कंप्यूटर विज्ञान
Computer Science

कक्षा / Class XII
2025-26

विद्यार्थी सहायक सामग्री
Student Support Material

केन्द्रीय विद्यालय संगठन~Kendriya Vidyalaya Sangathan

संदेश

विद्यालयी शिक्षा में शैक्षिक उत्कृष्टता प्राप्त करना एवं नवाचार द्वारा उच्च - नवीन मानक स्थापित करना केन्द्रीय विद्यालय संगठन की नियमित कार्यप्रणाली का अविभाज्य अंग है। राष्ट्रीय शिक्षा नीति 2020 एवं पी .एम .श्री विद्यालयों के निर्देशों का पालन करते हुए गतिविधि आधारित पठन-पाठन, अनुभवजन्य शिक्षण एवं कौशल विकास को समाहित कर, अपने विद्यालयों को हमने ज्ञान एवं खोज की अद्‌भुत प्रयोगशाला बना दिया है। माध्यमिक स्तर तक पहुँच कर हमारे विद्यार्थी सैद्धांतिक समझ के साथ-साथ, रचनात्मक -विशेषणात्मक एवं आलोचनात्मक चिंतन भी विकसित कर लेते हैं। यही कारण है कि वह बोर्ड कक्षाओं के दौरान विभिन्न प्रकार के मूल्यांकनों के लिए सहजता से तैयार रहते हैं। उनकी इस यात्रा में हमारा सतत योगदान एवं सहयोग आवश्यक है - केन्द्रीय विद्यालय संगठन के पांचों आंचलिक शिक्षा एवं प्रशिक्षण संस्थान द्वारा संकलित यह विद्यार्थी सहायक -सामग्री इसी दिशा में एक आवश्यक कदम है । यह सहायक सामाग्री कक्षा 9 से 12 के विद्यार्थियों के लिए सभी महत्वपूर्ण विषयों पर तैयार की गयी है। केन्द्रीय विद्यालय संगठन की विद्यार्थी सहायक -सामग्री अपनी गुणवत्ता एवं परीक्षा संबंधी -सामग्री संकलन की विशेषज्ञता के लिए जानी जाती है और शिक्षा से जुड़े विभिन्न मंचों पर इसकी सराहना होती रही है। मुझे विश्वास है कि यह सहायक सामग्री विद्यार्थियों की सहयोगी बनकर निरंतर मार्गदर्शन करते हुए उन्हें सफलता के लक्ष्य तक पहुँचाएगी।

शुभाकांक्षा सहित।

निधि पांडे
आयुक्त, केन्द्रीय विद्यालय संगठन

STUDENT SUPPORT MATERIAL, COMPUTER SCIENCE CLASS XII

PATRON

Smt. Nidhi Pandey, Commissioner, KVS

CO-PATRON

Dr. P. Devakumar, Additional Commissioner (Acad.), KVS (HQ)

CO-ORDINATOR

Ms. Chandana Mandal, Joint Commissioner (Training), KVS (HQ)

COVER DESION

KVS Publication Section

EDITORS:

1. Ms. Shahida Parveen, Director, ZIET Mumbai
2. Mr. B L Morodia, Director, ZIET Gwalior
3. Ms. Menaxi Jain, Director, ZIET Mysuru
4. Ms. Preeti Saxena, In-charge Director, ZIET Chandigarh
5. Mr. Birbal Dhinwa, In-charge Director, ZIET Bhubaneswar

CONTENT CREATORS:

MRS. SWATI VISHWAKARMA, PRINCIPAL, PM SHRI K V O F CHANDA

TEAM MEMBERS

NAME OF THE TEACHER	DESIGNATION	KV NAME	REGION
MRS. KSHAMA RAUT	PGT CS	PM SHRI KV AMBARNATH	MUMBAI
MRS. ARPITA DAS	PGT CS	PM SHRI KV MIRC AHMEDNAGAR	MUMBAI
MR. HEMANT	PGT CS	PM SHRI KV CME PUNE	MUMBAI
MR. DEEPAK WAJURKAR	PGT CS	PM SHRI KV CHANDRAPUR	MUMBAI
MR. VIKAS	PGT CS	PM SHRI KV OF CHANDA	MUMBAI
MR. ADITYA AURANGE	PGT CS	KV ONGC PANVEL	MUMBAI
MRS. BINDHIYA N	PGT CS	PM SHRI KV KANJIKODE	ERNAKULUM
MRS. ANJU MATHEW	PGT CS	PM SHRI KV RB KOTTAYAM	ERNAKULUM
MRS. HEMA C N	PGT CS	PM SHRI KV NO.1 NAVAL BASE KOCHI	ERNAKULUM
MRS. SOUMYAMOL D	PGT CS	PM SHRI K V KANHANGAD	ERNAKULUM
MRS. REENA P V	PGT CS	PM SHRI KV NILESHWAR	ERNAKULUM
MR. VINOD. B	PGT CS	KV NTPC KAYAMKULAM	ERNAKULUM
MR. SREEHARI S	PGT CS	PM SHRI KV AKKULAM	ERNAKULUM

REVIEW TEAM

MR. VIVEK SHRIVASTAVA	VICE PRINCIPAL, PM SHRI K V 2 COLABA, MUMBAI REGION
MRS. NAVNEET KAUR BHATTI	PGT CS, PM SHRI KV IIT POWAI, MUMBAI REGION
MRS PREETI SARKAR	PGT CS, PM SHRI KV O F BHANDARA, MUMBAI REGION

TABLE OF CONTENTS

SNO	TOPIC	PAGES
1	SYLLABUS & MARKS DISTRIBUTIONS	01-02
2	REVISION OF PYTHON TOPICS COVERED IN CLASS XI	03-16
3	WORKING WITH FUNCTIONS	17-27
4	EXCEPTION HANDLING	28-29
5	DATA FILE HANDLING	30-60
6	DATA STRUCTURE	61-67
7	COMPUTER NETWORKS	68-84
8	DATABASE MANAGEMENT	85-118
9	SOLVED CBSE QUESTION PAPER 2024-25	119-136
10	UNSOLVED SAMPLE PAPERS - (Set – 1,2 & 3)	137-146
11	REFERENCES	146

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

Unit wise Syllabus

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.

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.

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 I:COMPUTATIONAL THINKING AND PROGRAMMING

PYTHON REVISION TOUR

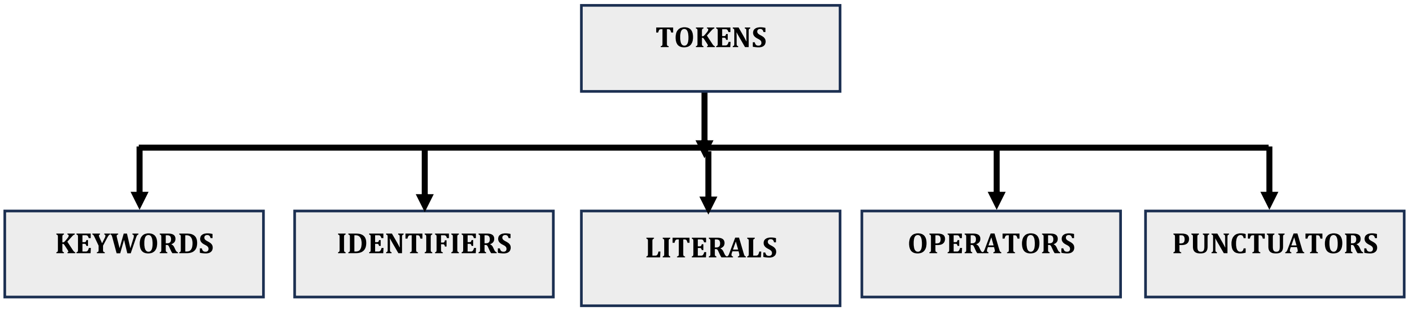
About Python:

- Python programming language was developed by Guido Van Rossum in February 1991.
- Python is based on two programming languages, ABC language and Modula-3
- It is a **case-sensitive** programming language.
- Python is one of the languages that are **not strict about data types**.
- Python works in to modes – **interactive and script**
- **Python is** Interpreted and Cross Platform Language, Free and Open Source, used for both procedural and object-oriented programming.

Basic terminologies:

- **TOKEN / LEXICAL UNIT:** Smallest individual unit in a Programming Language.
- **Types of Tokens** (Keywords, Identifiers, Literals, Operators, Punctuators)

NOTE: Mnemonic: "Kind Intelligent Lions Often Prey"



- **Keywords:** Reserved words having special meaning.

Literals	True, False, None
Operators	not, and, or, is, in
Empty statement	Pass
Conditional Statements	if, elif, else
Iterative statements	for, while, break, continue
Exception Handling	try, except, finally, assert, raise
Functions	def, return, global
File Handling	with
Others	del, import, from, as, lambda, nonlocal, yield, class

- **Identifiers:** These are the names given to **variables, objects, classes** or **functions** etc.

Identifier Naming Rules:

✓ Python identifiers **must begin** with a letter **(A-Z or a-z)** or an **underscore (_)**

✓ It can be followed by **letters, digits (0-9),** or **underscores**.

- ✓ However, Python keywords **cannot** be used as Identifiers
- ✓ Special characters like @, #, or $ **cannot be** while naming Identifiers.
- ✓ Length of identifier is endless and is case sensitive.

- **Literals / Constants:** Data items that have a fixed value are called Literals.
  Example: 5 (integer), 5.9 (float), True (Boolean), "Hello" (string)

- **Operators:** Symbols that trigger some action.

Arithmetic Operators	**, *, /, //, %
Relational Operators	<, <=, >, >=, ==, !=
Logical Operators	not, and, or
Identity Operator	is, is not
Membership Operator	in, not in
Assignment Operator	=, +=, -=, *=, /=, //=, %=

- **Operator Precedence:** Operator precedence defines the priority of operators in an expression.
- **Operator Associativity:** It define the direction in which operators of same precedence are evaluated either from left to right or right to left.

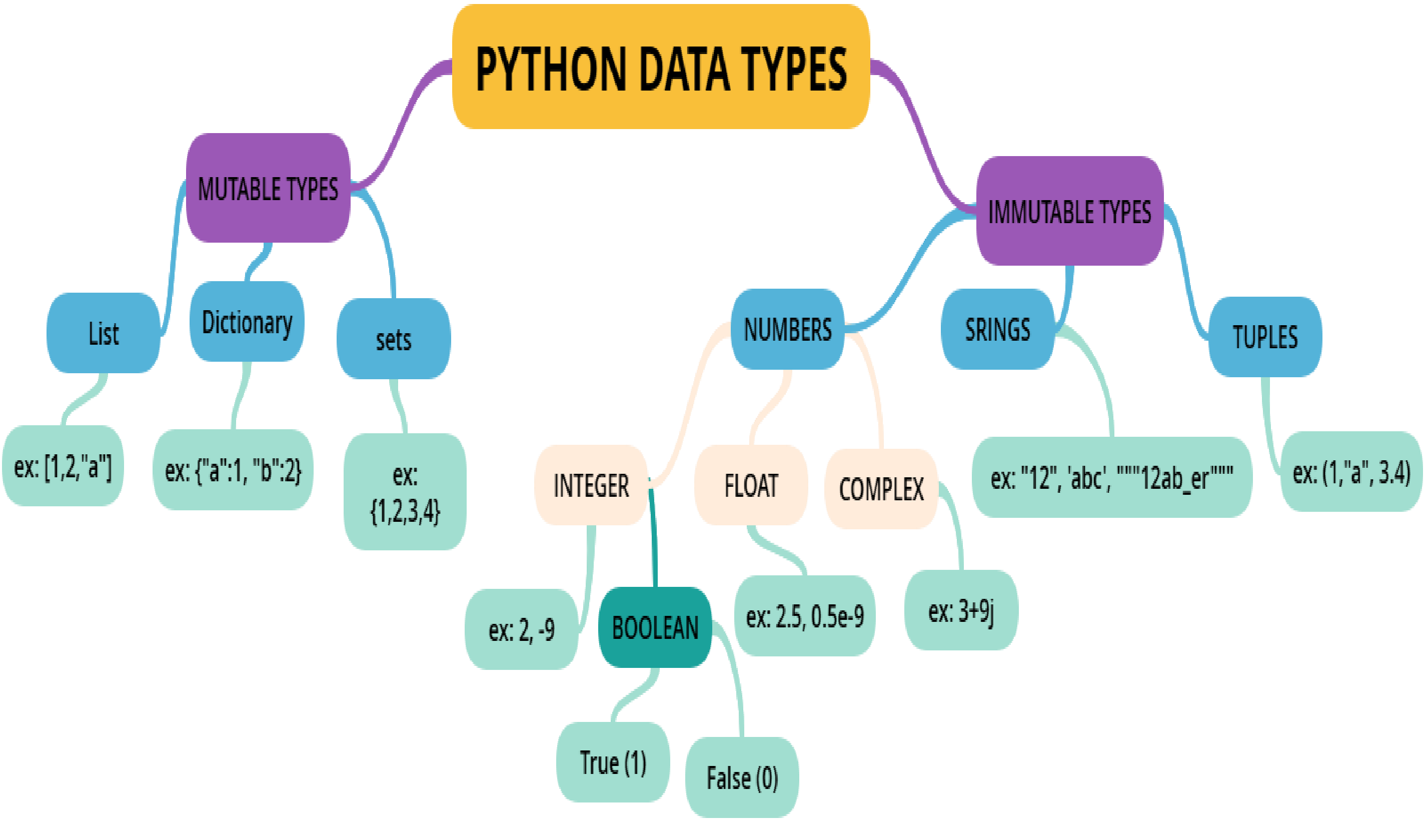
Precedence Level	**Operators**	**Description**	**Associativity**
1 (Highest)	( )	Parentheses (grouping)	N/A
2	x[index], x[attr], x(…), x[…]	Subscription, attribute reference, call	Left to Right
3	**	Exponentiation	Right to Left
4	+x, -x, ~x	Unary plus, minus, bitwise NOT	Right to Left
5	*, /, //, %	Multiplication, division, floor division, modulo	Left to Right
6	+, -	Addition, subtraction	Left to Right
7	<<, >>	Bitwise shift operators	Left to Right
8	&	Bitwise AND	Left to Right
9	^	Bitwise XOR	Left to Right
10	|	Bitwise OR	Left to Right
11	in, not in, is, is not, <, <=, >, >=, !=, ==	Comparisons, identity, membership tests	Left to Right
12	Not	Boolean NOT	Right to Left
13	And	Boolean AND	Left to Right
14	Or	Boolean OR	Left to Right
15(Lowest)	=	Assignment operator	Right to Left

Example : Evaluate 100 + 200 / 10 - 3 * 10
Here, the operators /, * have similar precedence and +, - have same precedence, hence the expression will be evaluated as ((100 + (200 / 10)) – (3 * 10)), which equals to 90.0

- **Punctuators:** Symbols that are used to organize sentence structure. These are used to give syntactic and semantic meaning to the program statement.
  Some commonly used punctuators are:

‘	“	#	\	(	)	[	]	{	}	@	,	:	.	=	;

- **Data Types in Python**



- **Variables:** Variables are containers for storing data values.
  Key points:
  - Variables do not need to be declared or defined in advance.
  - A variable is created when you first assign a value to it.
  - Python is dynamically typed language where data type of the variable is not required statically.
  - **Example:**

```
# Initially assigning an integer value
x = 88                              # x assigned as integer type
print ("x =", x, " Type:", type(x))
# Reassigning a string value to the same variable
x = "Dynamic Typing in Python"  # x assigned as string type
print ("x =", x, " Type:", type(x))
```

Note: Dynamic typing allows user to assign different values to same variable at different places in a program

- ## Conversion of Data Types in Python

Type Promotion (Implicit Type Conversion)	**Type Casting (Explicit Type Conversion):**
Python interpreter automatically converts/ promotes one data type to another higher type without any user involvement	The data type is manually changed by the user as per their requirement. With explicit type conversion, there is a risk of data loss since we are forcing an expression to be changed in some specific data type.

Example:
x = 100
y = 10.6
z = x + y
print(z)
print ("z is of type:”, type(z))
# z is implicitly converted to floating type 	
	Example:
	a, b=5.88, ‘4’
	c=int(a)
	print(“c=”,c) #c=5
	d=int(b)
	print(“d=”,d) #d=4

Python Modules:

- A **module** is a file containing Python code (functions, variables, classes) that can be imported and used in other Python programs.
- Modules help organize code and reuse functionality.

Random Number generation

random Module:

The random module in Python is a powerful tool for generating random numbers and performing random operations.

- To use the random module, **you must import it first**:

  import random

- It provides several functions to produce random values, simulate randomness, and shuffle data.

Name of method	**Method**	**Description**	**Example**
Random Float	random ( )	Generate a random float number between 0.0 and 1.0.	import random random. random ( )
Random Integer	randint ( )	Generate a random integer within a specified range (both values inclusive)	import random random. randint (1, 10)
Random Range	randrange( )	Generate a random number within a specified range with a step.	import random random. randrange (0, 100, 5) # Random number between 0 and 100, step 5

STRING

- A string is a sequence of characters enclosed in **single (' ')**, **double (" ")**, or **triple (''' ''' / """ """)** quotes.
- Strings are **immutable** — once created, they **cannot be changed**.

- Strings are **indexed**, starting from 0 (left to right) and -1 (right to left).
- Use triple quotes (''' ''' or """ """) for **multiline text**.
- Use \ to include special characters (e.g., \n for newline, \t for tab).

LIST

- A list is a collection of items (elements) which are ordered.
- Lists are mutable, meaning their contents can be changed after creation.
- Lists can contain elements of different data types: integers, strings, floats, even other lists.
- Lists are indexed, with the first element at index 0.
- Defined using square brackets [].
- Example:
- my_list = [10, "Hello", 3.14, True]

TUPLE

- **Ordered –** Elements are stored in a specific order.
- **Immutable** – Once created, elements cannot be changed.
- **Allow duplicates** – Tuples can contain repeated values.
- **Can hold different data types** – Integers, strings, lists, etc.
- **Indexing** – Elements can be accessed using indexes.
- **Faster than lists** – Due to immutability.
- Tuples written in **parenthesis ( ).**

Multiple element tuple	Single element tuple	

t=(10,20,30)
tup = 10, 20, 30
	t = (10,)
	Note: t=10
		# Comma is important
		#t is integer

- **Packing Unpacking of Tuple**

```
tup = 10, 20, 30
a, b, c = tup
print(a,b,c)  # 10 20 30
```

DICTIONARY

- A **dictionary** in Python is an **unordered collection** of items.
- Each item is a **key-value pair**.
- Dictionaries are **mutable** (can be changed).
- Defined using **curly braces** {} with key:value pairs.
- **Key**: Must be **unique** and **immutable** (string, number, tuple)
- **Value**: Can be **any datatype** and can be **duplicate**
- Keys are used to access values.

Data type	**Operations**				
	Traversal	**Concatenation (+)**	**Replication (*)**	**Membership (in, not in)**	**Slicing (start:stop:step)**

String ‘Hello’	**Traversing using a for loop**
	for char in s:
	print(char)
	
	Traversing using indexing
	for i in range(len(s)):
	print(s[i])
		"Hello" + " World!"
		
		"Hello World!"
			"Hello" *2
			
			"HelloHello”
				"W" in "World!"
				
				True
				
				“P" not in "Hello World!"
				
				False
					(a)
					s[start:end]
					"Hello"[1:4]
					"ell"
					
					(b)
					s [: end]
					"Hello"[:3]
					"Hel"
					
					(c)
					s[start:]
					"Hello"[2:]
					“Hel”
					
					(d)
					s[start:end:step]
					"Hello"[::2]
					"Hlo"

List [1,2,3]	lst=[10,20,30,40]
	for i in lst:
	print(i)
	
	for i in range(len(lst)):
	print(lst[i])
		[1,2] + [3,4]
		
		[1,2,3,4]
			[0] * 3
			
			[0,0,0]
				3 in [1, 2, 3]
				
				True
					L=[11,12,14,17]
					L[1:3]
					[12,14]
					
					L[:3]
					[11,12,14]

Tuple (1,2,3)	tpl=(10,20,30,40)
	for i in tpl:
	print(i)
	
	for i in range(len(tpl)):
	print(tpl[i])
		(1,2) + (3,4)
		
		(1,2,3,4)
			(1) * 3
			
			(1,1,1)
				3 in (1, 2, 3)
				
				True
					L=(11,12,14,17)
					L(1:3)
					(12,14)
					
					L(:3)
					(11,12,14)

String Functions and Methods

Method	**Explanation**	**Example**	**Output**
len( )	Returns number of characters in a string	len("Hello")	5
capitalize ( )	Capitalizes first letter, rest lowercase	"hello".capitalize( )	"Hello"
title ( )	Capitalizes the first letter of each word	"hello world".title( )	"Hello World"

Method	Explanation	Example	Output
lower ( )	Converts all characters to lowercase	"HELLO".lower( )	"hello"
upper ( )	Converts all characters to uppercase	"hello".upper( )	"HELLO"
count ( )	Counts occurrences of a substring	"banana".count("a")	3
find ( )	Finds first index of substring (-1 if not found)	"hello".find("e")	1
index ( )	Like find ( ) but error if not found	"hello".index("o")	4
endswith( )	Checks if string ends with specified value	"hello”. endswith("lo")	True
startswith( )	Checks if string starts with specified value	"hello”. startswith("he")	True
isalnum( )	Checks if all characters are alphanumeric	"abc123”. isalnum( )	True
isalpha( )	Checks if all characters are alphabetic	"abc”. isalpha( )	True
isdigit( )	Checks if all characters are digits	"123”. isdigit( )	True
islower( )	Checks if all characters are lowercase	"hello”. islower( )	True
isupper( )	Checks if all characters are uppercase	"HELLO”. isupper( )	True
isspace( )	Checks if string contains only whitespace	" “. isspace( )	True
lstrip( )	Removes leading spaces or characters	" hello”. lstrip( )	"hello"
rstrip( )	Removes trailing spaces or characters	"hello “. rstrip( )	"hello"
strip( )	Removes both leading and trailing spaces	" hello “. strip ( )	"hello"
replace( )	Replaces all occurrences of a substring	"hello".replace("l", "x")	"hexxo"
join ( )	Joins elements of a sequence with string as separator	“*”. join (["a", "b", "c"])	'a*b*c'
partition ( )	Splits string into 3 parts: before, separator, after	"hello world".partition(" ")	('hello', ' ','world')
split ( )	Splits string into list at separator	"a,b,c".split(",")	['a', 'b', 'c']

List Functions and Methods

Function / Method	Explanation	Syntax	Example	Output
len( )	Returns the number of elements in the list	len(list)	len([1, 2, 3])	3
list ( )	Converts an iterable (like string, tuple) to a list	list(iterable)	list("abc")	['a', 'b', 'c']

append ( )	Adds a single element at the end of the list	list.append(item)	a = [1, 2]
			a.append(3)
				[1, 2, 3]

extend ( )	Adds elements of another list to the end	list.extend(iterable)	a = [1]
			a.extend([2, 3])
				[1, 2, 3]

insert ( )	Inserts an element at a given position	list.insert(index, item)	a = [1, 3]
			a.insert(1, 2)
				[1, 2, 3]

count ( )	Returns number of times an element appears	list.count(item)	[1, 2, 2, 3]. count (2)	2

index ( )	Returns index of first occurrence of element	list.index(item)	a=[10, 20, 30]
			a.index(20)
				1

remove ( )	Removes first occurrence of a specified element	list.remove(item)	a = [1, 2, 3, 2];
			a.remove(2)
				[1,3,2]

pop ( )	Removes and returns element at given index (default: last)	list.pop([index])	a = [1, 2, 3];
			a.pop( )
				3
				list becomes [1, 2]

reverse ( )	Reverses the list in place	list.reverse( )	a = [1, 2, 3];
			a.reverse( )
				[3, 2, 1]

sort ( )	Sorts the list in ascending order (modifies original list)	list.sort( )	a = [3, 1, 2];
			a.sort( )
				[1, 2, 3]

sorted ( )	Returns a new sorted list (original list unchanged)	sorted(list)	sorted ([3, 1, 2])	[1, 2, 3]
min ( )	Returns the smallest item in the list	min(list)	min ([2, 3, 1])	1
max ( )	Returns the largest item in the list	max(list)	max ([2, 3, 1])	3
sum ( )	Returns the sum of all numeric items in the list	sum(list)	sum ([1, 2, 3])	6

Tuple Functions and Methods

Function	Description	Syntax	Example	Output
len( )	Returns the number of elements in the tuple	len(t)	len((10, 20, 30))	3
max ( )	Returns the maximum element	max(t)	max((5, 12, 8))	12
min ( )	Returns the minimum element	min(t)	min((5, 12, 8))	5
sum ( )	Returns the sum of all elements	sum(t)	sum((10, 20, 30))	60
sorted ( )	Returns a sorted list (not tuple) of elements	sorted(t)	sorted((50, 10,20))	[10, 20,50]
tuple( )	Converts a sequence into a tuple	tuple(seq)	tuple([1, 2, 3])	(1, 2, 3)
count(x)	Returns the **number of times** the value x appears in the tuple	tuple.count(x)	(1, 2, 2, 3).count(2)	2
index(x)	Returns the **index of the first occurrence** of the value x in the tuple	tuple.index(x)	(10, 20,30).index(20)	1

WORKING WITH DICTIONARIES

```
d = {'name': 'John', 'age': 16, 'grade': 'A'}
```

Operation	Code Example	Description	Output
Access a value	d['name']	Returns the value of the key	'John'
Add a new pair	D['city'] = 'Delhi'	Adds new key-value to the dictionary	{'city': 'Delhi'}
Modify a value	d['age'] = 17	Changes the value of existing key	{'age': 17}
Delete a pair	del d['grade']	Deletes key-value pair	Removes 'grade' key
Check key existence	'name' in d	Returns True if key exists	True
Loop through keys	for k in d:	Iterates over keys	'name', 'age', etc.
Loop through items	for k, v in d.items( ): print(k, v)	Iterates through key- value pairs	name John age 16 city Delhi

Dictionary Functions & Methods

Function/ Method	Example	Output
len( )	len({'a':1, 'b':2})	2
dict( )	dict(a=1, b=2)	{'a': 1, 'b': 2}
keys ( )	{'x':1, 'y':2}.keys( )	dict_keys(['x', 'y'])
values ( )	{'x':1, 'y':2}.values( )	dict_values([1, 2])
items ( )	{'x':1, 'y':2}.items( )	dict_items([('x', 1), ('y', 2)])
get ( )	{'name':'John'}.get('name')	'John'
update ( )	{'a':1}.update({'b':2})	{'a':1, 'b':2}
Del	d={'a':1}; del d['a']; print(d)	{}
clear ( )	d={'a':1}; d.clear( ); print(d)	{}
fromkeys( )	dict.fromkeys(['a', 'b'], 0) ** Creates a new dictionary from a sequence of keys with a common value.*	{'a': 0, 'b': 0}
copy ( )	d={'a':1}; d2=d.copy( ); print(d2)	{'a': 1}
pop ( )	d={'a':1,'b':2}; print(d.pop('a')); print(d) ** Removes the item with the specified key and returns its value.*	1 {‘b’:2}
popitem( )	{'a':1,'b':2}.popitem( )	('b', 2)
setdefault( )	d={'a':1}; d.setdefault('b',2); print(d) ** Returns value of key. If key not found, inserts it with a specified default value.*	{‘a’:1,’b’:2}
max ( )	max({'a':1, 'b':2})	'b'
min ( )	min({'a':1, 'b':2})	'a'
sorted ( )	sorted({'c':3, 'a':1, 'b':2})	['a', 'b', 'c']

Multiple Choice Questions

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

```
a = "10"
b = "20"
print (int (a + b) + int(a) * int(b))
```

a) 1220 b) 410 c) 620 d) 30

2 Which of the following code snippets correctly checks if a string s is a palindrome (case-insensitive)?

a) s == s[::-1] b) s.lower( ) == s.upper( )[::-1]

c) s.lower( ) == s.lower( )[::-1] d)s== s.reverse( )

3	What will be the output of the following code? x = 5 y = "2" print (x * int(y) + len(y + str(x))) a) 11 b) 12 c) 15 d) 10
4	Which of the following code snippets will raise an error for s = "Python"? a) print(s[10]) b) print(s[1:10]) c) print (s [-1: -3]) d) print(s[:2])

5	What will be the output of following python code:
	s = 'Programming'
	print(s.split("m"))
	a. ['Progra', ' ', 'ing'] b. ['Progra', 'ing']. c. ['Progra', 'm', 'ing'] d. [Progra', 'ming']

6	What will be the output of following python code:
	t1 = 1,2,3
	t2 = (1,2,3)
	print(t1 is t2)
	a. True b. 1 c. False d. 0

7	State whether the following statement is True or False: An exception may be raised even if the program is syntactically correct.
8	State whether the following statement is True or False: In Python, if an exception is raised inside a try block and not handled, the program will terminate without executing any remaining code in the finally block.

9	Select the correct output of the code:
	a = "foobar"
	a = a.partition("o")
	print(a)
	a. ["fo","","bar"] b. ["f","oo","bar") c. ["f","o","bar"] d. ("f","o","obar")

10	State whether the following statement is True or False: An exception may be raised even if the program is syntactically correct.
11	What will be the output of the following statement ? print(6+5/4**2//5+8) a.-14.0 b.14 c.-14 d. 1

12	Select the correct output of the code:
	S = "text#next"
	print(S.strip("t"))
	a. ext#nex b. ex#nex c. text#nex d. ext#next

13	Identify the valid Python identifier from the following : a. 2user b. user@2 c. user_2 d. user 2

14	Consider the statements given below and then choose the correct output from the given options :
	Game="World Cup 2023"
	print(Game[-6::-1])
	a. CdrW b. ce o c. puC dlroW d. Error

15	Predict the output of the following Python statements : import statistics as s s.mode ([10, 20, 10, 30, 10, 20, 30]) a. 30 b. 20 c.10 d. 18.5

ANSWERS									
1	A	2	C	3	b	4	A	5	a
6	a	7	False	8	False	9	D	10	False
11	B	12	A	13	C	14	C	15	C

Assertion/Reasoning Type Questions

Directions: In the following questions, A statement of Assertion (A) is followed by a statement of Reason (R). Mark the correct choice as:

(A) Both A and R are true and R is the correct explanation of A
(B) Both A and R are true and R is not the correct explanation of A
(C) A is true but R is false
(D) A is false but R is true

1	Assertion (A): Lists can store only elements of the same data type. Reason (R): Python is a dynamically typed language
2	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
3	Assertion. Dictionaries are mutable, hence its keys can be easily changed. Reason. Mutability means a value can be changed in place without having to create new storage for the changed value.
4	Assertion (A): You can add an element in a dictionary using key:value pair. Reasoning (R): A new (key:value) pair is added only when the same key doesn't exist in the dictionary. If the key is already present, then the existing key gets updated and the new entry will be made in the dictionary.
5	Assertion (A): Tuples hold a sequence of homogenous elements. Reason (R): Tuples are immutable

ANSWERS ASSERTION REASONING

1	**D**	**2**	**D**	**3**	**D**	**4**	**A**	**5**	**D**

Short Answer Questions/Long Answer Questions

1	a. Given is a Python list declaration : Listofnames=["Aman","Ankit","Ashish","Rajan","Rajat"] Write the output of : print (Listofnames [–1:–4:–1]) b. Consider the following tuple declaration : tup1=(10,20,30,(10,20,30),40) Write the output of : print(tupl.index(20))

2	a. Given is a Python string declaration :
	NAME = "Learning Python is Fun"
	Write the output of : print(NAME[-5:-10:-1])
	b. Write the output of the code given below :
	dict1={1:["Rohit",20], 2:["Siya",90]}
	dict2={1:["Rahul",95], 5:["Rajan",80]}
	dict1.update(dict2)
	print(dict1.values( ))

3	Write the output displayed on execution of the following Python code : LS=["HIMALAYA","NILGIRI","ALASKA","ALPS"]

```
D={}
for S in LS :
   if len(S)%4 == 0:
         D[S] = len(S)
for K in D :
   print(K,D[K], sep = "#")
```

4 Write the Python statement for each of the following tasks using built-in functions/methods only :

(i) To remove the item whose key is "NISHA" from a dictionary named Students. For example, if the dictionary Students contains {"ANITA":90, "NISHA":76, "ASHA":92}, then after removal the dictionary should contain{"ANITA":90,"ASHA":92}

(ii) To display the number of occurrences of the substring "is" in a string named message. For example if the string message contains "This is his book", then the output will be 3.

5 A tuple named subject stores the names of different subjects. Write the Python commands to convert the given tuple to a list and thereafter delete the last element of the list.

6 Predict the output of the following code :

```
d={"IND":"DEL","SRI”:"COL","CHI":"BEI"}
str1=""
for i in d:
   str1=strl+str(d[i])+"@"
   str2=str1[:−1]
print (str2)
```

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

(i) To delete an element 10 from the list lst.

(ii) To replace the string "This" with "That" in the string str1.

8 A dictionary dict2 is copied into the dictionary dict1 such that the common key’s value gets updated. Write the Python commands to do the task and after that empty the dictionary dict1.

9 Write difference between mutable and immutable data types? Write name of any two mutable data types.

10
i. Which operator in python has right to left associativity?
ii. Evaluate the following expression and write output: >>> 5//4**3%6+2

11 If str1= “foundation stone” and str2= “strong base”, then
Write the Python statements for each of the following tasks using Built-in functions/methods only:

- To count the occurrence of the letter ‘n’ in str1.
- To convert the string str2 in a list and each word of the string should become the element of the list

12 Write a user defined function in Python named showGrades(S) which takes the dictionary S as an argument. The dictionary, S contains Name:[Eng,Math,Science] as key:value pairs. The function displays the corresponding grade obtained by the students according to the following grading rules :

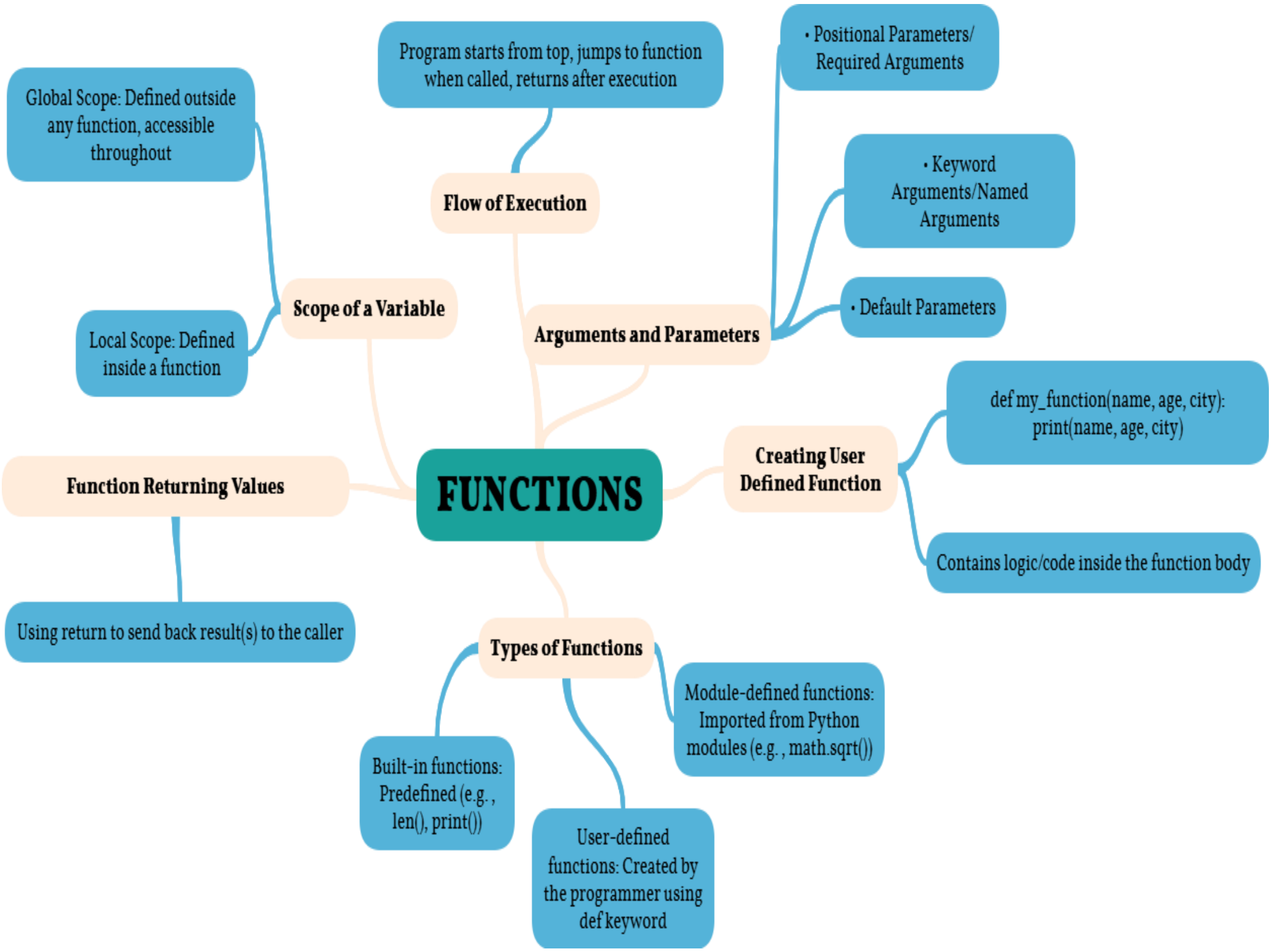
Average of Eng,Math,Science	Grade
>=90	A
<90 but >=60	B
<60	C

	For example : Consider the following dictionary S={"AMIT":[92,86,64],"NAGMA":[65,42,43],"DAVID":[92,90,88]} The output should be : AMIT – B NAGMA – C DAVID – A

Answers

<table>
<tr><td>1</td><td>a.['Rajat', 'Rajan', 'Ashish']
b. 1</td></tr>
<tr><td>2</td><td>a. si no
b. dict_values([['Rahul', 95], ['Siya', 90], ['Rajan', 80]])</td></tr>
<tr><td>3</td><td>HIMALAYA#8
ALPS#4</td></tr>
<tr><td>4</td><td>i) Students.pop("NISHA")
(ii) print(message.count("is"))
message.count("is")</td></tr>
<tr><td>5</td><td>subject=list(subject)
subject.pop( )</td></tr>
<tr><td>6</td><td>DEL@COL@BEI</td></tr>
<tr><td>7</td><td>(i) lst.remove(10)
ii) str1.replace("This","That")</td></tr>
<tr><td>8</td><td>dict1.update(dict2)
dict1.clear( )</td></tr>
<tr><td>9</td><td><table>
<tr><th>Mutable Data Type</th><th>Immutable Data Type</th></tr>
<tr><td>Value can be changed at particular index.
Examples:
list, dictionary</td><td>Value cannot be changed/assigned at particular index.
Examples
Numbers, strings</td></tr>
</table></td></tr>
<tr><td>10</td><td>i. Exponent operator (**)
ii. 2</td></tr>
<tr><td>11</td><td>A. print(str1.count('n'))
B. L=str2.split( )</td></tr>
<tr><td>12</td><td>def showGrades(S):
    for K, V in S. items( ):
        if sum(V)/3>=90:
            Grade="A"
        elif sum(V)/3>=60:
            Grade="B"
        else:
            Grade="C"
        print(K,"–",Grade)
S={"AMIT":[92,86,64],"NAGMA":[65,42,43],"DAVID":[92,90,88]}
showGrades(S)</td></tr>
</table>

WORKING WITH FUNCTIONS



FUNCTIONS
Flow of Execution
Program starts from top, jumps to function when called, returns after execution
Scope of a Variable
Global Scope: Defined outside any function, accessible throughout
Local Scope: Defined inside a function
Arguments and Parameters
• Positional Parameters/ Required Arguments
• Keyword Arguments/Named Arguments
• Default Parameters
Creating User Defined Function
def my_function(name, age, city): print(name, age, city)
Contains logic/code inside the function body
Function Returning Values
Using return to send back result(s) to the caller
Types of Functions
Built-in functions: Predefined (e.g., len(), print())
User-defined functions: Created by the programmer using def keyword
Module-defined functions: Imported from Python modules (e.g., math.sqrt())

- **FUNCTIONS:** Function can be defined as a named group of instructions that accomplish a specific task when it is invoked

- **ADVANTAGES** OF USING FUNCTIONS
  - Increases readability
  - Reduces code length
  - Increases reusability
  - Modularity

- **FUNCTIONS CAN BE CATEGORIZED AS:**
  1. Built-in function
  2. Functions defined in module
  3. User defined functions

a) **Built-in functions:** Built-in functions are the ready-made functions in Python that are frequently used in programs. We don't have to import any module to use these functions.

Name of the function	**Description**	**Example**
abs (x)	It returns the positive value of x.	>>>abs(-45) 45 >>>abs(119) 119

range (start, stop [, step])	It is used to generate a sequence of numbers.	>>> range(10)
		[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
		>>> range(1, 11)
		[1,2, 3, 4, 5,6, 7, 8, 9, 10]
		>>>range(0, 30, 5)
		[0, 5, 10,15,20, 25]

round (x [, n])	It returns float x rounded to n digits from the decimal point, where x and n are numeric expressions. If n is not provided then x is rounded to 0 decimal digits.	>>>round(80.23456, 2) 80.23 >>>round(-100.000056, 3) -100.0 >>> round (80.23456) 80

b) **Functions defined in module:** A module is a file containing Python definitions (i.e., functions) and statements. To use these modules in the program, a programmer needs to import the module by using either the import statement or the from statement.

- **import statement:** It is simplest and most common way to use modules in our code. syntax is:

```
import modulename1 [, modulename2,          ]
```

Example: *import math*

To use/ access/invoke a function, you will specify the module name and name of the function- separated by dot (.).

Example: >>> value= math.sqrt (25) # dot notation

- **From Statement:** It is used to get a specific function in the code instead of the complete module file. For modules having large no. of functions, it is recommended to use from instead of import.

Its syntax is:

```
>>> from modulename import functionname [, functionname…..]
>>>from modulename import * (Import everything from the file)
```

Example: >>> from math import sqrt

```
          value = sqrt (25)
```

Some functions from math module are:

Name of function	Description	Example
ceil (x)	It returns the smallest integer not less than x, where x is a numeric expression.	math.ceil(-45.17) -45.0 math.ceil(100.12) 101.0 math.ceil(100.72) 101.0
floor (x)	It returns the largest integer not greater than x, where x is a numeric expression.	math.floor(-45.17) -46.0 math.floor(100.12) 100.0 math.floor(100.72) 100.0
fabs (x)	It returns the absolute value of x, where x is a numeric value.	math.fabs(-45.17) 45.17 math.fabs(100.12) 100.12 math.fabs(100.72) 100.72
pow (x, y)	It returns the value of xy, where x and y are numeric expressions.	math.pow(100, 2) 10000.0 math.pow(100, -2) 0.0001 math.pow(2, 4) 16.0 math.pow(3, 0) 1.0
sqrt (x)	It returns the square root of x for x > 0, where x is a numeric expression.	math.sqrt(100) 10.0 math.sqrt(7) 2.6457513110645907

Some functions from statistics module are:

Name of the function	Description	Example
mean(x)	It returns arithmetic mean	>>>statistics.mean([11,24,32,45,51]) 32.6
median(x)	It returns median (middle value) of x	>>>statistics.median([11,24,32,45,51]) 32
mode(x)	It returns mode (the most repeated value)	>>>statistics.mode([11,24,11,45,11]) 11 >>>statistics.mode(("red","blue","red")) red

c) User defined functions

a. The keyword **def** is used to define a function.
b. After the keyword comes an identifier i.e. **name of the function**, followed by parenthesized **list of parameters** and the **colon** which ends up the line.
c. Next follows the **block of statement(s)** that are the part of function.

Syntax:

```
def function_name([parameter1, parameter2,....]) :
                statement1
                statement2
                ...
                [return <value1, value2, ...> ]
```

1. Formal Parameters or Parameters: These are the values provided in Parenthesis when we write a function Header.
2. Actual Parameters or Arguments: These are the values passed to the function when it is called/invoked.

Example:

```
def area_of_circle(radius ): # radius Formal Parameter
   print (“Area of circle = “,3.1416*radius*radius)

r=int (input (“Enter radius of circle”))
ar=area_of_circle(r)              # r - Actual Parameter or Argument
```

```
OUTPUT
Enter radius of circle 10
Area of circle = 314.16
```

TYPES OF ARGUMENTS / PARAMETERS

1. Positional Parameters/ Required Arguments
2. Keyword Arguments/Named Arguments
3. Default Parameters

a) Positional Parameters/ Required Arguments: The function call statement must match the number and positional order of arguments as defined in the function definition, then it is called the positional parameters.

Example:

```
def Test(x,y,z):       # x,y,z are positional parameters
```

Then we can call function using following statements p,q,r=3,4,6

```
Test(p,q,r)      #3 variables are passed
Test (4, q,r)    # 1 literal and 2 variables are passed
```

b) Keyword Arguments/Named Arguments: In Python, keyword arguments, also known as named arguments, are a way to pass arguments to a function by explicitly specifying the parameter name along with its value.

```
def my_function(name, age, city):
   print(name, age, city)
# Keyword arguments
my_function(age=25, city="London", name="Bob")
```

c) Default Parameters: The parameters which are assigned with a value in function header while defining the function are known as default parameters. If no value is provided in function call, then the default value will be taken by the parameter.

Example:

```
def SICal(amount, rate, time=10):     #time has default parameter
```

NOTE: Default parameters will be written in the end of the function header, which means positional parameter cannot appear to the right side of default parameter.

FUNCTION RETURNING VALUE(S)

A function can returns multiple values. The return values should be a comma separated list of values. **The multiple return values are returned as a tuple**. We can unpack the received value by specifying the same number of variables on the left side of the function call.

Example 1:
def add10(x,y,z):
return x+10, y+10, z+10
x, y, z=10, 20, 30
result=add10(x,y,z)
print(result)
	Example 2:
	def add10(x,y,z):
	return x+10, y+10, z+10
	x, y, z=10, 20, 30
	a,b,c=add10(x,y,z) #unpack
	print(a,b,c)

Output (20, 30, 40)	Output: 20 30 40

SCOPE OF VARIABLES*:*

The part of the program where a variable is accessible can be defined as the scope of that variable. There are two types of scope for variables:

1. Local Scope: A variable declared in a function-body is said to have local scope. It cannot be accessed outside the function.

2. Global Scope: A variable declared in top level segment (main) of a program is said to have a global scope.

Example:

```
def Sum (x, y) :  # x,y,z local
   z = x + y
   return z
a = 5           # a,b,s global
b = 7
s = Sum (a, b)
print(s)
```

Multiple Choice Questions

1	Which of the following is a valid function name? a) Start_game( ) b) start game ( ) c) start-game( ) d) All of the above
2	If the return statement is not used in the function, then which type of value will be returned by the function? a) int b) str c) float d) None

3	What is the minimum and maximum value of c in the following code snippet?
	import random
	a=random.randint(3,5)
	b = random.randint(2,3)
	c = a + b
	print(c)
	a) 3, 5 b) 5, 8 c) 2, 3 d) 3, 3

4	pow ( ) function belongs to which library? a) math b) string c) random d) maths
5	What is the data type of the object below? L = (1, 23, 'hello',1) a) list b) dictionary c) array d) tuple

6 What is returned by int (math.pow(3, 2))?
a) 6 b) 9 c) error, third argument required d) error, too many arguments

7 Which of the following is not a type conversion function?
a) int( ) b) str ( ) c) input ( ) d) float ( )

8 Identify the module to which the following function load ( ) belong to?
a) math b) random c) pickle d) sys

9 How many argument(s) a function can receive?
a) Only one b) 0 or many c) Only more than one d) At least one

10 Give the output

```
def fun ( ):
    global a
    a=10
    print(a)
a=5
fun ( )
print(a)
```

```
a)  10            b) 5        c) 5              d) 10
    10               10          5                 5
```

11 Value returning functions should be generally called from inside of an expression
a) True b) False

12 The variable declared inside the function is called a___________________variable
a) global b) local c) external d) none of the above

13 These are predefined functions that are always available for use. For using them we don't need to import any module
a) built in function b) pre-defined function
c) user defined function d) none of the above

14 The______________of a variable is the area of the program where it may be referenced
a) external b) global c) scope d) local

15 If you want to communicate between functions i.e. calling and called statement, then you should use
a) values b) return c) arguments d) none of the above

16 Find the flow of execution of the following code:

```
1.     def calculate (a, b):
2.          res=a**b
3.         return res
4.
5.     def study(a):
6.          ans=calculate(a,b)
7.         return ans
8.
9.     n=2
10.    a=study(n)
11.    print(a)
```

a) 1 > 5 > 9 > 10 >5>6 >1> 2 > 3 >6 7 > 10>11 b) 5 > 9 > 10 > 6 > 2 > 3 > 7 > 11
c) 9 > 10 > 5 > 1 > 6 > 2 > 3 > 7 > 11 d) None of the above

17 A_____________can be skipped in the function call statements

a) named parameter b) default parameter
c) keyword parameters d) all of the above

18 Write the output of the following:

```
a= (10, 12, 13, 12, 13, 14, 15)
print(max(a) + min(a) + a.count(2))
```

a) 13. b) 25 c) 26 d) Error

19 What is wrong with the following function definition?

```
def greet (name="Guest", age):
    print (name, "is", age, "years old"))
```

a). The syntax of print statement is wrong
b). Default parameter must follow required parameter
c). Function name cannot be greet
d). Nothing is wrong

20 Which of the following function definitions is INVALID?
a). def func(a, b=2, c=3): pass. b). def func(a=1, b, c=2): pass
c). def func(a, b, c=5): pass. d). def func(a=1, b=2, c=3): pass

ANSWERS

1	**A**	**2**	**D**	**3**	**B**	**4**	**A**	**5**	**D**	**6**	**B**
7	**C**	**8**	**C**	**9**	**B**	**10**	**A**	**11**	**A**	**12**	**B**
13	**A**	**14**	**C**	**15**	**C**	**16**	**A**	**17**	**B**	**18**	**B**
19	**B**	**20**	**B**								

ASSERTION AND REASONING 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

1 Assertion(A): If the arguments in a function call match the number and order of arguments as defined in the function definition, such arguments are called the positional arguments.
Reasoning(R): During a function call, the argument list first contains default arguments followed by positional arguments.

2 Assertion(A): The random module is a built-in module to generate the pseudo-random values.
Reason(R): The randrange( ) function is used to generate a random number between the specified range in its parameter.

3 Assertion (A): Global variable is declared outside the all the functions.
Reasoning (R): It is accessible through out all the functions.

4 Assertion (A): Built-in function is predefined in the language that are used directly.
Reason (R): print ( ) and input ( ) are built-in functions

5 Assertion (A): - In Python, statement return [expression] exits a function.
Reasoning (R): - Return statement passes back an expression to the caller.
A return statement with no arguments is the same as return None.

ANSWERS	
1	(c) A is True but R is False
2	(b) Both A and R are true and R is not the correct explanation for A
3	(a) Both A and R are true and R is the correct explanation for A
4	(a) Both A and R are true and R is the correct explanation for A
5	(a) Both A and R are true and R is the correct explanation for A

Short Answer Questions/Long Answer Questions

1 Rewrite the following Python program after removing all the syntactical errors (if any), underlining each correction:

```
def checkval
   x = input ("Enter a number")
   if x % 2 =0:
      print (x, "is even")
   elseif x<0:
      print (x, "should be positive")
   else:
      print (x, "is odd")
```

2 Mani Ayyar, a python programmer, is working on a project which requires him to define a function with name CalculateInterest( ).

He defines it as:

```
def CalculateInterest (Principal, Rate=.06, Time):      # Code
```

But this code is not working; Can you help Mani Ayyar to identify the error in the above function and with the solution?

3 Predict the possible output(s) of the following code. Also specify the maximum and minimum value that can be assigned to the variable R when K is assigned value as 2.

```
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#

4 Predict the output of the following code fragment

```
def display (x=2, y=3):
    x=x+y
    y+=2
    print(x,y)
display ( )
display (5,1)
display (9)
```

```
a)  5 5                 b)12 5                c)  5  6            d)  5   5
    6  3                   6  3                  12  5                7   7
    12 5                   5 5                    6   3               6   6
```

5 Predict the output of the following code snippet:

```
def Execute(M):
   if M%3==0:
      return M*3
   else:
    return M+10;
def Output(B=2):
  for T in range (0, B):
    print (Execute(T),"* ", end="")
  print ( )
Output (4)
Output ( )
Output (3)
```

6. Find the output of the following program:

```
def ChangeIt(Text,C):
  T=""
  for K in range(len(Text)):
    if Text[K]>='F' and Text[K]<='L':
      T=T+Text[K]. lower ( );
    elif Text[K]=='E' or Text[K]=='e':
      T=T+C;
    elif K%2==0:
      T=T+Text[K]. upper ( )
    else:
      T=T+T[K-1]
  print(T)
OldText="pOwERALone"
ChangeIt(OldText,"%")
```

7 What possible outputs are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum value that can be assigned to each of the variables L and U.

```
import random
Arr= [10,30,40,50,70,90,100]
L=random.randrange(1,3)
U=random.randrange(3,6)
for i in range (L, U+1):
    print (Arr[i],"@", end="")
```

i) 40 @50 @ ii) 10 @50 @70 @90 @

iii) 40 @50 @70 @90 @ iv) 40 @100 @

8 What will be the output of the following code

```
 total=0
def add(a,b):
        global total
        total=a+b
        print(total)
add (6,6)
print(total)
```

a) 12 b) 12 c) 0 d) None of these

9 What possible output(s) are expected to be displayed on screen at the time of execution of the following code? Also specify the maximum and minimum value that can be assigned to variable X.

```
import random
```

```
L= [10,7,21]
X=random.randint(1,2)
for i in range(X):
   Y=random.randint(1, X)
print(L[Y],"$", end=" ")
```

(i)10 $ 7 $ (ii) 21 $ 7 $ (iii) 21 $ 10 $ (iv) 7 $

10 Vivek has written a code to input a number and check whether it is even or odd number. His code is having errors. Rewrite the correct code and underline the corrections made.

```
Def checkNumber(N):
status = N%2
return
#main-code
num=int (input (“Enter a number to check :))
k=checkNumber(num)
if k = 0:
print (“This is EVEN number”)
else:
print (“This is ODD number”)
```

11 Find and write the output of the following Python code:

```
def changer(p,q=10):
   p=p/q
   q=p%q
   print(p,"#",q)
   return p
a=200
b=20
a=changer(a,b)
print(a,"$",b)
a=changer(a)
print(a,"$",b)
```

12 Write a function INDEX_LIST(L), where L is the list of elements passed as argument to the function. The function returns another list named ‘indexList’ that stores the indices of all Non-Zero Elements of L.

For example: If L contains [12,4,0,11,0,56] The indexList will have – [0,1,3,5]

ANSWERS

1

```
def checkval ( ):
  x = int (input ("Enter a number"))
  if x % 2 == 0:
     print (x, "is even")
  elif x<0:
     print (x, "should be positive")
  else:
       print (x, "is odd")
```

2 Non-default argument must appear before default argument

Correct definition: **def CalculateInterest(Principal, Time ,Rate=.06,):**

3	**b)** wait#stop# **Max value of R=1, Min value of R=0**
4	Ans. a)
5	Ans. 0 *11 *12 *9 * 0 *11 * 0 *11 *12 *
6	PPW%RRllN%
7	Ans. Options i and iii i)40 @50 @ iii) 40 @50 @70 @90 @ Maximum value of L and U : L=2 ,U=5
8	Ans : a)
9	Ans. iv) 7 $ Maximum value of x is 2 Minimum value of x is 1

10	**def** checkNumber(N): # Def should be def
	status = N%2
	return status
	#main-code
	num=int(input(" Enter a number to check : **"))** # Message not enclosed within quotation mark
	k=checkNumber(num)
	if k **==** 0:
	print("This is EVEN number")
	else:
	print("This is ODD number")

11	Ans. 10.0 # 10.0 10.0 $ 20 1.0 # 1.0 1.0 $ 20

12	def INDEX_LIST(L):
	indexList= []
	for i in range(len(L):
	if L[i]! =0:
	indexList.append(i)
	return indexList

EXCEPTION HANDLING IN PYTHON

Exception: An error that occurs **during program execution** is called an **exception**.
Exceptions interrupt the normal flow of a program.

Common examples of exceptions:

- Dividing by zero (ZeroDivisionError)
- Accessing a missing file (FileNotFoundError)
- Using an undefined variable (NameError)

Without handling exceptions, a program **crashes** when an error occurs.

Need for Exception Handling

- Prevents program from crashing abruptly.
- Helps in providing **user-friendly error messages**.
- Ensures **smooth program execution** even after an error.

Handling Exceptions using try-except-finally Blocks

Python provides a **structured way** to handle exceptions using three keywords:

Keyword	Purpose
try	Used to **write code** that may cause an exception.
except	Used to **catch and handle** the exception.
finally	Used to **execute code** whether an exception occurs or not (cleanup code).

```
try:
   # Code that might raise an exception
except ExceptionType:
   # Code that runs if the exception occurs
finally:
   # Code that will run no matter what (optional)
```

try block → Python runs the code inside try.
If an **exception occurs**:
The try block is exited immediately.
The control moves to the **except block**.
finally block always runs after the try/except.

Example 1: Handling a division by zero error

```
try:
   a = int(input("Enter numerator: "))
   b = int(input("Enter denominator: "))
   result = a / b
   print("Result =", result)
except ZeroDivisionError:
   print("Error: Cannot divide by zero!")
finally:
   print("Execution complete.")
```

Example 2: Multiple except blocks

```
try:
   num = int(input("Enter a number: "))
   print("Reciprocal is", 1/num)
except ValueError:
   print("Invalid input! Please enter an integer.")
except ZeroDivisionError:
   print("Cannot find reciprocal of zero.")
finally:
   print("Thank you!")
```

Output 1 (if denominator is 0): Enter numerator: 10 Enter denominator: 0 Error: Cannot divide by zero! Execution complete.	**Output 2 (normal input):** Enter numerator: 10 Enter denominator: 2 Result = 5.0 Execution complete.

Important Points

- try must be followed by at least one except, finally, or both.
- If no exception occurs, **except block is skipped**.
- finally block is **always executed**, even if:
    - An exception occurs
    - No exception occurs
    - return, break, or continue statements are used inside try/except

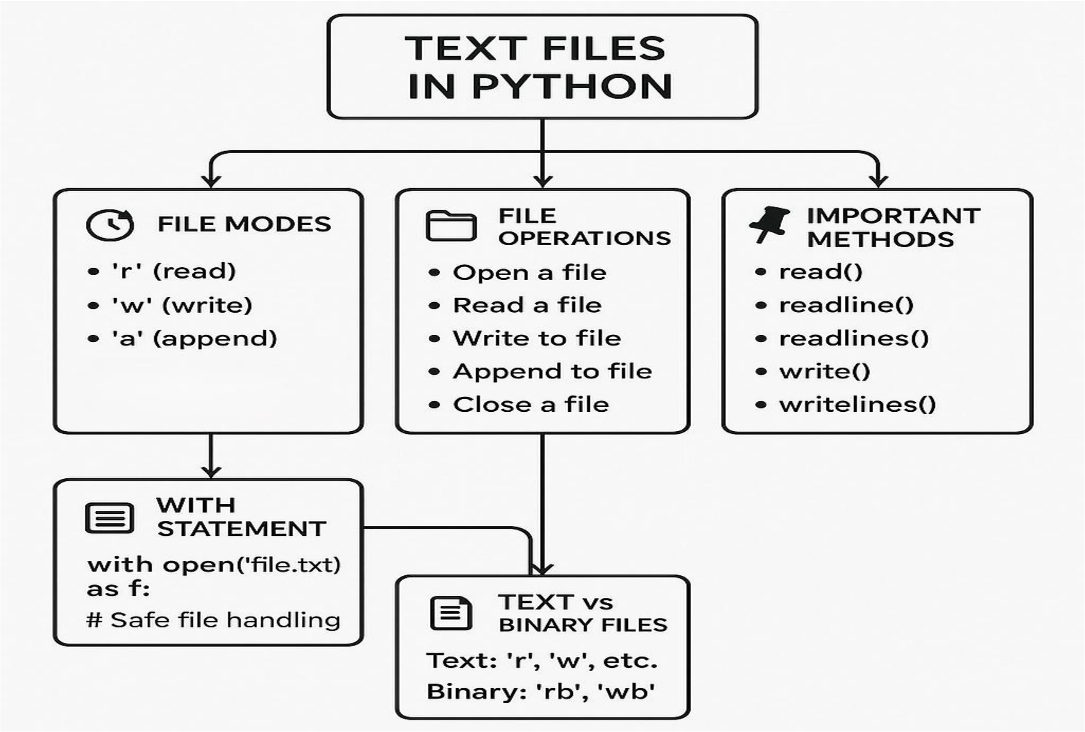
QUESTIONS:

1	Consider the statements given below and then choose the correct output from the given options: N='5' try: print('WORD' + N, end='#') except: print('ERROR',end='#') finally: print('OVER') (A) ERROR# (B) WORD5#OVER C) WORD5# (D) ERROR#OVER
2	State whether the following statement is True or False: While handling exceptions in Python, name of the exception has to be compulsorily added with **except** clause.
3	State whether the following statement is True or False: The finally block in Python is executed only if no exception occurs in the try block.
4	State whether the following statement is True or False: An exception may be raised even if the program is syntactically correct.
5	State True or False: “raise and assert statements are used to raise exceptions”

ANSWERS

1	(B) WORD5#OVER
2	False
3	False
4	True
5	True

DATA FILE HANDLING
TEXT FILE



- A file is a collection of data stored on a storage device.
- Used to store data permanently for later use.
- Files can be read from and written to using programming languages like Python, C++, Java, etc.

Types of Files

1. Text Files

- Store data in readable characters (ASCII or Unicode).
- Extensions: .txt, .py, .html
- Can be opened and edited with any text editor.
- **CSV Files (Comma-Separated Values)**
    - Special type of text files that stores tabular data.
    - Each line represents a row; values separated by commas.
    - Extension: .csv
    - Easily used with spreadsheet tools and data processing libraries.

2. Binary Files

- Store data in binary (0s and 1s).
- Not human-readable.
- Extensions: .exe, .jpg, .dat
- Used for images, audio, video, and compiled programs.

NOTE: A text file can be understood as a sequence of characters consisting of alphabets, numbers and other special symbols. Files with extensions like .txt, .py, .csv, etc. are some examples of text files.

- ✓ While opening a text file, the text editor translates each ASCII value and shows us the equivalent character that is readable by the human being.
- ✓ Each line of a text file is terminated by a special character, called the End of Line (EOL). For example, the default EOL character in Python is the newline (\n).

Steps to Process Text Files:

To open a file in Python, we use the open ( ) function. The syntax of open ( ) is as follows:

file_object= open (file_name, access_mode)

ex1. Fobj= open('abc.txt,'w')

ex2. Fobj= open('D:\\Computer\\abc.txt', 'w') #double slash should be used in file path

ex3. Fobj= open(r'D:\Computer\abc.txt, 'w') #if you don't want to use // then use r in front of file

Note: if file mode is not mentioned in open function, then default file mode, 'r' is used

File Open Modes

File Mode	Description	File Offset position
<r>	Opens the file in read-only mode.	Beginning of the file
<rb>	Opens the file in binary and read-only mode.	Beginning of the file
<r+>	Opens the file in both read and write mode.	Beginning of the file
<w>	Opens the file in write mode. If the file already exists, all the contents will be overwritten. If the file doesn't exist, then a new file will be created.	Beginning of the file
<wb>	Opens the file in binary and write mode. If the file already exists, all the contents will be overwritten. If the file doesn't exist, then a new file will be created.	Beginning of the file
<wb+>	Opens the file in read, write and binary mode. If the file already exists, the contents will be overwritten. If the file doesn't exist, then a new file will be created.	Beginning of the file
<a>	Opens the file in append mode. If the file doesn't exist, then a new file will be created.	End of the file
<a+>	Opens the file in append and read mode. If the file doesn't exist, then it will create a new file.	End of the file

Use of With Clause

Syntax:

with open ("file name", access mode) as file object:

example: with open("book.txt", 'r') as f:

NOTE: No need to close the file explicitly if we are using with clause

Closing a file

- Once done with the read/write operations on a file, close the file.
- Python provides a close ( ) method to do so.
- While closing a file, the system frees the memory allocated to it.
- The syntax of close ( ) is:

file_object.close( )

Writing to a Text File

- For writing to a file, open it in write or append mode.
- If we open an existing file in write mode, the previous data will be erased, and the file object will be positioned at the beginning of the file.
- On the other hand, in append mode, new data will be added at the end of the previous data as the file object is at the end of the file.
- After opening the file, we can use the following methods to write data in the file.

  • **write** ( ) - for writing a single string

  • **writelines**( ) - for writing a sequence of strings

  Example 1:

```
with open ("sample.txt", "w") as f:
        f.write("Hello World\n")
        f.write("This is a second line.\n")
        #You need to manually add \n if you want line breaks.
```

  Example 2:

```
 lines = ["Line 1\n", "Line 2\n", "Line 3\n"]
 with open ("sample_lines.txt", "w") as f:
        f.writelines(lines)
#It does not add newlines automatically – include \n in each string   if needed.
```

Reading from a Text File

- Before reading a file, we must make sure that the file is opened in “r”, “r+”, “w+” or “a+” mode.
- There are three ways to read the contents of a file:

read ( )	**readline( )**	**readlines( )**
Reads the **entire file** as **one string**	Reads **one line at a time**	Reads **all lines** into a **list of strings**
Use when you want to load the whole file at once	Call multiple times to read multiple lines	useful for looping through lines
f.read( ) **Output** "Hello\nWorld\nPython"	f.readline( ) **Output** "Hello\n"	f.readlines( ) **Output** ["Hello\n","World\n", Python"]
to read a specified number of bytes use f.read(n)	used to read a specified number (n) of bytes of data from a file but maximum up to the newline character (\n) f.readline(n)	using readlines( ) function, lines in the file become members of a list, where each list element ends with a newline character (‘\n’)

SETTING OFFSETS IN A FILE

• **The tell ( ) method**	• **The seek ( ) method**
This function returns an integer that specifies the current position of the file object in the file.	This method is used to position the file object at a particular position in a file.
The position specified is the byte position from the beginning of the file till the current position of the file object.	In the given syntax, offset is the number of bytes by which the file object is to be moved. reference_point indicates the starting position of the file object. That is, with reference to which position, the offset has to be counted. It can have any of the following values: 0 - beginning of the file 1 - current position of the file 2 - end of file
The syntax of using tell ( ) is: file_object.tell( )	The syntax of seek ( ) is: file_object.seek(offset [, reference_point])

RELATIVE AND ABSOLUTE PATH

Absolute Path:

- It gives the complete path to a file or folder from the root directory.
- Always starts from the drive name (Windows) or root "/" (Linux/Mac).
- Used when the exact location of the file is known.

  Example: f = open ("C:/Users/John/Documents/data.txt", "r")

Relative Path:

- It gives the **location relative to the current working directory** (where your Python script is).
- Useful for portability across systems.

  Example:

  f = open ("files/data.txt", "r")

Special Notations:

. → Current directory

.. → Parent directory

```
# Access file in parent directory
f = open ("../data.txt", "r")
#To get current working directory
import os
print(os.getcwd( ))
```

Multiple Choice Questions

1.	What does the 'r' mode do when open a file in Python? a) Opens a file for writing only b) Opens a file for reading only c) Opens a file for appending d) Opens a file in binary mode
2.	Which of the following functions changes the position of file pointer? a. flush( ) b. tell( ) c. seek( ) d. offset( )
3.	Which of the following function returns a list datatype? a. d=f.read( ) b. d=f.read(10) c. d=f.readline( ) d. d=f.readlines( )

4.	What will file.write("Hello\nWorld") do in a text file? a. Write the string without a newline b. Raise an error c. Write "Hello World" on a single line d. Write "Hello" and "World" on separate lines
5.	The correct syntax of seek( ) is: a. file_object.seek(offset [, reference_point]) b. seek(offset [, reference_point]) c. seek(offset, file_object) d. seek.file_object(offset)
6.	What will be the output of the following statement in python? (fh is a file handle) fh.seek(-30,2) Options:- It will place the file pointer:- a. at 30th byte ahead of current file pointer position b. at 30 bytes behind from end-of file c. at 30th byte from the beginning of the file d. at 5 bytes behind from end-of file.
7.	Which Python function is used to open a text file for reading? a. open ("file.txt", "w") b. open ("file.txt", "r") c. read("file.txt") d. write("file.txt")
8.	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
9.	What is the correct way to ensure a file is automatically closed after reading? a) file.close( ) b) with open('file.txt') as file: c) open('file.txt').close( ) d) read(file.txt)
10.	State True or False. The writelines( ) method automatically adds newline characters (\n) after each line.

ANSWERS:

1	**b)** Opens a file for reading only	2	**c)** seek( )
3	**d)** f.readlines( )	4	**d)** Write "Hello" and "World" on separate lines
5	**a)** file_object.seek(offset [, reference_point])	6	**b)** at 30 bytes behind from end-of file
7	**b)** open("file.txt", "r")	8	**c)** both iii and iv
9	**b)** with open('file.txt') as file:	10	False

ASSERTION (A) and REASONING (R)

	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):** Opening a file in 'a' mode will delete its previous content. **Reason (R):** The 'a' mode appends new data at the end of the existing file content.
2.	**Assertion (A):** The with statement ensures a file is properly closed after its block finishes. **Reason (R):** Using with avoids the need to explicitly call close ( ) on a file object.
3.	**Assertion (A):** Files must be closed using the close ( ) function to ensure data is saved properly. **Reason (R):** Not closing a file may result in data loss or corruption

4.	**Assertion (A):** Opening a file in write mode('w') will delete its existing contents. **Reason (R):** In Python, the 'w' mode creates a new file if it doesn't exist, but preserves old content if the file exists.
5.	**Assertion (A):** The writelines( ) method adds newline characters automatically after each line. **Reason (R):** writelines( ) writes a list of strings.

Answers

1	(a) Both (A) and (R) are true and (R) is the correct explanation for (A).
2	(d) (A) is false but(R) is true.
3	(a)Both A and R are true, and R is the correct explanation of A.
4	(c)A is true, R is false.
5	(d) A is false, R is true

Short Answer Type Questions

1.	Display words longer than 5 characters from para.txt
2.	Write a function in Python that counts the number of "the" or "this" words present in a text file "myfile.txt". Example: If the "myfile.txt" contents are as follows: This is my first class on Computer Science. File handling is the easiest topic for me and Computer Networking is the most interesting one. The output of the function should be: Count of the/this in file
3.	Write a Python program to count all the line having 'a' as last character.

4.	Observe the following code and answer the questions that follow.
	File=open("MyData","a")
	____________ #Blank1
	File.close( )
	a) What type (text/binary) of file is MyData ?
	b) Fill the Blank1 with statement to write "ABC" in the file "Mydata"

5.	What does the split( ) function do when used on a line read from a text file? Explain with example.

Answers

1

```
#Display words longer than 5 characters from para.txt
f=open("para.txt", "r")
lines = f.readlines( )
for line in lines:
    for word in line.split( ):
        if len(word) > 5:
            print(word)
```

2

```
def displayTheThis( ):
     num=0
     f=open("myfile.txt","r") N=f.read( ) M=N.split( )
     for x in M:
         if x=="the" or x== "this":
             print(x)
             num=num+1
     f.close( )
     print ("Count of the/this in file:",num)
```

3

```
#Number of lines having 'a' as last character
     count =0
```

```
    f=open('fracture.txt',"r")
    data=f.readlines( )
    for line in data:
        if line[-2] == 'a':
        count=count+1
    print("Number of lines having 'a' as last character is/are : " ,count)
    f.close( )
```

4 a) Text File b) File.write(“ABC”)

5 The split ( ) function breaks a string into a list of words using whitespace as the default separator.

```
        line = "Hello world"
        print(line.split( ))
```

output

```
['Hello', 'world']
```

Long Answer Type Questions

1. Program to count number of “ME” or “MY” in a text file story.txt

Ans

```
#Program to count number of "ME" or "MY" in a text file story.txt
def countwords( ):
        f=open("D:\\xiic\\story.txt","r")
        count=0
        1st=f.read( )
        for i in 1st.split( ):
                if i.upper( )=="Me".upper ( ) or i.upper ( )=="My".upper( ):
                        count=count+1
        print ("Number of me or my is", count)
        f.close( )
countwords( )
```

2. Program to count number of “a” or “m” in a text file story.txt

Ans

```
# def countAM( ):
        f=open("D:\\xiic\\story.txt","r"}
        countA=0
        countM=0
        CON=f.read( )
        for i in CON:
                if i.upper( )=='A':
                        countA=countA+1
                elif i.upper( ) == 'M':
                        countM=countM+1
        print ("Number of A is ", countA, "\nNumber M is:", countM)
        f.close( )
countAM ( )
```

3 Program to count number of lines starting with “M”

Ans

```
#Program to count number of lines starting with “M”
def countlines( ):
        f=open("D:\\xiic\\story.txt","r")
```

```
    count1=0
    LN=f.readlines( )
    for i in LN:
        If i[0]=="M":
            count1+=1
    print("Number of lines starting with M=", count1)
    f.close( )
countlines ( )
```

4 Program to print lines not staring with a vowel from a text file STORY.TXT

Ans

```
#Program to print lines not staring with a vowel from a text file TESTFILE.TXT
def COUNTLINES( ):
    file lines
    count=0
    open ('D:\\XIIC\\STORY.TXT', 'r') file.readlines ( )
    for w in lines:
        if (w[0]).lower( ) not in 'aeoiu':
            count count + 1
    print ("The number of lines not starting with any vowel: ", count)
    file.close( )
COUNTLINES( )
```

5 Write a function ETCount( ) in Python, which should read each character of a text file “TESTFILE.TXT” and then count and display the count of occurrence of alphabets E and T individually (including small cases e and t too)

Ans

```
#Write a function ETCount( ) in Python, which should read each character of a tex
def ETCount( ):
    file open ('D:\\XIIC\STORY.TXT', 'r')
    lines=file.read( )
    countE=0
    countT=0
    for ch in lines:
        if ch in 'Ee':
            countEcountE + 1
        if ch in 'Tt':
            countT-countT+ 1
    print ("The number of E or e: ", countE) print ("The number of T or t : ", countT)
    file.close( )
ETCount( )
```

6 Program to count the word “AND” in a text file STORY.TXT

Ans

```
#Program to count the word "AND" in a text file STORY.TXT.
def COUNT_AND( ):
    count=0
    file=open('D:\\XIIC\\STORY.TXT', 'r')
    line=file.read( )
    word=line.split( )
    for w in word:
```

	if w in ['AND', 'and', 'And']:
	            count=count+1
	    print ("Number of word And is", count)
	    file.close( )
	COUNT_AND ( )

7	Write a function in Pyton to read lines from a text file story.txt, and display only those lines, which are starting with an alphabet 'P'.

Ans	#Write a function in Pyton to read lines from a text file story.txt, and #display only those lines, which are starting with an alphabet 'p'.
	def rdlines( ):
	    file= open('D:\\XIIC\\story.txt','r')
	    lines=file.readlines ( )
	    for line in lines:
	        if line[0] == 'p':
	            print (line)
	    file.close( )
	rdlines ( )

8	Program to remove lines starting with @ and write it into another file.

Ans	def filter( ):
	    fin=open("d:\\xiic\\source.txt","r")
	    fout=open("d:\\xiic\\newfile2.txt", "w")
	    text=fin.readlines ( )
	    for i in text:
	        if i[0]!="@":
	            fout.write(i)
	    fin.close( )
	    fout.close( )
	def showcontent ( ) :
	    fin=open("d:\\xiic\\newfile2.txt","r"}
	    text=fin.read( )
	    print (text)
	    fin.close( )
	filter( )
	showcontent( )

BINARY FILE

A binary file is a file whose content is in a binary format (0s and 1s). It stores data as a sequence of bytes (each byte = 8 bits). Binary files include a wide range of file types, including executables, libraries, graphics, databases, archives and many others.

There are mainly two types of data files — text file and binary file.

Differences between text files and binary files.

S. No.	Text file	Binary File
1.	The text files can easily be transferred from one computer system to another.	Binary files cannot easily be transferred from one computer system to another due to variations.

2.	It stores data using ASCII format i.e. human-readable graphic characters.	It stores data in binary format i.e. with the help of 0 and 1.
3.	These files are easily readable and modifiable because the content written in text files is human readable.	These files are not easily readable and modifiable because the content written in binary files is not human-readable and it is encrypted content.

Steps to process a binary file

- Opening a file
- Writing data into a file
- Reading data from a file
- Closing a file

Opening a Binary file in Python

Opening a file refers to getting the file ready either for reading or for writing.
To open a file in Python, we use the open ( ) function.

The syntax of open ( ) is as follows:

<file_object>= open (<file_name>, <access_mode>)

Closing a Binary file

Python has a close( ) method to close a file.

The syntax of close ( ) is as follows:
<file_object>.close( )

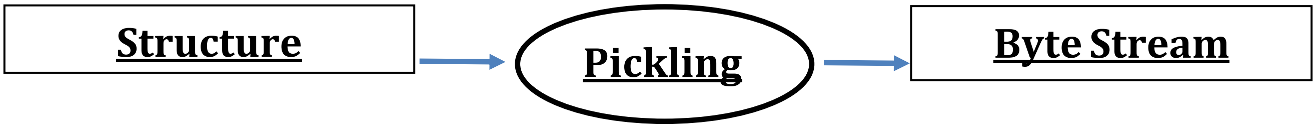
FILE MODES:

Mode	Description
rb	Open file in binary mode for reading only. The file pointer stands at the beginning of the file. Gives error if file does not exist
rb+(r+b)	Open file in binary mode for both reading and writing. The file pointer stands at the beginning of the file. Gives error if file does not exist
wb	Open file in binary mode for writing only. It creates the file if it does not exist. If the file exists, then it erases all the contents of the file. The file pointer stands at the beginning of the file.
wb+(w+b)	Open file in binary mode for both reading and writing. It creates the file if it does not exist. If the file exists, then it erases all the contents of the file. The file pointer stands at the beginning of the file.
ab	Open file in binary mode for appending data. Data is added to the end of the file. It creates the file if it does not exist. The file pointer stands at the end of the file.
ab+(a+b)	Open a file in binary mode for reading and appending data. Data is added to the end of the file. It creates the file if it does not exist. The file pointer stands at the end of the file.

Import pickle Module in Python

To write data to a binary file and read it subsequently, we need to use the Python module pickle.
The module pickle is used for serializing and de-serializing any Python object structure

Pickling/Serialization: The process of converting the structure (lists and dictionary etc.) into a byte stream just before writing to the file.
Unpickling/De-serialization: The reverse of pickling process where information from byte stream gets converted into object structure.



Methods of pickle module:

1. **pickle.dump( ):** Used to pickle the data object into byte stream and stores it in the binary file.
   Syntax
   pickle.dump(object , file_object) - used to write any object to the binary file.
2. **Pickle.load( ) :** Used to read data from a binary file. It converts the binary data back into the original python object (like a list, dictionary etc.).
   Syntax
   object=pickle.load(file_object) - used to read object from the binary file.

Writing data into a Binary File

Example:

```
import pickle
f=open("abc.dat","wb")
x="Hello 1"
pickle.dump(x,f)              # writing a string to file
x=[1,2,3,4,5]
pickle.dump(x,f)              #writing a list to file
x={"Name":"Ajay","Age":15,"Class":9}
pickle.dump(x, f)             # writing a Dictionary to file
f.close( )
```

Reading data from a Binary file

Follow these steps to read data:

1. Open the file in read mode using “rb” Ex.: f = open (“File.dat”, “rb”)
2. Use while loop with True statement to read the entire contents of the file individually.
3. Use try – except for Exception handling to handle runtime EOFError
4. Now load data into an object through the load( ) function
5. Print data as per need
6. Close the file

Example:

```
import pickle
f=open("abc.dat",'wb+')       # This will open the file in write and read mode
x=[1,2,3,4,5]
pickle.dump(x,f)              # This will write the object List into the file.
f.seek(0)                     #This will move the file pointer to the begining of the file
x=pickle.load(f)              #This will read the object from the file and store in variable.
print(x)
```

Search records from binary file

Follow these steps to search the record in the binary file:

1. Open the file in reading mode using “rb”
2. Prompt a message to ask unique field from data to search

3. Declare a boolean variable flag to store **False** for the record not found and **True** when the record found
4. Use a while loop to access records individually
5. Now load the data into the dictionary object using load( ) function
6. Use if condition to compare the data with the variable taken in step 2
7. Print the record found
8. Assign True to the flag variable declared in step 3
9. Use the except block to handle EOFError and terminate the loop using the break
10. Print record not found message when Flag is False
11. Finally, close the file using f.close( )

Example:

```
import pickle
def bf_search( ):
    try:
        f=open("Sports.dat","rb")
        pc=int(input("Player code to search:")) flag=False
        while True:
            try:
                rec= pickle.load(f)
                if rec['Pcode']==pc:
                    print("Player Name:",rec['Pname'])
                    print("Individual Score:",rec['Score'])
                    print("Rank:",rec['Rank'])
                    flag = True
            except Exception:
                break
    except FileNotFoundError:
        print("The file Sports.dat does not exist")
    if flag==False:
        print("Record not found...")
    f.close( )
bf_search( )
```

Append data in Binary File

To append data in binary follow these steps:

1. Open the file in append mode using “ab” Ex.: f = open (“file.dat”,”ab”)
2. Enter data to append
3. Append entered data into the dictionary/list object
4. Use pickle.dump( ) method to write the dictionary/list data
5. Close the file

Example:

```
def bf_append( ):
   import pickle
      f=open("sports.dat","ab")
      print("Append Data")
      pcode=int(input("Enter the Player code:"))
      pname = input("Enter Player Name: ")
      score =int(input("Enter individual score:"))
```

```
            rank =int(input("Enter Player Rank:"))
            rec={'Pcode':pcode,'Pname':pname,'Score':score,'Rank':rank}
            pickle.dump(rec,f)
            f.close( )
        bf_append( )
```

Update record in Binary file

To update record, you can use the search record code if you wish. To update the record follow these steps:

1. Open the file using read mode
2. Declare a variable for unique value to be updated
3. Use try-except and while loop as explained above
4. Add record fetched from binary file into a list
5. Enter the new record information to update
6. Compare the fetched records with entered record and assign the new values to update
7. Write the data using dump( ) function
8. Close the file

Example:

```
def bf_update( ):
    import pickle
    f=open('student.dat','rb')
    reclst=[]
    wihile True:
        try :
            rec = pickle.load(f)
            reclst.append (rec)
        except EOFError:
            break
    f.close( )
    pc=int(input("Enter player code to update:")
    pn=input("Enter new name:")
    ps=int(input("Enter Player Score:"))
    pr=int(input("Enter Player Rank:"))
    for i in range (len(reclst)):
        if reclst[i] Pcode =pc:
            reclst[i]['Pname'] = pn
            reclst[i]['Score']= ps
            reclst[i]['Rank']=pr
    f=open('sports.dat','wb')
    for i in reclst:
        pickle.dump(i,f)
    f.close( )
 bf_update( )
```

Multiple choice Questions

1.	Which file mode can be used to open a binary file in both append and read mode? a) w+ b) wb+ c) ab+ d) a+

2.	Nila wants to store a list of dictionaries into a binary file. Which of the following Python modules should she use? a) os b) csv c) pickle d) json
3.	Pick the correct syntax to read a binary file using pickle: a) pickle.read(file) b) pickle.load(file) c) pickle.open(file) d) pickle.input(file)
4.	What does the pickle.dump(obj, file) function do? a) Reads binary data from a file b) Writes text data to a file c) Converts a Python object into byte stream and writes it to a file d) Appends an object to a list
5.	Which file mode should be used to write a binary file in Python? a) 'w' b) 'r' c) 'wb' d) 'rb'
6.	What will happen if you try to read a binary file using 'r' mode instead of 'rb'? a) It will read data correctly. b) It will raise a SyntaxError. c) It will convert binary data into text d) It may raise an error or return incorrect data
7.	Which of the following is a key advantage of binary files over text files? a) Easy to read with any text editor b) Allows direct storage of complex Python objects b) Consumes more storage space d) Cannot be shared across system

Answers :

1	C	2	C	3	B	4	C
5	C	6	D	7	B		

Assertion and Reasoning

	Mark the correct choice as: 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) A is false, but R is true.
1.	**Assertion (A):** A binary file in python is used to store collection objects like lists and dictionaries that can be later retrieved in their original form using pickle module. **Reasoning (A):** A binary files are just like normal text files and can be read using a text editor like notepad
2.	**Assertion (A):** The pickle module in Python is used to store and retrieve Python objects in binary files. **Reason (R):** The pickle.dump( ) method reads data from a binary file and the pickle.load( ) method writes data to a binary file.
3.	**Assertion (A):** Binary files are used to store data in the same format as it is stored in memory. **Reason (R):** Binary files convert Python objects into byte streams which are machine-readable.

4.	**Assertion (A):** You can open a binary file using only 'r' or 'w' mode in Python. **Reason (R):** Binary files should be opened with 'rb' or 'wb' modes to ensure correct handling of byte streams.
5.	**Assertion (A):** Binary files are preferred when large volumes of data need to be stored with structure. **Reason (R):** Binary files are human-readable and can be opened using any text editor.

Answers:

1	C	2	C	3	A	4	D
5	C						

Short Answer Type Questions:

1	What is the purpose of the pickle module in Python?
2	State two advantages of using binary files over text files in Python.
3	How many times should pickle.load( ) be called while reading a binary file? Explain your answer with reference to the number of times pickle.dump( ) was used while writing the file.

Answers :

1	The pickle module is used for serializing and de-serializing Python objects into binary format. It allows saving complex data types like dictionaries, lists, etc., into a binary file and retrieving them later.
2	Binary files can store complex Python objects (like dictionaries, lists) directly. They are more secure and efficient in terms of storage space and speed.
3	The number of times pickle.load( ) should be called is **equal to the number of times** pickle.dump( ) **was called** when the file was written. If the file was written using pickle.dump( ) 3 times, then you must call pickle.load( ) 3 times to read all the data. Since we may not know the exact count, load( ) is usually called inside a loop until EOFError occurs.

Long Answer Type Questions:

1.	Karthik is a manager working in a recruitment agency. He needs to manage the records of various candidates. For this, he wants the following information of each candidate to be stored: Candidate_ID – integer Candidate_Name – string Designation – string Experience – float You, as a programmer of the company, have been assigned to do this job for Karthik. i. Write a function to input the data of a candidates and append it in a binary file ii. Write a function to update the data of candidates whose experience is less than 10 years and change their designation to "Assistant Manager". iii. Write a function to read the data from the binary file and display the data of all those candidates who are not "Assistant Manager".

Ans 1(i)	import pickle
	def input_candidates( ):

```
    f=open("candidate.dat","ab")
    n = int(input("Enter the number of candidates you want to add: "))
    for i in range(n):
        candidate_id = Candidate ID: "))
        candidate_name = input("Enter Candidate Name: ")
        designation = input("Enter Designation: ")
        experience = float(input("Enter Experience (in years):")):
        pickle.dump([candidate_id, candidate_name, designation, experience],f)
        print("Candidate data appended successfully. ")
```

(ii)

```
import pickle
def update_assistant_manager( ):
    updated_candidates = []
    try:
        with open('candidates.dat', 'rb') as file:
            while True:
                try:
                    Candidate= pickle.load(file)
                    if candidate[3] < 10: # If experience <10 years
                        candidate[2] = 'Assistant Manager'
                    updated_candidates.append(candidate)
                except EOFError:
                    break # End of file reached
    except FileNotFoundError:
        print("No candidate data found. Please add candidates first")
    with open('candidates.dat', 'wb') as file:
        for candidate in updated_candidates:
            pickle.dump(candidate, file)
```

1(iii)

```
import pickle
def display_non_assistant_managers( ):
    try:
        with open('candidates.dat', 'rb') as file:
            while True:
                try:
                    candidate pickle.load(file)
                        if candidate[2] != 'Assistant Manager':# Check if not Assistant Manager
                            print("Candidate ID: ",candidate[0])
                            print("Candidate Name:",candidate[1])
                            print("Designation:",candidate[2])
                            print("Experience:",candidate[3])
                            print("_____________")
                except EOFError:
                            break # End of file reached
    except FileNotFoundError:
        print("candidate data found. Please add candidates first.")
```

2

A school is maintaining student records in a binary file records.dat. Each record stores the following data:
Roll Number (int)
Name (str)

Percentage Marks (float)

Write Python functions to do the following

i. Create a binary file by entering student data in the form of dictionaries.

ii. Count and display the number of students who scored less than 40% (i.e., need academic improvement).

Ans 2(i)

```
import pickle
def create_file( ):
   with open('records.dat', 'wb') as file:
      n = int(input("Enter number of students:"))
      for I in range(n):
         student = {}
         student['roll'] = int(input("Enter Roll No:"))
         student['name'] =input("Enter Name:")
         student['percentage'] = float(input("Enter Percentage:"))
         pickle.dump(student, file)
```

2(ii)

```
import pickle
def count_low_score( ):
   count = 0
   try:
      with open('records.dat', 'rb') as file:
         while True:
            try:
               student = pickle.load(file)
               if student['percentage'] < 40:
                  Count+= 1
            except EOFError:
               break
         print("Number of students with less than 40% marks:", count)
   except FileNotFoundError:
      print("File not found!")
```

3

A binary file books.dat contains records of books. Each record is stored as a dictionary with the following fields:

bid- (Book ID – integer)
title- (Book Title – string)
price- (Book Price – float)

The librarian wants to perform the following operations efficiently:

i. Define a function add_books(n) that adds n book records to the binary file.

ii. Define a function update_price(Bid, new_price) that updates the price of the book whose Book ID is Bid.

iii. Define a function count_books_in_range(low, high) that returns the number of books whose price lies between low and high (inclusive).

3(i)

```
import pickle
def add_books(n):
   with open("books.dat","ab") as f:
      for i in range(n):
         book ={}
         book['bid'] int(input("Enter Book ID:")
         book['title'] = (input("Book Title:")
         book['price'] = float(input("Enter Book Price: "))
         pickle.dump(book, f)
```

(ii)

```
import pickle
def update_price(Bid, new_price):
    updated = False
    temp_list =[]
    try:
        with open("books.dat", "rb") as f:
            while True:
                try:
                    book = pickle.load(f)
                    if book['bid'] == Bid:
                        book['price'] = new_price
                        updated = True
                        temp_list.append(book)
                except EOFError:
                    break
        with open("books.dat", "wb") as f:
            for book in temp_list:
                pickle.dump(book, f)
            if updated:
                print("Price updated successfully.")
            else:
                print("Book ID not found.")
    except FileNotFoundError:
        print("File not found.")
```

(iii)

```
import pickle
def count_books_in_range(low,high):
    count = 0
    try:
        with open("books.dat","rb") as f:
            while True:
                try:
                    book = pickle.load(f)
                    if low <=book['price'] <=high:
                        count +=1
                except EOFError:
                    break
                print("Total books priced between",low,"and",high,":",count)
    except FileNotFoundError:
        print("File not found.")
```

4. A binary file patient.dat stores records of patients in a hospital. Each record is stored as a dictionary with the following fields:

pid- (Patient ID – integer)
pname- (Patient Name – string)
age- (integer)
bill- (float – hospital bill amount)

Perform the following tasks using three user-defined functions:

i. Define a function add_patients(n) to add n new patient records to patient.dat.
ii. Define a function update_bill(Pid, amount) that updates the bill of a patient by adding the given amount to the existing bill using their Patient ID.
iii. Define a function list_senior_patients( ) to display details of all patients aged 60 and above.

Ans 4(i)

```
import pickle
def add_patients(n):
    with open("patient.dat", "ab") as f:
        for i in range(n):
        patient = {}
        patient['pid'] = int(input("Enter Patient ID: "))
        patient['pname'] = input("Enter Patient Name: ")
        patient['age'] = int(input("Enter Age: "))
        patient['bill'] = float(input("Enter Bill Amount: "))
        pickle.dump(patient, f)
```

4(ii)

```
import pickle
def update_bill(Pid, amount):
    updated = False
    patients = []
    try:
        with open("patient.dat", "rb") as f:
            while True:
                try:
                    patient = pickle.load(f)
                    if patient['pid'] == Pid:
                    patient['bill'] += amount
                    updated = True
                    patients.append(patient)
                except EOFError:
                    break
            with open("patient.dat", "wb") as f:
                for p in patients:
                    pickle.dump(p, f)
            if updated:
                print("Bill updated successfully.")
            else:
                print("Patient ID not found.")
    except FileNotFoundError:
        print("File not found.")
```

4(iii)

```
import pickle
def list_senior_patients( ):
    try:
        with open("patient.dat", "rb") as f:
        print("Senior Patients (Age 60+):")
        found = False
        while True:
            try:
                patient = pickle.load(f)
                if patient['age'] >= 60:
                    print(patient)
                    found = True
            except EOFError:
                break
        if not found:
            print("No senior patients found.")
    except FileNotFoundError:
        print("File not found.")
```

5. A binary file, **EMP.DAT** has the following structure : **[Emp_Id, Name, Salary]** where

Emp_Id : Employee id
Name: Employee Name
Salary: Employee Salary

Write a user defined function, **disp_Detail( )**, that would read the contents of the file **EMP.DAT** and display the details of those employees whose salary is below **25000.**

Ans

```
import pickle
def Copy_new( ):
    F2=open("new_items.dat", "wb")
    try:
        F1=open("items.dat","rb")
        Datal=pickle.load(F1)
        Data2={}
        for K,V in Datal.items( ):
            if v[1]>1000:
                Data2 [K]=V
        pickle.dump (Data2, F2)
        F2.close( )
    except:
        print("File not found!")
   F1.close( )
```

6. Mr. Ashok is working on a Toy Shop project to manage toys records using Python. The toys data is stored in a binary file named **TOYDATA.DAT**. The binary file **TOYDATA.DAT** contains each record in given format:

{“Toy_ID”: T_ID, “TName”:toy_name, “Price”:price}
Where

- **Toy_ID**: Toy ID (string)
- **TName**: Toy name (string)
- **Price**: Price of toy (integer)

You as a programmer, help him to write following python functions:

i. **ADD_Data( )** : To write **n** records in binary file TOYDATA.DAT by taking the values for each record from user.
ii. **SHOW_Data( )** : Read all records from binary file and display them.
iii. **Remove_Toy( )** : that deletes the record of a toy in the file **TOYDATA.DAT** based on the **Toy ID** provided by the user. If the Toy ID does not exist in the file, display an appropriate message.

Ans 6(i)

```
import pickle
def ADD_Data( ):
    f=open("TOYDATA.DAT", "wb")
    L=[ ]
    n=int(input("How many records you want to enter: "))
    for i in range(n):
        T_ID=input("Enter Toy ID: ")
        toy_name=input("Enter Toy Name: ")
        price=int(input("Enter Price: "))
        D={"Toy_ID":T ID, "TName": toy name, "Price":price}
        L.append(D)
    pickle.dump(Lf)
    f.close( )
```

6(ii)

```
def  SHOW_Data( ):
    f=open("TOYDATA.DAT", "rb")
    L=pickle.load(f)
    print(L)
    f.close( )
```

6(iii)

```
def  Remove_Toy( ):
    Toy_ID=input("Enter Toy ID: ")
    f=open("TOYDATA.DAT", "rb+")
    found=0
    L=pickle.load(f)
    M=[ ]
    for D in L:
    if Toy ID not in D["Toy_ID"]:
        M.append(D)
    else:
        found=1
    if found==1:
        f.seek(0)
        pickle.dump(M,f)
        print("Record Deleted successfully")
    else:
        print("Record not found")
    f.close( )
```

7. A file, PASSENGERS.DAT, stores the records of passengers using the following structure : [PNR, PName, BRDSTN, DESTN, FARE]

where :

PNR – Passenger Number (string type)
PName – Passenger Name (string type)
BRDSTN – Boarding Station Name (string type)
DESTN – Destination Station Name (string type)
FARE – Fare amount for the journey (float type)

Write user defined functions in Python for the following tasks :

(i) Create( ) – to input data for passengers and write it in the binary file PASSENGERS.DAT.

(ii) SearchDestn(D) –to read contents from the file PASSENGERS.DAT and display the details of those Passengers whose DESTN matches with the value of D.

(iii) UpdateFare( ) – to increase the fare of all passengers by 5% and rewrite the updated records into the file PASSENGERS.DAT.

Ans

```
import pickle
```

7(i)

```
def Create( ):
        F=open ("PASSENGERS.DAT", "wb")
        PNR=input("PNR No: ")
        PName=input("Name: ")
        BRDSTN=input ("Boarding at: ")
        DESTN=input("Destination: ")
        FARE=float(input("Fare: "))
        Rec= [PNR, PName, BRDSTN, DESTN, FARE]
        pickle.dump (Rec, F)
        F.close( )
```

7(ii)

```
def SearchDestn (D):
        try:
                F=open ("PASSENGERS.DAT", "rb")
                Rec=pickle.load(F)
                for R in Rec:
                        if R[3]==D;
                                print (R)
                F.close( )
        except:
                print("File not found!")
```

7(iii)

```
def UpdateFare( ):
        try:
                FR=open ("PASSENGERS.DAT", "rb+")
                Rec=pickle.load(FR)
                for I in range (len (Rec)):
                        Rec [1] [4] += (Rec [I] [4] * 0.05)
                        print("Updation Done!")
                FR.seek(0)
                pickle.dump (Rec, FR)
                FR.close( )
        except:
                print("File not found!")
```

8. Consider a binary file, Cinema.dat containing information in the following structure :

[Mno, Mname, Mtype]

Write a function, search_copy( ), that reads the content from the file Cinema.dat and copies all the details of the "Comedy" movie type to file named movie.dat.

Ans

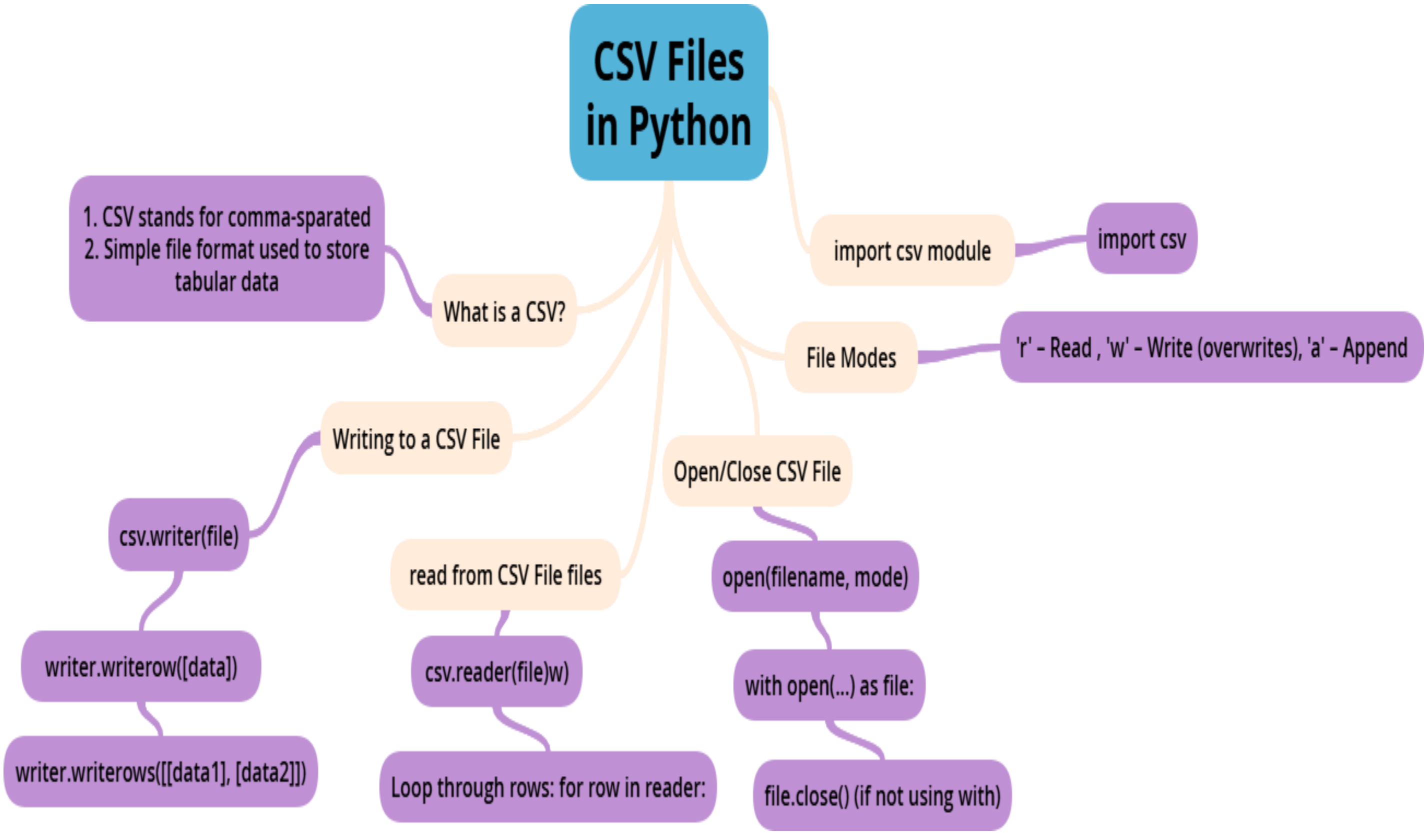
```
import pickle
def search_copy( ):
        try:
                F1=open("Cinema.dat","rb")
                F2=open("movie.dat", "wb")
                Datal=pickle.load(F1)
                Data2=[ ]
                for D in Datal:
                        if D[2]=="Comedy":
                             Data2.append(D)
                pickle.dump (Data2, F2)
                F1.close( )
```

	F2.close( )
	    except:
	        print("File not found!")

9.	Consider a binary file, items.dat, containing records stored in the given format : {item_id: [item_name,amount]} Write a function, Copy_new( ), that copies all records whose amount is greater than 1000 from items.dat to new_items.dat.

Ans	import pickle
	def Copy_new( ):
	    F2=open("new_items.dat", "wb")
	    try:
	        F1=open("items.dat","rb")
	        Datal=pickle.load(F1)
	        Data2={ }
	        for K,V in Datal.items( ):
	            if v[1]>1000:
	                Data2 [K]=V
	        pickle.dump (Data2, F2)
	        F2.close( )
	    except:
	        print("File not found!")
	    F1.close( )

COMMA SEPARATED VALUES(CSV) FILES



CSV Files in Python
What is a CSV?
1. CSV stands for comma-sparated
2. Simple file format used to store tabular data
import csv module
import csv
File Modes
'r' - Read , 'w' - Write (overwrites), 'a' - Append
Writing to a CSV File
csv.writer(file)
writer.writerow([data])
writer.writerows([[data1], [data2]])
read from CSV File files
csv.reader(file)w)
Loop through rows: for row in reader:
Open/Close CSV File
open(filename, mode)
with open(...) as file:
file.close() (if not using with)

CSV stands for Comma Separated Values CSV file is a type of plain text file means data stored in form of ASCII or Unicode characters Each line is a row and in row each piece of data is separated by a comma It is common format for data interchange.

<u>Steps to Process CSV Files</u>:

1. First of all we have to **import csv module** for file operation.

2. For writing to CSV Files ,

i. We open file in 'w' writing or 'a' append mode using open( ) method

```
f=open('filename.csv','w',newline=' ')
```

By setting newline='', Python don't modify the newline characters at all. This is commonly used when working with the csv module to ensure that the csv.writer handles the newline characters correctly (i.e., without adding extra blank lines between rows).

i. Creating a writer object, associated with the file object. The writer object allows you to write rows of data into the file in CSV format. Through csv.writer( ) method ,we create it.

```
w_obj=csv.writer(f)
```

ii. Perform writing operation (read data from user and write on to csv file)

writerow( ) used for writing single row

writerows( )used for writing multiple rows

3. For Reading from CSV File

i. Creating a reader object, associated with the file object we use csv.reader( ) method. It Creates a CSV reader object that can iterate over the rows of the file object.

```
r_obj=csv.reader(f)
```

ii. Perform reading operation (read data from csv file and display using python)

4.Close the file using close( ) method

```
f.close( )
```

csv.reader( )	The csv.reader( ) function in Python is used to read data from a CSV (Comma Separated Values) file. It is part of Python's built-in csv module. csv.reader(file_object, delimiter=',') **Parameters:** • file_object: A file object opened using open( ). • delimiter: (Optional) Specifies the character used to separate fields. Default is comma (,). It returns an **iterator** that returns each row in the CSV file as a **list of strings**. **Example** import csv with open('data.csv', 'r') as file: reader = csv.reader(file) for row in reader: print(row)
csv.writer( )	The csv.writer( ) function in Python is used to **write data to a CSV file**. It is part of Python's built-in csv module and writes rows as comma-separated values. csv.writer(file_object, delimiter=',')

Parameters:

- file_object: A file object opened in **write ('w') or append ('a')** mode.
- delimiter: (Optional) Character that separates values. Default is comma (,).
- It returns a **writer object** that lets you write rows (lists or tuples) to the CSV file.

Common Methods:

- writer.writerow(row) → Writes a **single row**.
- writer.writerows(list_of_rows) → Writes **multiple rows**.

Example:

```
import csv
with open('output.csv', 'w', newline='') as file:
   writer = csv.writer(file)
   # Write header
   writer.writerow(['Name', 'Age', 'City'])
    # Write multiple data rows
   writer.writerows([['Alice', 23, 'Delhi'],['Bob', 30, 'Mumbai'] ])
```

Multiple Choice Questions

1. Syntax for opening 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. Empty string gets appended to the first line.
c. Empty string gets appended to all lines. d. EOL translation is suppressed

2. Which of the following is not a function / method of csv module in Python?
a. read( ) b. reader( ) c. writer( ) d. writerow( )

3. A CSV file
a. is a text file b. can store images
c. is a type of python program d. is a Computer Software Validation file

4. What does the csv.reader( ) function do?
a. Reads JSON files b. Reads CSV files row by row as lists
c. Converts CSV to a dictionary d. Writes to CSV files

5. What mode should be used to open a file for reading in CSV format?
a. 'w' b. 'x' c. 'r' d. 'rw'

6. What is the default delimiter in a CSV file?
a. Semicolon (;) b. Tab (\t) c. Pipe (|) d. Comma (,)

7. State True or False:
CSV file is also known as delimited text file which can be opened in Notepad and Spreadsheet both.

8. Which of the following is not a correct way to open a CSV file for writing?
a. open('file.csv', 'r') b. open('file.csv', 'a')
c. open('file.csv', 'w') d. open('file.csv', 'rw')

9. Which method writes multiple rows at once in a CSV file?
a. writerows( ) b. writeall( )
c. write( ) d. writemultiple( )

10. When using the csv.reader( ) object to read from a CSV file, what type of data structure is each row of data represented as?
a) String b) Dictionary c) List d) Tuple

ANSWERS:

1	D	2	A	3	A	4	B	5	C
6	D	7	True	8	A	9	A	10	C

ASSERTION REASONING 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 file is a human readable text file where each line has a number of fields, separated by comma or some other delimiter. **Reason (R): writerow( )** method is used to write a single row in a CSV file.
2	**Assertion (A):** The csv.reader( ) object returns each row as a list of strings. **Reason (R):** The csv.reader( ) reads the entire file content at once into memory
3	**Assertion (A):** csv.writer( ) can be used to write data row by row into a CSV file. **Reason (R):** writerow( ) method of the csv.writer object writes a single row to the CSV file.
4	**Assertion (A):** The newline='' parameter is used while opening a CSV file in Python. **Reason (R):** It helps prevent insertion of blank lines between rows on Windows systems.
5	**Assertion (A):** csv.writer(file) returns an object that can be used to read data from a CSV file. **Reason (R):** csv.writer( ) is used for writing data into CSV files.

ANSWERS:

1	**Ans**: **(b)** Both (A) and (R) are true and (R) is not the correct explanation for (A).
2	**Answer:** c) A is true but R is false **Explanation:** csv.reader( ) reads the file line by line (not all at once), and returns rows as lists.
3	**Answer:** a) Both A and R are true, and R is the correct explanation of A
4	**Answer:** a) Both A and R are true, and R is the correct explanation of A
5	**Answer:** d) A is false but R is true **Explanation:** csv.writer( ) is for writing, not reading.

Long Answer Questions

1. Write a Program in Python that defines and calls the following user defined functions:

   (i) ADD( ) – To accept and add data of an employee to a CSV file ‘record.csv’. Each record consists of a list with field elements as empid, name and salary to store employee id, employee name and employee salary respectively.

   (ii) COUNTR( ) – To count the number of records present in the CSV file named ‘record.csv’.

Ans 1(i)

```
import csv
def ADD( ):
   f=open('record.csv','a')
   wr=csv.writer(f)
   eid=int(input("Enter Employee ID"))
   name=input("Enter name of employee")
   sal=float(input("enter salary"))
   data=[eid,name,sal]
   wr.writerow(data)
ADD( )
```

1(ii)

```
import csv
def COUNTR( ):
   f=open('record.csv','r')
   c=0
   data=csv.reader(f)
   for x in data:
      c=c+1
   print(c)
COUNTR( )
```

2. Create a function maxsalary( ) in Python to read all the records from an already existing file record.csv which stores the records of various employees working in a department. Data is stored under various fields as shown below:

E_code	E_name	Scale	Salary
A01	Bijesh Mehra	S4	65400
B02	Vikram Goel	S3	60000
C09	Suraj Mehta	S2	45300

Function should display the row where the salary is maximum.

Note: Assume that all employees have distinct salary.

Ans

```
import csv
def maxsalary( ):
   f=open('record.csv', 'r')
   reader=csv.reader (f)
   skip_header = True
   m= 0
   for row in reader:
      if skip_header:
         skip_header = False
      else:
         if(int(row[3])>m):
             m=int(row[3])
             rec=row
   print('Row with the highest salary: ', rec)
   f.close( )
maxsalary( )
```

3. A csv file "Happiness.csv" contains the data of a survey. Each record of the file contains the following data:

- Name of a country
- Population of the country
- Sample Size (Number of persons who participated in the survey in that country)
- Happy (Number of persons who accepted that they were Happy)

For example, a sample record of the file may be:

['Signiland', 5673000, 5000, 3426]

Write the following Python functions to perform the specified operations on this file:

(I) Read all the data from the file in the form of a list and display all those records for which the population is more than 5000000.

(II) Count the number of records in the file.

Ans 3(i)

```
def show( ):
    import csv
    f=open("happiness.csv",'r')
    records=csv.reader(f)
    next(records, None) #To skip the Header row
    for i in records:
        if int(i[1])>5000000:
            print(i)
    f.close( )
```

3(ii)

```
def Count_records( ):
    import csv
    f=open("happiness.csv",'r')
    records=csv.reader(f)
    next(records, None) #To skip the Header row
    count=0
    for i in records:
        count+=1
    print(count)
    f.close( )
```

4 Vedansh is a Python programmer working in a school. For the Annual Sports Event, he has created a csv file named Result.csv, to store the results of students in different sports events. The structure of Result.csv is :

[St_Id, St_Name, Game_Name, Result]
Where
St_Id is Student ID (integer)
ST_name is Student Name (string)
Game_Name is name of game in which student is participating(string)
Result is result of the game whose value can be either 'Won', 'Lost' or 'Tie'

For efficiently maintaining data of the event, Vedansh wants to write the following user defined functions:

Accept( ) – to accept a record from the user and add it to the file Result.csv. The column headings should also be added on top of the csv file.

wonCount( ) – to count the number of students who have won any event.

As a Python expert, help him complete the task.

Ans

```
def Accept( ):
    sid=int(input("Enter Student ID "))
    sname=input ("Enter Student Name ")
    game= input ("Enter name of game ")
    res=input ("Enter Result")
    headings=["Student ID", "Student Name", " Game Name", "Result"]
    data=[sid, sname, game, res]
    f=open('Result.csv', 'a', newline='')
    csvwriter=csv.writer (f)
    csvwriter.writerow(headings)
    csvwriter.writerow(data)
    f.close( )

def wonCount( ):
     f=open('Result.csv','r')
    csvreader=csv.reader (f, delimiter=',')
    head=list (csvreader)
    print (head [0])
    for x in head:
        if x[3]=="WON":
                  print (x)
    f.close( )
```

5 Mr. Mahesh is a Python Programmer working in a school. He has to maintain the records of the sports students. He has created a csv file named **sports.csv,** to store the details. The structure of **sports.csv** is :

[sport_id, competition, prize_won]

where

sport_id, is Sport id (integer)

competition is competition name (string)

prize_won is ("Gold", "Silver", "Bronze")

Mr. Mahesh wants to write the following user-defined functions :

Add_detail( ): to accept the detail of a student and add to a csv file, **"sports.csv"**.

Count_Medal( ): to display the name of competitions in which students have won **"Gold"** medal.

Help him in writing the code of both the functions.

Ans

```
import csv
def Add_detail( ):
    F=open("sports.csv","a")
    W=csv.writer(F)
    sport_id=int(input("Sport id:"))
    competition=input("Competition:")
    prize_won=input("Prize won:")
    L=[sport_id,competition,prize_won]
    W.writerow(L)
    F.close( )

def Count_Medal( ):
    F=open("sports.csv","r")
    L=list(csv.reader(F))
    for D in L:
        if D[2]=="Gold":
            print("Competition:",D[1])
    F.close( )
```

6. Sangeeta is a Python programmer working in a computer hardware company. She has to maintain the records of the peripheral devices. She created a csv file named **Peripheral.csv**, to store the details.

The structure of **Peripheral.csv** is:
[P_id,P_name,Price]
where
P_id is Peripheral device ID (integer)
P_name is Peripheral device name (String)
Price is Peripheral device price (integer)
Sangeeta wants to write the following user defined functions :
Add_Device( ) : to accept a record from the user and add it to a csv file, **Peripheral.csv**
Count_Device( ) : To count and display number of peripheral devices whose price is less than 1000.

Ans

```
import csv
def Add_Device( ):
    F=open("Peripheral.csv","a",newline='')
    W=csv.writer(F)
    P_id=int(input("Enter the Peripheral ID"))
    P_name=input("Enter Peripheral Name")
    Price=int(input("Enter Price"))
    L=[P_id,P_name,Price]
    W.writerow(L)
    F.close( )
def Count_Device( ):
    F=open("Peripheral.csv","r")
    L=list(csv.reader(F))
    Count=0
    for D in L:
        if int(D[2])<1000:
            Count+=1
    print(Count)
    F.close( )
```

7 Write a program in Python that defines and calls the following user defined functions:

(i) Add_Teacher( ) : It accepts the values from the user and inserts record of a teacher to a csv file 'Teacher.csv'. Each record consists of a list with field elements as T_id,Tname and desig to store teacher ID, teacher name and designation respectively.

(ii) Search_Teacher( ) : To display the records of all the PGT (designation) teachers.

Ans

```
import csv
def Add_Teacher( ):
    fout=open("Teacher.csv","a",newline="\n")
    T_id=int(input("Enter Teacher id: "))
    Tname=input("Enter Teacher name: ")
    desig=input("Enter Designation: ")
    rec=[T_id,Tname,desig]
    csvw=csv.writer(fout)
    csvw.writerow(rec)
    fout.close( )
```

```
def Search_Teacher( ):
    fin=open("Teacher.csv")
    csvr=csv.reader(fin)
    for record in csvr:
        if record[2]=="PGT":
            print(record)
    fin.close( )
Add_Teacher( )
Search_Teacher( )
```

8. Write a program in Python that defines and calls the following user defined functions:

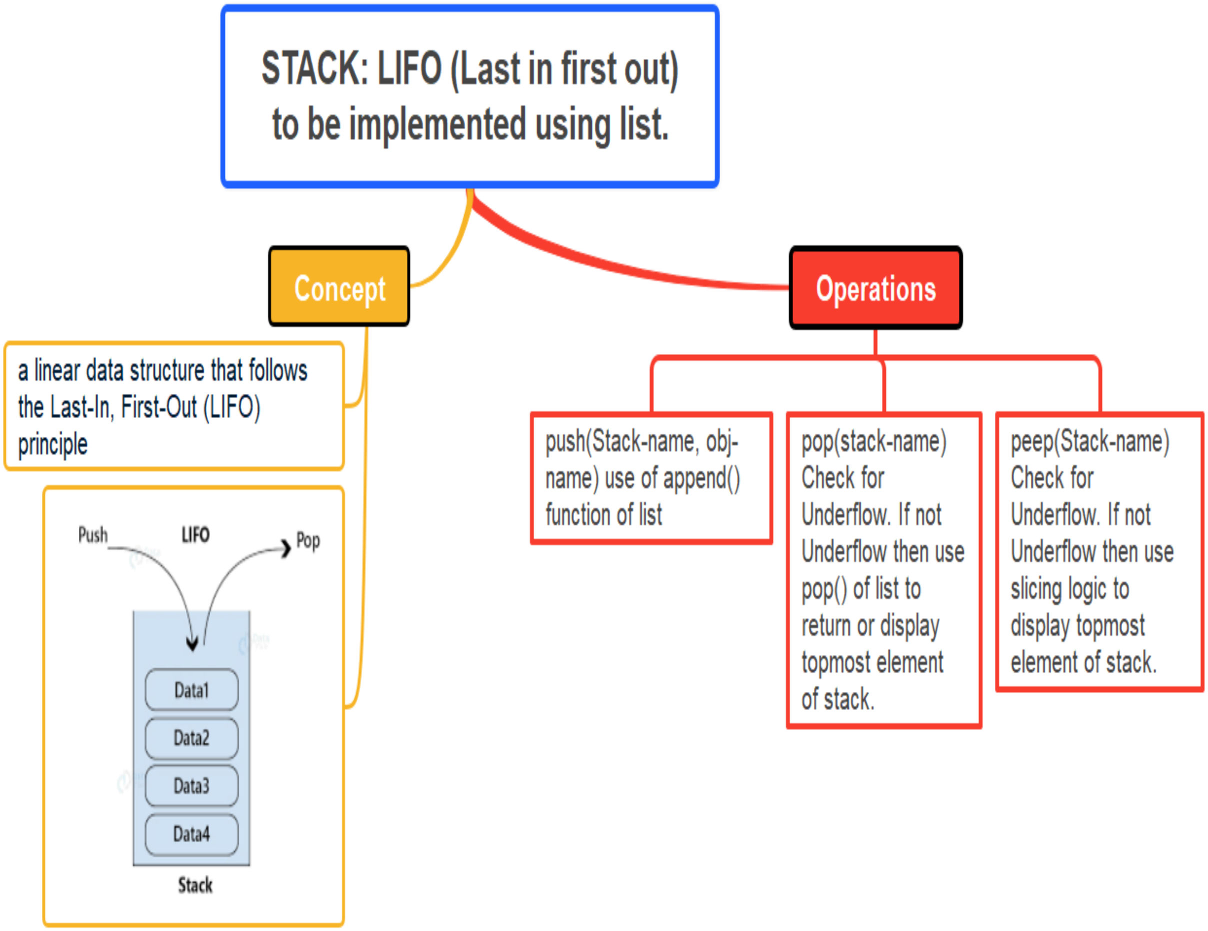
(i) Add_Device( ) : The function accepts and adds records of the peripheral devices to a csv file ‘peripheral.csv’. Each record consists of a list with field elements as **P_id, P_name** and **Price** to store peripheral device ID, device name, and price respectively.

(ii) Count_Device( ) : To count and display number of peripheral devices, whose price is less than ₹ 1000.

Ans

```
import csv
def Add_Device( ):
    fout=open("perpheral.csv","a",newline="\n")
    P_id=int(input("Enter Device Id: "))
    P_name=input("Enter Device name: ")
    Price=int(input("Enter Price: "))
    rec=[P_id,P_name,Price]
    csvw=csv.writer(fout)
    csvw.writerow(rec)
    fout.close( )
def Count_Device( ):
    fin=open("peripheral.csv")
    csvr=csv.reader(fin)
    ctr=0
    for record in csvr:
        if int(record[2])<1000:
            ctr=ctr+1
        print("Count of Price<1000 is",ctr)
    fin.close( )
Add_Device( )
Count_Device( )
```

DATA STRUCTURES



STACK: LIFO (Last in first out) to be implemented using list.

Concept

a linear data structure that follows the Last-In, First-Out (LIFO) principle

Operations

push(Stack-name, obj-name) use of append() function of list

pop(stack-name) Check for Underflow. If not Underflow then use pop() of list to return or display topmost element of stack.

peep(Stack-name) Check for Underflow. If not Underflow then use slicing logic to display topmost element of stack.

- Data structure is A set of rules and operations to organize and store data in an efficient manner. It is a way to store data in a structured way.
- Operations on data structure-
  - Traversal
  - Insertion
  - Deletion
- Types of data structures:

```
├── Linear Data Structures
│   ├── List (or Array)
│   ├── Stack
│   ├── Queue
│   └── Linked List
│
└── Non-Linear Data Structures
      ├── Tree
      └── Graph
```

- Built-in data structures available in Python: List, Tuple, Dictionary and Set.

- User Defined data structures in Python: Stack, Queue, Tree, Linked List etc.

STACK:

- 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
- Insertion and Deletion of elements will be done only from one end known as TOP.
- In Python, we can use 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 on Stack:

The Stack supports following operations:

1. Push: It adds an element to the TOP of the Stack.

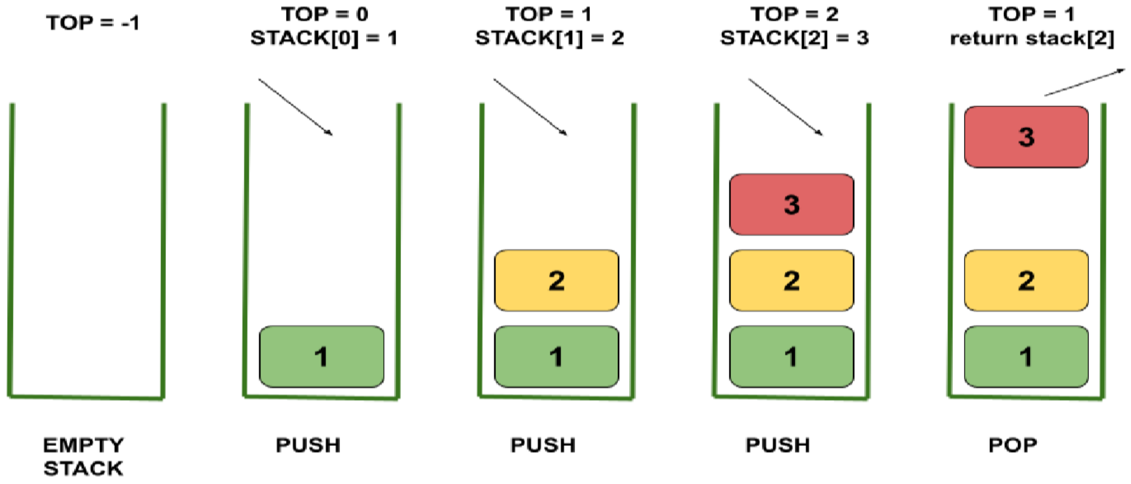
2. Pop: It removes an element from the TOP of the Stack.

3. Peek: It is used to know/display the value of TOP without removing it.

4. isEmpty: It is used to check whether Stack is empty.

OVERFLOW: It refers to the condition in which we try to PUSH an item in a Stack which is already FULL.

UNDERFLOW: It refers to the condition in which we are trying to POP an item from an empty Stack.



Stack Implementation in Python (Using List)

```
stack = []
# Function to push element into the stack
def push( ):
   element = input("Enter element to push: ")
   stack.append(element)
   print("Element  pushed to stack.")
# Function to pop element from the stack
```

```
def pop_element( ):
    if not stack:
        print("Stack is empty!")
    else:
        element = stack.pop( )
        print("Element popped from stack.")
# Function to display stack
def display( ):
    if not stack:
        print("Stack is empty!")
    else:
        print("Stack elements (top to bottom):")
        for item in reversed(stack):
            print(item)

# Menu-driven program
while True:
    print("\nSTACK OPERATIONS")
    print("1. Push")
    print("2. Pop")
    print("3. Display")
    print("4. Exit")
    choice = int(input("Enter your choice (1-4): "))
    if choice == 1:
        push( )
    elif choice == 2:
        pop_element( )
    elif choice == 3:
        display( )
    elif choice == 4:
        print("Exiting program...")
        break
    else:
        print("Invalid choice! Please try again.")
```

Multiple Choice Questions

1.	What is the principle of a stack? a) FIFO – First In First Out b) LIFO – Last In First Out c) FILO – First In Last Out d) LILO – Last In Last Out
2.	Which Python list method is used to add an element to the stack? a) insert( ) b) add( ) c) append( ) d) push ( )
3.	Which method is used to remove the top element from a stack implemented using a list? a) remove( ) b) delete( ) c) pop( ) d) discard ( )

4.	What will be the output of the following code?
	stack = [10, 20, 30]
	stack.pop( )
	print(stack)

	a) [10, 20, 30] c) [20, 30]	b) [10, 20] d) Error
5.	What happens if you call pop ( ) on an empty stack? a) Returns None c) Returns -1	 b) Raises IndexError d) Does nothing

Answers

1.	**B**	**2.**	**C**	**3.**	**C**	**4.**	**B**	**5.**	**B**

ASSERTION REASONING 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):** In **Python**, a stack can be implemented using a list. **Reason (R)*:*** A stack is an ordered linear list of elements that works on the principle of First In First Out (FIFO).
2.	**Assertion (A):** A stack can be used to reverse the contents of a text file line by line. **Reason (R):** In a stack, the last element inserted is the first to be removed (LIFO).
3.	**Assertion (A):** Using a stack is an efficient method for checking matching brackets in a file containing Python code. **Reason (R):** Stack allows multiple ends for insertion and deletion of elements.
4.	**Assertion (A):** A stack can be implemented using a list in Python to read a file and store each word for later processing. **Reason (R):** Lists in Python do not support push and pop operations.

Answers

1	**C**	**2**	**A**	**3**	**C**	**4**	**C**

Long Answer Questions

1.	A dictionary, **d_city** contains the records in the following format: **{state:city}** Define the following functions with the given specifications: (i) **push_city(d_city):** It takes the dictionary as an argument and pushes all the cities in the stack **CITY** whose states are of more than 4 characters. (ii) **pop_city( ):** This function pops the cities and displays **"Stack empty"** when there are no more cities in the stack.

Ans 1(i)	`CITY=[ ]`
	`def push_city(d_city):`
	`    for c in d_city:`
	`        if len(c) > 4:`
	`            CITY.append(d_city[c])`

1(ii)	`def pop_city( ):`
	`    while CITY:`

```
      print(CITY.pop( ))
   else:
      print("Stack empty")
```

2. Consider a list named Nums which contains random integers. Write the following user defined functions in Python and perform the specified operations on a stack named BigNums.

(i) PushBig( ): It checks every number from the list Nums and pushes all such numbers which have 5 or more digits into the stack, BigNums.

(ii) PopBig( ): It pops the numbers from the stack, BigNums and displays them. The function should also display "Stack Empty" when there are no more numbers left in the stack.

For example: If the list Nums contains the following data:

Nums = [213, 10025, 167, 254923, 14, 1297653, 31498, 386, 92765]

Then on execution of PushBig( ), the stack BigNums should store:

[10025, 254923, 1297653, 31498, 92765]

And on execution of PopBig( ), the following output should be displayed:

```
92765
31498
1297653
254923
10025
Stack Empty
```

Ans

```
def PushBig(Nums,BigNums):
   for N in Nums:
      if len(str(N)) >= 5:
      BigNums.append(N)
def PopBig(BigNums):
   while BigNums:
      print(BigNums.pop( ))
   else:
      print("Stack Empty")
```

3. A list contains following record of course details for a University:

[Course_name, Fees, Duration]

Write the following user defined functions to perform given operations on the stack named **'Univ'** :

(i) **Push_element( )** - To push an object containing the Course_name, Fees and Duration of a course, which has fees greater than 100000 to the stack.

(ii) **Pop_element( )** - To pop the object from the stack and display it. Also, display “Underflow” when there is no element in the stack.

For example:

If the lists of courses details are:

["MCA", 200000, 3]
["MBA", 500000, 2]
["BA", 100000, 3]
The stack should contain
["MBA", 500000, 2]
["MCA", 200000, 3]

Ans

```
Univ=[]
def Push_element(Course):
   for Rec in Course:
      if Rec[1]>100000:
         Univ.append(Rec)
def Pop_element( ):
   while len(Univ)>0:
      print(Univ.pop( ))
   else:
      print("Underflow")
```

4.

Write separate user defined functions for the following:

(i) PUSH(N) - This function accepts a list of names, N as parameter. It then pushes only those names in the stack named OnlyA which contain the letter 'A'.

(ii) POPA(OnlyA) - This function pops each name from the stack OnlyA and displays it. When the stack is empty, the message "EMPTY" is displayed.

For example :
If the names in the list N are
['ANKITA', 'NITISH', 'ANWAR', 'DIMPLE', 'HARKIRAT']
Then the stack OnlyA should store
['ANKITA', 'ANWAR', 'HARKIRAT']
And the output should be displayed as
HARKIRAT ANWAR ANKITA EMPTY

Ans

```
OnlyA=[ ]
def PUSH(N):
   for aName in N :
      if 'A' in aName :
         OnlyA.append(aName)
def POPA(OnlyA):
   while OnlyA :
      print(OnlyA.pop( ), end=' ')
   else :
      print('EMPTY')
```

5.

Write the following user defined functions:

(i) pushEven(N) - This function accepts a list of integers named N as parameter. It then pushes only even numbers into the stack named EVEN.

(ii) popEven(EVEN) - This function pops each integer from the stack EVEN and displays the popped value. When the stack is empty, the message "Stack Empty" is displayed.

For example:

If the list N contains:

[10,5,3,8,15,4]

Then the stack, EVEN should store

[10,8,4]

And the output should be

4 8 10 Stack Empty

Ans

```
EVEN=[ ]
def pushEven(N):
    for z in N :
        if z%2==0 :
            EVEN.append(z)
def popEven(EVEN):
    while EVEN :
        print(EVEN.pop( ), end=' ')
    else :
        print('Stack Empty')
```

6. Write the definition of a user defined function PushNV(N) which accepts a list of strings in the parameter N and pushes all strings which have no vowels present in it, into a list named NoVowel.

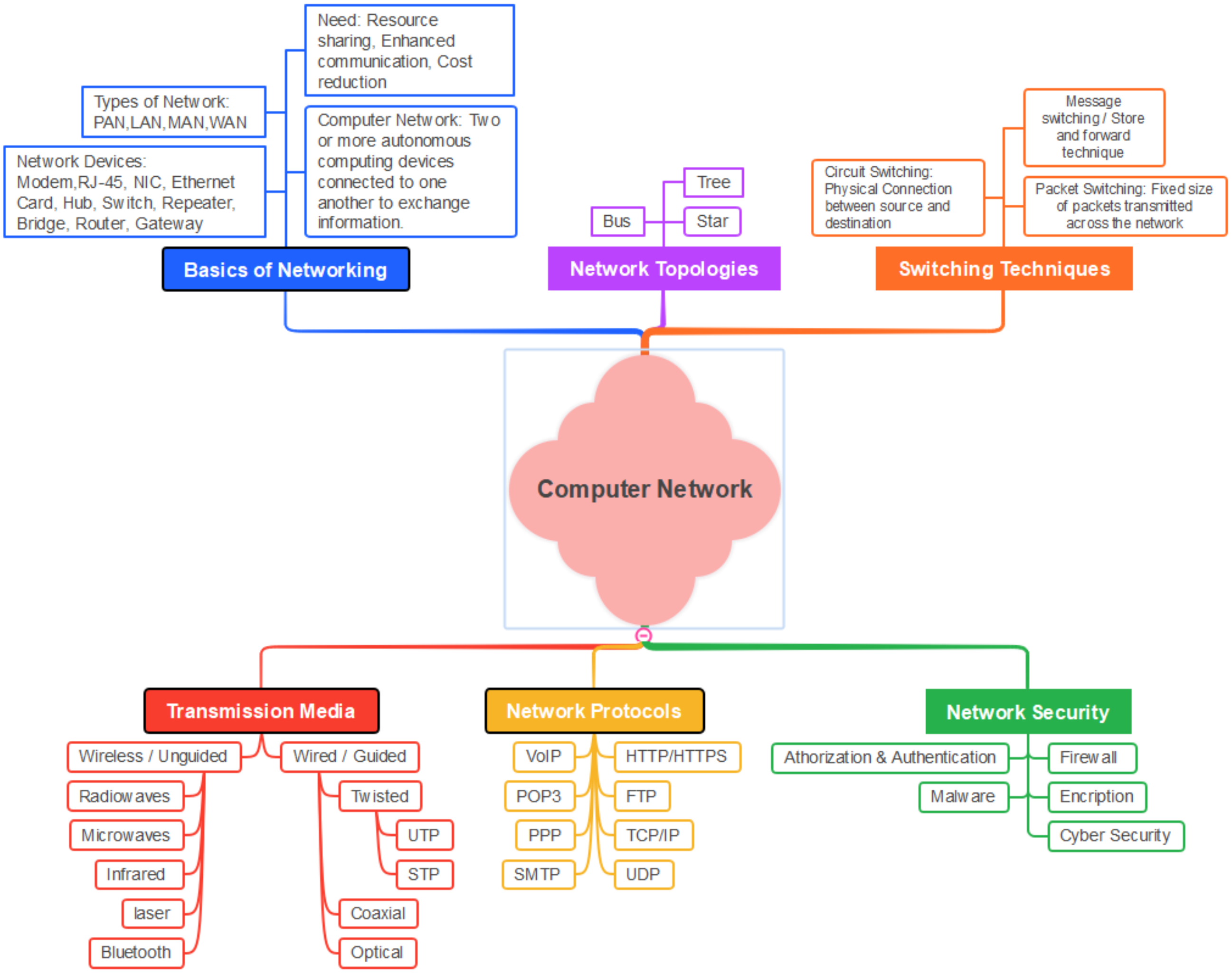
Write a program in Python to input 5 Words and push them one by one into a list named All.The program should that use the function PushNV( ) to create a stack of words in the list NoVowel so that it stores only those words which do not have any vowel present in it, from the list All.Thereafter, pop each word from the list NoVowel and display the popped word. When the stack is empty, display the message "EmptyStack".

Ans

```
def PushNV(N):
    for W in N :
        for C in W :
          if C.upper( ) in 'AEIOU':
              break
          else:
              NoVowel.append(W)
All=[ ]
NoVowel=[ ]
for i in range(5) :
    All.append(input('Enter a Word: '))
PushNV(All)
while NoVowel :
    print(NoVowel.pop( ), end=' ')
else :
    print('EmptyStack')
```

UNIT 2
COMPUTER NETWORKS

Concept Map:


Computer Network
Basics of Networking
Need: Resource sharing, Enhanced communication, Cost reduction
Computer Network: Two or more autonomous computing devices connected to one another to exchange information.
Types of Network: PAN,LAN,MAN,WAN
Network Devices: Modem,RJ-45, NIC, Ethernet Card, Hub, Switch, Repeater, Bridge, Router, Gateway
Network Topologies
Bus
Tree
Star
Switching Techniques
Circuit Switching: Physical Connection between source and destination
Message switching / Store and forward technique
Packet Switching: Fixed size of packets transmitted across the network
Transmission Media
Wireless / Unguided
Radiowaves
Microwaves
Infrared
laser
Bluetooth
Wired / Guided
Twisted
UTP
STP
Coaxial
Optical
Network Protocols
VoIP
POP3
PPP
SMTP
HTTP/HTTPS
FTP
TCP/IP
UDP
Network Security
Athorization &amp; Authentication
Malware
Firewall
Encription
Cyber Security

Network: A group of two or more similar things or people interconnected with each other is called network. Examples are social network and mobile network

Computer Network: A Computer network is an interconnection among two or more computers or computing devices. The advantages of computer networks are:

- Resource Sharing
- Collaborative Interaction
- Cost Saving
- Increased storage
- Time Saving

EVOLUTION OF NETWORK:

(I)ARPANET (Advanced Research Project Agency Network)

- It came into existence in 1960s
- A project for interconnecting, US department of defense with academic and research organization across different places for scientific collaboration.

(II)NSFNET (National Science Foundation Networks)

- It came into existence in 1986
- It was the first large-scale implementation of Internet technologies in a complex environment of many independently operated networks

(III) INTRANET

- It is a local or restricted communication system
- It is managed by a person or organization.
- Intranet users can avail services from internet but Internet user cannot access intranet directly

(IV) INTERNET

- It came into existence in 1960s
- It is known as Network of Networks
- A global computer network providing variety of information and communication facilities consisting of interconnected networks using standardized communication protocols.

DATA COMMUNICATION TERMINOLOGIES

DATA: Data means information in digital form such as text, audio, video which is stored processed and exchanged between digital devices like computer, mobile phones or laptop. Computers process the raw data into meaningful information. Information is processed data.

COMMUNICATION: The exchange of information between two or more networked or interconnected devices is called communication

COMPONENTS OF DATA COMMUNICATION

a) SENDER: Sender is a device which is capable of sending data over a communication network. In data communication Sender is also called Source.

b) RECEIVER: Receiver is a device which is capable of receiving data over a communication network. In data communication Receiver is also called Destination.

c) MESSAGE: message is the information being exchanged between a sender and a receiver over a communication network.

d) COMMUNICATION MEDIUM: Communication medium is the path or channel through which the information is moved from the sender to the receiver. A communication medium can be either wired/guided or wireless/unguided.

e) PROTOCOLS: The set of standard rules which are followed in data communication are known as Data Communication Protocols. All the communicating devices like sender receiver and other connected devices in the network should follow these protocols.

Why Protocols are needed?

The communicating devices may be in different geographical areas. The speed of these devices may be different. Also, the data transfer rates of different networks may be different. These complexities make it necessary to have a common set of rules to ensure the secure communication of data. Some commonly used Protocols in data communication are:

- Transmission Control Protocol (TCP)
- Internet Protocol (IP)
- File Transfer Protocol (FTP)
- Simple Mail Transport Protocol (SMTP)
- Hyper Text Transfer Protocol (HTTP)

MEASURING CAPACITY OF COMMUNICATION MEDIA

Capacity of a communication channel means the maximum quantity of signals that a communication channel can carry. The capacity of a communication medium is measured by its bandwidth and data transfer rate.

BANDWIDTH: Bandwidth is the difference between the highest and lowest frequencies a transmission media can carry. The unit of bandwidth is Hertz.

DATA TRANSFER RATES: Data transfer rate is the number of bits transmitted through a channel per unit of time. Data transfer rate is measured in bits per second (bps). It is also measured in Kilobits per second (Kbps), Megabits per second (Mbps) or Gigabits per second (Gbps).

IP ADDRESS: IP address or Internet Protocol address is a unique numeric address assigned to every device connected to a network. It uniquely identifies every node connected to a local network or internet. Example IP address: 24.171.248.170

SWITCHING TECHNIQUES

In large networks, there may be more than one paths for transmitting data from sender to receiver. The process of selecting a path of data out of the available paths is called switching. There are two popular switching techniques – circuit switching and packet switching.

1. Circuit Switching : In circuit switching, whenever a source end node wants to send a message to the destination end node a physical link is first established between the source and the destination. Then only the data transmission takes place. Example: telephone network

2. Packet Switching : In the packet switching technique, the whole message is split into small packets. Now, these packets are transmitted one by one from sender to the receiver through the intermediary switches in the network. The packets will take shortest path as possible.

TRANSMISSION MEDIA

- Transmission media are the channels used to carry data signals.
- They are broadly classified into
    - ○ **Wired / Guided Media**

      Includes:
        - ▪ Twisted pair cable
        - ▪ Coaxial cable
        - ▪ Fiber-optic cable

      Features:
        - ▪ High Speed

        - Secure
        - Used for comparatively shorter distances
    - **Wireless/Un-guided media**

      Includes:
        - Radio waves
        - Microwaves
        - Infrared waves.

      Features:
        - The signal is broadcasted through air
        - Less Secure
        - Used for larger distances

Wired Communication Media:

- **Twisted Pair Cable:**
    - Consists of two separately insulated copper wires twisted together to reduce electromagnetic interference.
    - It is commonly used for telephone lines and local area networks (LANs).
    - Twisted Pair is of two types:

1. Unshielded Twisted Pair (UTP):

- UTP consists of two insulated copper wires twisted around one another.
- This type of cable has the ability to block interference and does not depend on a physical shield for this purpose.
- It is used for telephonic applications.

Advantages of Unshielded Twisted Pair

- Least expensive
- Easy to install
- High-speed capacity

Disadvantages of Unshielded Twisted Pair

- Lower capacity and performance in comparison to STP
- Short distance transmission due to attenuation

2. Shielded Twisted Pair

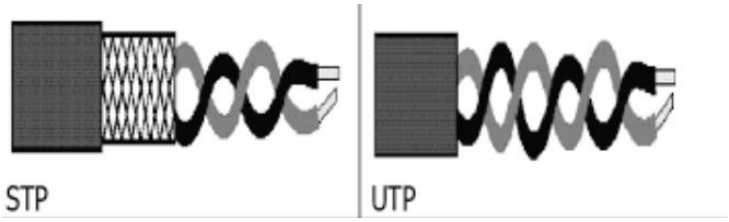
- Shielded Twisted Pair (STP): Shielded Twisted Pair (STP) cable consists of a special jacket (a copper braid covering or a foil shield) to block external interference.
- It is used in fast data rate Ethernet and in voice and data channels of telephone lines.

Advantages of Shielded Twisted Pair

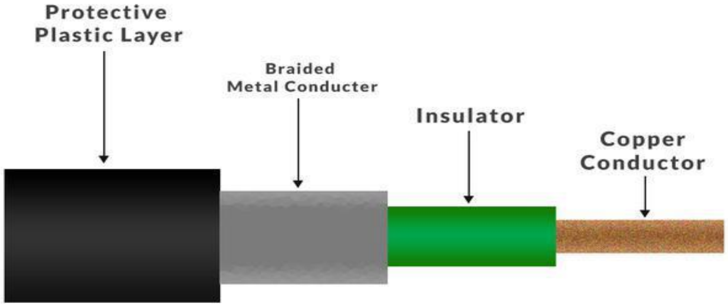
- Better performance at a higher data rate in comparison to UTP
- Eliminates crosstalk
- Comparatively faster

Disadvantages of Shielded Twisted Pair

- Comparatively difficult to install and manufacture
- More expensive
- Bulky



- **Coaxial Cable:**
  - Coaxial cable has an outer plastic covering containing an insulation layer made of PVC or Teflon and 2 parallel conductors each having a separate insulated protection cover.
  - The coaxial cable transmits information in two modes: Baseband mode(dedicated cable bandwidth) and Broadband mode(cable bandwidth is split into separate ranges).
  - Cable TVs and analog television networks widely use Coaxial cables.



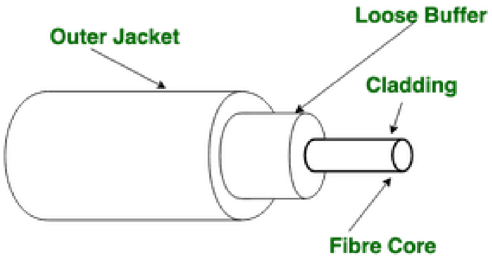
Advantages of Coaxial Cable

- Coaxial cables have high bandwidth.
- It is easy to install.
- Coaxial cables are more reliable and durable.
- Less affected by noise or cross-talk or electromagnetic inference.
- Coaxial cables support multiple channels

Disadvantages of Coaxial Cable

- Coaxial cables are expensive.
- The coaxial cable must be grounded in order to prevent any crosstalk.
- As a Coaxial cable has multiple layers it is very bulky.
- There is a chance of breaking the coaxial cable and attaching a “t-joint” by hackers, this compromises the security of the data.

- **Optical Fiber Cable**
  - Optical Fibre Cable uses the concept of total internal reflection of light through a core made up of glass.
  - The core is surrounded by a less dense glass or plastic covering called the coating.
  - It is used for the transmission of large volumes of data.
  - The cable can be unidirectional or bidirectional.
  - The WDM (Wavelength Division Multiplexer) supports two modes, namely unidirectional and bidirectional mode.



Advantages of Optical Fibre Cable

- Increased capacity and bandwidth
- Lightweight
- Less signal attenuation
- Immunity to electromagnetic interference
- Resistance to corrosive materials

Disadvantages of Optical Fibre Cable

- Difficult to install and maintain
- High cost

Applications of Optical Fibre Cable

- **Medical Purpose:** Used in several types of medical instruments.
- **Defence Purpose:** Used in transmission of data in aerospace.
- **For Communication:** This is largely used in formation of internet cables.
- **Industrial Purpose:** Used for lighting purposes and safety measures in designing the interior and exterior of automobiles.

Wireless Communication Media:

Radio Waves

Radio waves are easy to generate and can penetrate through buildings. The sending and receiving antennas need not be aligned.

Frequency Range:3KHz - 1GHz. AM and FM radios and cordless phones use Radio waves for transmission.

Types of Radio Waves:

- **Short Wave:** AM Radio
- **VHF (Very High Frequency):** FM Radio/TV
- **UHF (Ultra High Frequency):** TV

Radio Wave Components:

- **Transmitter:** Responsible for encoding the signal.
- **Receiver:** Responsible for decoding the signal.

Microwaves

It is a line-of-sight transmission i.e. the sending and receiving antennas need to be properly aligned with each other. The distance covered by the signal is directly proportional to the height of the antenna.

Frequency Range:1GHz - 300GHz.

Micro waves are majorly used for mobile phone communication and television distribution.

Advantages:

- Cheaper than using cables
- Freedom from land acquisition
- Ease of communication in difficult terrains

- Communication over oceans

Disadvantages:

- Insecure communication.
- Out of phase signal.
- Susceptible to weather conditions.
- Bandwidth is limited.
- High cost of design, implementation, and maintenance.

Infrared

Infrared waves are used for very short distance communication. They cannot penetrate through obstacles. This prevents interference between systems.

Frequency Range:300GHz - 400THz.

It is used in TV remotes, wireless mouse, keyboard, printer, etc.

Difference Between Radio Waves, Micro Waves, and Infrared Waves

Basis	**Radiowave**	**Microwave**	**Infrared wave**
Direction	omni-directional	unidirectional	unidirectional
Penetration	At low frequency, they can penetrate through solid objects and walls but high frequency they bounce off the obstacle.	At low frequency, they can penetrate through solid objects and walls. at high frequency, they cannot penetrate.	They cannot penetrate through any solid object and walls.
Frequency range	3 KHz to 1GHz.	1 GHz to 300 GHz.	300 GHz to 400 GHz.
Security	These offers poor security.	These offers medium security.	These offers high security.
Attenuation	is high.	is variable.	is low.
Government License	Some frequencies in the radio-waves require government license to use these.	Some frequencies in the microwaves require government license to use these.	There is no need of government license to use these waves.
Usage Cost	moderate.	high.	very less.
Communication	used in long distance communication.	Used in long distance communication.	not used in long distance communication.

NETWORK DEVICES

Modem: Stands for "modulator-demodulator.", converts digital data from a computer into analog signals for transmission over telephone lines or cable systems. Also it converts incoming analog signals back into digital data for the computer. Used to connect to the internet via ISP (Internet Service Provider).

Ethernet Card: Also known as a network interface card (NIC)., enables a computer to connect to an Ethernet network using Ethernet cables, Essential for wired network connections._Provides a physical interface for networking using an RJ45 connector

RJ45 Connector: Registered Jack 45 connector, used to connect Ethernet cables to devices such as computers, switches, and routers, Ensures a secure and reliable physical connection. **Repeater**: Amplifies and retransmits signals in a network, extends the range of network signals, especially in large or congested environments. Used to combat signal loss over long distances.

Hub: A basic networking device that connects multiple devices in a network, Broadcasts data to all connected devices, causing network congestion and inefficiency, which can lead to collisions.

Switch: Intelligent device that connects devices in a network, Forwards data only to the device that needs it, improving network performance and efficiency by reducing collisions.

Router: Manages traffic between different networks, such as your home network and the internet, performs functions like assigning IP addresses, directing data, and providing security.

Gateway: Acts as an entry and exit point for data traveling between different networks or protocols (e.g., LAN to WAN), translates data between different formats or protocols to ensure smooth communication.

Wi-Fi Card: A wireless network adapter that allows a computer to connect to Wi-Fi networks. Commonly found in laptops and mobile devices for wireless internet access

COMPUTER NETWORK TYPES

A computer network can be categorized by their size, complexity and geographical spread. A computer network is mainly of four types:

- LAN (Local Area Network)
- PAN (Personal Area Network)
- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)

PAN (Personal Area Network): Personal Area Network is a network of information technology devices (laptop, mobile phones, media player and play stations) arranged within the range of an individual person, typically within a range of 10 meters / covers an area of 30 feet.

LAN (Local Area Network): Local Area Network is a group of computers connected to each other in a small area such as a building, office through a communication medium such as twisted pair, coaxial cable, etc to share resources.

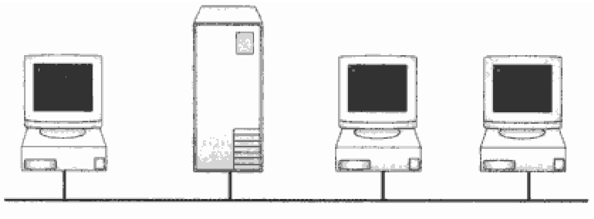
MAN (Metropolitan Area Network): A metropolitan area network is a network that covers a larger geographic area that is spread over an area as big as a city by interconnecting different LAN to form a larger network through a telephone exchange line.

WAN (Wide Area Network): A Wide Area Network is a network that extends over a large geographical area such as states or countries through a telephone line, fiber optic cable or satellite links. The internet is one of the biggest WAN in the world.

NETWORKING TOPOLOGIES

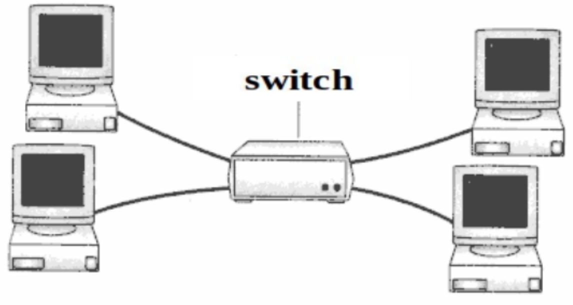
Bus

It uses a single cable, called a trunk or segment, along which all the computers of the network are connected



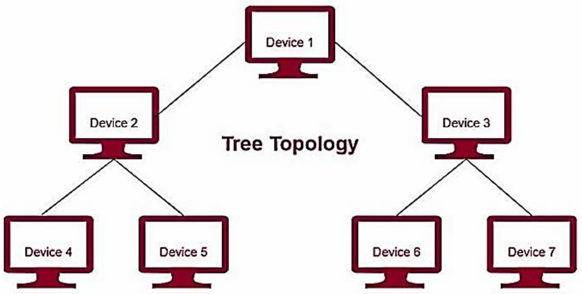
Star

All computers are connected using cable segments to a central component called a switch. The signals from the transmitting computer go through the switch to all the others.



Tree

Tree Topology is a topology which is having a tree structure in which all the computers are connected like the branches which are connected with the tree.



NETWORK PROTOCOL: A protocol means the rules that are applicable for a network. Protocols defines standardized formats for data packets, techniques for detecting and correcting errors etc.

TCP/IP (Transmission Control Protocol/ Internet Protocol)

- The IP protocol ensures that each computer or node connected to the Internet is assigned an IP address, which is used to identify each node independently.
- TCP ensures that the message or data is broken into smaller chunks, called IP packets. Each of these packets are routed (transmitted) through the Internet, along a path from one router to the next, until it reaches the specified destination. TCP guarantees the delivery of packets on the

designated IP address. It is also responsible for ordering the packets so that they are delivered in sequence.

FTP (File Transfer Protocol): It is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another. It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet. It is also used for downloading the files to computer from other servers.

SMTP (Simple Mail Transfer Protocol.): SMTP is a set of communication guidelines that allow software to transmit an electronic mail over the internet is called Simple Mail Transfer Protocol.

Point-to-Point Protocol (PPP) is protocol that is used to directly connect one computer system to another. Computers use PPP to communicate over the telephone network or the Internet.

Post Office Protocol version 3 (POP3) Protocol: (POP3) is a standard mail protocol used to receive emails from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline. J, and JVM

Telnet: Telnet is a program that allows a user to log on to a remote computer. Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side.

VoIP: VoIP stands for **Voice over Internet Protocol**. It is also referred to as IP telephony, internet telephony, or internet calling.

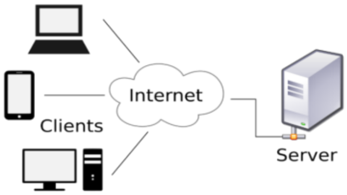
Web Services: Web Services means the services provided by World Wide Web. The World Wide Web provides services like chatting, emailing, video conferencing, e-learning, e-shopping, e-reservation, e-groups and social networking.

World Wide Web (WWW): The World Wide Web commonly referred to as WWW, W3, or the Web is an interconnected system of public webpages accessible through the Internet. It was invented by Tim Berners-Lee in 1989. Major components of WWW are:

1. Web Server – It is a computer that stores website resources (web pages, images, videos, etc.).
2. Web Browser (Client) - A software application used to access the web resources.
3. Webpage - Hypertext documents formatted in Hypertext Mark-up Language (HTML) and displayed in a web browser.
4. Website - A website is a collection of inter-linked web pages that is identified by a common domain name (website name) and stored on a web server.
5. HTTP Protocol - It governs data (web page) transfer between a server and a client.
6. HTML- A mark-up language used to specify the structure of a webpage.
7. URL- Address used to identify documents and other web resources on the internet.

WEB ARCHITECTURE

Web is working based on a client-server architecture.



Client: It is a computer capable of requesting, receiving & displaying information in the form of web pages or using a particular service from the service providers (Servers).

Servers: It is a remote computer which provides/transfers information to the client (in the form of web pages) or access to particular services.

Difference between Internet and WWW

Internet	**World Wide Web(WWW)**
Internet stands for Interconnected Networks	WWW stands for World wide Web
Internet is a means of connecting a computer to any other computer anywhere in the world.	World Wide Web which is a collection of information which is accessed via the Internet.
Internet is infrastructure.	WWW is service on top of that infrastructure.
Internet is primarily hardware-based	WWW is more software-oriented as compared to the Internet.
Internet uses TCP/IP protocol.	WWW uses HTTP Protocol.

HTML (Hypertext Mark-up Language): It is a mark-up language that tells web browsers how to structure the web pages you visit. It has a variety of tags and attributes for defining the layout and structure of the web document. A HTML document has the extension .htm or .html. Hypertext is a text which is linked to another html document via clickable links known as hyperlinks.

XML (eXtensible Mark-up Language): XML is a mark-up language like HTML but it is designed to transport or store data. It does not have predefined tags but allows the programmer to use customized tags. An XML document has the extension .xml.

HTML v/s XML

HTML	**XML**
HTML stands for Hyper Text Mark-up Language	XML stands for eXtensible Mark-up Language
HTML is a case insensitive.	XML is case sensitive.
Predefined tags (commands).	User defined tags (commands).
It is used for presentation of the Data.	It is used for transportation of the Data.
Small errors can be ignored.	Errors not allowed.
Closing tags are optional.	Compulsory to use closing tags.

HTTP - Hyper Text Transfer Protocol: HTTP is used to transfer data across the web. HTTP specifies how to transfer hypertext (linked web documents) between two computers. It allows users of the World Wide Web to exchange information found on web pages.

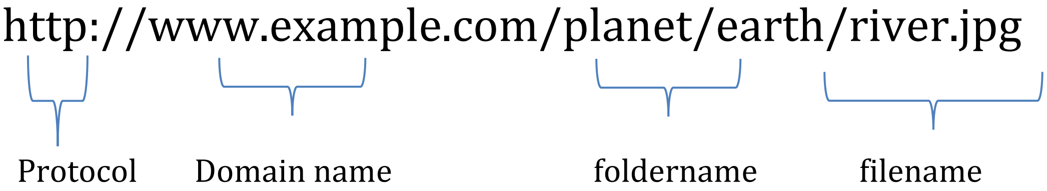
Hypertext Transfer Protocol Secure (HTTPS) is an extension of the HTTP which is used for secure communication over a computer network.

Domain Names: Every device connected to the Internet has a numeric IP address which is very difficult to remember. Each computer server hosting a website or web resource is given a name known as Domain Name corresponding to unique IP addresses. For example, IP addresses and domain names of some websites are as follows:

Domain Name	IP Address
ncert.nic.in	164.100.60.233
cbse.nic.in	164.100.107.32

The process of converting a hostname (such as www.google.com) into the corresponding IP address (such as 172.217.14.196) is called domain name resolution. Specialized DNS servers are used for domain name resolution (DNS resolution).

URL-Uniform Resource Locator: Every web page that is displayed on the Internet has a specific address associated with it, this address is known as the URL. The structure of a URL can be represented as follows:



http://www.example.com/planet/earth/river.jpg

Protocol Domain name foldername filename

The URL consists of four basic parts, namely, protocol, hostname, folder name and the filename. Each one of these has a specific function.

1) The "protocol" indicates the type of Protocol (http/https/ftp etc.) being used. The protocol is always followed by "://" and the host name.
2) The host name/domain name is the Internet address of a remote computer on which the files reside.
3) The folder name indicates the name of the directory in which the files are located.
4) The filename specifies the name of the specific document to be displayed in the browser. The filename itself consists of two pieces of information, the name of the file to be displayed and the file extension, which specifies the file type (.htm for HTML file, .txt for a text file, .bmp for a bitmap image, etc.)

The structure of a URL can be represented as follows:

Websites: A website is a collection of linked web pages (plus their associated resources) that share a unique domain name.

Web page: Web page is an electronic document designed using HTM linked with hyperlinks.

Web Browser: Web browser is software program to navigate the web pages on the internet. E.g., Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera etc.

Cookie: A cookie is a small text file that stores information stored on your computer. Cookies often store your settings for a website, such as your preferred language or location, pages visited.

Web Server: A web server is a computer or a group of computers hosting one or more websites. E.g., Apache, IIS etc.

Web Hosting: Web hosting is the process of uploading/saving the web content on a web server to make it available on WWW.

Multiple Choice Questions

1	What are the three common types of computer networks? a. ROM, MAN, LAN b. RAM, WAN, LAN c. MAN, LAN, WAN d. None of the above
2	What is the Full form of LAN? a. Local Area Network b. Local Access Network c. Line And Networking d. Line-less Networking
3	Define what a LAN is?

	a. Connected devices share the resources of a single processor or server within a small geographic area b. Normally find within a business and school c. These are computers that share resources over a large area d. None of the above
4	Mr. John is a small businessman who runs Hardware store. He has been experiencing problems with his small accounting department, which he depends on to provide sales reports. Mr. John wants to share information between his 7 computer stations and have one central printing area. What type of network would you recommend to Mr. John? a. MAN b. LAN c. WAN d. SAN
5	WAN covers a larger geographical area than MAN? a. True b. False
6	A network that consists of both LANs and MANs is called a Wide area network? a. True b. False
7	Arrange the Following Types of Networks according to their size, from largest to smallest? a. LAN, WAN, MAN b. WAN, LAN, MAN c. MAN, LAN, WAN d. WAN, MAN, LAN
8	You are a member of a club that deals with computer networks. The club has to take a project to build a MAN. Where would this project likely take place? a. A small building/organization b. University or college c. Home d. None of the above
9	What is the full form of MAN ? a. Magnetic Access Network b. Metropolitan Area Network c. Multi-Area Network d. Multi-Access net
10	In your school there is a library, and you can use the internet to do research, this library will most likely be a WAN network? a. True b. False
11	Types of Networks are Categories by their Geographical Area cover? a. True b. False
12	What's a web browser? a) A kind of spider b) A computer that store www files c) A person who likes to look at websites d) A software program that allows you to access sites on the World Wide Web
13	A ____ is a document commonly written and is accessible through the internet or other network using a browser? a) Accounts b) Data c) Web page d) Search engine
14	Which of the following is used to read HTML code and to render Webpage? a) Web Server b) Web Browser c) Web Matrix d) Weboni
15	Which of the following is a Web Browser? a) MS-office b) Notepad c) Firefox d) Word 2007

16	Which of the following is not a web browser? a) Chrome b) Microsoft Edge c) Safari d) Ubuntu
17	URL stands for (a) Uniform Research Limited (b) Uniform Resource Locator (c) Uniform Resource Labs (d) Uniform Research Locator
18	LinkedIn is an example for ___________________ website. (a) E-learning (b) E-commerce (c) Video conferencing (d) Social networking
19	Which of the following is not a web service? (a) Distance Learning (b) E-mailing (c) Video conferencing (d) Social networking
20	Web browsers are also called as _____________ (a) Web Servers (b) Web Clients (c) Web Hosting (d) Web Designing
21	Working of WWW based on _________________ architecture. (a) Peer-To-Peer architecture (b) Client-Client architecture (c) Client-Server architecture (d) Server-Server architecture
22	______________ is computer software capable of requesting, receiving & displaying information in the form of webpages. (a) Web Servers (b) Web Browser (c) Web Designers (d) Web Camera
23	______________ is a mark-up language that helps in developing web pages. (a) HTTP (b) HTML (c) XML (d) C++
24	______________ is a language used to transport data over internet. (a) HTTP (b) HTML (c) XML (d) C++
25	.__________ is a set of rules for communication between two computers over a network. (a) Modem (b) Protocol (c) Switch (d) IP address

ANSWERS

1. **c**	2. **a**	3. **a**	4. **b**	5. **a**
6. **a**	7. **d**	8. **d**	9. **b**	10. **a**
11. **a**	12. d	13. **c**	14. **b**	15. **c**
16. d	17. b	18. d	19. **a**	20. **b**
21. c	22. **b**	23. **b**	24. **c**	25. **b**

Assertion Reason 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
1	**Assertion (A):** A Repeater is a device that amplifies the network over geographical distance. **Reasoning (R):** A Hub is a device which is used to connect more than one device in the network.
2	**Assertion (A):** VoIP stands for Voice over Internet Protocol.

	Reasoning (R): VolP is a technology that allows us to make voice calls using a broadband connection instead of a regular phone line.
3	**Assertion (A):** A protocol defines the set of rules that are applicable for data sharing and communication over a network. **Reasoning (R):** Local Area Network (LAN) is an example of protocol.
4	**Assertion (A):** A Gateway is a network device that connects dissimilar networks. **Reasoning (R):** A Gateway establishes a connection between local network and external network.
5	**Assertion(A):** On a computer network the users work on network nodes only. **Reason(B):** A server cannot act as a network node.

Assertion Reason Answers

1	**B**	**2**	**A**	**3**	**C**	**4**	**A**	**5**	**C**

Very Short Answer Type Questions

1	Name the transmission media best suitable for connecting to hilly areas
2	How many pair of wires are there in twisted pair cable (Ethernet)?
3	Name a device that forwards data packets along networks.
4	What is the full form of WWW?
5	What is the full form of Internet?
6	Who invented the WWW in 1989?
7	Special software's that is used to view webpages are ___________
8	_______ are used to store webpages, so whenever a request, it will serve the request.
9	___________ are programs /computers used to store information's in the form of webpages.
10	Web pages that are linked to each other via _________
11	___________ protocol is used to transfer web pages over internet.
12	Full form of HTTP?
13	.____________ is a massive collection of digital pages to access information over the Internet
14	Write any 2 differences between HTML & XML?
15	.__________ is a real-time communication between two or more users via computer.

ANSWERS

1	Microwave / Radio wave
2	Two insulated copper wires
3	Router
4	World Wide Web
5	Interconnected Networks
6	Tim Berners-Lee
7	Web browsers
8	Web servers
9	Web servers
10	Hyperlinks

<table>
<tr><td>11</td><td>HTTP- HyperText Transfer Protocol</td></tr>
<tr><td>12</td><td>HyperText Transfer Protocol</td></tr>
<tr><td>13</td><td>World Wide Web(WWW) or Web</td></tr>
<tr><td>14</td><td>
<table>
<tr><th>HTML</th><th>XML</th></tr>
<tr><td>HTML stands for Hyper Text Mark-up Language</td><td>XML stands for eXtensible Mark-up Language</td></tr>
<tr><td>HTML is a case insensitive.</td><td>XML is case sensitive.</td></tr>
<tr><td>Predefined tags (commands).</td><td>User defined tags (commands).</td></tr>
<tr><td>It is used for presentation of the Data.</td><td>It is used for transportation of the Data.</td></tr>
<tr><td>Small errors can be ignored.</td><td>Errors not allowed.</td></tr>
<tr><td>Closing tags are optional.</td><td>Compulsory to use closing tags.</td></tr>
</table>
</td></tr>
<tr><td>15</td><td>Chat</td></tr>
</table>

Short Answer Type Questions

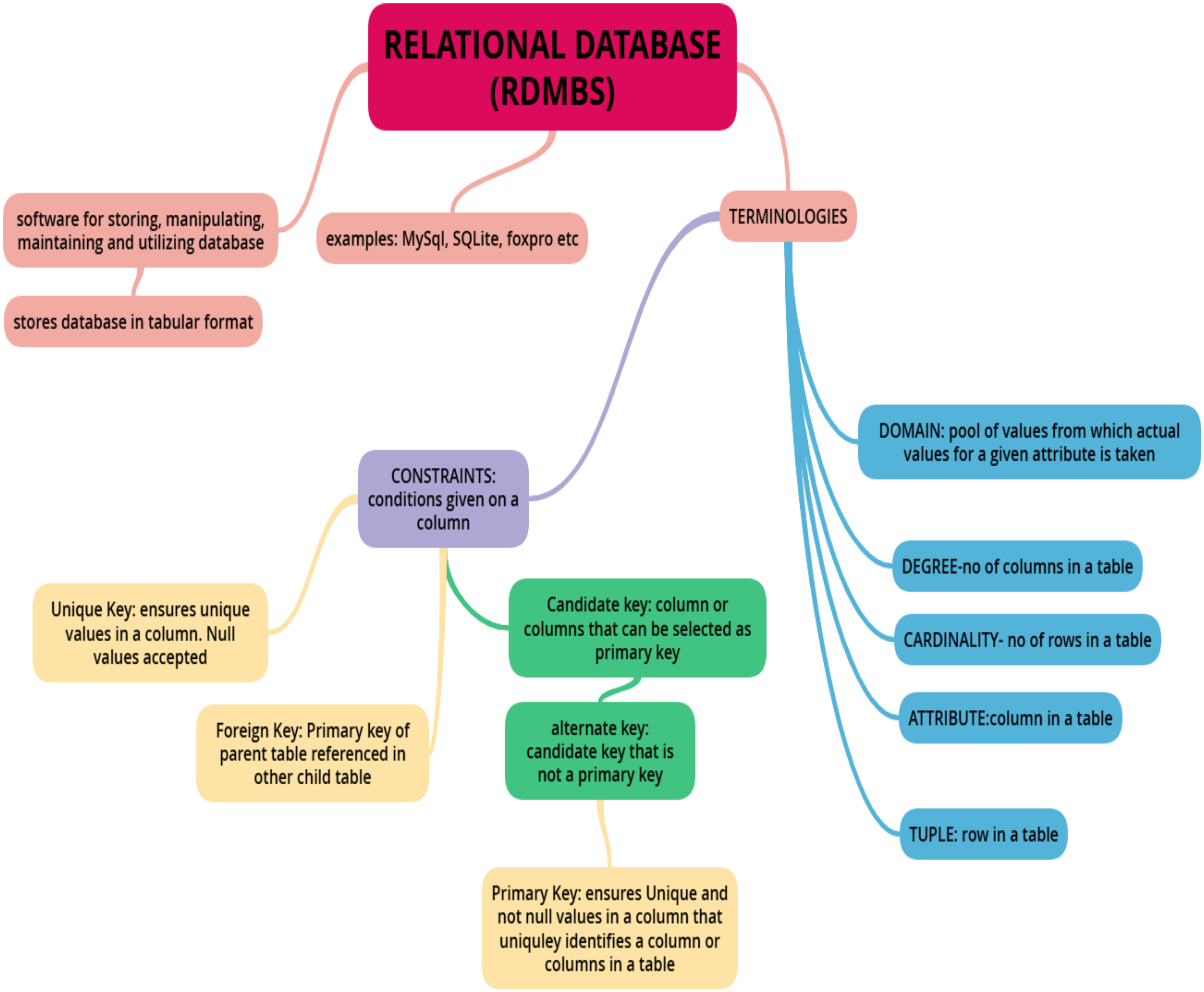
1	What are Protocols? Name the protocol used to transfer a file from one device to the other.
2	What is meant by an IP Address? Give an example for IP Address
3	Explain how an IP Address become helpful in investigating cyber-crimes
4	Why Protocols are needed in the case of Data Communication?
5	What is the difference between World Wide Web & Internet?
6	What is a protocol, give some examples?
7	What is the difference between E-mail and chat?
8	What are cookies?
9	What is the difference between domain name and IP address?
10	Give one suitable example of each URL and domain name?

ANSWERS

1	Protocols are set of rules that are followed while transmitting data through a computer network. Protocols determines how to data can be moved securely from a source device to a destination device. The protocol used for transferring a file from one device to another is the File Transfer Protocol (FTP)
2	An IP Address is a numerical address that uniquely identifies every device connected to a network or internet. The user's physical location can be tracked by using an IP Address. IP V4 (IP Version 4) is a popular version of IP Address. IP Address (in IP V4) consists of four set of numbers separated by a dot. These numbers can range from 0 to 255. An example IP Address format is given below: 192.158.12.38
3	IP address can be used to trace the physical location of a user connected to a network. By this many cyber crime can be investigated and traced out efficiently tracking the exact location from where the cybercrime is carried out.

4	The communicating devices may be in different geographical areas. The speed of these devices may be different. Also, the data transfer rates of different networks may be different. These complexities make it necessary to have a common set of rules i.e., Protocols to ensure the secure communication of data
5	Internet means interconnected networks that spread all over the world (i.e. the physical infrastructure), while WWW means the information's (available in the form of webpages) that can be accessed through internet.
6	Protocols are set of rules that are followed while transmitting data through a computer network. Protocols determines how to data can be moved securely from a source device to a destination device. The protocol used for transferring a file from one device to another is the File Transfer Protocol (FTP)
7	In order to chat, you need to have an account on the same service as the person you are chatting with. e.g. on the other hand, in case of E-mail, it is not necessary, i.e. you can have an account from any provider and you can establish your own.
8	Cookies are files stored temporarily on www browser's computer, which allow the www server to store persistent information associated with browsing user on user's system
9	IP addresses look like this: 192.168.12.134. Domain names look like this: "www.google.com" Domain names are easier for us to remember and use, while computers are quite handy with numbers. Thus, we use DNS (Domain Naming System) to translate domain names into the IP addresses. IP address is a unique identifier for a computer or device on internet. A domain name (website name) is a name that identifies one or more IP addresses (when hosted at different servers for load balancing).
10	URL: https://kvsangathan.nic.in/hq-gyan-kosh Domain name: kvsangathan.nic.in

UNIT 3
DATABASE MANAGEMENT


RELATIONAL DATABASE (RDMBS)
software for storing, manipulating, maintaining and utilizing database
stores database in tabular format
examples: MySql, SQLite, foxpro etc
TERMINOLOGIES
DOMAIN: pool of values from which actual values for a given attribute is taken
DEGREE-no of columns in a table
CARDINALITY- no of rows in a table
ATTRIBUTE:column in a table
TUPLE: row in a table
CONSTRAINTS: conditions given on a column
Unique Key: ensures unique values in a column. Null values accepted
Foreign Key: Primary key of parent table referenced in other child table
Candidate key: column or columns that can be selected as primary key
alternate key: candidate key that is not a primary key
Primary Key: ensures Unique and not null values in a column that uniquley identifies a column or columns in a table

A database is an organized collection of interrelated data that serves many applications. Its is generally a computer record keeping system. In a database we can not only store the data but we can also change the data as per the user requirement. These databases are generally managed by special software called DBMS (Database Management System)

Database Management System (DBMS):

It is software which is responsible for storing, manipulating, maintaining and utilizing the databases.

Database System:

A database along with the DBMS is referred to as database system.
Eg: Oracle, MySQL Server, MySQL, Sybase, SQLite, PostgreSQL, FoxPro, SAP , dBase

Need for DBMS:

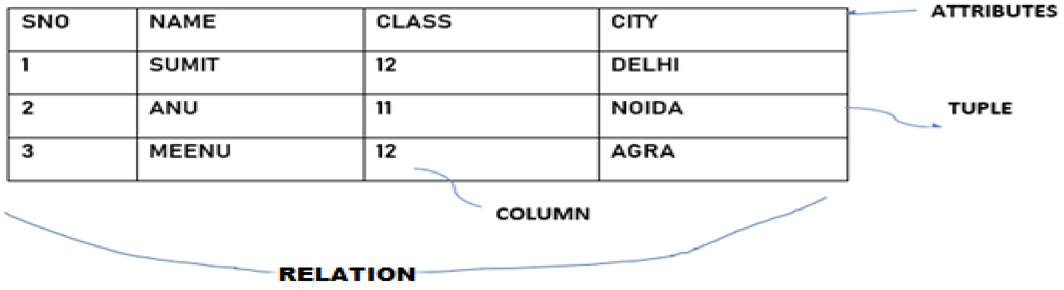
- Databases reduce redundancy i.e. it removes the duplication of data.

- Database controls inconsistency i.e. when two copies of the same data do not agree to each other it is called inconsistency. By controlling redundancy, the inconsistency is also controlled
- Databases allows sharing of data
- Database ensures data security by the process of authentication and does not allow unauthorized access. Database Maintains integrity
- Database is maintained in a standard format which helps to interchange the data between two systems.

RELATIONAL DATABASE MODEL

In Relational Database Model, the data is stored in the form of tables i.e. rows and columns.

- In Relational Database Model a table is referred to as a Relation.
- In Relation Database Model a column is referred to as an attribute
- In relational database model a row is referred to as a tuple.



SNO	NAME	CLASS	CITY
1	SUMIT	12	DELHI
2	ANU	11	NOIDA
3	MEENU	12	AGRA

Relation: A Relation is logically related data organized in the form of tables.
Attribute/ Field: Column of a table is called Attribute or Field.
Tuple/ Entity/ Record: Rows of a table is called Tuple or Record.
Domain: It is collection of values from which the value is derived for a column.
Degree - Number of columns (attributes) in a table.
Cardinality - Number of rows (Records) in a table.
Keys:

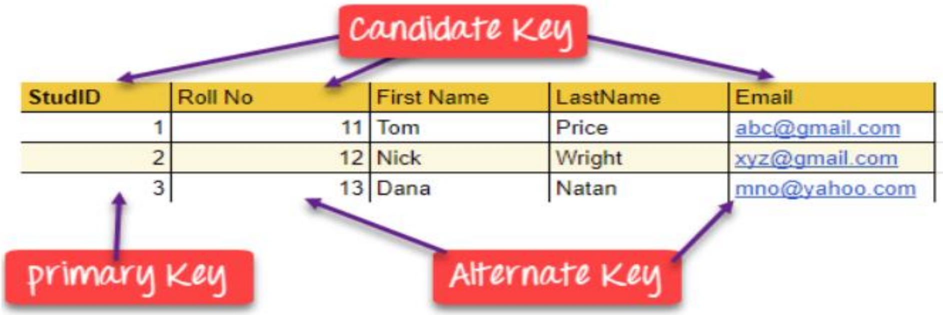
In a relation each record should be unique i.e. no two records can be identical in a database. A key attribute identifies the record and must have unique values.

Primary Key – A primary is an attribute or set of attributes in a relation that uniquely identifies tuples (rows) in that relation.

Candidate Key –It is an attribute or a set of attributes or keys participating for Primary Key, to uniquely identify each tuples in that relation.

Alternate Key – A candidate key that is not the primary key is called alternate key or secondary key.

Foreign Key – Foreign keys are the attributes of a relation that points to the primary key of another relation



StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Multiple choice Questions(MCQ):

1	DBMS stands for____________ a)Data Base Management Software b) Data Base Maintenance System c)Data Basic Management System d) Data Base management system
2	In RDBMS, R stands for________ a)Relational b) Rotational c) Rational d)None of the above
3	A Database contains one or more______ a)Data b) Tables c) Files d)Links
4	What is not true in respect of DBMS? a)Database enforces standards b)Database increases redundancy c)Database facilitates sharing of data d) Database helps to maintain integrity
5	Cardinality is total __________ a)number of rows in a table b)number of columns in a table c)number of data items in a table d) none of the above
6	Degree refers to total________ a) number of rows in a table b) number of columns in a table c) number of data items in a table d) none of the above
7	Data about data is ________ a) Data redundancy b) Meta Data b) Database schema d) None of the above
8	Repetition of data is called ________ a) Data redundancy b) Data Description c) Data inconsistency d) None of the above
9	Mismatched redundant copies of data is known as data __________ a)Dependence b) Inconsistency c) Isolation d) Redundancy
10	A ______________is an organized collection of structured data. a)Database b) File c) DBMS d) Information
11	A data _____________ is a set of rules that define valid data. a)Query b) Constraint c) Dictionary d) All of the above
12	A relational database consists of a collection of _____________ a)Fields b) Records c) Keys d) Tables
13	A row in a database is called _____________ a)Fields b) Records c) Keys d) Tables
14	The term ___________ is used to refer to a field in a table. a)Attribute b) Row c) Tuple d) Instance
15	Which of the following statements is not true about relational database? a) Relational data model is the most widely used data model. b) The data is arranged as a collection of tables in relational database. c) Relational database increases data redundancy and inconsistency. d) None of the above.

ANSWERS:

1	d	2	a	3	b	4	b
5	a	6	b	7	b	8	a
9	b	10	a	11	b	12	d
13	b	14	a	15	c		

Very Short Answer Questions

1	What is meant by a database?
2	Define primary key?Give an example.
3	What do you mean by candidate key?
4	What is meant by degree and cardinality of a table?
5	What is meant by DBMS?
6	What is meant by database schema?
7	What is meant by data constraint?
8	What is meant by relation?

Very Short Answer Questions: Answers

1	A database is an organized collection of structured information, or inter-related data, typically stored in a computer system.
2	A primary key is a column or set of columns that contain values that uniquely identify each row in a table. For example Rno can be primary key of the table student. Table:Student RNO | NAME | MARK 100 | Tanay | 30 101 | Kiran | 50 102 | Manu | 30
3	It is an attribute or a set of attributes capable of being the Primary Key, to uniquely identify each record in that table.
4	Degree refers to the number of attributes/columns in a relation. Cardinality refers to the number of tuples/rows in a relation.
5	RDBMS (Relational Database Management System) is the software used to store, manage, query, and retrieve data stored in a relational database.
6	Database schema is also called the visual or logical architecture as it tells us how the data are organized in a database.
7	Restrictions or limitations on the type of data that can be inserted in one or more columns of a table to ensure accuracy and reliability of data in the database.
8	A relation is a named, two dimensional table storing logically related data.

STRUCTURED QUERY LANGUAGE(SQL)

SQL(Structured Query Language) is a language that is used to manage data that is held in a relational database management system. It uses tables to manipulate and retrieve information from databases for analysis.

By using SQL commands, one can search for any data in the database and perform other functions like creating tables, adding records, modifying data, removing rows, dropping tables etc.

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert, update and delete records in a database
- SQL can create new databases,new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database

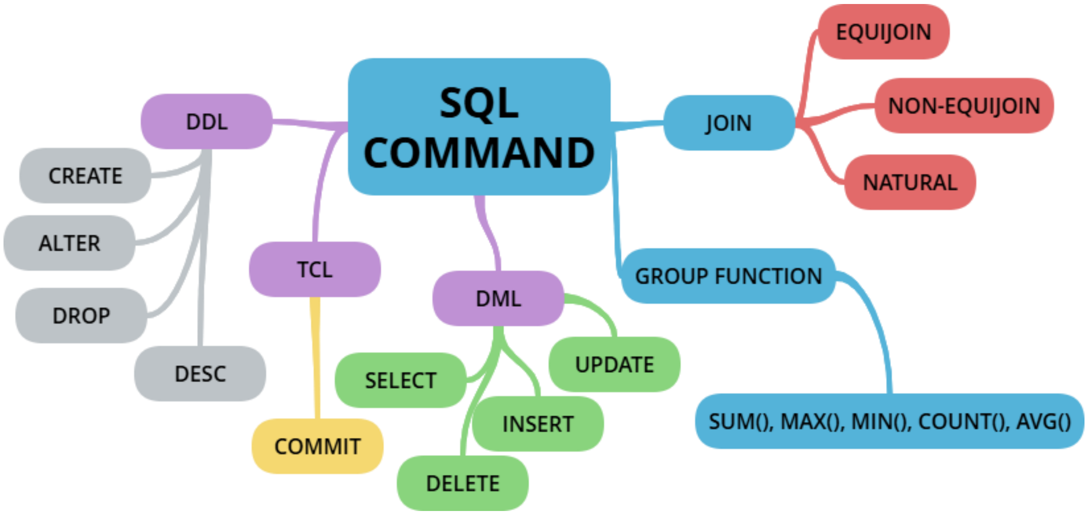
- SQL can set permissions on tables, procedures, and views

SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

- There are five types of SQL commands: DDL, DML, DCL, TCL



DDL or Data Definition Language

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. DDL is a set of SQL commands used to create, modify, and delete database structures but not data.

List of DDL commands:

CREATE: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).

DROP: This command is used to delete objects from the database.

ALTER: This is used to alter the structure of the database.

DML (Data Manipulation Language):

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

List of DML commands:

INSERT : It is used to insert data into a table.

UPDATE: It is used to update existing data within a table.

DELETE : It is used to delete records from a database table.

Difference between DDL and DML:

DDL	DML
It stands for Data Definition Language.	It stands for Data Manipulation Language.

It is used to create database schema and can be used to define some constraints as well.	It is used to add, retrieve or update the data.
It basically defines the column (Attributes) of the table.	It add or update the row of the table. These rows are called as tuple.

DATATYPES

- **Text Data types**
  Char(size) – fixed length of *size* bytes
  Varchar(size)-variable length of *size* bytes
  Varchar2(size)-variable length of *size* bytes
- **Number Data types**
  Integer(size)or Int- It represents a number without decimal point
  Float(Size)-It represents a floating point number
  Real-Same as float but no size argument is used
- **Date data type**
  Date , Time

CONSTRAINTS

A Constraint is a condition or check applicable on a field or set of fields.

Types of Constraints:

- **Unique Constraint** :-This ensures that no rows have the same value in the specified column(s)
  Example

  ```
  CREATE TABLE EMP (ecode integer   unique,
  ename   char(20),sex  char(1),
  grade   char(2));
  ```

  Unique constraint applied on ecode of EMP table ensures that no rows have the same ecode value **.**
- **Primary key Constraint**:-
  This declares a column as the primary **key** of the table. This is similar to **unique** constraint except that one column (or one group of columns) can be applied in this constraint .
  The primary key cannot allow NULL values but Unique key allows NULL values.
  The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created:

  ```
  CREATE TABLE Persons
  (ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int  PRIMARY KEY (ID));
  ```

- **Not null**: -This constraint ensures column should not be **NULL**
  Example:

  ```
  CREATE TABLE EMP(
  ecode integer   Not  null  unique,
  ename   char(20),
  sex  char(1),
  grade   char(2));
  ```

DATABASE COMMANDS IN MYSQL

- **CREATE DATABASE**
  CREATE DATABASE is the SQL command used for creating a database in MySQL.
  Imagine you need to create a database with name “movies”. You can create a database in MySQL by executing following SQL command
  Syntax: mysql>CREATE DATABASE movies;
- **SHOW DATABASES**
  You can see list of existing databases by running following SQL command.
  Syntax: mysql>SHOW DATABASES;
- **USE**
  You can use SQL command **USE** to select a particular database.
  Syntax: mysql>USE database_name;
- **DROP DATABASE**
  The DROP DATABASE statement is used to drop an existing SQL database.
  Syntax: mysql>DROP DATABASE database_name*;*

CREATE TABLE

The CREATE TABLE statement is used to create a new table in a database.

Syntax:

CREATE TABLE *table_name* (*column1 datatype*, *column2 datatype*, *column3 datatype*, ...);

Example: The following example creates a table called "Persons" that contains five columns: PersonID, LastName, FirstName, Address, and City:

```
CREATE TABLE Persons (
   PersonID int,
   LastName varchar(255),
   FirstName varchar(255),
   Address varchar(255),
   City varchar(255)        );
```

SHOW TABLES

We can get the number of table information of a database using the following statement:

```
mysql> SHOW TABLES;
```

DESCRIBE TABLE

Use the **DESCRIBE** command to show the structure of the table, such as column names, constraints on column names, etc. The **DESC** command is a short form of the DESCRIBE command. Both DESCRIBE and DESC commands are equivalent.

Syntax The following are the syntax to display the table structure:

mysql> DESCRIBE | **DESC** table_name;

ALTER TABLE

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

ALTER TABLE - ADD Column/Attribute

- **ALTER TABLE - ADD A COLUMN**

ALTER TABLE *table_name* ADD *column_name datatype*;

Example **ALTER TABLE Customers**
ADD Email varchar(255);

- **ALTER TABLE - DROP COLUMN**

To delete a column in a table, use the following syntax

ALTER TABLE *table_name* DROP COLUMN *column_name*;

Example ALTER TABLE Customers DROP COLUMN Email;

- **ALTER TABLE- ADD PRIMARY KEY**
  To create a PRIMARY KEY constraint on the "ID" column when the table is already created, use the following SQL:
  ALTER TABLE table_name ADD PRIMARY KEY (Column_name);
  Example : ALTER TABLE Persons ADD PRIMARY KEY (ID);

- **ALTER TABLE-DROP PRIMARY KEY**

To drop a PRIMARY KEY constraint, use the following SQL:

ALTER TABLE table_name DROP PRIMARY KEY;

Example : **ALTER TABLE Persons DROP PRIMARY KEY;**

DROP TABLE

The DROP TABLE statement is used to drop an existing table in a database.

Syntax **DROP TABLE *table_name*;**

DROP TABLE Shippers;

INSERT:

The INSERT INTO statement is used to insert new records in a table.

INSERT INTO Syntax:

It is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:
INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...) VALUES (*value1*, *value2*, *value3*, ...);

2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the INSERT INTO syntax would be as follows:
INSERT INTO *table_name* VALUES (*value1*, *value2*, *value3*, ...);

DELETE:

The DELETE statement is used to delete existing records in a table.

DELETE Syntax:

DELETE FROM *table_name* WHERE *condition*;

Note: Be careful when deleting records in a table! Notice the WHERE clause in the DELETE statement. The WHERE clause specifies which record(s) should be deleted. If you omit the WHERE clause, all records in the table will be deleted!

The following SQL statement deletes all rows in the "Customers" table, without deleting the table:

DELETE FROM Customers;

UPDATE

The UPDATE statement is used to modify the existing records in a table.

UPDATE Syntax: UPDATE *table_name* SET *column1* = *value1*, *column2* = *value2*, ...
WHERE *condition*;

UPDATE Table

The following SQL statement updates the first customer (CustomerID = 1) with a new contact person *and* a new city.

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```

SELECT

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

SELECT Syntax: SELECT *column1*, *column2, ...* FROM *table_name*;

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

```
SELECT * FROM table_name;
```

WHERE Clause:

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

WHERE Syntax: SELECT *column1*, *column2, ...*FROM *table_name* WHERE *condition*;

Operators in The WHERE Clause

The following operators can be used in the WHERE clause:

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. Note: In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.
- The NOT operator displays a record if the condition(s) is NOT TRUE.

AND Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

OR Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT Syntax

```
SELECT column1, column2, ...
```

FROM *table_name*
WHERE NOT *condition*;

IN Operator

The IN operator allows you to specify multiple values in a WHERE clause.
The IN operator is a shorthand for multiple OR conditions.
IN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

BETWEEN Operator

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.
The BETWEEN operator is inclusive: begin and end values are included.
BETWEEN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

LIKE Operator

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
There are two wildcards often used in conjunction with the LIKE operator:

- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (_) represents one, single character

The percent sign and the underscore can also be used in combinations!

LIKE Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name.
Aliases are often used to make column names more readable.
An alias only exists for the duration of that query.
An alias is created with the AS keyword.

Alias Column Syntax: SELECT *column_name* AS *alias_name* FROM *table_name;*

Alias Table

Syntax: SELECT *column_name(s)* FROM *table_name* AS *alias_name;*

DISTINCT

The SELECT DISTINCT statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SELECT DISTINCT Syntax:

SELECT DISTINCT *column1*, *column2, ...*FROM *table_name*;

SELECT Example Without DISTINCT

The following SQL statement selects all (including the duplicates) values from the "Country" column in the "Customers" table:

Eg: **SELECT Country FROM Customers;**

Now, let us use the **SELECT DISTINCT statement** and see the result.

SELECT DISTINCT Examples

The following SQL statement selects only the DISTINCT values from the "Country" column in the "Customers" table:

SELECT DISTINCT Country FROM Customers;

The following SQL statement lists the number of different (distinct) customer countries:

SELECT COUNT(DISTINCT Country) FROM Customers;

NULL value

A field with a NULL value is a field with no value.

If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value. It is not possible to test for NULL values with comparison operators, such as =, <, or <>.

We will have to use the IS NULL and IS NOT NULL operators instead.

IS NULL Syntax

SELECT *column_names* FROM *table_name* WHERE *column_name* IS NULL;

IS NOT NULL Syntax

SELECT *column_names* FROM *table_name* WHERE *column_name* IS NOT NULL;

The IS NULL Operator

The IS NULL operator is used to test for empty values (NULL values).

The following SQL lists all customers with a NULL value in the "Address" field:

SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NULL;

The IS NOT NULL Operator

The IS NOT NULL operator is used to test for non-empty values (NOT NULL values).

The following SQL lists all customers with a value in the "Address" field:

SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NOT NULL;

ORDER BY

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

ORDER BY Syntax

SELECT *column1*, *column2, ...*
FROM *table_name*

ORDER BY *column1, column2, ...* ASC|DESC;

Example

SELECT * FROM Customers

ORDER BY Country;

ORDER BY DESC Example

The following SQL statement selects all customers from the "Customers" table, sorted DESCENDING by the "Country" column:

SELECT * FROM Customers

ORDER BY Country DESC;

ORDER BY Several Columns Example

The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column. This means that it orders by Country, but if some rows have the same Country, it orders them by CustomerName:

Eg: SELECT * FROM Customers ORDER BY Country, CustomerName;

AGGREGATE (GROUP) FUNCTIONS

- Aggregate functions are the functions that operate on a set of rows to give one result per group.
- These sets of rows on which group function is applied may be the whole table or the table split into groups.

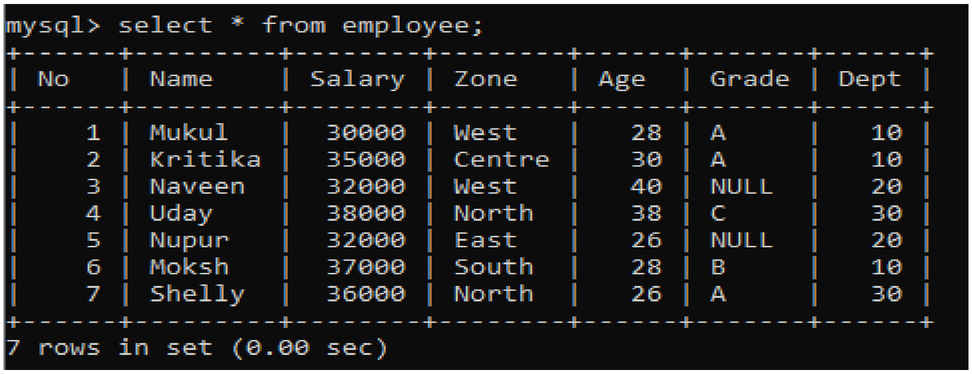
Types of Group Functions

Function	**Description**
sum( )	Find the sum of numeric values
avg( )	Find the average of numeric values
count( )	Counts the number of rows in a table
max( )	Find the maximum among all the values in a column
min( )	Find the minimum among all the values in a column

Remember the following points about group functions:

- All group functions, except count(*) ignore NULL values
- Functions -sum( ),avg( ) are used with NUMERIC data.
- Functions -min( ) and max( ) can be used with any data type.

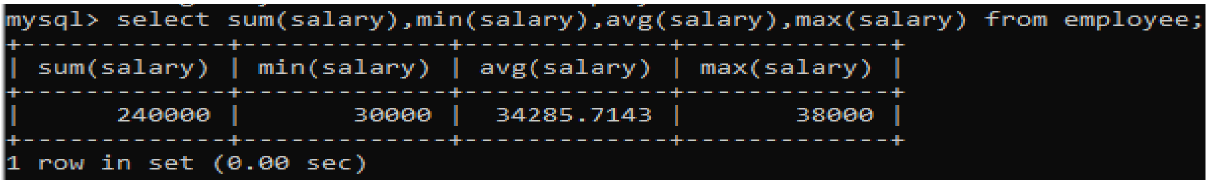
Example: Consider the following table employee



```
mysql> select * from employee;
+------+---------+--------+--------+------+-------+------+
| No   | Name    | Salary | Zone   | Age  | Grade | Dept |
+------+---------+--------+--------+------+-------+------+
|    1 | Mukul   |  30000 | West   |   28 | A     |   10 |
|    2 | Kritika |  35000 | Centre |   30 | A     |   10 |
|    3 | Naveen  |  32000 | West   |   40 | NULL  |   20 |
|    4 | Uday    |  38000 | North  |   38 | C     |   30 |
|    5 | Nupur   |  32000 | East   |   26 | NULL  |   20 |
|    6 | Moksh   |  37000 | South  |   28 | B     |   10 |
|    7 | Shelly  |  36000 | North  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.00 sec)
```

sum( ), average( ), minimum( ), maximum( ) function

Q: Find the sum, average, minimum, maximum value of salaries of employees in the employee table



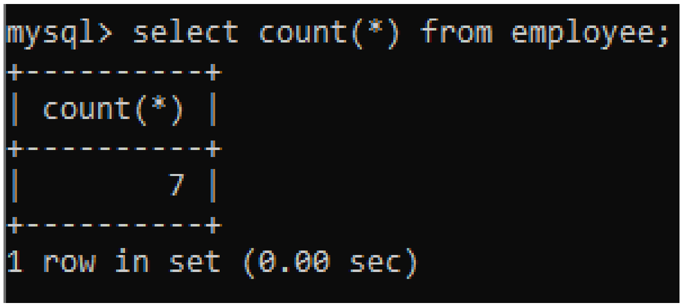
```
mysql> select sum(salary),min(salary),avg(salary),max(salary) from employee;
+-------------+-------------+-------------+-------------+
| sum(salary) | min(salary) | avg(salary) | max(salary) |
+-------------+-------------+-------------+-------------+
|      240000 |       30000 |  34285.7143 |       38000 |
+-------------+-------------+-------------+-------------+
1 row in set (0.00 sec)
```

count( ) function

Count ( ) has got three formats:

count(*)

This function returns the number of rows in the table that satisfy the criteria of select statement.
In its counting, it includes duplicate rows and rows with NULL values in any of the column
Example:
Q: Count the number of employees in the employee table.

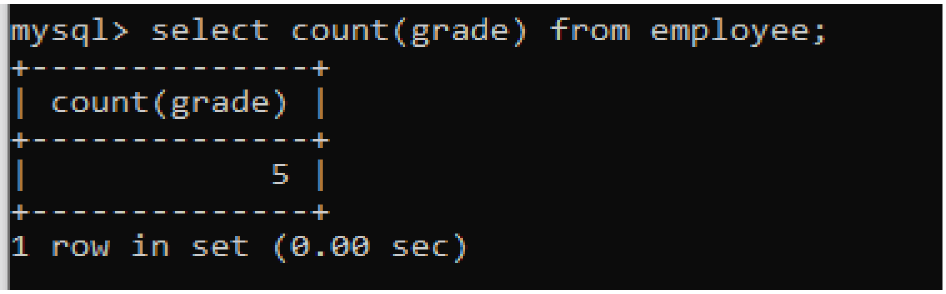


```
mysql> select count(*) from employee;
+----------+
| count(*) |
+----------+
|        7 |
+----------+
1 row in set (0.00 sec)
```

count(<col name>)

This function returns the number of not null values in the specified column, but includes duplicate values in counting
Example

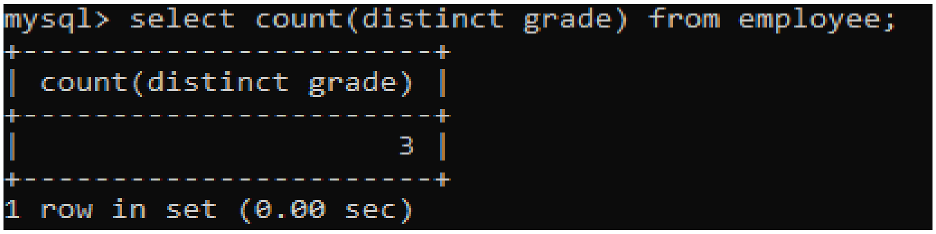
Q: Count the number of grades of employees in the employee table.



```
mysql> select count(grade) from employee;
+--------------+
| count(grade) |
+--------------+
|            5 |
+--------------+
1 row in set (0.00 sec)
```

count(DISTINCT <col name>)

This function returns the number of unique, not null values in the specified column.
Example
Q: Count the number of different grades of the employee



```
mysql> select count(distinct grade) from employee;
+-----------------------+
| count(distinct grade) |
+-----------------------+
|                     3 |
+-----------------------+
1 row in set (0.00 sec)
```

Grouping Records (Group by clause)

- To divide the rows in a table into smaller groups of information, group by clause is used.
- It combines all identical rows in a group of fields.
- A column name is used for grouping

Syntax: -

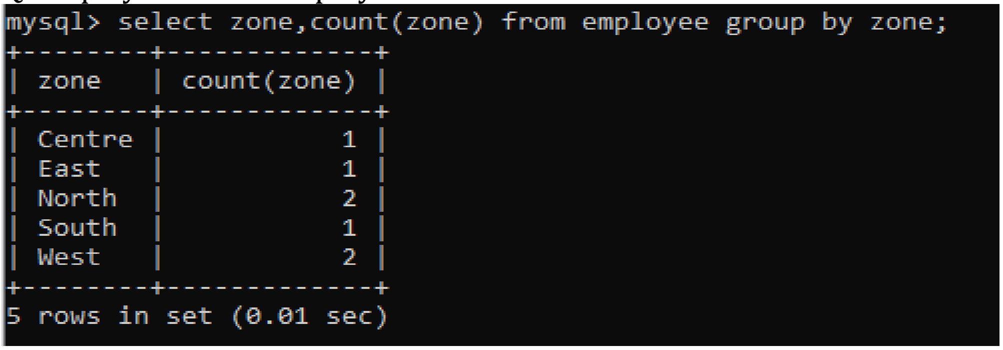
SELECT [DISTINCT] <COL LIST> FROM <TABLE NAME>
[WHERE <CONDITION>]
[GROUP BY < GROUP BY EXPR>]
[HAVING <CONDITION>]
ORDER BY <COL NAME>/<EXPR> ASC/DESC];

NOTE -

- Group by expression specifies the columns whose values determine the basics for grouping rows
- WHERE clause is always before GROUP BY if required.

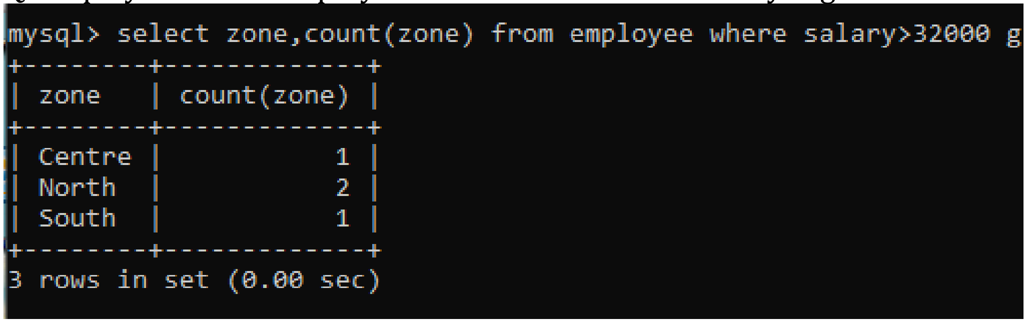
Example

Q. Display the no of employees in each zone.



```
mysql> select zone,count(zone) from employee group by zone;
+--------+-------------+
| zone   | count(zone) |
+--------+-------------+
| Centre |           1 |
| East   |           1 |
| North  |           2 |
| South  |           1 |
| West   |           2 |
+--------+-------------+
5 rows in set (0.01 sec)
```

Q. Display the no of employees in each zone whose salary is greater than 32000



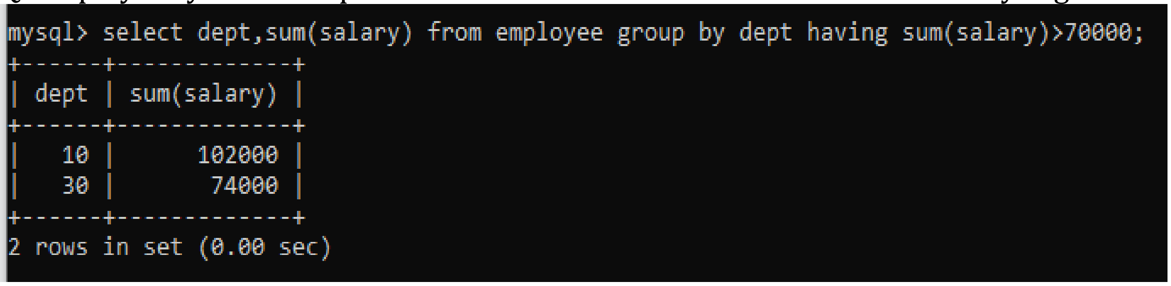
```
mysql> select zone,count(zone) from employee where salary>32000 g
+--------+-------------+
| zone   | count(zone) |
+--------+-------------+
| Centre |           1 |
| North  |           2 |
| South  |           1 |
+--------+-------------+
3 rows in set (0.00 sec)
```

Having clause

- This clause is used to restrict rows resulting after grouping.
- Steps followed in execution of select with group by and having clause-
    1. Rows are grouped according to the columns in the group by clause.
    2. Then the group function is applied.
    3. Groups matching with having clauses are displayed.

Example

Q. Display only whose departments with sum of salaries whose total salary is greater than 70000.

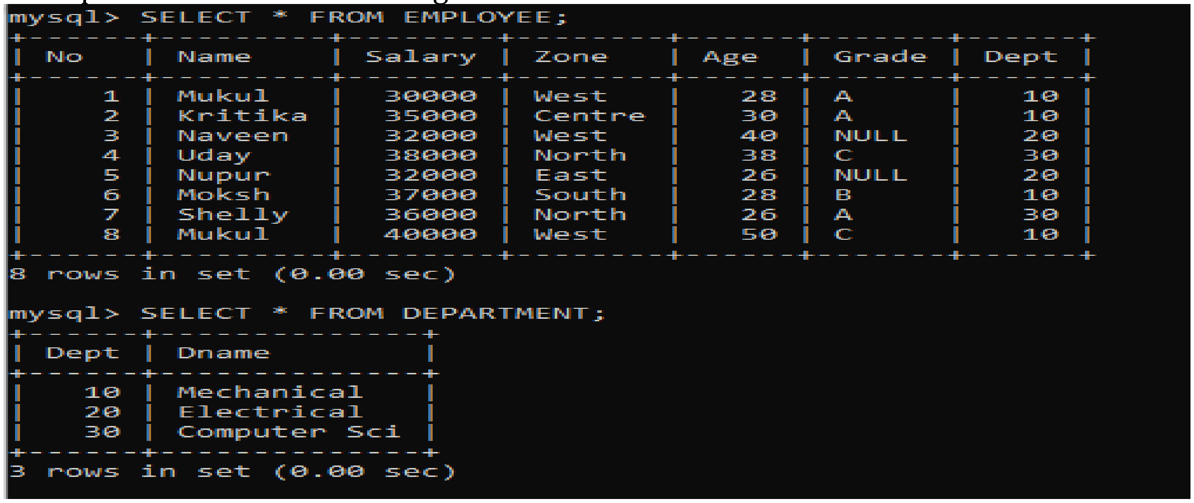


```
mysql> select dept,sum(salary) from employee group by dept having sum(salary)>70000;
+------+-------------+
| dept | sum(salary) |
+------+-------------+
|   10 |      102000 |
|   30 |       74000 |
+------+-------------+
2 rows in set (0.00 sec)
```

Cartesian Product (Cross Join or Unrestricted Join)

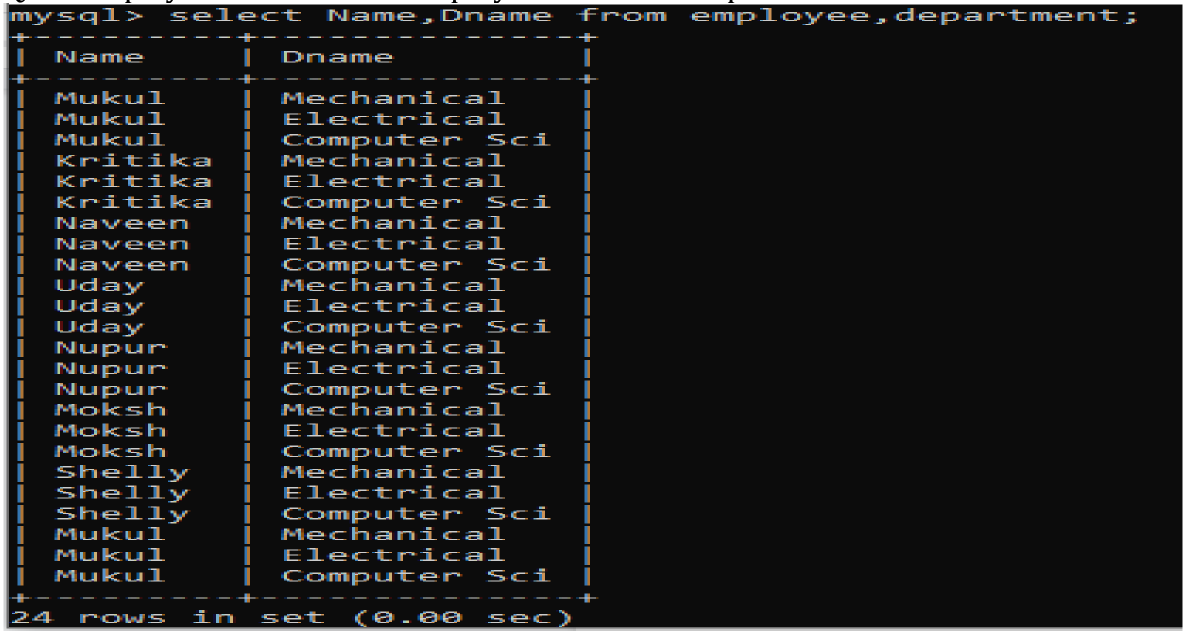
- Returns all the rows in the two tables listed in the query.
- Each row of the first table is paired with all the rows in the second table.
- This happens when there is no relationship between two tables.

Example- Consider the following tables



```
mysql> SELECT * FROM EMPLOYEE;
+------+----------+--------+--------+------+-------+------+
| No   | Name     | Salary | Zone   | Age  | Grade | Dept |
+------+----------+--------+--------+------+-------+------+
|    1 | Mukul    |  30000 | West   |   28 | A     |   10 |
|    2 | Kritika  |  35000 | Centre |   30 | A     |   10 |
|    3 | Naveen   |  32000 | West   |   40 | NULL  |   20 |
|    4 | Uday     |  38000 | North  |   38 | C     |   30 |
|    5 | Nupur    |  32000 | East   |   26 | NULL  |   20 |
|    6 | Moksh    |  37000 | South  |   28 | B     |   10 |
|    7 | Shelly   |  36000 | North  |   26 | A     |   30 |
|    8 | Mukul    |  40000 | West   |   50 | C     |   10 |
+------+----------+--------+--------+------+-------+------+
8 rows in set (0.00 sec)

mysql> SELECT * FROM DEPARTMENT;
+------+--------------+
| Dept | Dname        |
+------+--------------+
|   10 | Mechanical   |
|   20 | Electrical   |
|   30 | Computer Sci |
+------+--------------+
3 rows in set (0.00 sec)
```

Q: To display the name of the employees and their department name.



```
mysql> select Name,Dname from employee,department;
+---------+--------------+
| Name    | Dname        |
+---------+--------------+
| Mukul   | Mechanical   |
| Mukul   | Electrical   |
| Mukul   | Computer Sci |
| Kritika | Mechanical   |
| Kritika | Electrical   |
| Kritika | Computer Sci |
| Naveen  | Mechanical   |
| Naveen  | Electrical   |
| Naveen  | Computer Sci |
| Uday    | Mechanical   |
| Uday    | Electrical   |
| Uday    | Computer Sci |
| Nupur   | Mechanical   |
| Nupur   | Electrical   |
| Nupur   | Computer Sci |
| Moksh   | Mechanical   |
| Moksh   | Electrical   |
| Moksh   | Computer Sci |
| Shelly  | Mechanical   |
| Shelly  | Electrical   |
| Shelly  | Computer Sci |
| Mukul   | Mechanical   |
| Mukul   | Electrical   |
| Mukul   | Computer Sci |
+---------+--------------+
24 rows in set (0.00 sec)
```

JOINS IN MYSQL

- A join is used when data from two or more tables is required.
- Rows in one table can be joined to the rows in another table based on the common values existing in corresponding columns of two tables.
- Joins are used to retrieve data from tables related to each other with primary- foreign key relationships.
- There are many types of joins:

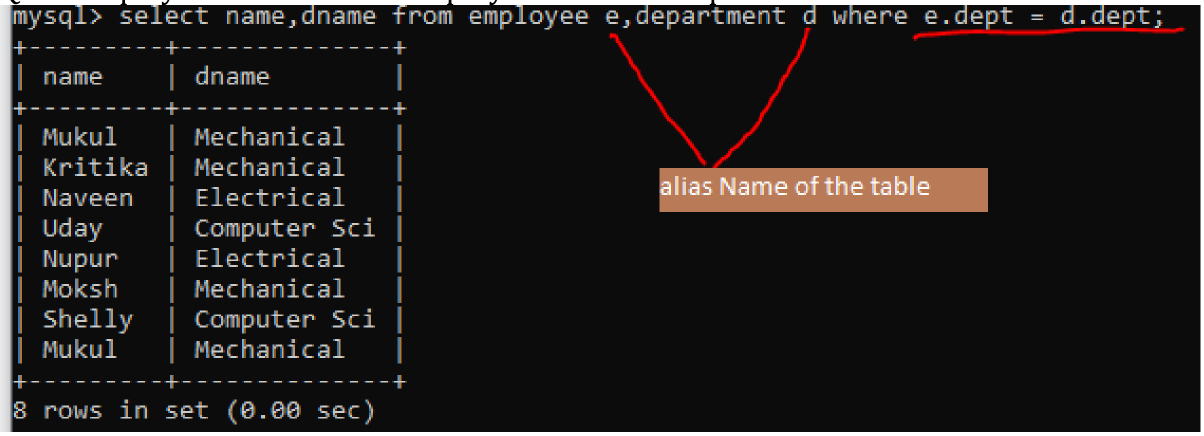
EQUI JOIN

- Specified columns from the joining tables are checked for equality.
- Values from joining tables are retrieved only if the condition in where clause is satisfied.

SYNTAX:-

```
SELECT <column_name (s)>
FROM <table_name1>, <table_name2>, ...., <table_nameN>
WHERE <table_name1>.<column_name> = <table_name2>.<column_name>;
```

Q: To display the name of the employee and their department



```
mysql> select name,dname from employee e,department d where e.dept = d.dept;
+---------+--------------+
| name    | dname        |
+---------+--------------+
| Mukul   | Mechanical   |
| Kritika | Mechanical   |
| Naveen  | Electrical   |
| Uday    | Computer Sci |
| Nupur   | Electrical   |
| Moksh   | Mechanical   |
| Shelly  | Computer Sci |
| Mukul   | Mechanical   |
+---------+--------------+
8 rows in set (0.00 sec)
```

Note-You should always qualify the columns when joining tables having the same name as corresponding columns. To qualify the columns we use "." (dot) operator.

Natural Join

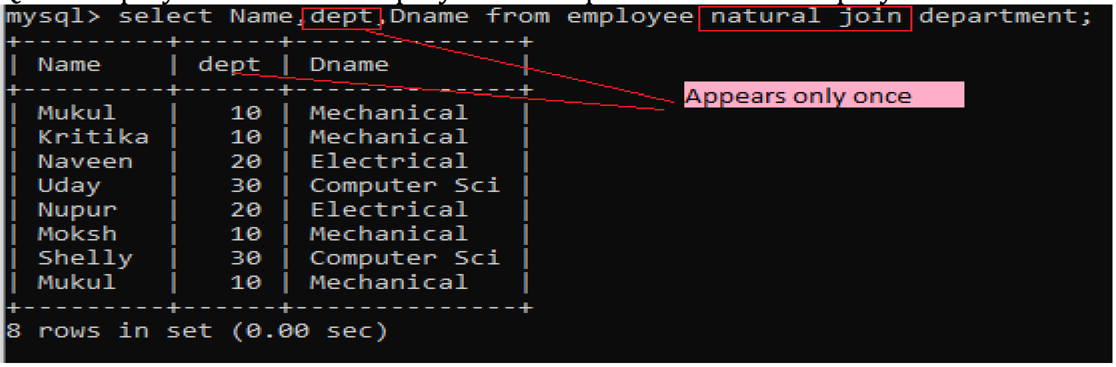
This clause is based on all the columns in the two tables that have the same name. It selects the rows from two tables that have equal values in the matched columns.

SYNTAX:-

```
SELECT [column_names / *] FROM table_name1   NATURAL JOIN table_name2;
```

Example- consider the same tables employee and department.

Q: To display the name of employee and department of all employee.



```
mysql> select Name,dept,Dname from employee natural join department;
+---------+------+--------------+
| Name    | dept | Dname        |
+---------+------+--------------+
| Mukul   |   10 | Mechanical   |
| Kritika |   10 | Mechanical   |
| Naveen  |   20 | Electrical   |
| Uday    |   30 | Computer Sci |
| Nupur   |   20 | Electrical   |
| Moksh   |   10 | Mechanical   |
| Shelly  |   30 | Computer Sci |
| Mukul   |   10 | Mechanical   |
+---------+------+--------------+
8 rows in set (0.00 sec)
```

Note-No need to specify the column names to join. Works with same column name in both the tables. The Resulting table has unique columns.

MULTIPLE CHOICE QUESTIONS

1	Which of the following SQL commands is used to use/select a particular database? a. use b. select c. view d. project
2	Which SQL command is used to define and maintain physical structure or schema of table in database like creating, altering and deleting database object such as table and constraints? a. DDL b. DML c. DCL d. TCL
3	Which commands is used to show all table in current using database? a. display tables; b. show tables; c. view tables; d. select all tables;
4	Identify the MySQL Commands that belongs to DML category : a. ALTER b. DROP c. DELETE d. CREATE
5	Which command is used in where clause to search NULL values in a particular column? a. IS NULL b. IN NULL c. NOT NULL d. IS NOT NULL
6	Wild card operator (%,_) are used with? a. count b. max c. like d. min

7	Prapti is presently working in the database SUBJECT. She wants to change and go to the database RECORD. Choose the correct statement in MySQL to go to the database RECORD.
	a. GO TO DATABASE RECORD; b. USE DATABASE RECORD;
	c. CHANGE DATABASE RECORD; d. USE RECORD;

8	Which SQL clause is used in database table to eliminate duplicate rows from the query result? a. group by b. distinct c. describe d. duplicate
9	Which SQL function is used to count the entire number of row in database table? a. count b. count(*) c. max d. min
10	Which SQL function is used to determine the no. of row or non-null values? a. min b. max c. count d. sum

ANSWERS

1	a	2	a	3	b	4	c	5	a
6	c	7	b	8	b	9	b	10	c

ASSERTION AND REASONING QUESTIONS

Directions: In the following questions, A statement of Assertion (A) is followed by a statement of Reason (R). Mark the correct choice as:

(A) Both A and R are true and R is the correct explanation of A
(B) Both A and R are true and R is not the correct explanation of A
(C) A is true but R is false
(D) A is false but R is true

1	Assertion(A): The resultset refers to a logical set of records that are fetched from the database executing an SQL Query. Reason (R): Resultset stored in a cursor object can be extracted by using fetch(…) functions.
2	Assertion(A): In SQL, aggregate function avg( )calculates the average value on a set of values and produce a single result. Reason (R): The aggregate functions are used to perform some fundamental arithmetic tasks such as min( ), max( ), sum( ) etc.

3	Assertion(A): Primary key is a set of one or more attributes that identify tuples in a relation. Reason (R): The primary key constraint is a combination of the NOT NULL and UNIQUE constraints.
4	Assertion(A): Foreign key is a non-key attribute whose value is derived from primary key of another table. Reason (R): Each foreign key refers a candidate key in a relation.
5	Assertion(A): The SELECT statement in SQL is used to retrieve data from one or more tables. Reason(R): The SELECT statement can be used to retrieve all columns or a subset of columns from a table.

ANSWERS

1	b	2	B	3	a	4	b	5	a

SHORT ANSWER TYPE QUESTIONS

1	Deepika wants to remove all rows from the table BANK. But she needs to keep the structure of the table. Which command is used to implement the same?
2	While creating table 'customer', Rahul forgot to add column 'price'. Which command is used to add new column in the table. Write the command to implement the same.
3	Mitali is a database programmer, She has to write the query from EMPLOYEE table to search for the employee who are not getting any commission, for this she has written the query as: SELECT * FROM EMPLOYEE WHERE commission=null; But the query is not producing the correct output, help her and correct the query so that she gets the desired output.
4	Which clause is used to eliminate the duplicate rows from output?
5	Which command is used to see information like name of columns, data type, size.
6	Differentiate between order by and group by clause in SQL with appropriate example.
7	Categorize the following commands as DDL or DML: INSERT, UPDATE, ALTER, DROP.
8	Muneer has created a database "school" and table "student". Now he wants to view all the databases present in his laptop. Help him to write SQL command for that , also to view the structure of the table he created
9	Ms. Minakshi has just created a table named "Staff" containing Columns Sname, Department and Salary. After creating the table, she realized that she has forgotten to add a primary key in the table. Help her in writing an SQL command to add a primary key - StaffId of integer type to the table Staff.Thereafter, write the command to insert the following record in the table: StaffId – 111,Sname- Shalu,Department: Marketing,Salary: 45000
10	Meera working as database developer in Syntel Pvt Ltd Company Agra. She is designing as SQL table names A & B .If a MySQL table A has 5 columns and 6 rows and another table B has 3 columns and 4 rows, then what will be the degree and cardinality of the cartesian product of A and B?
11	Sunil decides to delete phoneno column from a table student . Write the SQL command to remove the column in the student table. Also mention the type of SQL command.
12	Write SQL command to remove the Primary Key constraint from a table, named M_ID is the primary key of the table MOBILE.
13	Write SQL command to make the column M_ID the Primary Key of an already existing table, named MOBILE.
14	What constraint should be applied on a table column so that duplicate values are not allowed in that column, but NULL is allowed.

ANSWERS:

1	delete from BANK;
2	ALTER TABLE customer add price int;
3	SELECT * FROM EMPLOYEE WHERE commission is null;
4	Distinct
5	DESCRIBE or DESC
6	ORDER BY is used to sort the result set based on one or more columns, either in ascending (ASC) or descending (DESC) order whereas group by is used to group rows that have the same values in specified columns into summary rows, often used with aggregate functions. Eg: SELECT name, salary FROM employees ORDER BY salary DESC; This will list all employees and their salaries, sorted by salary from highest to lowest. SELECT department, AVG(salary) AS average_salary FROM employees GROUP BY department; This will group employees by department and show the average salary for each department.
7	DDL –ALTER , DROP DML-INSERT , UPDATE

8	SHOW DATABASES;
	SHOW TABLES;

9	ALTER TABLE Staff ADD StaffId INT;
	ALTER TABLE Staff ADD CONSTRAINT pk_staff PRIMARY KEY (StaffId);
	INSERT INTO Staff (StaffId, Sname, Department, Salary) VALUES (111, 'Shalu', 'Marketing', 45000);

10	Degree = 8 Cardinality = 24
11	ALTER TABLE student DROP phoneno It's a DDL command.
12	ALTER TABLE mobile DROP PRIMARY KEY;
13	ALTER TABLE mobile ADD PRIMARY KEY (M_ID);
14	UNIQUE constraint makes sure that duplicate values are not allowed in that column, but NULL will be allowed.

SHORT ANSWER QUESTIONS II

1 Meenu has been entrusted with the management of NSE Database. She needs to access some information from STOCK and TRADERS tables for a survey analysis.
Help him extract the following information by writing the desired SQL queries as mentioned below.

Table Name: TRADERS

TCODE	TNAME	CITY
T01	RELIANCE DIGITAL	MUMBAI
T02	TATA DIGITAL	BHUBANESWAR
T03	BIRLA DIGITAL	NEW DELHI

Table name: STOCK

SCODE	SNAME	QTY	PRICE	BRAND	TCODE
1001	COMPUTER	90	45000	DELL	T01
1006	LCD PROJECTOR	40	42000	NEC	T02
1004	IPAD	100	55000	APPLE	T01
1003	DIGITAL CAMERA	160	15000	SAMSUNG	T02

1005	LAPTOP	600	35000	HP	T03

Write SQL queries for the following:

(i) Display the SNAME, QTY, PRICE, TCODE, and TNAME of all the stocks in the STOCK and TRADERS tables.

(ii) Display the details of all the stocks with a price >= 35000 and <=50000 (inclusive).

(iii) Display the SCODE, SNAME, QTY*PRICE as the “TOTAL PRICE” of BRAND “NEC” or “HP” in ascending order of QTY*PRICE.

iv) Display the number of stock items in each TCODE.

OR

To display the Cartesian Product of these two tables

2 Consider the following tables employees, empsalary.

Table : Employees

Empid	Firstname	Lastname	Address	City
010	Ravi	Kumar	Raj nagar	GZB
105	Harry	Waltor	Gandhi nagar	GZB
152	Sam	Tones	33 Elm St.	Paris
215	Sarah	Ackerman	440 U.S. 110	Upton
244	Manila	Sengupta	24 Friends street	New Delhi
300	Robert	Samuel	9 Fifth Cross	Washington
335	Ritu	Tondon	Shastri Nagar	GZB
400	Rachel	Lee	121 Harrison St.	New York
441	Peter	Thompson	11 Red Road	Paris

Table: EmpSalary

Empid	Salary	Benefits	Designation
010	75000	15000	Manager
105	65000	15000	Manager
152	80000	25000	Director
215	75000	12500	Manager
244	50000	12000	Clerk
300	45000	10000	Clerk
335	40000	10000	Clerk
400	32000	7500	Salesman
441	28000	7500	salesman

Write the SQL commands for the following:

a) To show first name, last name, address and city of all employees who lives in Paris.

b) To display the details of Employees table in descending order of First name.

c) To display the first name, last name and salary of all employees from the tables Employee and EmpSalary, who are working as Manager.

Give the Output of following SQL commands:

d) Select designation, sum(salary) from empsalary group by designation having count(*) > 2;

OR

d) Select sum(benefits) from empsalary where designation =’clerk’;

3 **Consider the table PRODUCT and CLIENT given below:**

PR_ID	PR_NAME	MANUFACTURER	PRICE	QTY
BS101	BATH SOAP	PEARSE	45.00	25

SP210	SHAMPOO	SUN SILK	320.00	10
SP235	SHAMPOO	DOVE	455.00	15
BS120	BATH SOAP	SANTOOR	36.00	10
TB310	TOOTH BRUSH	COLGATE	48.00	15
FW422	FACE WASH	DETOL	66.00	10
BS145	BATH SOAP	DOVE	38.00	20

C_ID	C_NAME	CITY	PR_ID
01	DREAM MART	COCHIN	BS101
02	SHOPRIX	DELHI	TB310
03	BIG BAZAR	DELHI	SP235
04	LIVE LIFE	CHENNAI	FW422

Write SQL Queries for the following:

i) Display the details of those clients whose city is DELHI

ii) Increase the Price of all Bath soap by 10

iii) Display the details of Products having the highest price

iv) Display the product name, price, client name and city with their corresponding matching product Id.

4 **Write SQL Commands for (a) to (e) and write the outputs for (f) on the basis of table: FURNITURE**

NO	ITEM NAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT
1	White Lotus	Double Bed	2002-02-23	3000	25
2	Pink feathers	Baby Cot	2002-01-29	7000	20
3	Dolphin	Baby Cot	2002-02-19	9500	20
4	Decent	Office Table	2002-02-01	25000	30
5	Comfort zone	Double Bed	2002-02-12	25000	30
6	Donald	Baby cot	2002-02-24	6500	15
7	Royal Finish	Office Table	2002-02-20	18000	30
8	Royal tiger	Sofa	2002-02-22	31000	30
9	Econo sitting	Sofa	2001-12-13	9500	25
10	Eating Paradise	Dinning Table	2002-12-19	11500	25

1. To show all the information about the Baby cots from the furniture table.

2. To list the itemname which are priced at more than 15000 from the furniture table.

3. To list itemname and type of those items, in which dateofstock is after 2002-02-01 from the furniture table in descending order of itemname

4. To display itemname and dateofstock of those items, in which the discount percentage is more than 25 from the furniture table.

ANSWERS:

1

```
i)select sname, qty, price , stock.tcode, tname
from stock, traders where  stock.tcode = traders.tcode;
ii) select * from stock  where  price between 35000 and 50000;
iii) select scode, sname, qty * price as "total price"  from stock
     where brand in ('nec', 'hp')  order by qty * price asc;
iv) select tcode, count(*) as  stock_count from stock  group by tcode;
```

2

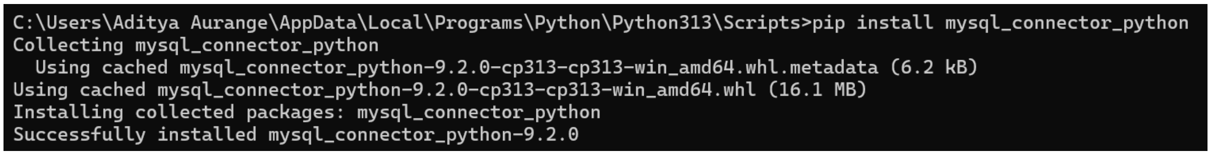
```
i) select firstname, lastname, address, city from employees   where city = 'paris';
```

	ii) select * from employees order by firstname desc; iii) select e. firstname, e. lastname, s. salary from employees e join empsalary s on e.empid = s.empid where s.designation = 'manager'; iv) manager 215000 clerk 135000 or 32000
3	A) **D_NAME** GUPTA HANEEF B) **D_DEPT** ENT MEDICINE ORTHO CARDIOLOGY SKIN C) **D_NAME** **EXPERIENCE** DEEPTI 6 SUMAN 7 JOSEPH 10 GUPTA 12 HANEEF 12 VEENA 12
4	i) select * from client where city = 'delhi'; ii) update product set price = price + 10 where pr_name = 'bath soap'; iii) select * from product where price = (select max(price) from product); iv) select p.pr_name, p.price, c.c_name, c.city from product p join client c on p.pr_id = c.pr_id;
5	i) select * from furniture where lower(type) = 'baby cot'; ii) select item_name from furniture where price > 15000; iii) select item_name, type from furniture where dateofstock > '2002-02-01' order by item_name desc; iv) select item_name, dateofstock from furniture where discount > 25;

INTERFACE PYTHON WITH MySQL DATABASE

1. Installation of mysql.connector module

a) USE COMMAND : pip install mysql_connector_python



```
C:\Users\Aditya Aurange\AppData\Local\Programs\Python\Python313\Scripts>pip install mysql_connector_python
Collecting mysql_connector_python
  Using cached mysql_connector_python-9.2.0-cp313-cp313-win_amd64.whl.metadata (6.2 kB)
Using cached mysql_connector_python-9.2.0-cp313-cp313-win_amd64.whl (16.1 MB)
Installing collected packages: mysql_connector_python
Successfully installed mysql_connector_python-9.2.0
```

b) After Installation, check by connecting to MYSQL.

```
import mysql.connector as m
con = m.connect(host='localhost',user='root', passwd='1234', database='class')
```

print(con.is_connected( ))

>>> True (OUTPUT)

The output came True, meaning successful connection with MYSQL has been established

2. Steps involved in MySQL-Python connectivity

1. Open python editor
2. Import the package required (import mysql.connector)
3. Open CONNECTION to the database
4. Create a CURSOR instance
5. Execute the QUERY via CURSOR object
6. Extract data from CURSOR object
7. Clean up the environment

3. Importing mysql.connector in python

- import mysql.connector

Or

- import mysql.connector as m

Note: "m" is an alias, which can be used in place of "mysql.connector" whenever a function of module "mysql.connector" is required. We can use any valid IDENTIFIER as an alias.

4. Open a connection to MySQL Database

To create connection, connect( ) function is used

Its syntax is:

import mysql.connector as m

con = connect (host= H , user= U, passwd= P , database= D)

- host 'H' means where the MySQL database is hosted, generally it is given as "localhost"
- user 'U' means user by which we connect with mysql generally it is given as "root"
- passwd 'P' is the password of MySQL for user "root"
- database 'D' is the name of database whose data(table) we want to use

After forming connection, check by using is_connected( )

is_connected( ) : This function returns True if connection has been formed.

import mysql.connector as m

con = m.connect(host='localhost',user='root', passwd='1234', database='class')

print(con.is_connected( ))

>>> True (OUTPUT)

The output came True, meaning successful connection with MYSQL has been established.

5. Creating the CURSOR Object :

- cursor object in MySQL-Python connectivity acts as a middleman between your Python code and the MySQL database.
- It's used to execute SQL queries and fetch results from the database.
- Cursor object can be created using Connection object.

import mysql.connector as m

con = m.connect(host='localhost',user='root', passwd='1234', database='class')

```
cur = con.cursor( )
```

Here 'cur' is the cursor object created on 'con' connection object. What the Cursor Object Does:

a) Execute SQL Commands.
b) Fetch Result from Database.

6. Executing Queries using execute( ) function with cursor object.

- The execute( ) function accepts the QUERY in the form of a STRING.
- The RESULT of the query is received in the cursor object.

```
import mysql.connector as m

con= m.connect(host='localhost', user='root', passwd='1234', database='class')

cur = con.cursor( )
query = "SELECT * FROM users"
cur.execute(query)
```

7. Fetching the Result

- Fetching/Extracting the Data/Result stored inside CURSOR object after Executing the QUERY.
- For this we have 3 functions which are applied on the CURSOR object 'cur'.
  a) fetchall( ): It will return list of all the records retrieved in tuple form.
  b) fetchone( ): It will return one record from the result set.
  c) fetchmany(n): It will return number of records as per value of n and by default only one records in tuple form.

The SYNTAX for extracting data –

```
import mysql.connector as m
con= m.connect(host='localhost', user='root', passwd='1234', database='class')
cur = con.cursor( )
query = "SELECT * FROM users"
cur.execute(query)
data = cur.fetchall( )
        or
data = cur.fetchone( )
        or
data = cur.fetchmany(n)

for row in data:
        print(row)  # Displaying Each Record of the TABLE.
```

As ROWS of the table are received inside a TUPLE, we can display each ROW by iterating over data/result extracted from the cursor in python.

8. ROWCOUNT

- It is the property of the cursor object that returns the number of rows fetched from the cursor object till that moment and not the number of records the executed query will give.

SYNTAX –

```
count = cur.rowcount
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system",   database="student")

cur=con.cursor( )               # ASSUME TABLE HAS 10 ROWS
```

```
cur.execute("select * from student") # All 10 ROWS are now in CURSOR object
data = cur.fetchone( )        # Only 1 ROW is fetched from CURSOR till now
print(cur.rowcount)           # Output = 1
data = cur.fetchmany(4)       # Total 5 ROWS have been fetched from CURSOR
print(cur.rowcount)           # Output = 5
data = cur.fetchone( )        # Total 6 ROWS have been fetched from CURSOR
print(cur.rowcount)           # Output = 6
data = cur.fetchall( )        # Remaining 4 ROWS have been fetched as well
print(cur.rowcount)           # Output = 10
```

9. **commit( )**

   After executing insert or update query we must commit our transaction using commit method of connection object.

   Eg: con.commit( )

10. **con.close( )**

    Closing the connection, Since the database can keep limited number of connections at a time, we must close the connection using **connection_object**.close( ).

    Eg: con.close( )

STATIC QUERIES

Queries which are formed without passing any python object/variable/data.

Now we can execute any MySQL query through Python. Below are few examples static queries.

a) TO CREATE A TABLE IN MYSQL USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("CREATE TABLE FEES (ROLLNO INT, NAME VARCHAR(20), AMOUNT INT);")
```

b) TO SHOW THE TABLES IN MYSQL USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("SHOW TABLES")
for data in cur:
    print(data)
```

c) TO DESCRIBE TABLE STRUCTURE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
mycursor.execute("DESC STUDENT")
for data in cur:
print(data)
```

d) TO EXECUTE SELECT QUERY USING A PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("select * from student")
```

```
r=cur.fetchone( )
while r is not None:
    print(r)
    r=cur.fetchone( )
```

e) TO EXECUTE SELECT QUERY WITH WHERE CLAUSE USING A PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("select * from student where marks>90")
r=cur.fetchall( )
count=cur.rowcount
print("total no of rows:",count)
for row in r:
    print(row)
```

f) TO UPDATE A DATA IN A TABLE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("UPDATE STUDENT SET MARKS=100 WHERE MARKS=40")
con.commit( )
print(cur.rowcount,"RECORD UPDATED")
```

g) TO DELETE A RECORD FROM THE TABLE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("DELETE FROM STUDENT WHERE MARKS<50")
con.commit( )
print(cur.rowcount,"RECORD DELETED")
```

h) TO DROP AN ENTIRE TABLE FROM MYSQL DATABASE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("DROP TABLE STUDENT")
```

i) TO ADD A COLUMN IN THE EXISTING TABLE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("ALTER TABLE STUDENT ADD AGE NT”)
con.commit( )
```

j) TO DROP A COLUMN FROM THE TABLE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("ALTER TABLE DROP AGE ”)
con.commit( )
```

k) TO ALTER THE DATATYPE OF A COLUMN IN A TABLE USING PYTHON INTERFACE

```
import mysql.connector as m
```

```
con =m.connect(host="localhost",user="root", passwd="system", database="student")
cur=con.cursor( )
cur.execute("ALTER TABLE STUDENT MODIFY GRADE CHAR(3)")
```

PARAMETERIZED QUERIES

Parameterize the query to add python variables/object into the string query to access values as per the user's input.

Example: to display record of a particular rollno.

There are two ways to use parameterized queries:

a) with { }.format pattern
b) with fstring & { } braces

1) Using {}.format pattern

(a)

```
rn = int(input('Enter Roll no. ')) # SUPPOSE USER ENTERS 10 AS ROLL NO
query  = "select * from student where rollno  = { }".format(rn)
cur.execute(query)
```

Note: "Here the format function will assign the value 10 in place of { } braces inside the string query. python will convert it to-> query = "select * from student where rollno = 10"

(b)

Suppose we want to display the data based on the column name & value given by user

```
col_name = eval(input('Enter Column Name ')) # User entered Rollno
col_value = eval(input('Enter Column Value ')) # User entered 12
query  = "select * from student where { }  = { }".format(col_name , col_value)
# python will convert it to ->>> query  = "select * from student where Rollno = 12 "
cur.execute(query)
```

(c)

Suppose we want to display the data from a specific class & section given by user

```
cls = eval(input('Enter Class '))            # User entered 12
sec = eval(input('Enter Section '))          # User entered A
query  =   "select * from student
              where class = { } and section  = '{ }' ".format(cls , sec)
# python will convert it to ->>> query  = "select * from student
                                          where class = 12 and section  = 'A' "
cur.execute(query)
```

2) Using fstring & { } braces

(a)

```
rn = int(input('Enter Roll no. ')) # SUPPOSE USER ENTERS 10 AS ROLL NO
query  = f"select * from student where rollno  = {rn}"
cur.execute(query)
```

Note: Here we add 'f' before the string & write the variable inside { } braces which assigns the value 10 in place of {rn} inside the string query. python will convert it to-> query= "select * from student where rollno = 10"

(b)

Suppose we want to display the data based on the column name & value given by user

```
col_name = eval(input(‘Enter Column Name ‘)) # User entered Rollno
col_value = eval(input(‘Enter Column Value ‘)) # User entered 12
query  = f’select * from student where { col_name } = { col_value } “
cur.execute(query)
```

Note: python will convert it to ->>> query = "select * from student where Rollno = 12 "

(c)

Suppose we want to display the data from a specific class & section given by user

```
cls = eval(input(‘Enter Class ‘))           # User entered 12
sec = eval(input(‘Enter Section ‘))        # User entered A
query  = f ”select * from student where class = {cls} and section = ‘{sec}’ “
cur.execute(query)
```

Note: python will convert it to ->>> query = "select * from student where class = 12 and section = ‘A’ "

Multiple Choice Questions

1	Which of the following command is used to connect Python with MySQL? a) mysql.connect( ) b) MySQL.connector.connect( ) c) mysql.connector.connect( ) d) connect.mysql( )
2	Which module needs to be imported to use MySQL in Python? a) mysql b) sqlite3 c) MySQL d) mysql.connector
3	What does the cursor( ) method do in MySQL Python connector? a) Executes the SQL commands directly b) Establishes a connection to the database c) Creates a cursor object to interact with the database d) Creates a database
4	Which method is used to execute SQL queries in Python? a) executeQuery( ) b) run( ) c) execute( ) d) query( )
5	Which method is used to retrieve all rows from the executed query? a) fetchone( ) b) fetchall( ) c) getrows( ) d) read( )
6	After performing an INSERT operation, which method is used to save the changes in the database? a) commit( ) b) save( ) c) flush( ) d) update( )
7	Which of the following is not a valid parameter in mysql.connector.connect( )? a) host b) username c) user d) password
8	What is the purpose of the close( ) method? a) To close the database file b) To shut down the MySQL server c) To close the cursor or connection d) To delete the database
9	Which of the following is correct for selecting a database after connection? a) cursor.select("mydb") b) connection.database("mydb") c) cursor.execute("USE mydb") d) connect.use("mydb")
10	What is the output type of fetchall( )? a) List of strings b) Tuple of strings c) List of tuples d) Dictionary

Answers

1	**C**	**2**	**D**	**3**	**C**	**4**	**C**	**5**	**B**
6	**A**	**7**	**B**	**8**	**C**	**9**	**C**	**10**	**C**

Assertion Reasoning Questions

Each question contains Assertion (A) and Reason (R). Choose the correct option from:
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) A is false but R is true.

1	Assertion (A): The cursor( ) method is used to execute SQL queries in Python. Reason (R): The cursor object allows interaction with the database.
2	Assertion (A): cursor.fetchall( ) is used to delete all records from a table. Reason (R): fetchall( ) retrieves all records returned by a SELECT query.
3	Assertion (A): mysql.connector.connect( ) requires parameters like host, user, password, and database to connect successfully. Reason (R): mysql.connector.connect( ) is used to establish a connection between Python and MySQL.
4	Assertion (A): commit( ) is used after every SELECT query to save the fetched results. Reason (R): commit( ) saves changes made by INSERT, UPDATE, and DELETE operations.
5	Assertion (A): The close( ) method is mandatory after the database operations are completed. Reason (R): Keeping database connections open unnecessarily may lead to resource leakage.

Answers

1	**A**	**2**	**D**	**3**	**A**	**4**	**D**	**6**	**A**

VERY SHORT ANSWER QUESTIONS

1	Which command is use to install MySQL library in python?
2	Which method we use to establish the connection?
3	Which statement we use to access the MySQL module?
4	What is the Database Connector?
5	Which function is used to check the successful connection?

Answers

1	pip install MySQL. Connector
2	connect( ) method with connection object.
3	import mysql.connector
4	A database connector is a software that connects an application to any database.
5	.is_connected( ) method

SHORT ANSWER QUESTIONS

1	What are the steps for creating database connectivity applications?
2	What is a connection? What is its role?
3	What is a resultset?

4	What is a database cursor?
5	How to retrieve data from a table?

Answers

1	To create database connectivity, follow the given steps: Step 1: Start Python Step 2: Import mysql.connector Step 3: Open a connection to the database Step 4: Create a cursor instance Step 5: Execute a query Step 6: Extract data from result set Step 7. Clean up the environment
2	A connection (represented by the connection object) is the session between the application program and database. To do anything with database, one must have a connection object.
3	A result set refers to a logical set of records that are fetched from the database by executing a query and made available to the application program.
4	A database cursor is a special control structure that facilitates row by row processing of records in the result set, i.e., the set of records retrieved as per the query.
5	There are multiple ways to retrieve data: i. fetchall( ) – fetches all the remaining rows of a query result, current pointer position forwards ii. fetchone( ) – fetches the next row as a sequence; returns None when no more data iii. fetchmany(n) :It will return number of records as per value of n and by-default only one record.

LONG ANSWER QUESTIONS

1	Write a Python code to connect to a database
2	How to create a database in MySQL through Python ?
3	Write the Python code to display the present databases in MySQL
4	How to create a table in MySQL through Python ?
5	Write the Python code to insert data into student table of database kvs .
6	How to fetch data in Python from a student table of education database?
7	Write the Python code to update a record in the student table of education database.
8	Write the Python code to delete a record from the student table of education database

9	Mr.Harsh want to interface python with mysql and write some code help him to write the code
	import____________.connector as m #Line1
	mydb=m.________(host=”localhost”,user=”root”,
	passwd=”tiger”,database=” choo ”) # Line2
	cursor=mydb.__________( ) #Line3
	cursor.______________(“select * from student”) #Line4
	data=cursor._________( ) #Line5 To retrieved all records
	count=cursor.__________ #Line6

10	What is the difference between fetchone( ), fetchmany( ), fetchall( )?

Answers

1	import mysql.connector as m
	Mycon=m.connect(host=”localhost”,user=”root”, password=”tiger”,database=”project”)
	print(mycon)

2	import mysql.connector as m
	mycon= m.connect(host=”localhost”,user=”root”,password=”tiger”)
	cursor=mycon.cursor( )
	cursor.execute(“create database education”)
	mycon.commit( )

3	import mysql.connector as m
	mycon= m.connect(host=”localhost”,user=”root”,password=”tiger”)
	cursor=mycon.cursor( )
	cursor.execute(“show databases”)
	for i in cursor:
	      print(i)

4	import mysql.connector as m
	mycon= m.connect(host=”localhost”,user=”root”, password=”tiger”)
	cursor=mycon.cursor( )
	query = “create table student(admn_no int primary key, sname varchar(30), gender char(2), DOB date, stream varchar(10), mark float”
	cursor.execute(query)
	mycon.commit( )

5	import mysql.connector as m
	mycon= m.connect(host=”localhost”,user=”root”,password=”tiger”)
	cursor=mycon.cursor( )
	no=int(input(“Enter admission no “))
	n=input(“Enter name “)
	g=input(“Enter gender ”)
	dob=input(“Enter DOB “)
	st=input(“Enter stream “)
	m=float(input(“Enter m rk ”))
	query= “insert into student values({}, ‘{}’ , ‘{}’ , ’{}’ , ’{}’ , {})”.format(no,n,g,dob, st,m)
	cursor.execute(query)
	mycon.commit( )

6	import mysql.connector as m
	mycon= m.connect(host=”localhost”,user=”root”,password=”tiger”,
	                                   database = ‘education’)
	cursor=mycon.cursor( )
	cursor.execute(“select * from student”)
	for row in cursor:
	            print(row)

7	import mysql.connector as m
	mycon= m.connect(host=”localhost”,user=”root”,password=”tiger”, database = ‘education’)
	cursor=mycon.cursor( )
	cursor.execute(“update student set marks =67 where admn_no=306”)
	mycon.commit( )

8	import mysql.connector as m
	mycon= m.connect(host=”localhost”,user=”root”,password=”tiger”, database = ‘education’)
	cursor=mycon.cursor( )
	cursor.execute(“delete from student where admn_no=308 ”)
	mycon.commit( )

9	Line1:-mysql, Line2:-connect, Line3:cursor, Line4: execute, Line5: fetchall, Line6: rowcount

10	**fetchone( )**
	• **Returns:** A single row from the result set as a tuple (or None if no more rows).
	fetchmany(size)
	• **Returns:** A list of **up to size rows**, each as a tuple.
	fetchall( )
	• **Returns:** A list of **all remaining rows** as tuples in the result set.

LONG ANSWER QUESTIONS

1	A table, named STATIONERY, in ITEMDB database, has the following structure: Field Type itemNo int(11) itemName varchar(15) price float qty int(11) Assume the following for Python-Database connectivity: Host: localhost, User: root, Password: Pencil Write the following Python function to perform the specified operation: **Add( ):** To input details of an item and store it in the table STATIONERY. **Display( ):** The function should then retrieve and display all records from the STATIONERY table where the Price is greater than 120.
2	Sartaj has created a table named CLASS in MYSQL database, CLASS: rno(Roll number)- integer name(Name) - string DOB (Date of birth) – Date Fee – float Note the following to establish connectivity between Python and MySQL: Username – root, Password – tiger, Host – localhost, DB – SC Sartaj, now wants to display the records of students whose fee is more than 5000. Help Sartaj to write the program in Python.
3	Kabir wants to write a program in Python to insert the following record in the table named CLASS in MYSQL database SC, CLASS: rno(Roll number)- integer name (Name) - string DOB (Date of birth) – Date Fee – float Note the following to establish connectivity between Python and MySQL: Username – root, Password – tiger, Host – localhost, DB – SC The values of fields rno, name, DOB and fee has to be accepted from the user. Help Kabir to write the program in Python.
4	Rehaan wants to write a program in Python to insert the following record in the table named EMP in MYSQL database COMPANY: a. eno(Empno)- integer b. ename(Name) - string c. DOB (Date of birth) – Date d. Salary – float

Note the following to establish connectivity between Python and MySQL:
Username – root, Password – password, Host - localhost
The values of fields eno, name, DOB and salary has to be accepted from the user. Help Rehaan to write the program in Python.

Answers:

1

```
import mysql.connector as m
mycon= m.connect(host=”localhost”,user=”root”,password=”Pencil”)
mycursor=mycon.cursor( )
def Add(mycursor):
            mycursor.execute(‘use ITEMDB’)
            n = int(input(‘Enter no. of Items to be entered: ’))
            for i in range(n):
                ino = int(input(‘Enter Item no: ’))
                inm = input(‘Enter Item name: ’)
                ipr = float(input(‘Enter Item price: ’))
                iq = int(input(‘Enter Item quantity: ’))
                query = f ”insert into STATIONARY values({ino},’{inm}’,{ipr},{iq})”
                mycursor.execute(query)
                    mycon.commit( )
def Display(mycursor):
            mycursor.execute(‘use ITEMDB’)
            query = ‘Select * from STATIONARY where price > 120’
            mycursor.execute(query)
            data = mycursor.fetchall( )
            for record in data:
                       print(‘Item No = ‘ , record[0])
                       print(‘Item Name = ‘ , record[1])
                       print(‘Item Price = ‘ , record[2])
                       print(‘Item Qty = ‘ , record[3])
 Add(mycursor)
 Display(mycursor)
 mycon.close( )
```

2

```
def Display(mycursor):
           query = ‘Select * from CLASS where Fee > 5000’
           mycursor.execute(query)
           data = mycursor.fetchall( )
           for record in data:
                      print(‘Roll No = ‘ , record[0])
                      print(‘Name = ‘ , record[1])
                      print(‘DOB = ‘ , record[2])
                      print(‘Fee = ‘ , record[3])
import mysql.connector as m
mycon=m.connect(host=”localhost”,user=”root”,password=”tiger”, database=’SC’)
mycursor=mycon.cursor( )
Display(mycursor)
```

	mycon.close( )

3	def Add(mycursor):
	    n = int(input(‘Enter no. of Student Records to be entered: ’))
	    for i in range(n):
	        rn = int(input(‘Enter Student Roll No: ’))
	        nm = input(‘Enter Student name: ’)
	        dob = input(‘Enter Student DOB as ‘YYYY-MM-DD’ : ’))
	        fee = float(input(‘Enter Student Fee: ’))
	        query = f ”insert into CLASS values({rn},’{nm}’,’{dob}’,{fee})”
	        mycursor.execute(query)
	        mycon.commit( )
	import mysql.connector as m
	mycon=m.connect(host=”localhost”,user=”root”,password=”tiger”, database=’SC’)
	mycursor=mycon.cursor( )
	Add(mycursor)
	mycon.close( )

4	def Add(mycursor):
	    mycursor.execute(‘use COMPANY’)
	    n = int(input(‘Enter no. of Records to be entered: ’))
	    for i in range(n):
	        en = int(input(‘Enter Student Roll No: ’))
	        enm = input(‘Enter Student name: ’)
	        dob = input(‘Enter Student DOB as ‘YYYY-MM-DD’ : ’))
	        sal = float(input(‘Enter Student Fee: ’))
	        query = f ”insert into EMP values({en},’{enm}’,’{dob}’,{sal})”
	        mycursor.execute(query)
	    mycon.commit( )
	import mysql.connector as m
	mycon=m.connect(host=”localhost”,user=”root”,password=”password”)
	mycursor=mycon.cursor( )
	Add(mycursor)
	mycon.close( )

CBSE Question Paper with Solution: 2024-25
Computer Science (083)

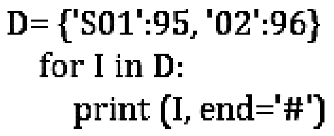
Subject: Computer Science (Theory) **Class: XII**

Time Allowed: 3:00 Hours **Max. Marks - 70**

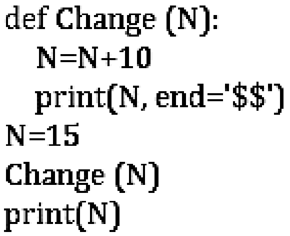
General Instructions:

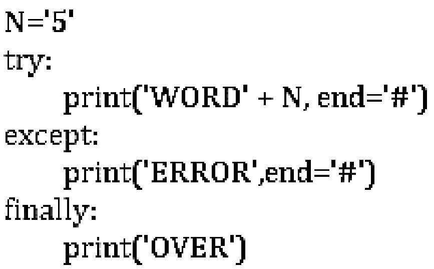
- This question paper contains 37 questions.
- All questions are **compulsory**. However, internal choices have been provided in some questic Attempt only one of the choices in such questions
- The paper is divided into **5 Sections**- A, B, C, D and E.
- Section A consists of **21** questions (**1 to 21**). Each question carries **1 Mark**.
- Section B consists of **7** questions (**22 to 28**). Each question carries **2 Marks**.
- Section C consists of **3** questions (**29 to 31**). Each question carries **3 Marks**.
- Section D consists of **4** questions (**32 to 35**). Each question carries **4 Marks**.
- Section E consists of **2** questions (**36 to 37**). Each question carries **5 Marks**.
- All programming questions are to be answered using Python Language only.
- **In case of MCQ, text of the correct answer should also be written**.

Q. No.	**SECTION-A (21 x 1 = 21 Marks)**	**Marks**
1.	State True or False: "A Python List must always contain all its elements of same data type "	**1**
Ans – 1	False **Explanation**: A Python List may Contains different kind of data values.	
2.	What will be the output of the following statement? **print(14%3**2*4)** (A) 16 (B) 64 (C) 20 (D) 256	**1**
Ans-2	Option (C) 20 **Explanation:** Expression 14 % 3 ** 2 * 4 in Python. Python follows **operator precedence**: ** (exponentiation) first,% (modulus) and * (multiplication) are next, evaluated left to right.	
3.	Identify the correct output of the following code snippet: **game="Olympic2024"** **print (game.index ("C"))** (A) 0 (B) 6 (C) -1 (D) ValueError	**1**
Ans-3	Option (D) ValueError	

	Explanation: ValueError because the character "C" (uppercase) does **not** exist in the string "Olympics2024".	
4.	Which of the following is the correct identifier? (A) global (B) Break (C) def (D) with	**1**
Ans-4	Option (B) Break ● global → Not valid (it's a Python keyword) ● Break → Valid (not a keyword — only break is) ● def → Not valid (it's a Python keyword) • with → Not valid (it's a Python keyword)	
5.	Identify the invalid Python statement out of the following options: (A) **print("A",10.end="*")** (B) **print("A",sep="*",10)** (C) **print("A",10,sep="*")** (D) **print("A"*10)**	**1**
Ans-5	Option (B) print (“A”, sep=”*”, 10) **Explanation**: Keyword arguments **sep** must come **after** all positional arguments.	
6.	consider the statements given below and then choose the correct output from the given options: **L = ['TIC', 'TAC']** **print (L[::-1])** (A) ['CIT', 'CAT'] (B) ['TIC', 'TAC'] (C) ['CAT', 'CIT'] (D) ['TAC ', 'TIC']	**1**
Ans-6	Option (D) ['TAC', 'TIC']	
7.	Which of the following operator evaluates to True if the variable on either side of the operator points towards the same memory location and False otherwise? (A) **is** (B) **is not** (C) **and** (D) **or**	**1**
Ans-7	Option (A) is **Explanation**: The **is** operator evaluates to **True** if **both variables point to the same memory location.** o a == b checks **value equality** o a is b checks **object identity**	
8.	Consider the statements given below and then choose the correct output from the given options:  D= {'S01':95, '02':96} for I in D: print (I, end='#') (A) **S01#S02#** (B) **95#96#** (C) **S01,95#502,96#** (D) **S01#95#S02#96#**	**1**

Ans-8	Option (A) S01#02# **Explanation**: In a for loop over a dictionary, **the keys are iterated** (not the values). First iteration: I = 'S01' → prints S01# Second iteration: I = '02' → prints 02#	
9.	While creating a table, which constraint does not allow insertion of duplicate values in the table? (A) **UNIQUE** (B) **DISTINCT** (C) **NOT NULL** (D) **HAVING**	**1**
Ans-9	Option (A) UNIQUE **Explanation**: The **UNIQUE** constraint ensures that all values in a column are **distinct means no duplicate value.** **DISTINCT**: It is not a constraint. **NOT NULL**: Ensures that a column cannot have NULL values, but it does not prevent duplicates. **HAVING**: It is not a constraint	

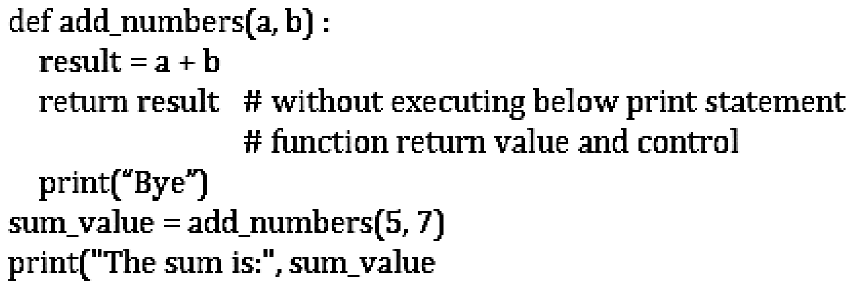
10.	Consider the statements given below and then choose the correct output from the given options:
	
	def Change (N):
	    N=N+10
	    print(N, end='$$')
	N=15
	Change (N)
	print(N)
	(A) **25$$15** (B) **15$$25** (C) **25$$25** (D) **2525$$**
		1

Ans-10	Option (A) 25$$15 **Explanation**: The print statement inside the function outputs 25$$. The value of N outside the function remains unchanged (since integers are passed by value in Python). Thus, the print (N) statement will print the original N = 15.	
11.	Consider the statements given below and then choose the correct output from the given options:  N='5' try: print('WORD' + N, end='#') except: print('ERROR',end='#') finally: print('OVER') (A) **ERROR#** (B) **WORD5#OVER** (C) **WORD5#** (D) **ERROR#OVER**	**1**

Ans-11	Option (B) WORD5#OVER **Explanation**: except block will **not** be executed because there is no error in the try block. The finally block is always gets executed, regardless of whether an exception occurred or not.	
12.	Which of the following built-in function/method returns a dictionary? **(A) dict ( ) (B) keys ( ) (C) values ( ) (D) items ( )**	**1**
Ans-12	Option (A) dict( ) **Explanation**: The **dict( )** function is used to create a new dictionary. It returns a **new empty dictionary** if no arguments are passed.	
13.	Which of the following is a DML command in SQL? (A) **UPDATE** (B) **CREATE** (C) **ALTER** (D) **DROP**	**1**
Ans-13	Option (A) UPDATE	
14.	Which aggregate function in SQL displays the number of values in the specified column ignoring the NULL values? (A) len ( ) (B) count( ) (C) number( ) (D) num( )	**1**
Asn-14	Option (B) count( ) **Explanation**: The **COUNT( )** aggregate function in SQL returns the number of **non-NULL** values in a specified column. len ( ) , number( ) and num( ) – there are not SQL functions.	
15.	In MYSQL, which type of value should not be enclosed within quotation marks.? (A) **DATE** (B) **VARCHAR** (C) **FLOAT** (D) **CHAR**	**1**
Ans-15	Option (C) FLOAT **Explanation**: Except Numeric values, all other values are enclosed in quotation.	
16.	State True or False: If table **A** has 6 rows and 3 columns, and table **B** has 5 rows and 2 columns, the Cartesian product of A and B will have 30 rows and 5 columns.	**1**
Ans-16	**True** **Explanation**: The **Cartesian product** of these tables will: **Rows = 6 × 5 = 30 rows.** **Columns = 3 + 2 = 5 columns.**	
17.	Which of the following networking devices is used to regenerate and transmit the weakened signal ahead? (A) Hub (B) Ethernet Card (C) Repeater (D) Modem	**1**
Ans-17	Option (C) Repeater	

18.	Which of the following options is the correct protocol used for phone calls over the internet? (A) PPP (B) FTP (C) HTTP (D) VoIP	**1**
Ans-18	Option (D) VoIP (Voice over Internet Protocol) **PPP (Point-to-Point Protocol)**: A data link protocol used to establish a direct connection between two nodes over a network. It is not used for voice communication. **FTP (File Transfer Protocol)**: Used for transferring files over a network. It does not handle voice communication. **HTTP (Hypertext Transfer Protocol)**: A protocol used for transferring hypertext documents on the web (like loading websites), not for voice communication.	
19.	Expand ARPANET	**1**
Ans-19	**Advanced Research Projects Agency Network**	
	Q.20 and Q.21 are **Assertion(A)** and **Reason(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	
20.	**Assertion (A):** For a binary file opened using **'rb'** mode, the **pickle.dump ( )** method will display an error. **Reason (R)** : The pickle.dump( ) method is used to read from a binary file.	**1**
Asn-20	(C) Assertion (A) is true but, Reason (R) is false. **Explanation: Assertion (A) is true**: The pickle.dump( ) method would cause an error if the file is opened in **'rb'** mode. **Reason (R) is false**: The **pickle.dump( )** method is for writing to a file, not reading from it.	
21.	**Assertion (A)** : We can retrieve records from more than one table in MYSQL. **Reason (R)** : Foreign key is used to establish a relationship between two tables.	**1**
Ans-21	A) Both Assertion (A) and Reason (R) are true and Reason (R) is the correct explanation for Assertion (A).	
	SECTION-B (7 x 2=14 Marks)	
22.	What does the return statement do in a function? Explain with the help of an example.	**2**
Ans-22	The return statement forcefully return **the control and value** along with the control without executing further statements in the function. By default, function return control and None value after executing all statements of the function.	

Example:



```
def add_numbers(a, b) :
   result = a + b
   return result   # without executing below print statement
                   # function return value and control
   print(“Bye”)
sum_value = add_numbers(5, 7)
print("The sum is:", sum_value
```

23. Write one example of each of the following in Python: **2**

(i) Syntax Error

(ii) Implicit Type Conversion

Ans-23 (i) **Syntax Error:** A **syntax error** occurs when the rules of the Python language are not followed.

```
if 5 > 2
        print("Five is greater than two")
```

Explanation: Python expects a colon: after if 5 > 2. Without it, it generates **Syntax Error**.

(ii) **Implicit Type Conversion**: **Implicit Type Conversion** happens when Python automatically converts one data type into another during an operation.

```
x = 5       # integer
y = 2.5     # float

result = x + y
print(result)
print(type(result))
```

Explanation:

- Here, x (integer) is automatically converted to a float during addition with y.

result becomes 7.5, which is a **float**.

24. Consider the following dictionaries, D and D1: **2**

D=("Suman": 40, "Raj":55, "Raman":60)

D1={"Aditi": 30, "Amit":90, "Raj":20}

(Answer using built-in Python functions only)

(i) (a) Write a statement to display/return the value corresponding to the key “Raj" in the dictionary D.

OR

(b) Write a statement to display the length of the dictionary D1.

(ii) (a) Write a statement to append all the key-value pairs of the dictionary D to the dictionary D1.

OR

(b) Write a statement to delete the item with the given key "Amit" from the dictionary D1.

Ans-24

(i) (a) print(D.get("Raj"))

OR

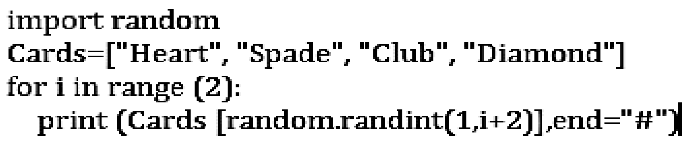
(b) print(len(D1))

(ii) (a) D1.update(D)

OR

(b) D1.pop("Amit")

25. What possible output from the given options is expected to be displayed when the following code is executed? **2**



```
import random
Cards=["Heart", "Spade", "Club", "Diamond"]
for i in range (2):
   print (Cards [random.randint(1,i+2)],end="#")
```

(A) Spade #Diamond# (B) Spade#Heart#

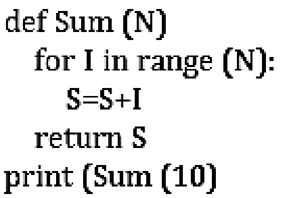
(C) Diamond#Club# (D) Heart#Spade#

Ans-25

Option (A) Spade #Diamond#

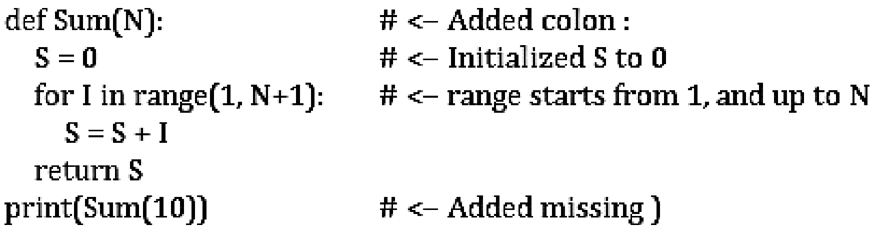
Explanation: random.randint(1,n) - Generates values from 1 to n , both values are included

26. The code given below accepts N as an integer argument and returns the sum of all integers from 1 to N. Observe the following code carefully and rewrite it after removing all syntax and logical errors. Underline all the corrections made. **2**



```
def Sum (N)
   for I in range (N):
      S=S+I
   return S
print (Sum (10)
```

Ans-26



```
def Sum(N):                  # <– Added colon :
   S = 0                     # <– Initialized S to 0
   for I in range(1, N+1):   # <– range starts from 1, and up to N
      S = S + I
   return S
print(Sum(10))               # <– Added missing )
```

27. Nisha is assigned the task of maintaining the staff data of an organization. She has to store the details of the staff in the SQL table named **EMPLOYEES** with attributes **as EMPNO, NAME, DEPARTMENT, BASICSAL** to store Employee's Identification Number, Name, Department, and Basic Salary respectively. **2**

There can be two or more Employees with the same name in the organization.

	(i) (a) Help Nisha to identify the attribute which should be designated as the PRIMARY KEY. Justify your answer. **OR** (b) Help Nisha to identify the constraint, which should be applied to the attribute **NAME** such that the Employees' Names cannot be left empty or NULL while entering the records but can have duplicate values. (ii) (a) Write the SQL command to change the size of the attribute BASICSAL in the table EMPLOYEES to allow the maximum value of 99999.99 to be stored in it. **OR** (b) Write the SQL command to delete the table EMPLOYEES.	
Ans-27	(i) a) The attribute **EMPNO** (Employee Number) should be the **PRIMARY KEY.** **Justification:** ● A **Primary Key** must be **unique** and **not NULL.** ● Employee Numbers (EMPNO) are **unique** for each employee, even if two employees have the same Name. **OR** (b) **NOT NULL** constraint (ii) (a) ALTER TABLE EMPLOYEES **MODIFY** BASICSAL DECIMAL(7,2); **OR** ALTER TABLE EMPLOYEES **CHANGE** BASICSAL BASICSAL DECIMAL(7,2); **OR** (b) DROP TABLE EMPLOYEES;	
28.	(a) Expand and explain the term URL. **OR** (b) Expand the term PPP. What is the use of PPP?	**2**
Ans-28	**(a) URL** stands for **Uniform Resource Locator**. A URL is the **address** used to access resources (like web pages, images, videos) on the **internet.** It tells the browser **where** to find the resource and **how** to retrieve it. A typical URL includes: Protocol (like http, https) , Domain name (like www.example.com) , Path to the resource (like /folder/page.html) **Example**: https://www.googel.com/research **OR** **(b) PPP** stands for **Point-to-Point Protocol**. PPP is a communication protocol used to **directly connect two computers** over a serial cable, phone line, fiber, or wireless link. It is commonly used for **establishing internet connections** over telephone lines — like in old **dial-up internet.**	

SECTION-C (3 x 3 = 9 Marks)

29.

(a) Write a Python function that displays all the lines containing the word **'vote'** from a text file **"Elections.txt"**.

For example, if the file contains :

In an election, many people vote to choose their representative.
The candidate getting the maximum share of votes stands elected.
Normally, one person has to vote once.
The process of voting may vary with time and region.

Then the output should be:

In an election, many people vote to choose their representative.
Normally, one person has to vote once.

OR

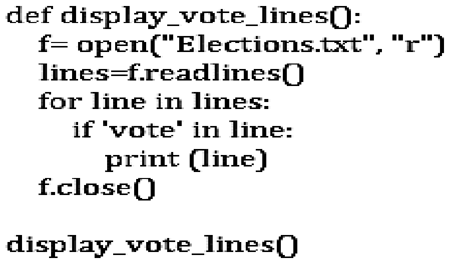
(b) Write a Python function that displays all the words starting and ending with a vowel from a text file "**Report.txt**". The consecutive words should be separated by a space in the output. For example, if the file contains:

Once there was a wise man in a village.
He was an awesome story-teller.
He was able to keep people anchored while listening to him.

Then the output should be:

Once a a awesome able

3

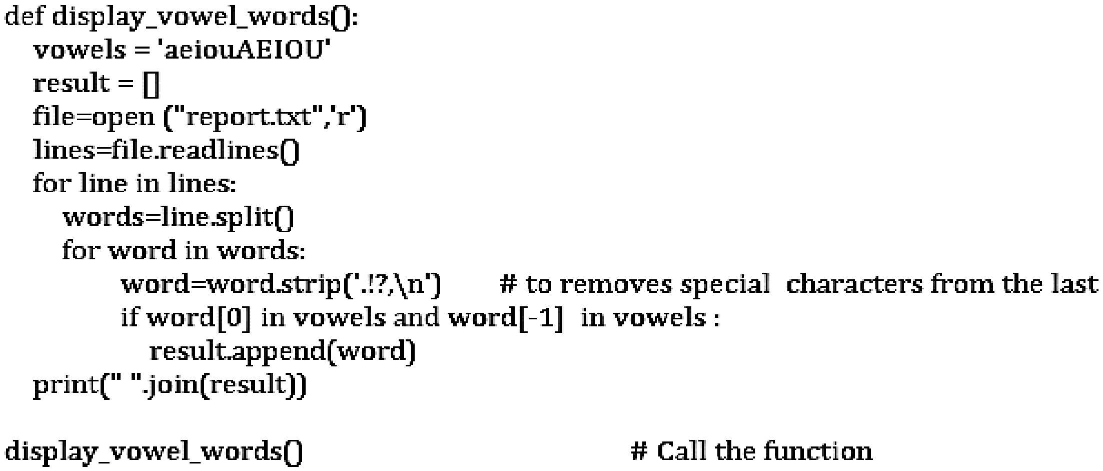


```
def display_vote_lines():
    f= open("Elections.txt", "r")
    lines=f.readlines()
    for line in lines:
        if 'vote' in line:
            print (line)
    f.close()

display_vote_lines()
```

(a)

OR

(b)



```
def display_vowel_words():
   vowels = 'aeiouAEIOU'
   result = []
   file=open ("report.txt",'r')
   lines=file.readlines()
   for line in lines:
      words=line.split()
      for word in words:
            word=word.strip('.!?,\n')       # to removes special  characters from the last
            if word[0] in vowels and word[-1]  in vowels :
               result.append(word)
   print(" ".join(result))

display_vowel_words()                                   # Call the function
```

30. (a) A stack, named **ClrStack**, contains records of some colors. Each record is represented as a tuple Containing four elements **ColorName**, **RED**, **GREEN**, **BLUE**. **ColorName** is a string, and **RED**, **GREEN**, **BLUE** are integers. **3**

For example, a record in the stack may be **('Yellow', 237, 250, 68)**

Write the following user-defined functions in Python to perform the specified operations on **ClrStack**:

(i) **push_Clr (ClrStack, new_Clr)**: This function takes the stack **ClrStack** and a new record **new_Clr** as arguments and pushes this new record onto the stack.

(ii) **pop_Clr (ClrStack)**: This function pops the topmost record from the stack and returns it. If the stack is already empty, the function should display the message "**Underflow**".

(iii) **isEmpty (ClrStack)**: This function checks whether the stack is empty. If the stack is empty, the function should return **True**, otherwise the function should return **False**.

OR

(b) Write the following user-defined functions in Python:

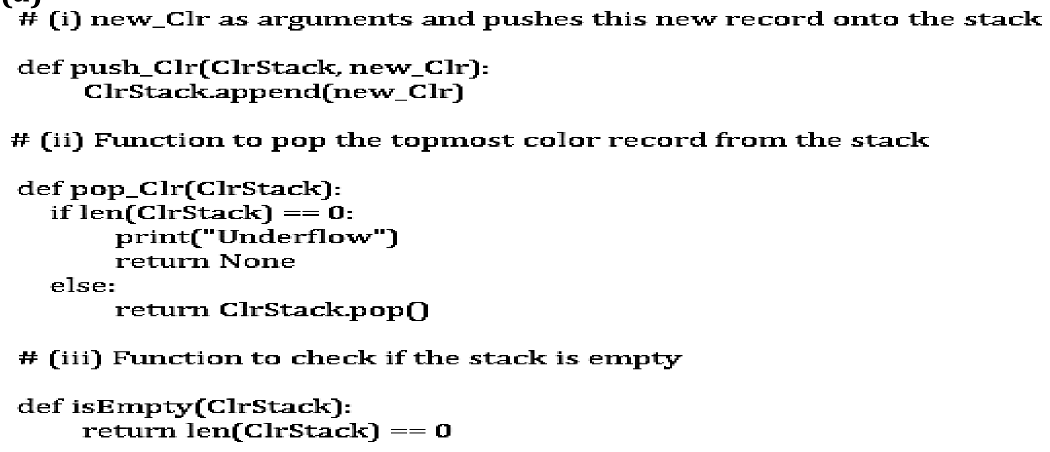
(i) **push_trail (N, myStack)**: Here **N** and **mystack** are lists, and **myStack** represents a stack. The function should push the last 5 elements from the list N onto the stack **myStack.**

For example, if the list N is [1,2,3,4,5,6,7], then the function push_trail( ) should push the elements 3,4,5,6,7 onto the stack. Therefore, the value of stack will be [3,4,5,6,7]. Assume that N contains at least 5 elements.

(ii) **pop_one (myStack)**: The function should pop an element from the stack **myStack**, and return this element. If the stack is empty, then the function should display the message '**Stack Underflow**', and return **None**.

(iii) **display_all (myStack)**: The function should display all the elements of the stack **myStack**, without deleting them. If the stack is empty, the function should display the message **'Empty Stack'**.

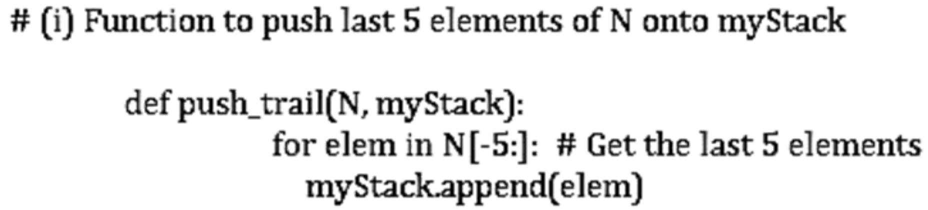
Ans-30 **(a)**



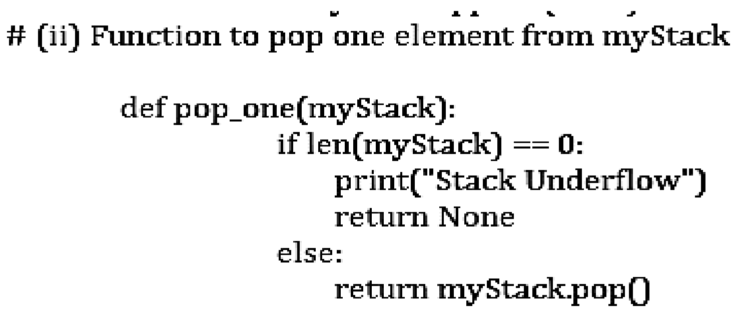
```
 # (i) new_Clr as arguments and pushes this new record onto the stack

 def push_Clr(ClrStack, new_Clr):
       ClrStack.append(new_Clr)

# (ii) Function to pop the topmost color record from the stack

 def pop_Clr(ClrStack):
    if len(ClrStack) == 0:
          print("Underflow")
          return None
    else:
          return ClrStack.pop()

 # (iii) Function to check if the stack is empty

 def isEmpty(ClrStack):
       return len(ClrStack) == 0
```

OR

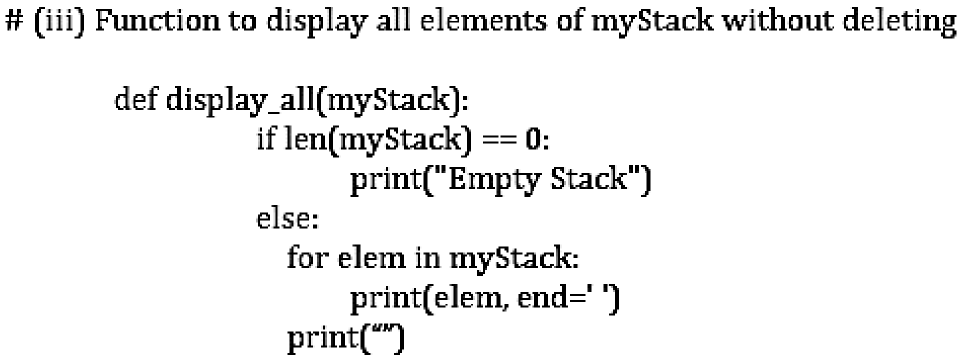
(b)



```
# (i) Function to push last 5 elements of N onto myStack

        def push_trail(N, myStack):
                for elem in N[-5:]:  # Get the last 5 elements
                  myStack.append(elem)
```

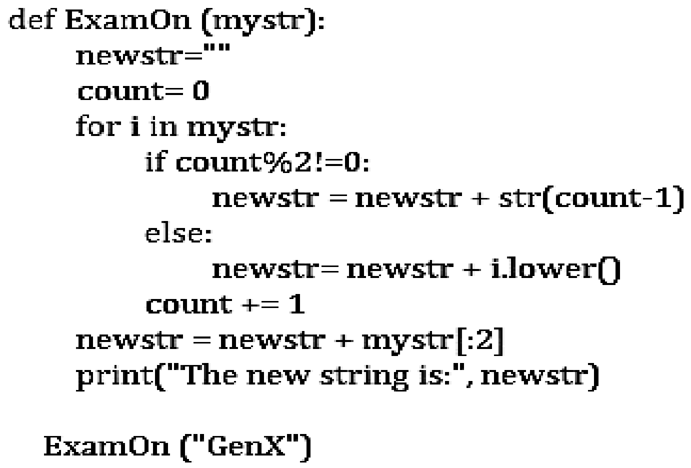


```
# (ii) Function to pop one element from myStack

        def pop_one(myStack):
                if len(myStack) == 0:
                      print("Stack Underflow")
                      return None
                else:
                      return myStack.pop()
```



```
# (iii) Function to display all elements of myStack without deleting

        def display_all(myStack):
                if len(myStack) == 0:
                      print("Empty Stack")
                else:
                  for elem in myStack:
                      print(elem, end=' ')
                  print("")
```

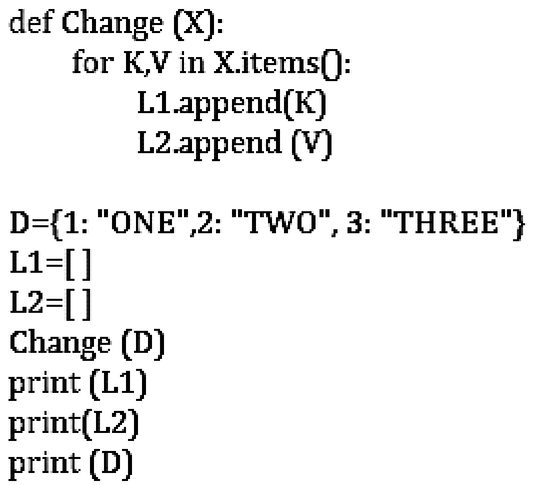
31. (a) Predict the output of the following code:



```
def ExamOn (mystr):
     newstr=""
     count= 0
     for i in mystr:
          if count%2!=0:
               newstr = newstr + str(count-1)
          else:
               newstr= newstr + i.lower()
          count += 1
     newstr = newstr + mystr[:2]
     print("The new string is:", newstr)

  ExamOn ("GenX")
```

OR

(b) Write the output on execution of the following Python code:



```
def Change (X):
    for K,V in X.items():
        L1.append(K)
        L2.append (V)

D={1: "ONE",2: "TWO", 3: "THREE"}
L1=[ ]
L2=[ ]
Change (D)
print (L1)
print(L2)
print (D)
```

Ans-31 (a) The new string is: g0n2Ge

OR

(b)

```
[1, 2, 3]
['ONE', 'TWO', 'THREE']
{1: 'ONE', 2: 'TWO', 3: 'THREE'}
```

SECTION-D (4 x 4 = 16 Marks)

32. Suman has created a table named **WORKER** with a set of records to maintain the data of the construction sites, which consists of **WID**, **WNAME**, **WAGE**, **HOURS**, **TYPE**, and **SITEID**. After creating the table, she entered data in it, which is as follows: **4**

WID	WNAME	WAGE	HOURS	TYPE	SITEID
W01	Ahmed J	1500	200	Unskilled	103
W11	Naveen S	520	100	Skilled	101
W02	Jacob B	780	95	Unskilled	101
W15	Nihal K	560	110	Semiskilled	NULL
W10	Anju S	1200	130	Skilled	103

(a) Based on the data given above, answer the following questions:

(i) Write the SQL statement to display the names and wages of those workers vhose wages are between 800 and 1500.

(ii) Write the SQL statement to display the record of workers whose **SITEID** is ot known.

(iii) Write the SQL statement to display WNAME, WAGE and HOURS of all those vorkers whose **TYPE** is **'Skilled'**.

(iv) Write the SQL statement to change the **WAGE** to 1200 of the workers where he **TYPE** is “**Semiskilled**".

OR

(b)Considering the above given table **WORKER**, write the output on execution of the following SQL commands:

(i) SELECT WNAME, WAGE*HOURS FROM WORKER WHERE SITEID = 103;

(ii) SELECT COUNT (DISTINCT TYPE) FROM WORKER;

(iii) SELECT MAX (WAGE), MIN (WAGE), TYPE FROM WORKER GROUP BY TYPE;

(iv) SELECT WNAME, SITEID FROM WORKER WHERE TYPE="Unskilled" ORDER BY HOURS;

Ans-32

(a) (i) SELECT WNAME, WAGE FROM WORKER WHERE WAGE BETWEEN 800 AND 1500;

(ii) SELECT * FROM WORKER WHERE SITEID IS NULL;

(iii) SELECT WNAME, WAGE, HOURS FROM WORKER WHERE TYPE = 'Skilled';

(iv) UPDATE WORKER SET WAGE = 1200 WHERE TYPE ='Semiskilled';

OR

(b) (i)

WNAME	**WAGE*HOURS**
Ahmed J	300000
Anju S	156000

(ii)

COUNT(DISTINCT TYPE)
3

MAX(WAGE)	**MIN(WAGE)**	**TYPE**
1500	780	Unskilled
1200	520	Skilled
560	560	Semiskilled

(iii)

WNAME	**SITEID**
Jacob B	101
Ahmed J	103

(iv)

33. A csv file **"P_record.csv"** contains the records of patients in a hospital. Each record of the file contains the following data: **4**

- Name of a patient
- Disease

- Number of days patient is admitted
- Amount

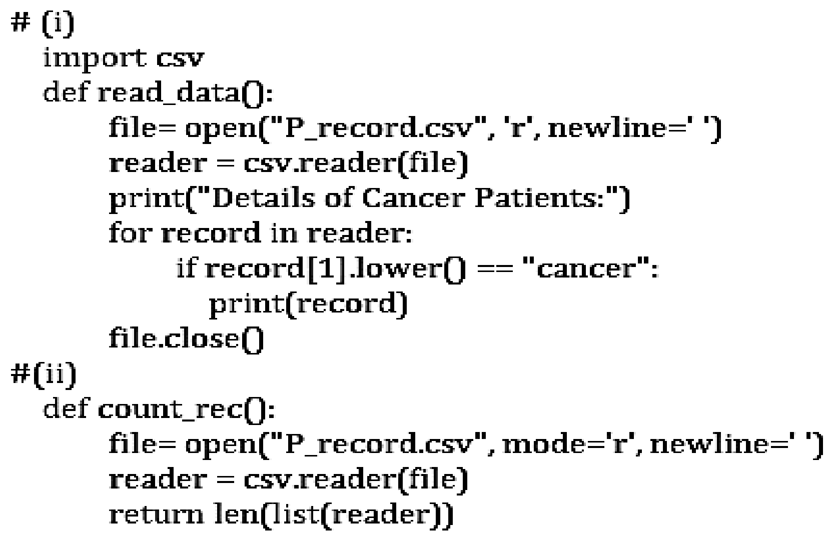
For example, a sample record of the file may be:
["Gunjan", "Jaundice", 4,15000]

Write the following Python functions to perform the specified operations on this file:

(i) Write a function **read_data( )** which reads all the data from the file and displays the details of all the **'Cancer'** patients.

(ii) Write a function **count_rec( )** which counts and returns the number of records in the file.

Ans-33



```
# (i)
   import csv
   def read_data():
         file= open("P_record.csv", 'r', newline=' ')
         reader = csv.reader(file)
         print("Details of Cancer Patients:")
         for record in reader:
               if record[1].lower() == "cancer":
                  print(record)
         file.close()
#(ii)
   def count_rec():
         file= open("P_record.csv", mode='r', newline=' ')
         reader = csv.reader(file)
         return len(list(reader))
```

34. Assume that you are working in the IT Department of a Creative Art Gallery (CAG), which sells different forms of art creations like Paintings. Sculptures etc. The data of Art Creations and Artists are kept in tables **Articles** and **Artists** respectively. Following are few records from these two tables: **4**

Table: Articles

Code	A Code	Article	DOC	Price
PL001	A0001	Painting	2018-10-19	20000
SC028	A0004	Sculpture	2021-01-15	16000
QL005	A0003	Quilling	2024-04-24	3000

Table: Artists

A Code	Name	Phone	Email	DOB
A0001	Roy	595923	reCrAG.com	1986-10-12
A0002	Ghosh	1122334	ghosh@CrAG.com	1972-02-05
A0003	Gargi	121212	Gargi@CrAG.com	1996-03-02
A0004	Mustafa	33333333	Mf@CrAg.com	2000-01-01

Note :

- The tables contain many more records than shown here.
- DOC is Date of Creation of an Article.

As an employee of CAG, you are required to write the SQL queries for the following:

(i) To display all the records from the Articles table in descending order of price.
(ii) To display the details of Articles which were created in the year 2020.
(iii) To display the structure of Artists table.
(iv) (a) To display the name of all artists whose Article is Painting through Equi Join.

OR

(b) To display the name of all Artists whose Article is 'Painting' through Natural Join.

Ans-34

(i) SELECT * FROM Articles ORDER BY Price DESC;
(ii) SELECT * FROM Articles WHERE YEAR(DOC) = 2020;
(iii) DESC Artists; OR DESCRIBE Artists;
(iv) (a) SELECT Artists.Name FROM Articles, Artists WHERE Articles.ACode = Artists.ACode AND Articles.Article = 'Painting';

OR

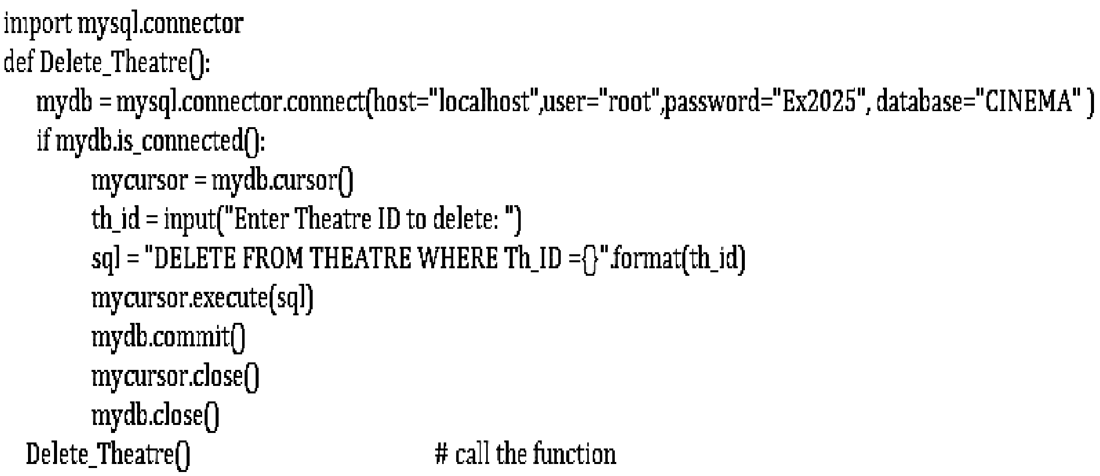
(b) SELECT Name FROM Articles NATURAL JOIN Artists WHERE Article = 'Painting';

35. A table, named THEATRE, in CINEMA database, has the following structure: **4**

Field	**Type**
Th_ID	char (5)
Name	varchar (15)
City	varchar (15)
Location	varchar (15)
Seats	Int

Write a function Delete_Theatre( ), to input the value of Th_ID from the user and permanently delete the corresponding record from the table.
Assume the following for Python-Database connectivity:
Host: localhost, User: root, Password: Ex2025

Ans-35



```
import mysql.connector
def Delete_Theatre():
    mydb = mysql.connector.connect(host="localhost",user="root",password="Ex2025", database="CINEMA" )
    if mydb.is_connected():
            mycursor = mydb.cursor()
            th_id = input("Enter Theatre ID to delete: ")
            sql = "DELETE FROM THEATRE WHERE Th_ID ={}".format(th_id)
            mycursor.execute(sql)
            mydb.commit()
            mycursor.close()
            mydb.close()
   Delete_Theatre()                              # call the function
```

SECTION-E (2 X 5 = 10 Marks)

36. A file, PASSENGERS. DAT, stores the records of passengers using the following structure: **[PNR, PName, BRDSTN, DESTN, FARE]** **5**

Where:

PNR	**-**	**Passenger Number (string type)**
PName	**-**	**Passenger Name (string type)**
BRDSTN	**-**	**Boarding Station Name (string type)**
DESTN	**-**	**Destination Station Name (string type)**
FARE	**-**	**Fare amount for the journey (float type)**

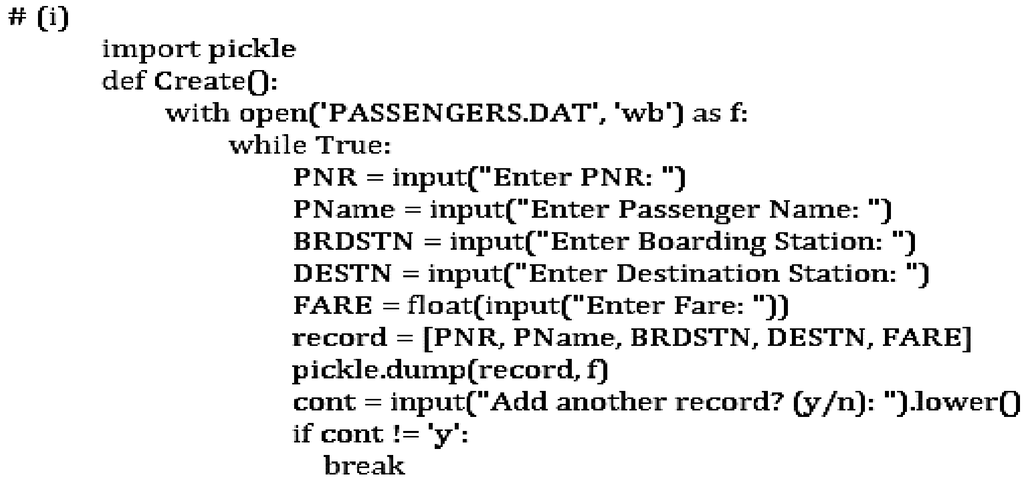
Write user defined functions in Python for the following tasks:

(i) **Create ( )** - to input data for passengers and write it in the binary file **PASSENGERS.DAT**.

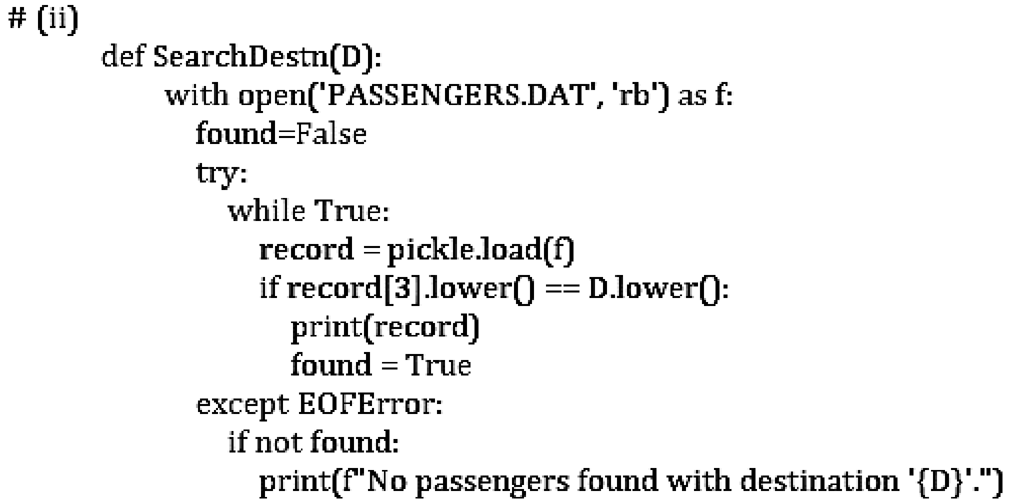
(ii) **SearchDestn (D)** - to read contents from the file PASSENGERS. DAT and display the details of those Passengers whose DESTN matches with the value of D.

(iii) **UpdateFare( )** to increase the fare of all passengers by 5% and rewrite the updated records into the file **PASSENGERS.DAT**.

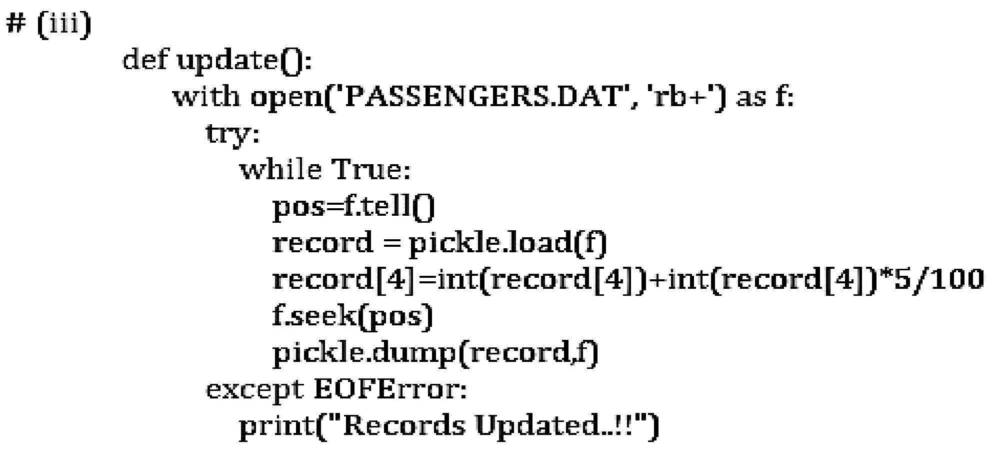
Ans 36



```
# (i)
import pickle
def Create():
    with open('PASSENGERS.DAT', 'wb') as f:
        while True:
            PNR = input("Enter PNR: ")
            PName = input("Enter Passenger Name: ")
            BRDSTN = input("Enter Boarding Station: ")
            DESTN = input("Enter Destination Station: ")
            FARE = float(input("Enter Fare: "))
            record = [PNR, PName, BRDSTN, DESTN, FARE]
            pickle.dump(record, f)
            cont = input("Add another record? (y/n): ").lower()
            if cont != 'y':
              break
```



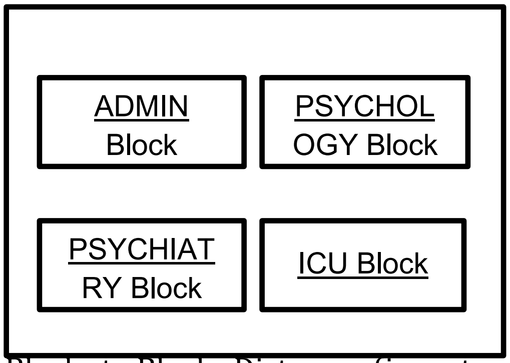
```
# (ii)
def SearchDestn(D):
    with open('PASSENGERS.DAT', 'rb') as f:
      found=False
      try:
        while True:
          record = pickle.load(f)
          if record[3].lower() == D.lower():
            print(record)
            found = True
      except EOFError:
        if not found:
          print(f"No passengers found with destination '{D}'.")
```



```
# (iii)
        def update():
            with open('PASSENGERS.DAT', 'rb+') as f:
              try:
                while True:
                  pos=f.tell()
                  record = pickle.load(f)
                  record[4]=int(record[4])+int(record[4])*5/100
                  f.seek(pos)
                  pickle.dump(record,f)
              except EOFError:
                print("Records Updated..!!")
```

37. Swabhaav' is a big NGO working in the field of Psychological Treatment and Counselling, having its Head Office in Nagpur. It is planning to set up a center in Vijayawada. The Vijayawada Center will have four blocks -ADMIN, PSYCHIATRY, PSYCHOLOGY, and ICU. You, as a Network Expert, need to suggest the best network-related solutions for them to resolve the issues/problems mentioned in questions (i) to (v), keeping the following parameters in mind: **5**

Vijayawada Center



Blocks to Blocks Distances (in meters):

From	To	Distance
ADMIN	PSYCHIATRY	65 m
ADMIN	PSYCHOLOGY	65 m
ADMIN	ICU	65 m
PSYCHIATRY	PSYCHOLOGY	100 m
PSYCHIATRY	ICU	50 m
PSYCHOLOGY	ICU	50 m

Distance of Nagpur Head Office from Vijayawada Center = 700 KM

Number of Computers in each block is as follows -

Block	No. of Computers
ADMIN	16
PSYCHIATRY	40
PSYCHOLOGY	19

ICU	20

(i) Suggest the most appropriate location of the server inside the Vijayawada Center. Justify your choice.

(ii) Which hardware device will you suggest to connect all the computers within each block of Vijayawada Center?

(iii) Draw a cable layout to efficiently connect various blocks within the Vijayawada Center.

(iv) Where should the router be placed to provide internet to all the computers in the Vijayawada Center?

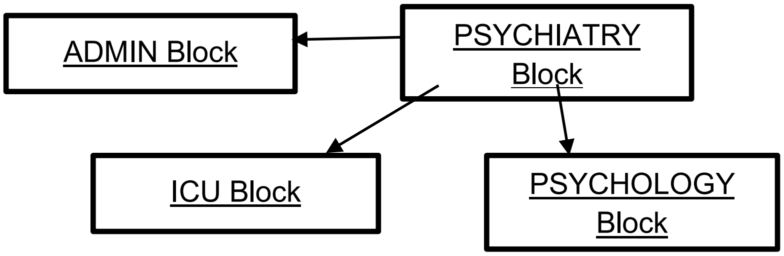
(v) (a) The Manager at Nagpur wants to remotely access the computer in Admin block in Vijayawada. Which protocol will be used for this?

OR

(b) Which type of Network (PAN, LAN, MAN or WAN) will be set up among the computers connected with Vijayawada Center?

Ans-37

(i) The **PSYCHIATRY block** is the most appropriate location to place the server.



Justification:

- PSYCHIATRY block has the maximum number of computers (40 computers), so maximum data communication is required there.
- It is centrally located with minimum average distance to all other blocks.

(ii) A **Switch** should be used to connect all the computers within each block.

(iii) Connect all blocks using a **star topology** with **PSYCHIATRY block** as the central node.

(iv) The **Router** should be placed in the **PSYCHIATRY block** to provide internet connectivity to all blocks.

(V) (a) The **RDP (Remote Desktop Protocol)** protocol will be used for remote access.
(Alternatively, **SSH (Secure Shell)** can also be used.)

OR

(b) The network set up in Vijayawada Center will be a **LAN (Local Area Network)**

KENDRIYA VIDYALAYA SANGATHAN
UNSOLVED SAMPLE QUESTION PAPER-1

Subject: Computer Science (Theory) **Class: XII**

Time Allowed: 3:00 Hours **Max. Marks - 70**

General Instructions:

- This question paper contains 37 questions.
- All questions are compulsory. However, internal choices have been provided in some questions. Attempt only one of the choices in such questions
- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A consists of 21 questions (1 to 21). Each question carries 1 Mark.
- Section B consists of 7 questions (22 to 28). Each question carries 2 Marks.
- Section C consists of 3 questions (29 to 31). Each question carries 3 Marks.
- Section D consists of 4 questions (32 to 35). Each question carries 4 Marks.
- Section E consists of 2 questions (36 to 37). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.
- In case of MCQ, text of the correct answer should also be written.

Q. No.	**SECTION-A (21 x 1 = 21 Marks)**	**Marks**
1.	State True or False: "In a nested loop, a break statement terminates all the nested loops in one go."	**1**

2.	Identify the output of the following code snippet:
	line="successful entrepreneurs"
	r=line.partition("en")
	print(r)
	(A) 'successful ', 'en', 'trepreneurs' (B) ('successful', 'en', 'trepreneurs')
	(C) ['successful ', 'en', 'trepreneurs'] (D) ('successful ', 'en', 'trepreneurs')
		1

3.	Which of the following expressions evaluates to Non-zero value? (A) 3 and 0 (B) True==1 and 6-6 (C) 0==1 and 0 or 5 (D) 2**(2**0)+5%7-50//7	**1**

4.	What is the output of the given expression?
	remark = "NLP-Natural Language Processing"
	note = remark(2:18].split( )
	print(note)
	(A) ['P-Natural', 'Languag'] (B) ['LP-Natural', 'Langua']
	(C) ['P-Natural', 'Langua'] (D) ['LP-Natural', 'Languag']
		1

5.	Which among the following errors will occur at the time of execution of given python code:
	>>> Oceans=('pacific', 'arctic', 'Atlantic', 'southern')
	>>> print(Oceans[4])
		1

	(A) ValueError (B) IndexError (C) NameError (D) TypeError	
6.	What will be the output of the following code? Bparts=['ear','nose','eye'] print('2'.join(Bparts)) (A) ear2nose2eye (B) ear2nose2eye2 (C) ['ear','nose','eye', 'ear','nose','eye'] (D) 2ear2nose2eye2	**1**
7.	Which of the following statement(s) would give an error after executing the following code ? Stud= { "Kiran": 70, "Jaya": 95} # Statement 1 print (Stud[95]) # Statement 2 Stud ["Arvind"]=60 # Statement 3 print(Stud.pop( )) # Statement 4 print(Stud) # Statement 5 (A) Statement-2 (B) Statement-3 (C) Statement-4 (D) Statement-2 and 4	**1**
8.	Which of the following is not correct? (A) del deletes the list or tuple from the memory (B) remove deletes the list or tuple from the memory (C) pop is used to delete an element at a certain position (D) pop(<index>) and remove(<element>) performs the same operation	**1**
9.	If a table has one Primary key, two alternate keys and one foreign key. How many Candidate keys will this table have? (A) 1 (B) 2 (C) 3 (D) 4	**1**
10.	The correct syntax of dump( ) is: (A) file_object=dump(data_object) (B) file_object.dump(data_object) (C) pickle.dump(data_object,file_object) (D) pickle.dump(file_object,data_object)	**1**
11.	State True or False: "The keys of a dictionary must be of immutable types."	**1**

12.	What will be the output of the following code?
	v = 50
	def Change(n):
	    global v
	    v, n = n, v
	    print(v, n, sep = "#", end = "@")
	Change(20)
		1

	print(v) (A) 20#50@20 (B) 50@20#50 (C) 50#50#50 (D) 20@50#20	
13.	Which SQL command can decrease cardinality of a relation?	**1**
14.	Which aggregate function in SQL displays the number of values in the specified column ignoring the NULL values? (A) len( ) (B) count( ) (C) number( ) (D) num( )	**1**
15.	In MySQL, Which type of value should not be enclosed within quotation marks? (A) Date (B) Varchar (C) Float (D) Char	**1**
16.	State True or False: If table A has 6 rows and 3 columns, and a table B has 5 rows and 2 columns, the Cartesian product of A and B will have 30 rows and 5 columns.	**1**
17.	Which device is primarily used to amplify and regenerate signals in a network, allowing data to travel longer distances? (A) Gateway (B) Repeater (C) Router (D) Switch	**1**
18.	________ is used for point-to-point communication or unicast communication such as radar and satellite. (A) Infrared (B) Bluetooth (C) Microwaves (D)Radio waves	**1**
19.	In __________ switching technique, data is divided into chunks of packets and travels through different paths and finally reach the destination.	**1**
	Q.20 and Q.21 are **Assertion(A)** and **Reason(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	
20.	**Assertion (A):** The return statement in a Python function is optional. **Reason (R):** If no return statement is used, the function returns None by default.	**1**
21.	**Assertion (A):** In SQL, the GROUP BY clause is used to arrange identical data into groups. **Reason (R):** The GROUP BY clause is mandatory when using aggregate functions like SUM( ) or AVG( ).	**1**
	SECTION-B (7 x 2=14 Marks)	
22.	Write difference between mutable and immutable data types? Write name of any two mutable data types.	**2**
23.	i. Which operator in python has right to left associativity? ii. Evaluate the following expression and write output: >>> 5//4**3%6+2	**2**

24. Consider T=(10,20,30) and L=[60,50,40] and answer the question I and II. **2**

(I)

A. Write command(s) to add tuple **T** in list **L**.

OR

B. Write command to find and delete element 20 from tuple **T**

(II)

A. Write command to add **50** in **L** at position **2**.

OR

B. Write command to delete the variable **T**.

25. Look at the following python code and find the possible output(s) from the options (i) to (iv) following it. Also, write the highest and lowest values that can be pointed by label **p**. **2**

```
import random
text = "NewYear"
p=random.randint(0,2)
q=random.randint(1,2)
for j in range(0, p+q):
   print(text[j], end='@')
```

i. e@w@Y@ **ii.** N@

iii. N@e@w@Y@ **iv.** N@e@w@Y@e@

26. The code provided below is intended to check a number whether it is perfect number or not. However, there are **syntax and logical errors** in the code. Rewrite the code in python after removing all error(s). Underline each correction done in the code. **2**

```
def Perfect_Number( ):
   n=input(int("Enter a number: "))
   sum=0
   for i in range(1,n):
      if n%i==0:
         sum=+i
   if N==sum:
      print("Perfect number")
   else:
      print("Not a perfect number")

Perfect_Number( ):
```

27. **2**

i.

A. What constraint should be applied on a table column so that **NULL** is not allowed in that column, but duplicate values are allowed.

OR

	B. Categorize the following commands as DDL and DML: INSERT, UPDATE, ALTER, DROP **ii.** **A.** Write the SQL command to change the size of the attribute **Salary** in the table **Employee** to allow the maximum value of 99999.99 to be stored in it. **OR** **B.** Write an SQL command to change a table name from **Employee** to **Emp**.	
28.	Write two points of difference between **XML** and **HTML.** **OR** Write two points of difference between **Circuit Switching** and **Packet switching.**	**2**
	SECTION-C (3 x 3 = 9 Marks)	
29.	Mr. Ashutosh has written a poem in a text file named as **“POEM.TXT”.** Now he wants to count number of lines which start with the character ‘W’ (including uppercase and lowercase) in the text file “POEM.TXT”. Help him to write a user defined function **CountLineW( )** in python to solve the problem. Example: If the “POEM.TXT” contents are as follows: A tiny ant goes marching by, With little legs and wings to fly. Carrying crumbs, oh so small, Working together, one and all. The output of the function should be: No. of lines start with character W are: 2 **OR** Write a function **CountYouMe( )** in python which reads the content of a text file “BIOGRAPHY.TXT” and counts the words ‘You’ and ‘Me’ separately. (Not case sensitive). *Example:* If the contents in “BIOGRAPHY.TXT” are as follows: You are a hero for me you gifted a book to me which helped me a lot to learn the programming It gave me a chance to develop business software The output of the function should be: Count of You in file: 2 Count of Me in file: 4	**3**
30.	You have a stack named **MovieStack** that contains records of movies. Each movie record is represented as a list containing **movie_title**, **director_name**, and	**3**

release_year.

Write the following user-defined functions in Python to perform the specified operations on the stack **MovieStack**:

i. **add_movie(MovieStack, new_movie):** This function takes the stack MovieStack and a new movie record new_movie as arguments and pushes the new movie record onto the stack and display the stack.

ii. **remove_movie(MovieStack)**: This function removes the topmost movie record from the stack and returns it. If the stack is already empty, the function should display "Underflow".

iii. **view_top(MovieStack)**: This function displays the topmost element of the stack without deleting it. If the stack is empty, the function should display 'None'.

OR

Stationery_Item is a dictionary containing the details of stationary items.

Write a user defined function **PUSH_DATA(Stationery_Item),** to push the name of those items into the stack named as **PriceStack** which have price **more than 75** and display stack. Also display the number of elements pushed into the stack.

For example: If the dictionary contains the following data:

Stationery_Item={"Pen":106,"Pencil":59,"Notebook":80,"Eraser":25}

The stack should contain

['Pen', 'Notebook']

The output should be:

Number of elements in stack: 2

31. Predict the output of the following code: **3**

```
d={ }
V="programs"
for x in V:
   if x in d.keys( ):
      d[x]=d[x]+1
   else:
      d[x]=1
print(d)
```

OR

Predict the output of the following code:

```
V="interpreter"
L=list(V)
L1=""
for x in L:
   if x in ['e','r']:
      L1=L1+x
print(L1)
```

SECTION-D (4 x 4 = 16 Marks)

32. Consider a table **TRAIN** as given below: **4**

TNo	TName	Train_Type	Source	Destination	Fare
12001	Rajdhani Express	Superfast	Delhi	Mumbai	3000
12625	Shatabdi Express	Superfast	Delhi	NULL	1200
12501	North East Express	Express	Guwahati	Delhi	1500
16159	Kanyakumari Exp	Express	Kanyakumari	Delhi	2500
20814	Jodhpur-Puri Exp	Express	Jodhpur	Puri	2800
19412	Sabarmati Express	Express	Sabarmati	NULL	1100

A) Write the following queries:

i. Display type of train and total fare of each type of train.

ii. Display Train Number, Train Name and Fare of those trains whose name starts with the alphabet letter **'S'**.

iii. Display train number, train name and source of those trains whose destination is **NULL**.

iv. Count the number of trains which have train type as superfast and source is '**Delhi**'**.**

OR

B) Write the output:

i. Select TNAME, Fare from TRAIN Where Fare<2000 order by TName;

ii. Select avg(fare) from TRAIN group by Train_Type;

iii. Select TName, Destination, Fare from TRAIN where fare <>2500 and destination IS NULL;

iv. Select min(fare) from TRAIN where Train_Type= "Express";

33. Ms. Pallavi is a python programmer working in a software company. She has to develop a simple inventory management system of all books in a library. She has created a csv file named **bookdata.csv**, to store the details of books. The structure of **bookdata.csv** is: **4**

[Book_Number, Book_Name, Author, Price]

Ms. Pallavi wants to write a Program in Python that defines and calls the following user defined functions:

i. **Insert_Books( )** – To accept and add data of **n** number of books to a CSV file **'bookdata.csv'.**

ii. **Display( )** – To show the record of those books which have the price more than 400 in the CSV file named **'bookdata.csv'**.

34. Ms. Nishi has been entrusted with the bank Database. She needs to access some information from **LOAN** and **BORROWER** tables for a survey analysis. Help her to extract the following information by writing the desired SQL queries as mentioned below. **4**

Table: LOAN

loan_number	branch_name	Amount
L123	Nagpur	45000
L456	Pune	60000
L347	Delhi	80000
L987	Delhi	25000
L901	Pune	45000

Table : BORROWER

customer_name	loan_number
Ajit Das	L456
Rohan Yadav	L901
Suman Verma	L123
Ayesha Tiwari	L987
Saurav	L347

i. To display customer name and branch name of those customers who have taken loan from Delhi branch.

ii. To display loan number, customer name and amount of those customers who have taken loan more than 40000.

iii. To display branch name and average amount of that branch which has given average loan amount more than 50000.

iv.

A. To display customer name and amount in descending order of amount.

OR

B. What will be degree of resultant table after performing natural join of these two tables.

35. Consider a database named 'DB' containing a table named '**Vehicle**' with the following structure **4**

Field	Type
Model	char(10)
Make_year	Int(4)
Qty	Int(3)
Price	Number(8,2)

Write the following Python functions to perform the following operation as mentioned:

i. **Add_Vehicle( )** - which takes input of data and store it to the table

ii. **Search_vehicle( )** – which can search a model given by user and show it on screen

* Assume the following for Python – Database connectivity:
Host: localhost, **User:** root, **Password:** work

SECTION-E (2 X 5 = 10 Marks)

36. Mr. Ashok is working on a Toy Shop project to manage toys records using Python. The toys data is stored in a binary file named **TOYDATA.DAT**. The binary file **TOYDATA.DAT** contains each record in given format: **5**

{“Toy_ID”: T_ID, “TName”:toy_name, “Price”:price}

Where

- **Toy_ID**: Toy ID (string)
- **TName**: Toy name (string)
- **Price**: Price of toy (integer)

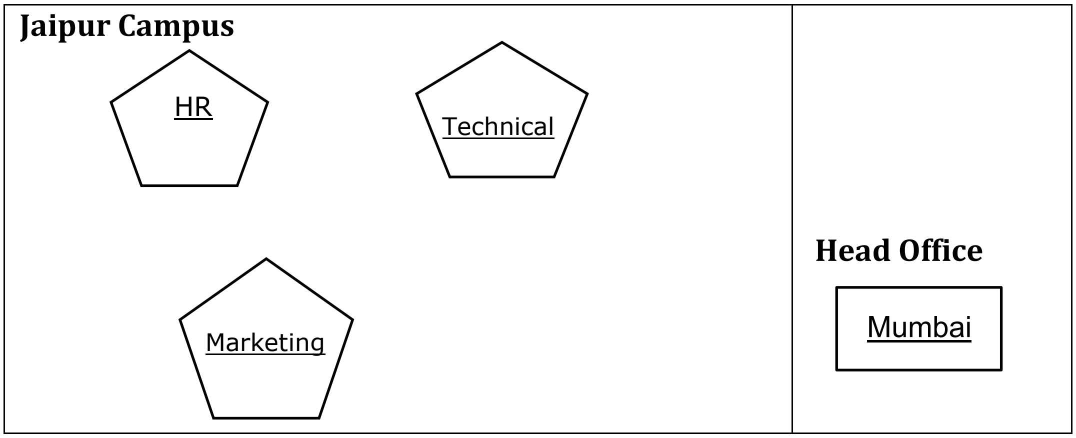
You as a programmer, help him to write following python functions:

i. **ADD_Data( )** : To write **n** records in binary file TOYDATA.DAT by taking the values for each record from user.

ii. **SHOW_Data( )** : Read all records from binary file and display them.

iii. **Remove_Toy( )** : that deletes the record of a toy in the file **TOYDATA.DAT** based on the **Toy ID** provided by the user. If the Toy ID does not exist in the file, display an appropriate message.

37. **“MyTech Services”** is planning to set up its India campus at Jaipur with its Head Office at Mumbai. The Jaipur campus has 3-main blocks-**HR, Technical** and **Marketing**. You as a network expert have to suggest the best network related solutions for their problems raised in **(i)** to **(v)**. **5**



Distance between various building blocks:

HR BLOCK to TECHNICAL BLOCK	45 m
HR BLOCK to MARKETING BLOCK	98 m
TECHNICAL BLOCK to MARKETING BLOCK	107 m
Head Office to JAIPUR Campus	1275 KM

Number of computers in each Block:

HR BLOCK	10
TECHNICAL BLOCK	105
MARKETING BLOCK	45

Suggest the most appropriate location of the server inside the JAIPUR campus (out of the 3 blocks), to get the best connectivity for maximum number of computers. Justify your answer.

i. Which among the following devices will you suggest to be procured by the company for connecting all the computers within each of their offices?
   - Switch/Hub
   - Modem
   - Bridge

iii. Suggest network type (out of LAN, MAN, WAN) for connecting each of the following set of their offices:
   a. HR and Marketing Block
   b. Head Office and Jaipur office

iv. Which of the following communication medium, you will suggest to be procured by the company for connecting their local offices in Jaipur for very effective and fast communication?
   - Telephone cable
   - Optical fiber
   - Ethernet cable

v.
   A. In JAIPUR Campus, in between which offices repeater should be installed? Justify the answer.

   OR

   B. Suggest and draw the cable layout to efficiently connect various blocks within the JAIPUR campus for connecting the computers.

LINK TO UNSOLVED SAMPLE PAPERS II & III

****

REFERENCES

1. **NCERT Textbook for Computer Science – Class XII**
   National Council of Educational Research and Training (NCERT)
   https://ncert.nic.in

2. **CBSE Curriculum and Guidelines for Class XII Computer Science (083)**
   Central Board of Secondary Education
   https://cbseacademic.nic.in

3. **Python Official Documentation**
   Python Software Foundation
   https://docs.python.org/3/