What is output of following code segment T= min(1,6,0,-3,100) (a) 1 (b) -3 (c) 0 (d) None of above	1
10	What is the purpose of the random module? a.) To generate random numbers for cryptography. b.) To generate random numbers for statistical analysis c.) To generate random numbers for various applications. To generate truly random numbers for secure applications .	1
11	The internet facility that facilitates remote login is: HTTP b. FTP c. TELNET d. LAN	1
12	A ______________ is a block of organized and reusable code. a. Parameter b. Argument c. Definitio n d. Function	1
13	When will the else part of try-except-else be executed? a. Always b. When an exception occurs c. When no exception occurs When an exception occurs in to except block	1
14	Which of the following is not an aggregate function? AVG b. MAX c. JOIN d. COUNT	1
15	Which of the following transmission media has the highest bandwidth? a. Co-axial cable b. Fiber optic cable c. Twisted pair cable d. None of these	1

<table>
<tr><td>16</td><td>Which of the following function is used with the csv modules in Python to read the content of a csv file into an object?
readrow() b. readrows() c. reader() d. load()</td><td>1</td></tr>
<tr><td></td><td>Q17 and 18 are ASSERTION (A) and REASONING (R) based questions. Mark the correct choice as
(a) Both A and R are true and R is the correct explanation for A
(b) Both A and R are true and R is not the correct explanation for A
(c) A is True but R is False.
A is false but R is True</td><td></td></tr>
<tr><td>17</td><td><b>Assertion:</b> The order of execution of the statements in a program is known as flow of control.
<b>Reason:</b> The flow of control can be implemented using control structures.</td><td>1</td></tr>
<tr><td>18</td><td><b>Assertion:</b> Keyword arguments are related to the function calls.
<b>Reason:</b> When you use keyword arguments in a function call, the caller identifies the arguments by the parameter name.</td><td>1</td></tr>
<tr><td colspan="3"><b>Section B</b></td></tr>
<tr><td>19</td><td>Name the protocol:
1. Used to transfer voice using packet switched network
Used for chatting between two groups or between two individuals</td><td>2</td></tr>
<tr><td>20</td><td>What will be the output of the following code?
def Func1(x):
    try:
        print(5/x)
    except:
        print("Error...")</td><td>2</td></tr>
<tr><td>21</td><td>What is the difference between parameters and arguments?</td><td>2</td></tr>
<tr><td>22</td><td>Define Stack. Give any two applications of Stacks</td><td>2</td></tr>
<tr><td>23</td><td>Write a Python function that takes two lists and returns True if they have at least one common member.</td><td>2</td></tr>
<tr><td>24</td><td>Write the code in Python to create a table Student in database School with following fields:
<table>
<tr><th>Fields</th><th>Datatype</th></tr>
<tr><td>Student_id</td><td>varchar(6)</td></tr>
<tr><td>S_name</td><td>varchar(20)</td></tr>
<tr><td>Class</td><td>int(4)</td></tr>
<tr><td>Section</td><td>char(2)</td></tr>
</table></td><td>2</td></tr>
<tr><td>25</td><td>What will be the output of the following program?
a. def INCREASE(x):
        a = a + x
        return
    a = 15</td><td>2</td></tr>
</table>

```
    b = 10
    increase(b)
    print(a)
```

b.
```
    z = 1
    def FUNC():
        z = 10
    print(z)
```

Section C

26 How can we import a module in Python? What are the possible outcomes(s) executed from the following code? Also specify the maximum and minimum values that can be assigned to variable FLOWER. 3

```
FLOWER = random.randint(0,3)
NAME = ["Rose", "Lily", "Tulip", "Marigold"]
for x in NAME:
    for y in range (1, FLOWER):
        print(x, end= " ")
    print()
```

27 Write the output of the queries (a) to (c) based on the table, Furniture given below. 3

FID	Name	Dt_of_Purchase	Cost	Disc.
B001	Double Bed	03-Jan-2018	45000	10
T010	Dinning Tabel	10-Mar-2020	51000	5
B004	Single Bed	19-Jul-2016	22000	0
C003	Long Back Chair	30-Dec-2016	12000	3
T006	Console Table	17-Nov-2019	15000	12
B006	Bunk Bed	01-Jan-2021	28000	14

i.) SELECT SUM(DISCOUNT) FROM FURNITURE WHERE COST > 15000;

ii.) SELECT MAX(Dt_of_Purchase) FROM FURNITURE;

iii.) SELECT * FROM FURNITURE WHERE DISCOUNT > 5 AND FID LIKE "T%";

28 Write a program that uses a function which takes to string arguments and returns the string comparison result of the two passed strings. 3

29 Consider the table Personal given below: 3

P_ID	Name	Desig	Salary	Allowance
P01	Rohit	Manager	89000	4800
P02	Kashish	Clerk	NULL	NULL
P03	Mahesh	Supervisor	48000	NULL

P04	Salil	Clerk	31000	1900
P05	Ravina	Supervisor	NULl	2100

Based on the given table, write SQL queries for the following:

a. Increase the salary by 10% of persons whose allowance is given.

b. Display Name and Total salary (sun of salary and allowance) of all persons. The column heading 'TOTAL SALARY' should also be displayed.

c. Delete the record of Supervisors who have salary greater than 25000.

30 A list, NList, contains following record as list elements: 3

[City, Country, distance from Delhi]

Each of these records are nested together to form a nested list. Write the following user defined functions in Python to perform the specified operations on the stack named travel.

i. **Push_element(NList):** It takes the nested list as an argument and pushes a list object containing name of the city and country, which are not in India and distance is less than 3500 km from Delhi.

Pop_element(): It pops the objects from the stack and displays them. Also, the function should display "Stack Empty" when there are no elements in the stack.

Section D

31 Anmol maintain that database of medicines for his pharmacy using SQL to store the data. 4

The structure of the table PHARMA for the purpose is as fo llows:

- Name of the table -PHARMA
- The attributes of PHARMA are as fo llows:

MID - numeric
MNAME - character of size 20
PRICE - numeric
UNITS - numeric
EXPIRY – date

Table: PHARMA

MID	MNAME	PRICE	UNITS	EXPIRY
M1	PARACETAMOL	12	120	2022-12-25
M2	CETRIZINE	6	125	2022-10-12

M3	METFORMIN	14	150	2022-05-23
M4	VITAMIN B-6	12	120	2022-07-01
M5	VITAMIN D3	25	150	2022-06-30
M6	TELMISARTAN	22	115	2022-02-25

(a) Identify the attribute best suitable to be declared as a primary key.

(b) Anmol has received a new medicine to be added into his stock, but for which he does not know the number of UNITS. So, he decides to add the medicine without its value for UNITS. The rest of the values are as follows:

MID	MNAME	PRICE	EXPIRY
M7	SUCRALFATE	17	2022-03-20

Write the SQL command which Anmol should execute to perform the required task.

(c) Anmol wants to change the name of the attribute UNITS to QUANTITY in the table PHARMA. Which of the following commands will he use for the purpose?

I. UPDATE
II. DROP TABLE
III. CREATE TABLE
IV. ALTER TABLE

(d) Now Anmol wants to increase the PRICE of all medicines following commands will he use for the purpose?

32 — 4

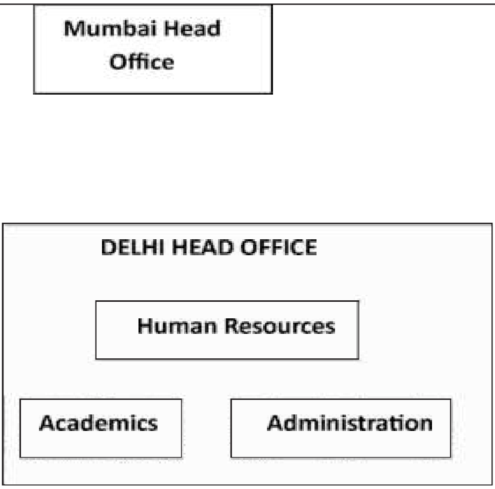
Write a program in Python to get student details (Rollno, Name and marks) for multiple students from the user and create a CSV file by writing all the student details in one go. Also read and display the data of CSV file.

33

Section E

VishInternational Inc. is planning to connect its Bengaluru office setup with its Head Office inDelhi. The Bengaluru Office G.R.K. International Inc. is spread across an area of approx. 1 Square Killometer, consisting of 3 blocks : Human Resource, Academic and Administration. — 5

You as a network expert have to suggest answers to the five quaries (i) to (v) raised by them.



Shortest distances between various Blocks :

Human Resources to Administration	200 m
Human Resources to Academics	60 m
Academics to Administration	150 m
Delhi Head Office to Bangaluru office Setup	2000 km

The Number of Computers installed at various blocks are as follows:

BLOCK	No. of Computers
Human Resources	55
Administration	60
Academics	200
Delhi Head Office	30

(i) Suggest the network type (out of LAN, MAN, WAN) for connecting each of the followingset of their Offices:
Mumbai Head office and Administration Office
Human Resources Office and Academics office
ii) Which device will you suggest to be procured by the company for connecting all thecomputers within each of their offices out of the following devices?
MODEM (b) Switch/Hub (c) Repeater
iii) Suggest a cable /wiring layout among the various blocks within the Delhi OfficeSetup for connecting the Blocks.
iv) Suggest the most suitable in the Delhi Office Setup, to host the Server. Give a suitablereason with your suggestion.

v) Suggest the most suitable media to provide secure, fast and reliable data connectivity between Mumbai Head Office and the Delhi Office

Setup.

34 Write any two differences between text and binary files. 5

Write a program in python with function **LineRead()**, which reads the contents of a text file "***Memory.txt***" and displays those lines from the file which have at least 10 words in it.

For example, if the content of "Memory.txt" is as follows:

Five cows lived in a little forest.
They ate fresh grass in a large green meadow.
They were kind friends.
They decided to do everything together, so the lions couldn't attack them for food.
One day, five cows fought and each one started to eat grass in a different place.
The lions decided to seize the opportunity and killed them one by one

Then the output should be,

They decided to do everything together, so the lions couldn't attack them for food.
One day, five cows fought and each one started to eat grass in a different place.
The lions decided to seize the opportunity and killed them one by one

OR

Can a function return multiple values? If yes then explain with suitable example.

A Binary file ***Book.dat*** has structure [BookNo, BookName, Author]

1. Write a user defined function FileData() to input data for a record and add to book.dat.

Write a function CountData(Author) in Python which accepts the AuthorName as parameter and count and return number of books by the given author are stored in the binary file *Book.dat*.

35 Consider the following DEPT and EMPLOYEE tables. 5

Table: DEPT

D_CODE	D_NAME	CITY
D001	INFRASTRUCTURE	DELHI
D002	MARKETING	DELHI
D003	MEDIA	MUMBAI
D005	FINANCE	KOLKATA
D004	HUMAN RESOURCE	MUMBAI

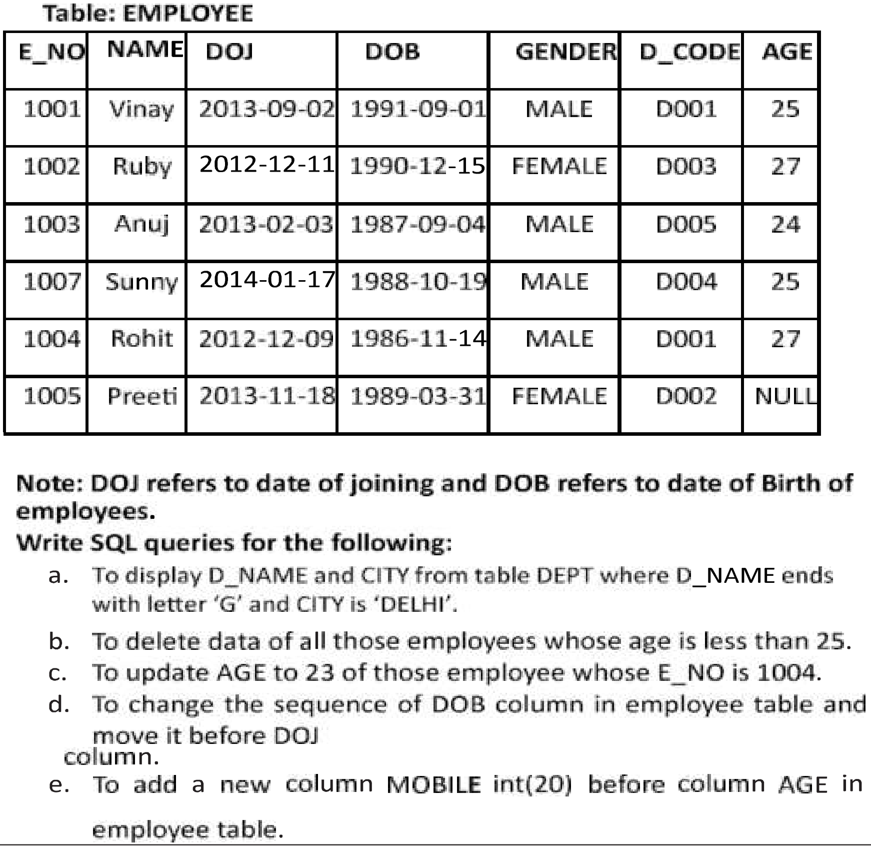


Table: EMPLOYEE

E_NO	NAME	DOJ	DOB	GENDER	D_CODE	AGE
1001	Vinay	2013-09-02	1991-09-01	MALE	D001	25
1002	Ruby	2012-12-11	1990-12-15	FEMALE	D003	27
1003	Anuj	2013-02-03	1987-09-04	MALE	D005	24
1007	Sunny	2014-01-17	1988-10-19	MALE	D004	25
1004	Rohit	2012-12-09	1986-11-14	MALE	D001	27
1005	Preeti	2013-11-18	1989-03-31	FEMALE	D002	NULL

Note: DOJ refers to date of joining and DOB refers to date of Birth of employees.

Write SQL queries for the following:

a. To display D_NAME and CITY from table DEPT where D_NAME ends with letter 'G' and CITY is 'DELHI'.

b. To delete data of all those employees whose age is less than 25.

c. To update AGE to 23 of those employee whose E_NO is 1004.

d. To change the sequence of DOB column in employee table and move it before DOJ column.

e. To add a new column MOBILE int(20) before column AGE in employee table.

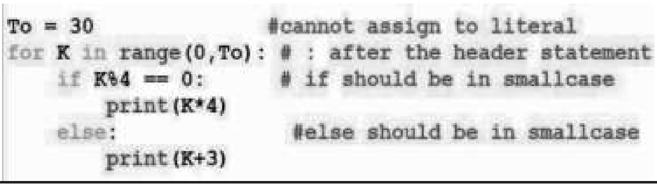
ANSWER KEY

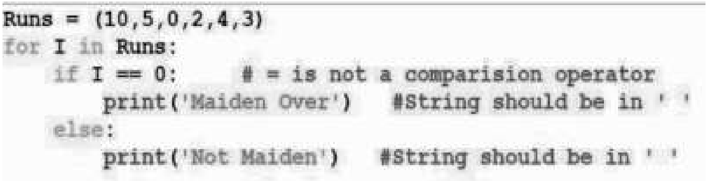
Chapter 1

PART A

1. d)	2. b)	3. c)	4. b)	5. True	6. a)
7. c)	8. True	9. b)	10. b)	11. b)	12. a)
13. c)	14.d)	15. d)	16. b)	17. c)	18. c)
19. c)	20. b)	21. c)	22. a)	23. b)	24. d)
25. d)	26. puterSc	27. d)	28. b)	29. ['Rajat','Rajan',‘Ashish']	30. d)
31. c)	32. a= 12 b= 4 a= 14 b= 2.0	33. d)	34. d)	35. a)	36. dict={1:”Monday,2:”Tuesday”,3:”Wednesday”}
37. d)		38. a)	39. c)		

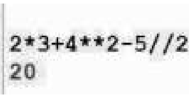
PART B

1	 `To = 30 #cannot assign to literal`
	`for K in range(0,To): # : after the header statement`
	`if K%4 == 0: # if should be in smallcase`
	`print(K*4)`
	`else: #else should be in smallcase`
	`print(K+3)`

2	 `Runs = (10,5,0,2,4,3)`
	`for I in Runs:`
	`if I == 0: # = is not a comparision operator`
	`print('Maiden Over') #String should be in ' '`
	`else:`
	`print('Not Maiden') #String should be in ' '`

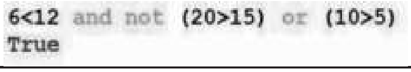
3

a.



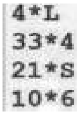
```
2*3+4**2-5//2
20
```

b.



```
6<12 and not (20>15) or (10>5)
True
```

4

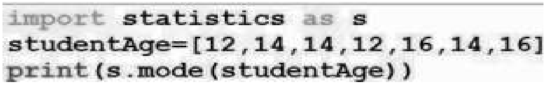


```
4*L
33*4
21*S
10*6
```

5

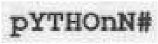
(i) L1.insert(3,200)
(ii) message.endswith('.')

6



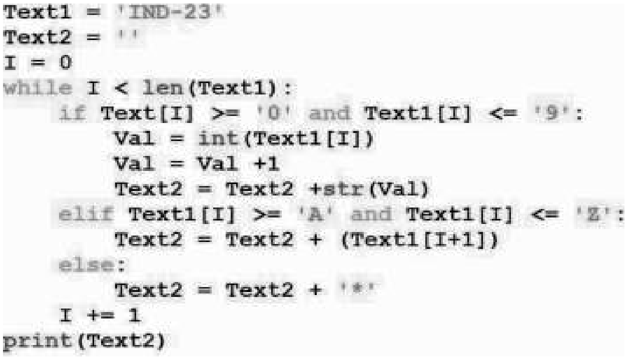
```
import statistics as s
studentAge=[12,14,14,12,16,14,16]
print(s.mode(studentAge))
```

7



```
pYTHOnN#
```

8

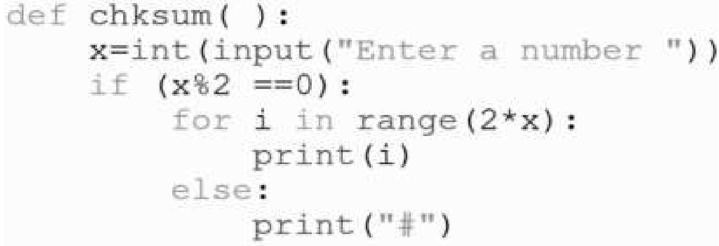


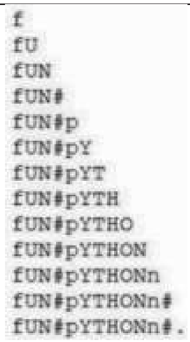
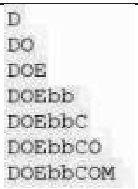
```
Text1 = 'IND-23'
Text2 = ''
I = 0
while I < len(Text1):
    if Text[I] >= '0' and Text1[I] <= '9':
        Val = int(Text1[I])
        Val = Val +1
        Text2 = Text2 +str(Val)
    elif Text1[I] >= 'A' and Text1[I] <= 'Z':
        Text2 = Text2 + (Text1[I+1])
    else:
        Text2 = Text2 + '*'
    I += 1
print(Text2)
```

Chapter 2:Functions

1. b	2. b	3. a	4. d	5. a	6. a	7. a	8. d
9. d	10. b	11. c	12. a	13. d	14. c	15. c	16. b
17. d	18. b	19. d	20. c	21. d	22. b	23. b	24. d
25. b	26. b	27. b	28. d	29. b	30. (i). b	30. (ii) b	30. (iii). c
31. (i). a	31. (ii) a	31. (iii) d	31.(iv) c	31 (v) a			

32.	None
33.	50 @ 50 # 75 @ 75 #
34.	Inside func1: 10 Outside any function: 10 After func2: 20
35.	10.0$20

36.	
	def chksum( ):
	    x=int(input("Enter a number "))
	    if (x%2 ==0):
	        for i in range(2*x):
	            print(i)
	        else:
	            print("#")

37.	250 300
38.	50#5
39.	['CSCS', 'HINDIHINDI', 'PHYSICSPHYSICS', 'CHEMISTRYCHEMISTRY', 'MATHSMATHS']
40. i.	65#70@
40. ii.	5#8#5#4#
41.	3 5 5
42.	 f fU fUN fUN# fUN#p fUN#pY fUN#pYT fUN#pYTH fUN#pYTHO fUN#pYTHON fUN#pYTHONn fUN#pYTHONn# fUN#pYTHONn#.
43. i.	 D DO DOE DOEbb DOEbbC DOEbbCO DOEbbCOM
43. ii.	250 # -150 250 # 100 130 # -100
44	190
45. i.	27.0
45. ii.	78.53981633974483

45. iii.	9
45. iv.	26
45. v.	30
45. vi.	1
46.	-1
47. i.	3

47. ii.	3 3
	3 2
	5 3

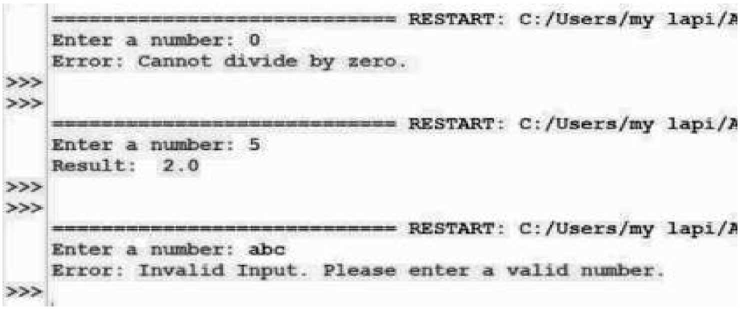
47. iii.	Hello, Anuj None
47. iv.	5
48.	20#50@20
49.	10.0 Division Successful You can't divide any number by Zero Division Successful
50. i.	17 4 17 19 19
50. ii.	0 @6 @7 @8 @9 @25 @11 @ 0 @6 @7 @8 @9 @ 0 @6 @7 @
50. iii.	Hi
50. iv.	150

Chapter 3: EXCEPTION HANDLING

PART A

1. (c)	2. (c)	3. (a)	4. (c)	5. (a)	6. (a)	7. (a)	8. (d)	9. (a)

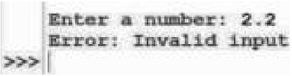
10.



```
    ============================== RESTART: C:/Users/my lapi/A
    Enter a number: 0
    Error: Cannot divide by zero.
>>>
>>>
    ============================== RESTART: C:/Users/my lapi/A
    Enter a number: 5
    Result:  2.0
>>>
>>>
    ============================== RESTART: C:/Users/my lapi/A
    Enter a number: abc
    Error: Invalid Input. Please enter a valid number.
>>>
```

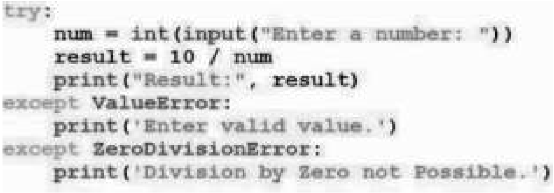
11. TRUE

12.



```
Enter a number: 2.2
Error: Invalid input
>>>
```

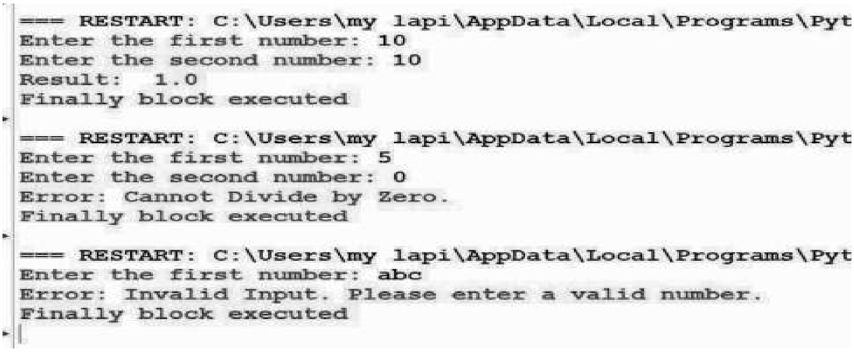
13.



```
try:
    num = int(input("Enter a number: "))
    result = 10 / num
    print("Result:", result)
except ValueError:
    print('Enter valid value.')
except ZeroDivisionError:
    print('Division by Zero not Possible.')
```

14. (c) Type Error

15.



```
=== RESTART: C:\Users\my lapi\AppData\Local\Programs\Pyt
Enter the first number: 10
Enter the second number: 10
Result:  1.0
Finally block executed

=== RESTART: C:\Users\my lapi\AppData\Local\Programs\Pyt
Enter the first number: 5
Enter the second number: 0
Error: Cannot Divide by Zero.
Finally block executed

=== RESTART: C:\Users\my lapi\AppData\Local\Programs\Pyt
Enter the first number: abc
Error: Invalid Input. Please enter a valid number.
Finally block executed
```

16. (c)

17. (c)

Chapter 4: File Handling

1.a	2.a	3. b	4.a	5.d	6.c	7.a	8.b
9.b	10.c	11.a	12.d	13.a	14.c	15.a	16.c
17.c	18.c	19.a	20.b	21.a	22.c	23.d	24.b
25.a	26.a	27.a	28.b	29.a	30.b	31.a	32.a
33.d	34.d	35.b	36.a	37.c	37.d	39 .c	40 .b
41.c	42.b	43.b	44.a	45.a	46.b		

Short answer questions (2marks)

1.	27
2.	3

3.	12
4.	5
5.	2
6.	100
7.	12
8.	7
9.	pickle.dump(tup1, myfile)
10.	3
11.	i. Text file ii. File.write(“ABC”)
12.	i. Text file ii. content = File.read() # Blank1

Assertion and reasoning based questions

1.	(a) Both A and R are true and R is the correct explanation for A.
2.	(a) Both A and R are true and R is the correct explanation for A.
3.	(a) Both A and R are true and R is the correct explanation for A.
4.	(d) A is false but R is True
5.	(d) A is false but R is True
6.	(c) A is True but R is False

Long answer type questions

1.	a. csv b. ’w’or ’w+’ or ‘a’or ‘a+’ c. reader d. close()
2.	1. import csv #line1 2. f=open(“Teachers.csv”,”w”) #line2 3. f=open(“Teachers.csv”,”r”) #line3 4. csv.reader() #line4 5. f.close() #line5

3.	a. csv b. ’w’or ’w+’ or ‘a’or ‘a+’ c. reader d. close() e. Arjun 123@456 Arunima aru@nima Frieda myname@FRD
4.	i. bin_file=open(“Cust_file.dat”,”wb”) ii. if qty<10 : iii. pickle.dump(c_detail,bin_file) #statement 3 break #statement 4 iv. bin_file.close() #statement 5 v. write_bin() #statement 6
5.	i. pickle ii. fout=open(“temp.dat”,”wb”)#statement 2 iii. rec=pickle.load(fin) #statement 3 pickle.dump(rec,fout)#statement 4

Chapter 5: DATA STRUCTURE

PART A

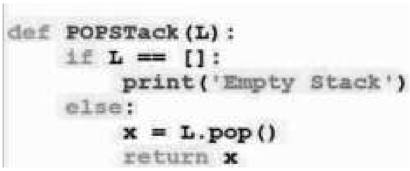
1. (c)	**2. (a)**	**3. (b)**	**4. (a)**	**5. (c)**	**6. (b)**

PART B

1. (d)	**2. (b)**	**3. (d)**	**4. (a)**	**5. (c)**	**6. (a)**

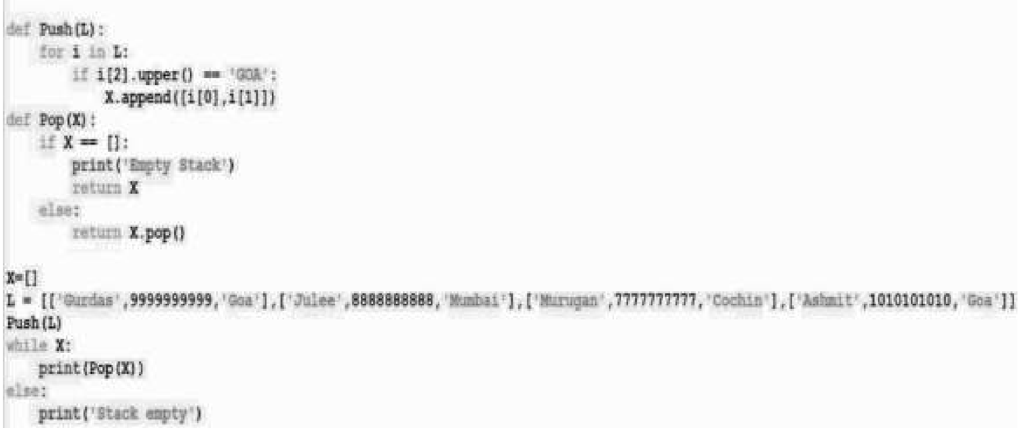
PART C

1.



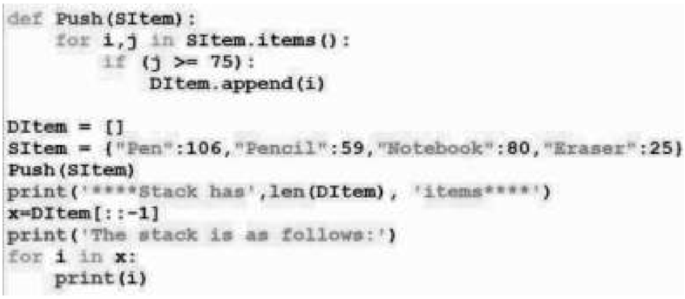
```
def POPSTack(L):
    if L == []:
        print('Empty Stack')
    else:
        x = L.pop()
        return x
```

2.



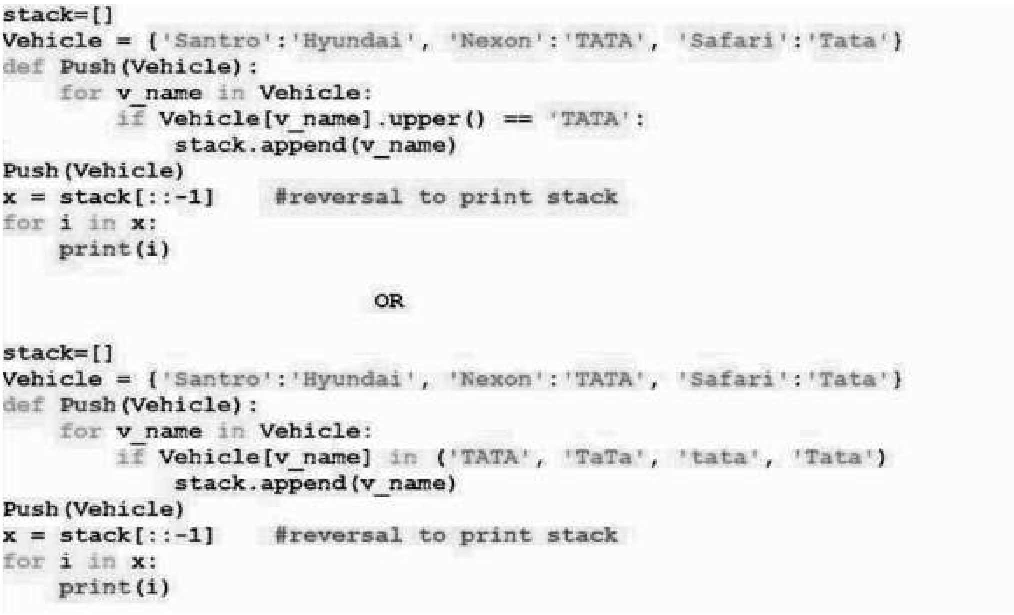
```
def Push(L):
    for i in L:
        if i[2].upper() == 'GOA':
            X.append([i[0],i[1]])
def Pop(X):
    if X == []:
        print('Empty Stack')
        return X
    else:
        return X.pop()

X=[]
L = [['Gurdas',9999999999,'Goa'],['Julee',8888888888,'Mumbai'],['Murugan',7777777777,'Cochin'],['Ashmit',1010101010,'Goa']]
Push(L)
while X:
    print(Pop(X))
else:
    print('Stack empty')
```

3.



```
def Push(SItem):
    for i,j in SItem.items():
        if (j >= 75):
            DItem.append(i)

DItem = []
SItem = {"Pen":106,"Pencil":59,"Notebook":80,"Eraser":25}
Push(SItem)
print('****Stack has',len(DItem), 'items****')
x=DItem[::-1]
print('The stack is as follows:')
for i in x:
    print(i)
```

4.

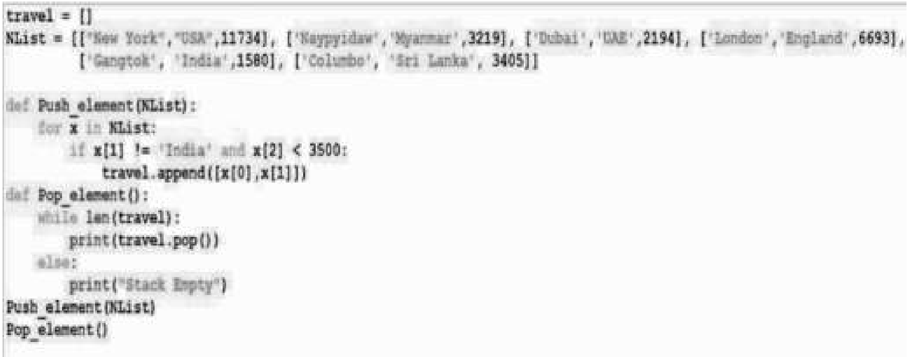


```
stack=[]
Vehicle = {'Santro':'Hyundai', 'Nexon':'TATA', 'Safari':'Tata'}
def Push(Vehicle):
    for v_name in Vehicle:
        if Vehicle[v_name].upper() == 'TATA':
            stack.append(v_name)
Push(Vehicle)
x = stack[::-1]    #reversal to print stack
for i in x:
    print(i)
```

OR

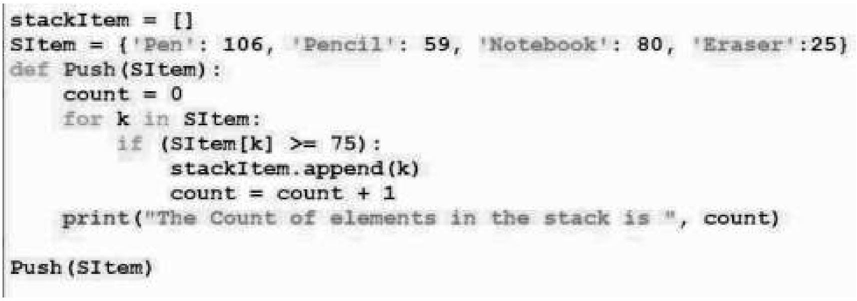
```
stack=[]
Vehicle = {'Santro':'Hyundai', 'Nexon':'TATA', 'Safari':'Tata'}
def Push(Vehicle):
    for v_name in Vehicle:
        if Vehicle[v_name] in ('TATA', 'TaTa', 'tata', 'Tata')
            stack.append(v_name)
Push(Vehicle)
x = stack[::-1]    #reversal to print stack
for i in x:
    print(i)
```

5.



```
travel = []
NList = [["New York","USA",11734], ['Naypyidaw','Myanmar',3219], ['Dubai','UAE',2194], ['London','England',6693],
        ['Gangtok', 'India',1580], ['Columbo', 'Sri Lanka', 3405]]

def Push_element(NList):
    for x in NList:
        if x[1] != 'India' and x[2] < 3500:
            travel.append([x[0],x[1]])
def Pop_element():
    while len(travel):
        print(travel.pop())
    else:
        print("Stack Empty")
Push_element(NList)
Pop_element()
```

6.



```
stackItem = []
SItem = {'Pen': 106, 'Pencil': 59, 'Notebook': 80, 'Eraser':25}
def Push(SItem):
    count = 0
    for k in SItem:
        if (SItem[k] >= 75):
            stackItem.append(k)
            count = count + 1
    print("The Count of elements in the stack is ", count)

Push(SItem)
```

COMPUTER NETWORK SOLUTIONS

Multiple Choice Questions

1. Star
2. The entire network becomes inoperative
3. Mesh
4. Token Passing
5. High installation cost
6. Star
7. The entire network becomes inoperative
8. Star
9. Ring
10. Twisted-pair cable
11. Fiber-Optic Cable

Short Answer Questions

1. Network topology is the process that describes the way that different parts of a computer network, like computers and cables, are connected and arranged.
2. The most commonly used topology in Ethernet LANs (Local Area Networks) is the star topology.
3. The advantages of mesh topology in terms of reliability include:
   a. **Redundancy:** Each device is connected to multiple other devices, so if one link fails, data can still travel through alternate paths.
   b. **Fault Tolerance:** The network can continue to operate smoothly even if some connections fail, reducing the risk of complete network downtime.
   c. **Robustness:** It is resilient against failures because each device has multiple connections, ensuring consistent network performance and stability.
4. The topology commonly used in wireless networks is the **mesh topology**. In a wireless mesh network, each node connects directly, dynamically, and non-hierarchically to as many other nodes as possible, collaborating to route data efficiently.
5. The tree topology offers several advantages over the bus topology:
   a. **Scalability**: Tree topology allows for the expansion of the network by adding more branches or levels, whereas bus topology may become limited in scalability due to the linear nature of the bus.
   b. **Fault Isolation**: In tree topology, if one branch fails, it does not affect the entire network. However, in bus topology, if the main bus line is damaged or faulty, the entire network can be disrupted.
   c. **Performance**: Tree topology can potentially offer better performance as data does not have to contend with other devices on a shared bus, reducing the likelihood of data collisions and improving overall network efficiency.
   d. **Organizational Structure**: Tree topology can mirror organizational structures more effectively, allowing for better management and organization of network resources, whereas bus topology may not align as well with hierarchical structures.
   Overall, the tree topology provides more flexibility, fault tolerance, and performance advantages compared to the bus topology.
6. The different types of transmission media include:
   a. Twisted Pair Cable
   b. Coaxial Cable
   c. Fiber Optic Cable
   d. Wireless Transmission (e.g., Radio Waves, Microwaves, Infrared)
7. Twisted pair cables offer several advantages and drawbacks:

Advantages:

1. **Cost-Effective:** Twisted pair cables are relatively inexpensive compared to other types of cables like fiber optics, making them a cost-effective choice for many networking applications.
2. **Flexibility:** They are flexible and easy to install, making them suitable for various network layouts and configurations.
3. **Widely Available:** Twisted pair cables are widely available and compatible with most network devices, making them easy to procure and use.

4. **Suitable for Short to Medium Distances:** They are suitable for transmitting data over short to medium distances, making them ideal for local area networks (LANs) and telephone systems.
5. **Resistance to Electromagnetic Interference (EMI):** Twisted pair cables are less susceptible to EMI compared to unshielded cables, especially when properly installed and shielded.

Drawbacks:

1. **Limited Bandwidth:** Twisted pair cables have limited bandwidth compared to fiber optic cables, which can restrict the speed and amount of data that can be transmitted over long distances.
2. **Susceptible to Crosstalk:** Crosstalk can occur when signals from adjacent pairs interfere with each other, potentially degrading signal quality and causing data errors.
3. **Limited Distance:** Twisted pair cables have a limited transmission distance compared to fiber optic cables, which may require additional equipment like repeaters to extend the reach of the network.
4. **Security Concerns:** Twisted pair cables can be vulnerable to eavesdropping and wiretapping if not adequately secured or encrypted, posing potential security risks to the network.
5. **Signal Degradation:** Signal degradation can occur over long distances or in noisy environments, leading to reduced signal quality and potential data loss.

8. The main drawbacks of coaxial cables include:

1. **Limited Bandwidth**: Coaxial cables have a limited bandwidth compared to fiber optic cables, which can restrict the speed and amount of data that can be transmitted over long distances.
2. **Susceptibility to Signal Attenuation**: Coaxial cables can experience signal attenuation, especially over long distances or at higher frequencies, leading to signal degradation and potential data loss.
3. **Difficulty in Installation**: Coaxial cables are thicker and less flexible than twisted pair cables, making them more challenging to install, especially in tight spaces or over long distances.
4. **Limited Transmission Distance**: Coaxial cables have a limited transmission distance compared to fiber-optic cables, which may require additional equipment like repeaters to extend the reach of the network.
5. **Vulnerability to Electromagnetic Interference (EMI)**: Coaxial cables can be susceptible to EMI from external sources, such as power lines or electronic devices, which can degrade signal quality and cause data errors.
6. **Cost**: Coaxial cables can be more expensive than twisted pair cables, especially for higher-quality or specialized types, which can impact the overall cost of network installation and maintenance.

9. Fiber optic cables have several advantages over coaxial cables such as higher bandwidth, faster speeds. Longer distance, better signal quality, lightweight and thin, greater security and durability.
10. Several devices use the Infra-Red technology. Some common devices include: remote controls, IR blasters, security systems, healthcare devices, computers and peripherals, toys.

11. Guided transmission media offer more security as compared to un-guided transmission media since they offer limited signal propagation, higher encryption, resistance to interference and enhanced physical control over the signals.
12. Twisted pair cables, coaxial cables, fiber optic cables, Wi-Fi, Bluetooth, and Infra-Red Communication (IRC) are commonly used transmission media for short distances.

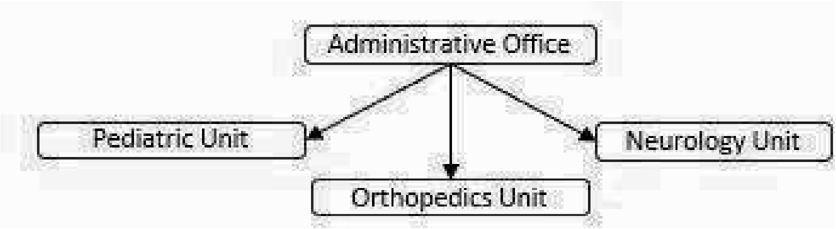
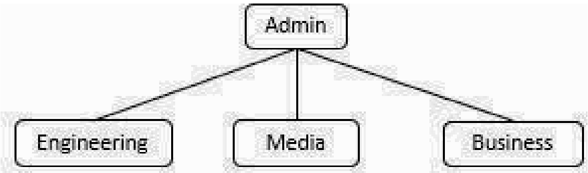
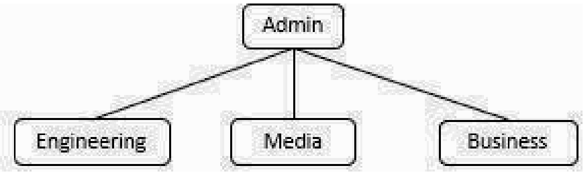
    Fiber optic cables, Microwaves, Satellite Communication, and Radio Waves are commonly used transmission media for long distances.
13. Distance, Bandwidth requirement, cost, Installation and maintenance, environmental factors, security, scalability, and application specific (mobility/fixed infrastructure) requirements, Industry specific compliance are some important factors that influence the choice of transmission media in a network.
14. A Local Area Network (LAN) is a network that connects computers and other devices within a relatively small and specific geographic area, such as a home, office, or building.

    A Personal Area Network (PAN) is a network that connects devices within the immediate vicinity of a single user, typically within a range of a few meters. PANs are designed for personal use.

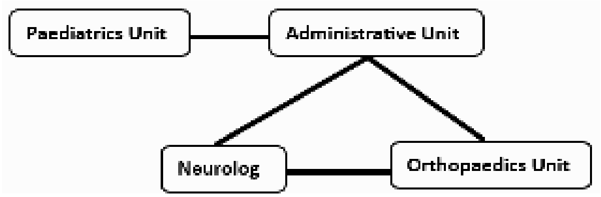
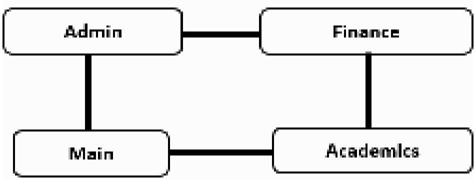
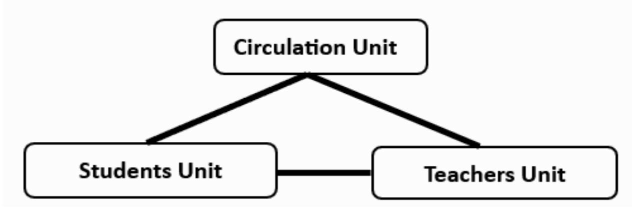
    A Metropolitan Area Network (MAN) is a network that connects multiple local area networks (LANs) within a metropolitan area, such as a city or a large campus. MANs are designed to extend the reach of a LAN, enabling efficient communication and resource sharing over a larger geographic area than a LAN.

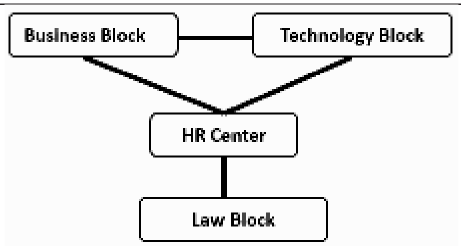
    A Wide Area Network (WAN) is a telecommunications network that extends over a large geographic area, often encompassing cities, countries, or even continents. WANs are used to connect multiple smaller networks, such as local area networks (LANs) and metropolitan area networks (MANs).
15. A Virtual Private Network (VPN) is a technology that creates a secure, encrypted connection over a public network such as the internet. It allows users to access and transmit data securely over an untrusted network as if they were directly connected to a private network.
16. Domain Name Service (DNS) is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It translates human- readable domain names (e.g., www.example.com) into numeric IP addresses (e.g., 192.0.2.1) needed for locating and identifying devices and services on the network.

Case Study Based Questions

1.	1. Administrative Office 2. The suitable layout is  Administrative Office Pediatric Unit Neurology Unit Orthopedics Unit 3. Modem: in all the units Gateway in Administrative Office Switch with the server unit in the Administrative Office 4. Bus topology in all the individual Units using the Ethernet cable Star topology to inter-connect all the units using the coaxial cable
2.	1.  2. Switch 3. Admin, since it has the highest number of computers and workload. 4. Satellite Communication 5. Not required as the farthest office is the Hyderabad Office and it is connected via satellite communication.
3.	1. Administrative building *2.*  3. LAN

	4. Satellite communication 5. Satellite communication is a type of wireless communication to connect computers around the world. It is a type of WAN (Wide Area Network).

4.	1. Administrative office *2.*  Paediatrics Unit Administrative Unit Neurolog Orthopaedics Unit 3. Switch 4. Bus topology, ethernet cable
5.	*a.*  b. Admin c. Switch can be placed in all the buildings easily, but, the repeater must be placed in the **Main** building to connect with the **Finance** building as these are farthest as compared to other buildings. d. Coaxial cable must be used due to its low cost of installation and maintenance, and Data Security.
6.	1. Circulation Unit *2.*  3. Modem should be placed in the circulation unit, and the internet should be provided to all the units via installing Sub-Modem in teachers and student's unit, thereby distributing using the ethernet LAN cable. 4. Coaxial Ethernet cable. 5. LAN should be established using the Coaxial cable as the range of LAN is up to 100 meters and Coaxial cables provide reduced installation and maintenance cost with comparatively high data security and durability in all weather conditions.
7.	a. The server must be placed in the technology unit for ease of maintenance. *b.*

	 Business Block Technology Block HR Center Law Block c. Switch can be installed using the Ethernet Coaxial Cables in a bi-directional ring. d. The repeater must be installed in the Law Block while establishing connection with the HR block as the HR block is located at the farthest distance and it needs the most accurate data for decision making. e. WAN must be set-up using Satellite Communication due to long range distance.

DBMS1 - ANSWERS DATABASE CONCEPT

1 Database is collection of interrelated data or record in organized form so it can easily be accessed, managed and updated**.** On the other hand, DBMS is a software system that is used to manage databases. DBMS acts as an interface between a user and database that enables the user to create, insert, retrieve, update and delete the data.

2 Data

3 We do need database for following reasons:

1. **Centralized Storage**: Storage of data in a single location or central database.
2. **Data Integrity**: Enforces data integrity rules which ensures that information stored is accurate, valid and consistent.
3. **Data Security**: Control access to sensitive data and protecting data from unauthorized access.
4. **Data Retrieval**: Authorized User/Application can access and retrieve the information as per their need.
5. **Efficient Data Retrieval**: Database helps user to retrieve data in an efficient way.

4 Relational Model

DBMS2 - ANSWERS RELATIONAL DATA MODEL

1 Relation Data Model is proposed by E.F. Codd in 1970

2 Conceptual Data Model is used to capture the meaning of data from the viewpoint of the user and try to represent it using Data Model

3 ER stands for Entity-Relationship Model. ER Model tools are used to represent Conceptual Data Model.

4 Tuple represent the horizontal form of Relation. Attributes represent the vertical form of Relation.

5 Total number of tuples in a relation is called Cardinality.

6 Total number of attributes in a relation is called a Degree.

7 A Domain of database is set of atomic value (which can't further be distributed) of a particular attribute/column.

8 In relations, Datatypes is used to declare what type of data that will be stored in particular column. Some commonly used datatype in SQL are follows :

1. Numeric Type:

- INT: Integer type
- FLOAT: Floating-point number
- DECIMAL or NUMERIC: Fixed-point number

2. Character String Type:

- CHAR(n): Fixed-length character string with maximum length of n
- VARCHAR(n): Variable-length character string with maximum length of n
- TEXT Type : Variable-length character string with no specified maximum length

3. Data and Time Type:

- DATE : for date only
- TIME: for time only
- DATETIME or TIMESTAMP: for date and time combined

9 CHAR(n): Fixed-length character string with maximum length of n

VARCHAR(n): Variable-length character string with maximum length of n

10 Keys is column/attribute which is used to fetch/extract/retrieve row in a table

11
1. Primary Key: Primary Key is a unique identifier which identify unique record in a particular table. It must contain unique values for each record. And Primary key attribute/column/field can't be NULL.
2. Candidate Key: Candidate key are those key which are eligible for primary key and can be used as primary key in a table.
3. Alternate Key: After selecting the primary key from candidate key, the remaining keys (which are also eligible for primary key) are called Alternate Key.
4. Foreign Key: A Foreign key is a column or group of columns in a table that provides a link between two tables.

12 SQL is a query language not a database system.
Example – Oracle, MySQL, MongoDB, PostgreSQL, SQL Server, DB2 etc.
SQL perform following operation:

- Create a database
- Create a table
- Create view
- Insert data
- Update data
- Delete data
- Execute Query
- Set Permission or Constraints in table

13 Clause in SQL

14
1. create database class12;
2. show databases;
3. use class12;
4. show tables;
5. create table student12(
   student_id int,
   student_name char(30),
   age int,
   phone int,
   address varchar(50));
6. desc student12 or describe student12
7. insert into student12(1,'Rohit',9874563210,25,'Delhi');
8. insert into <table name> values (<value>, <value> , <value> ...);
9. select * from student12;
10. drop table student12;

15 Constraints in SQL are a set of rules that are applied to the data in a relation/table. Constraints are used to ensure the accuracy and reliability of the data.
Types of constraints:

1. Primary key
2. Unique
3. Not Null
4. Foreign Key

16 DDL commands are used to make any changes in the structure of the table/database. These commands don't change the data of the table.
Example: create table, alter table, drop table, create database, create view etc.

17 DML commands are used to make any changes in the data of the table.
Example: Insert, delete, update, select etc.

18 Aliasing in SQL

19 Drop table command is used to delete data as well as structure of a table, drop command is used but Delete Command used to delete the existing rows in a table that matches the condition.

20 Distinct clause

21 Where clause

22 Char

23 Primary Key

24 Candidate Key

25 Order by Clause

26 Like operator

27 Primary key constraints

28 NOT NULL Constraints

29 Alter table

30 having

DBMS3 – ANSWERS MULTIPLE CHOICE QUESTIONS (MCQ)

Q. No.	Answer	Q. No.	Answer	Q. No.	Answer	Q. No.	Answer
1	D	14	B	27	A	40	C
2	D	15	A	28	B	41	B
3	A	16	A	29	B	42	B
4	B	17	B	30	C	43	C
5	C	18	C	31	C	44	D
6	C	19	B	32	C	45	D
7	B	20	C	33	B	46	A
8	D	21	B	34	B	47	C
9	B	22	A	35	A	48	A
10	C	23	C	36	B	49	C
11	A	24	C	37	C	50	B
12	D	25	B	38	B		
13	D	26	C	39	B		

DBMS4 – ANSWERS SHORT ANSWER TYPE QUESTIONS

1. SQL stands for structured query language, is a standard programming language used for

managing and manipulating relational databases.

2. DML is used for managing data within existing databases. DML commands are SELECT, INSERT, UPDATE AND DELETE. DDL command is used for defining and modifying database structures. DDL Commands are CREATE, ALTER AND DROP.

3. **Datatype in SQL**:

1. Numeric Type:
- INT: Integer type
- FLOAT: Floating-point number
- DECIMAL or NUMERIC: Fixed-point number

2. Character String Type:
- CHAR(n): Fixed-length character string with maximum length of n
- VARCHAR(n): Variable-length character string with maximum length of n
- TEXT Type: Variable-length character string with no specified maximum length

3. Data and Time Type:
- DATE: for date only
- TIME: for time only
- DATETIME or TIMESTAMP: for date and time combined

4. Other Data type:
- NULL: to represent a missing/unknown/empty value
- ENUM: An enumeration type for a set of predefined values

4. The WHERE statement in SQL is used to filter the results of a SELECT statement by specifying one or more conditions. Having statement is used to filter the result set of group by clause in select statement.

5. Primary Key is a unique identifier which identify unique record in a particular table. It must contain unique values for each record. And Primary key attribute/column/field can't be NULL. A table can have only ONE primary key

6. Database table is a structured collection of data organized into rows and columns. Database record in a single, complete set of related data items in a table, representing a single entity. Each record is a row in a table which consists of one value for each column.

7. A Foreign key is a column or group of columns in a table that provides a link between two tables. Foreign key are essential for defining relationships between tables in a relational databases, which help in maintaining integrity and accuracy of dataDomain is a set of possible values or range of valid values or set of all unique values that an attribute/column can hold.

8. Clause are built in functions which is use to deal with data inside the table that help SQL to filter and analyses data quickly.

9. Aliasing in SQL is the process of assigning a nick name or a temporary name to a table or column. We create aliases to make queries more readable and easier to use. Alias created using as keyword. Creating aliases don't change name of any table or column permanently.

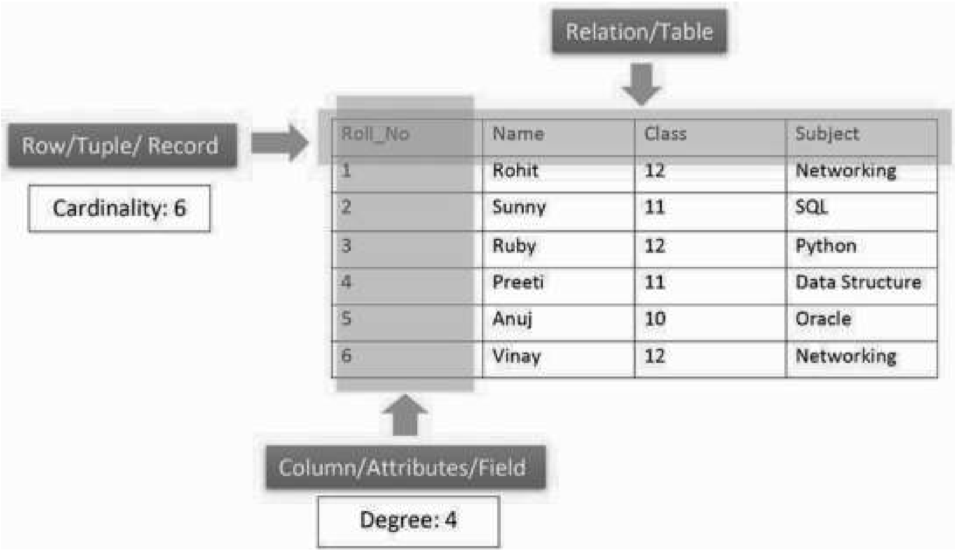
10. CHAR(n): Fixed-length character string with maximum length of n

VARCHAR(n): Variable-length character string with maximum length of n

11. Candidate key are those key which are eligible for primary key and can be used as primary key in a table. After selecting primary key from candidate key, the remaining keys (which are also eligible for primary key) are called Alternate Key.

12. Cardinality: Total number of row/record/tuple in a relation is called as Cardinality.

Degree: Total number of column/attributes in a relation is called as Degree.



Roll_No	Name	Class	Subject
1	Rohit	12	Networking
2	Sunny	11	SQL
3	Ruby	12	Python
4	Preeti	11	Data Structure
5	Anuj	10	Oracle
6	Vinay	12	Networking

13. In database, keys is column/attribute which is used to fetch/extract/retrieve row in a table. Or we can say that Keys are used to uniquely identify records in a table through column or combination of column.

Primary Key: Primary Key is a unique identifier which identify unique record in a particular table.
Candidate Key: Candidate keys are those keys which are eligible for primary key and can be used as primary key in a table, but are not selected to be the same.
Alternate Key: After selecting primary key from candidate key, the remaining keys (which are also eligible for primary key) are called Alternate Key.

14. Constraints in SQL are set of rules that are applied to the data in a relation/table. Constraints are used to ensure the accuracy and reliability of the data.

Types of constraints:

1. Primary key
2. Unique

3. Not Null
4. Foreign Key

15. Like operator is used to match a pattern. The like operator is used with were clause. Like operator has 2 wildcards:

1. _ (underscore): It is used to match one character.
2. % (percentage sign): It is used to match zero or more characters.

16. Joins are used to combine rows from multiple tables.

17. **Equi Join**- It joins the tables based on one common column. However, final result will consists of common column from both the tables.
Natural Join-It joins the tables based on one common column. However, final result will consists of common column only once.

18. Aggregate functions are those functions that operates on a list of values and returns a single digit value or we can summarize the data using aggregate functions.

Max(): It is used to find out the maximum value from a column.

Min(): It is used to find out the minimum value from a column.

19. A cursor is SQL is database object used to retrieve, manipulate and navigate through a result set tow by row. It is a pointer or iterator which points towards the resultset of the SQL query. Whenever a SQL query runs, It give the entire result set in one go.

20. **Commit()**: Whenever we perform update, delete or insert query, commit() function must be run before closing the connection.

21. Cartesian product gives all possible combinations from more than one table. It combines every row from one table with every row from another table.
Total number of row after applying cartesian product is 90.

22. This is a DDL command and it is used to modify a table. This command can be used to add, delete, or modify columns, add or drop constraints etc.

Example: alter table student add class varchar(50) not null;

fetchall(): It returns all the records from resultset. Each individual record will be in the form of a tuple whereas the entire resultset will be in the form of a list.

Syntax: <variable name>=<cursor>.fetchall()

fetchmany(): It returns n number of records from resultset in the form of a list where each individual record is in the form of a tuple. It returns empty tuple, if no more records are there.

Syntax: <variable name>=<cursor>.fetchmany(n)

23. In SQL, Define aggregate function and write the name of the aggregate function which will display the cardinality of a table.

Aggregate functions are those functions that operates on a list of values and returns a single digit value or we can summarize the data using aggregate functions.

count(*) is often used to count all rows in a table. Example:

select count(*) as cardinality from employees;

DBMS5 – ANSWERS ASSERTION AND REASONING

1. (c)	**2.** (c)	**3.** (b)	**4.** (b)	**5.** (a)
6. (d)	**7.** (b)	**8.** (c)	**9.** (a)	**10.** (c)

DBMS6 – ANSWERS LONG ANSWER TYPE QUESTIONS

1. create table Employee (Emp_ID int(20) primary key, Emp_Name char(100) not null, salary int(20) not null, Department char(30), Age int(15) not null, Address varchar(200) unique);

2. (i) create table Student (Student_ID varchar(20) primary key, Student_Name char(80) not null, Gender char(20) not null, Class varchar(30), Age int(20) not null, Address varchar(150) unique, Phone int(15) not null unique);
   (ii) create table Activities (Student_ID varchar(20), Activity_Name char(80) not null, Position char(30) not null, foreign key(Student_ID) references Student(Student_ID));
3. (a) Degree – 5 and Cardinality – 6

   (b) MID

   (c) insert into PHARMA values (MID ,MNAME, PRICE, EXPIRY) values (‘M7’, ’SUCRALFATE’, 17, ’2022-03-20’);

   (d) IV

   (e) I

4. RAVI KUMA

   NISHANT JAIN DEEPAK
   PRAKASH

5. (1) select * from customers where name like ‘A%’;
   (2) select name, balance from customers, transaction where gender=’F’ and year(TDATE)=2019);
   (3) select gender, count(*) as no_of_customer from customer group by gender;

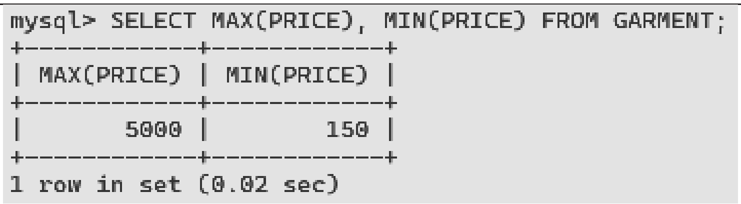
(4) select name, balance from customer order by gender;
(5) select name, balance*0.08 as interest from customer;

6. (1) E_NO
(2) D_CODE
(3) Degree – 7 and Cardinality – 6
(4) Degree – 3 and Cardinality – 5
(5) (1) insert into employee values (1006, ’Rahul’, ‘2019-11-06’, ‘1992-01-04’, ‘MALE’, ‘D003’, 156000);
insert into employee values (1008, ’Sonam’, ‘2022-01-06’, ‘1991-04-06’, ‘FEMALE’ , ‘D005’, 167000);
(2) select E_NO,NAME,GENDER from EMPLOYEE order by E_NO desc;
(3) select NAME from EMPLOYEE where GENDER=’FEMALE’;
(4) select E_NO,NAME from EMPLOYEE where DOB between ‘1987-01-01’ and ‘1991-12-01’.
(5) select NAME,CITY from EMPLOYEE where DEPARTMENT in (‘MEDIA’,’FINANCE’).
(6) select name from employee where name like ‘R%’;
(7) select name from employee where name like ‘%n%’;
(8) select name from employee where name like ‘_ _ _ _ _’;
(9) select d_name, city from dept where d_name like ‘%G’ and city=’delhi’;
(10) select max(salary) from employee;
(11) delete from employee where age<25;
(12) update employee set salary=230000 where e_no=1004;
(13) alter table employee modify dob date after name;
(14) alter table employee add column mobile int(20) after d_code;
(15) update employee set salary=300000 where age is null;
(16) update employee set salary=salary+30000;
(17) select avg(salary) from employee;
(18) select name from employee where salary>200000 order by name;
(19) select d_name, avg(salary) from employee natural join dept group by d_name;
(20) select count(d_name) from dept;
(21) delete from employee where d_code != ’d001’;
(22) select e_no, name, salary from employee where city not in (‘delhi’);
(23) alter table dept change column city d_city char(20);
(24) drop table employee;
(25) alter table dept drop column d_name;
(26) select name,city from dept D,employee E where D.d_code=E.d_code.

7. (1) DDL – alter, create, drop
DML – insert, update, delete
(2) GCODE
(3) Degree – 4 and Cardinality – 6
(4) create database store;
(5) show databases;
(6) use store;
(7) show tables;

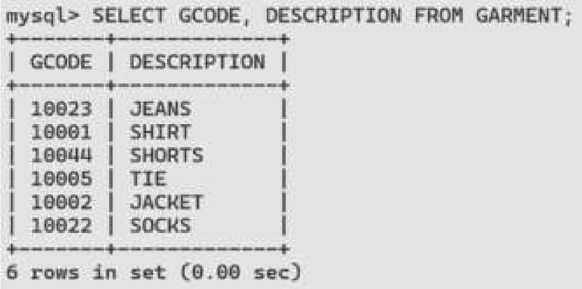
(8) create table garment (gcode int, description char(50), price int, fcode varchar(10));

(i)



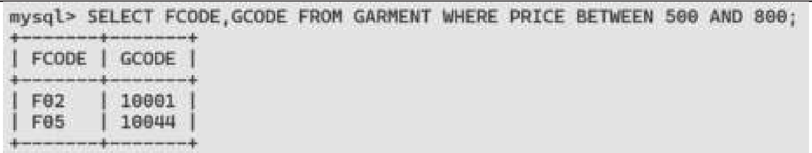
```
mysql> SELECT MAX(PRICE), MIN(PRICE) FROM GARMENT;
+------------+------------+
| MAX(PRICE) | MIN(PRICE) |
+------------+------------+
|       5000 |        150 |
+------------+------------+
1 row in set (0.02 sec)
```

(ii)



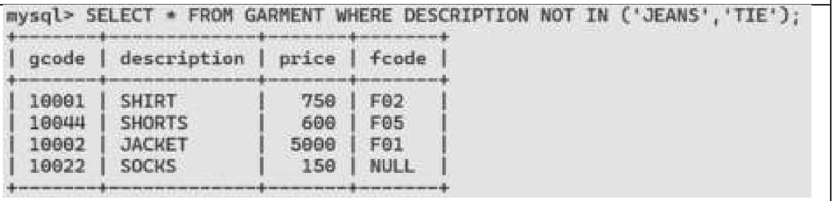
```
mysql> SELECT GCODE, DESCRIPTION FROM GARMENT;
+-------+-------------+
| GCODE | DESCRIPTION |
+-------+-------------+
| 10023 | JEANS       |
| 10001 | SHIRT       |
| 10044 | SHORTS      |
| 10005 | TIE         |
| 10002 | JACKET      |
| 10022 | SOCKS       |
+-------+-------------+
6 rows in set (0.00 sec)
```

(iii)



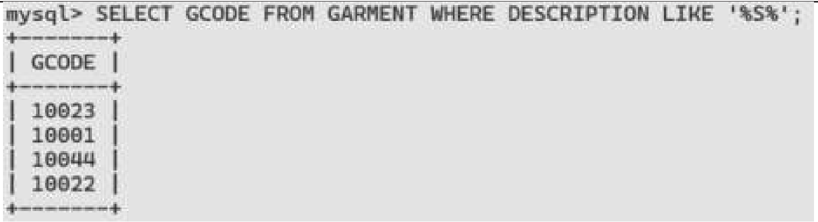
```
mysql> SELECT FCODE,GCODE FROM GARMENT WHERE PRICE BETWEEN 500 AND 800;
+-------+-------+
| FCODE | GCODE |
+-------+-------+
| F02   | 10001 |
| F05   | 10044 |
+-------+-------+
```

(iv)



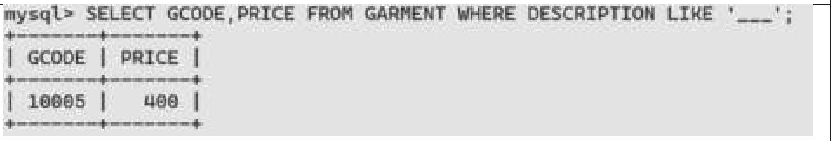
```
mysql> SELECT * FROM GARMENT WHERE DESCRIPTION NOT IN ('JEANS','TIE');
+-------+-------------+-------+-------+
| gcode | description | price | fcode |
+-------+-------------+-------+-------+
| 10001 | SHIRT       |   750 | F02   |
| 10044 | SHORTS      |   600 | F05   |
| 10002 | JACKET      |  5000 | F01   |
| 10022 | SOCKS       |   150 | NULL  |
+-------+-------------+-------+-------+
```

(v)



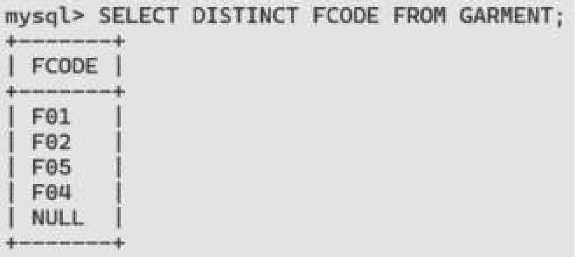
```
mysql> SELECT GCODE FROM GARMENT WHERE DESCRIPTION LIKE '%S%';
+-------+
| GCODE |
+-------+
| 10023 |
| 10001 |
| 10044 |
| 10022 |
+-------+
```

(vi)



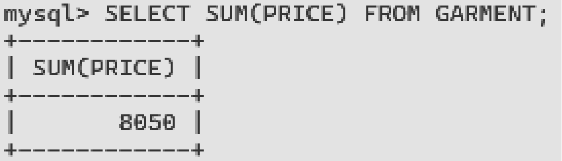
```
mysql> SELECT GCODE,PRICE FROM GARMENT WHERE DESCRIPTION LIKE '___';
+-------+-------+
| GCODE | PRICE |
+-------+-------+
| 10005 |   400 |
+-------+-------+
```

(vii)



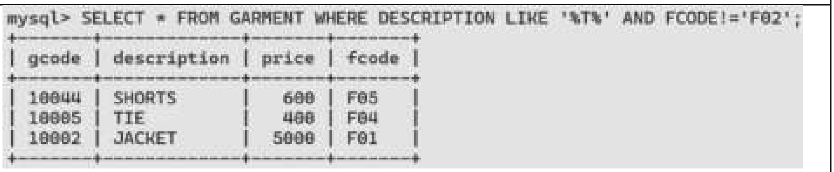
```
mysql> SELECT DISTINCT FCODE FROM GARMENT;
+-------+
| FCODE |
+-------+
| F01   |
| F02   |
| F05   |
| F04   |
| NULL  |
+-------+
```

(viii)



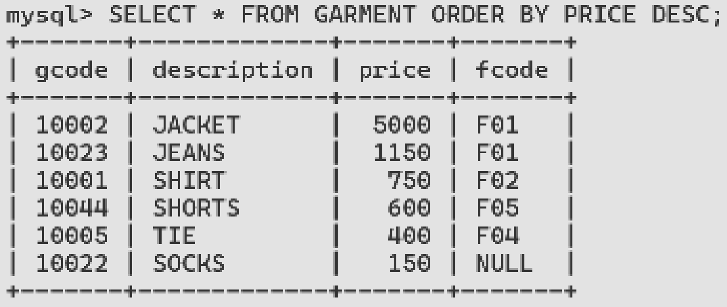
```
mysql> SELECT SUM(PRICE) FROM GARMENT;
+------------+
| SUM(PRICE) |
+------------+
|       8050 |
+------------+
```

(ix)



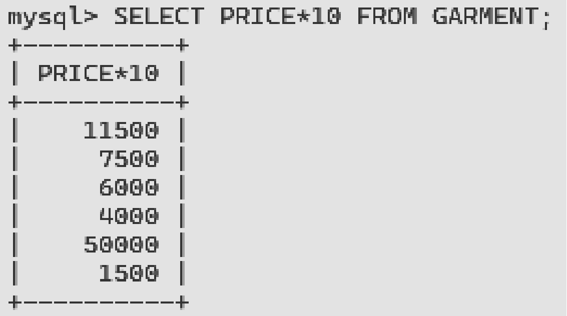
```
mysql> SELECT * FROM GARMENT WHERE DESCRIPTION LIKE '%T%' AND FCODE!='F02';
+-------+-------------+-------+-------+
| gcode | description | price | fcode |
+-------+-------------+-------+-------+
| 10044 | SHORTS      |   600 | F05   |
| 10005 | TIE         |   400 | F04   |
| 10002 | JACKET      |  5000 | F01   |
+-------+-------------+-------+-------+
```

(x)



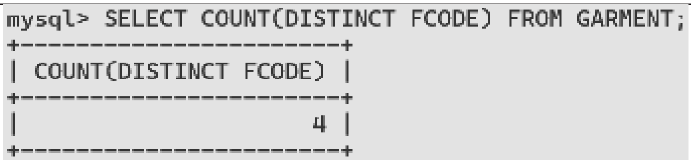
```
mysql> SELECT * FROM GARMENT ORDER BY PRICE DESC;
+-------+-------------+-------+-------+
| gcode | description | price | fcode |
+-------+-------------+-------+-------+
| 10002 | JACKET      |  5000 | F01   |
| 10023 | JEANS       |  1150 | F01   |
| 10001 | SHIRT       |   750 | F02   |
| 10044 | SHORTS      |   600 | F05   |
| 10005 | TIE         |   400 | F04   |
| 10022 | SOCKS       |   150 | NULL  |
+-------+-------------+-------+-------+
```

(xi)



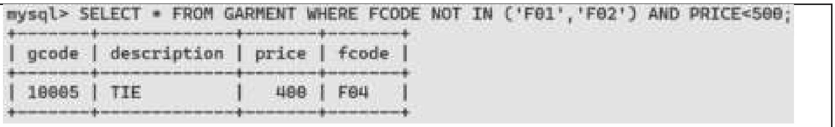
```
mysql> SELECT PRICE*10 FROM GARMENT;
+----------+
| PRICE*10 |
+----------+
|    11500 |
|     7500 |
|     6000 |
|     4000 |
|    50000 |
|     1500 |
+----------+
```

(xii)



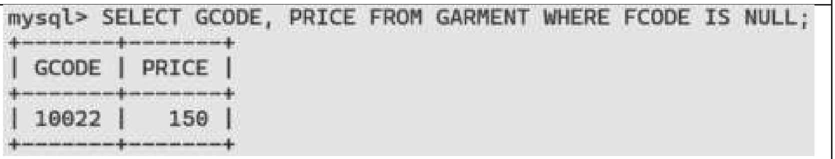
```
mysql> SELECT COUNT(DISTINCT FCODE) FROM GARMENT;
+-----------------------+
| COUNT(DISTINCT FCODE) |
+-----------------------+
|                     4 |
+-----------------------+
```

(xiii)



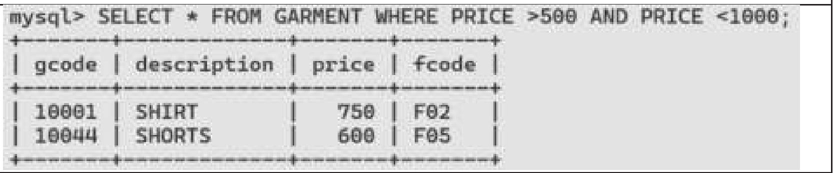
```
mysql> SELECT * FROM GARMENT WHERE FCODE NOT IN ('F01','F02') AND PRICE<500;
+-------+-------------+-------+-------+
| gcode | description | price | fcode |
+-------+-------------+-------+-------+
| 10005 | TIE         |   400 | F04   |
+-------+-------------+-------+-------+
```

(xiv)



```
mysql> SELECT GCODE, PRICE FROM GARMENT WHERE FCODE IS NULL;
+-------+-------+
| GCODE | PRICE |
+-------+-------+
| 10022 |   150 |
+-------+-------+
```

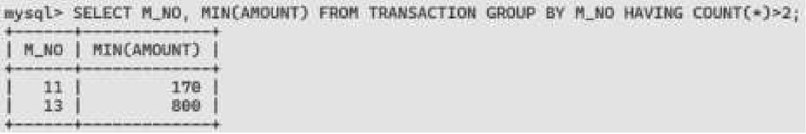
(xv)



```
mysql> SELECT * FROM GARMENT WHERE PRICE >500 AND PRICE <1000;
+-------+-------------+-------+-------+
| gcode | description | price | fcode |
+-------+-------------+-------+-------+
| 10001 | SHIRT       |   750 | F02   |
| 10044 | SHORTS      |   600 | F05   |
+-------+-------------+-------+-------+
```

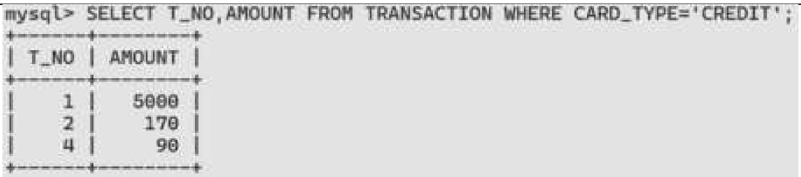
8. Queries

(i)



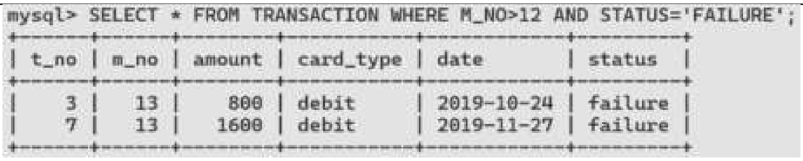
```
mysql> SELECT M_NO, MIN(AMOUNT) FROM TRANSACTION GROUP BY M_NO HAVING COUNT(*)>2;
+------+-------------+
| M_NO | MIN(AMOUNT) |
+------+-------------+
|   11 |         170 |
|   13 |         800 |
+------+-------------+
```

(ii)



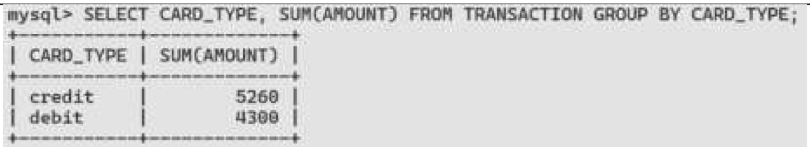
```
mysql> SELECT T_NO,AMOUNT FROM TRANSACTION WHERE CARD_TYPE='CREDIT';
+------+--------+
| T_NO | AMOUNT |
+------+--------+
|    1 |   5000 |
|    2 |    170 |
|    4 |     90 |
+------+--------+
```

(iii)



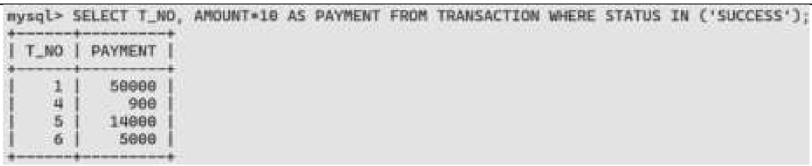
```
mysql> SELECT * FROM TRANSACTION WHERE M_NO>12 AND STATUS='FAILURE';
+------+------+--------+-----------+------------+---------+
| t_no | m_no | amount | card_type | date       | status  |
+------+------+--------+-----------+------------+---------+
|    3 |   13 |    800 | debit     | 2019-10-24 | failure |
|    7 |   13 |   1600 | debit     | 2019-11-27 | failure |
+------+------+--------+-----------+------------+---------+
```

(iv)



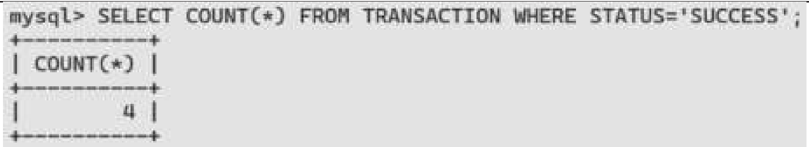
```
mysql> SELECT CARD_TYPE, SUM(AMOUNT) FROM TRANSACTION GROUP BY CARD_TYPE;
+-----------+-------------+
| CARD_TYPE | SUM(AMOUNT) |
+-----------+-------------+
| credit    |        5260 |
| debit     |        4300 |
+-----------+-------------+
```

(v)



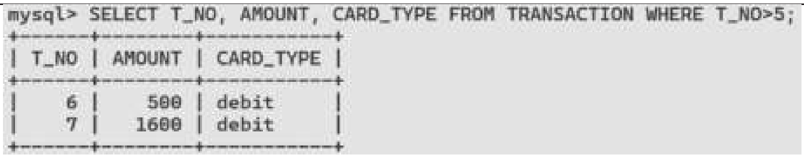
```
mysql> SELECT T_NO, AMOUNT*10 AS PAYMENT FROM TRANSACTION WHERE STATUS IN ('SUCCESS');
+------+---------+
| T_NO | PAYMENT |
+------+---------+
|    1 |   50000 |
|    4 |     900 |
|    5 |   14000 |
|    6 |    5000 |
+------+---------+
```

(vi)



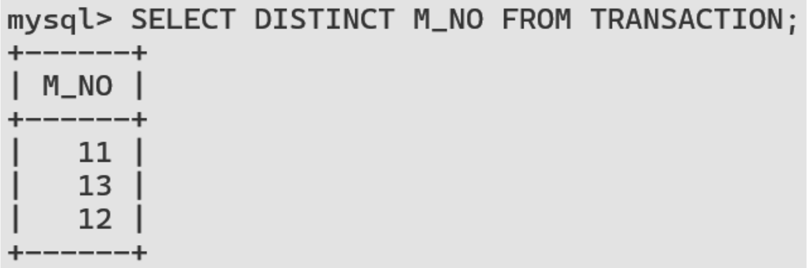
```
mysql> SELECT COUNT(*) FROM TRANSACTION WHERE STATUS='SUCCESS';
+----------+
| COUNT(*) |
+----------+
|        4 |
+----------+
```

(vii)



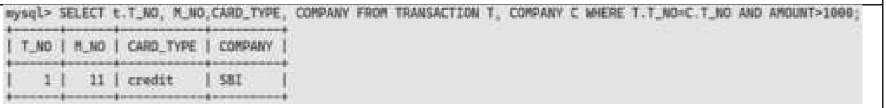
```
mysql> SELECT T_NO, AMOUNT, CARD_TYPE FROM TRANSACTION WHERE T_NO>5;
+------+--------+-----------+
| T_NO | AMOUNT | CARD_TYPE |
+------+--------+-----------+
|    6 |    500 | debit     |
|    7 |   1600 | debit     |
+------+--------+-----------+
```

(viii)



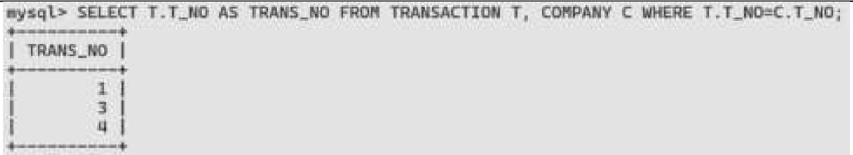
```
mysql> SELECT DISTINCT M_NO FROM TRANSACTION;
+------+
| M_NO |
+------+
|   11 |
|   13 |
|   12 |
+------+
```

(ix)



```
mysql> SELECT t.T_NO, M_NO,CARD_TYPE, COMPANY FROM TRANSACTION T, COMPANY C WHERE T.T_NO=C.T_NO AND AMOUNT>1000;
+------+------+-----------+---------+
| T_NO | M_NO | CARD_TYPE | COMPANY |
+------+------+-----------+---------+
|    1 |   11 | credit    | SBI     |
+------+------+-----------+---------+
```

(x)



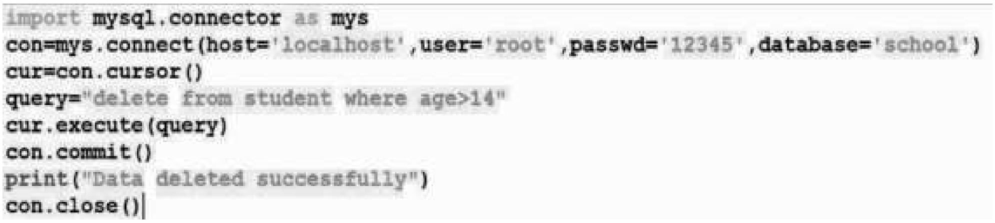
```
mysql> SELECT T.T_NO AS TRANS_NO FROM TRANSACTION T, COMPANY C WHERE T.T_NO=C.T_NO;
+----------+
| TRANS_NO |
+----------+
|        1 |
|        3 |
|        4 |
+----------+
```

9. Statement 1 = mysql.connector

Statement 2 = a.cursor()
Statement 3 = select * from employee where city not in (‘delhi’); Statement 4
= b.execute(query)
Statement 5 = print(z)

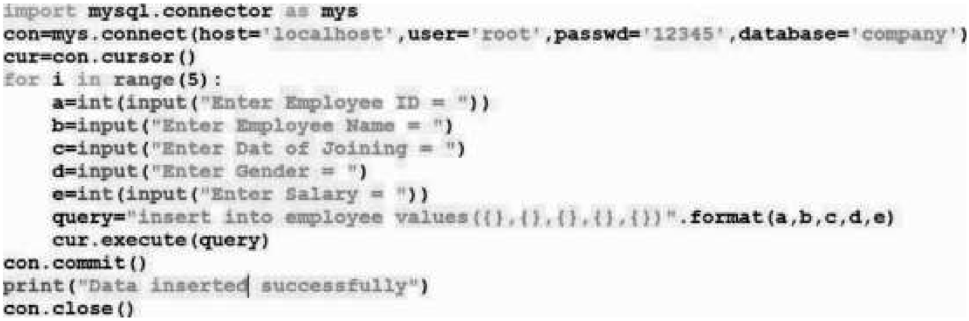
10. Statement 1 = mysql.connector
Statement 2 = mysql.connect
Statement 3 = a.cursor()
Statement 4 = delete from employee where e_id=’a101’ and e_name like ‘D%’; Statement 5
= b.commit()

11.



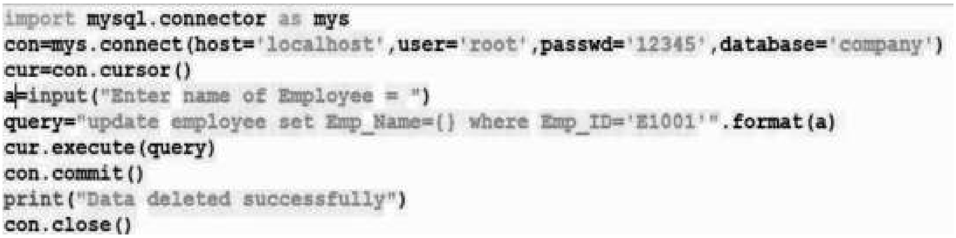
```
import mysql.connector as mys
con=mys.connect(host='localhost',user='root',passwd='12345',database='school')
cur=con.cursor()
query="delete from student where age>14"
cur.execute(query)
con.commit()
print("Data deleted successfully")
con.close()
```

12.



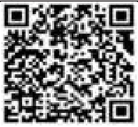
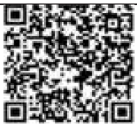
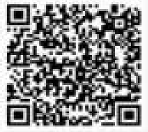
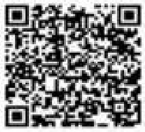
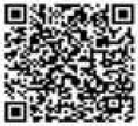
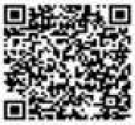
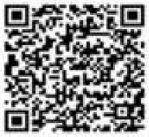
```
import mysql.connector as mys
con=mys.connect(host='localhost',user='root',passwd='12345',database='company')
cur=con.cursor()
for i in range(5):
    a=int(input("Enter Employee ID = "))
    b=input("Enter Employee Name = ")
    c=input("Enter Dat of Joining = ")
    d=input("Enter Gender = ")
    e=int(input("Enter Salary = "))
    query="insert into employee values({},{},{},{},{})".format(a,b,c,d,e)
    cur.execute(query)
con.commit()
print("Data inserted successfully")
con.close()
```

13.



```
import mysql.connector as mys
con=mys.connect(host='localhost',user='root',passwd='12345',database='company')
cur=con.cursor()
a=input("Enter name of Employee = ")
query="update employee set Emp_Name={} where Emp_ID='E1001'".format(a)
cur.execute(query)
con.commit()
print("Data deleted successfully")
con.close()
```

QR Code For Computer Science Previous Year Question Papers, Sample Paper And Their Marking Schemes

S.No.	Title	QR Code
1.	CBSE Annual Exam 2024-25 Question Paper	
2.	CBSE Sample Question Paper 2024-2025	
3.	CBSE Marking Scheme of Sample Question Paper 2024-2025	
4.	CBSE Annual Exam 2023-24 Question Paper	
5.	Marking Scheme 2024	
6.	Compartment Question Paper 2023	
7.	Marking Scheme of Compartment 2023	
8.	CBSE Annual Exam 2022-23 Question Paper	

9.	**Marking Scheme 2023**	
10.	**Compartment Question Paper 2022**	
11.	**Marking Scheme of Compartment 2022**	
12.	**CBSE Annual Exam 2021-22 Question Paper**	
13.	**Marking Scheme 2022**	