SAMPLE QUESTION PAPER - 4
Computer Science (083)
Class XII (2024-25)

Time Allowed: 3 hours **Maximum 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.

Section A

1. State true or false: **[1]**

   Like **while**, the **for** loop also works with conditions and truth values.

2. A relational database consists of a collection of ________. **[1]**

   a) Attributes b) Tuples

   c) Keys d) Relations

3. What does the special file called data dictionary contains? **[1]**

   a) The data types of all data in all files. b) The names of all fields in all files.

   c) All of these d) The widths of all fields in all files.

4. Given a function that does not return any value, what value is thrown by default when executed in shell? **[1]**

a) void b) None

c) bool d) int

5. Out of the following, find those identifiers, which cannot be used for naming Variables or Functions in a Python program: **[1]**
Price*Qty, class, For, do, 4thCol, totally, Row31, _Amount

6. If value of checksum is 0, then message is **[1]**

a) accepted b) resend

c) rejected d) sent back

7. Raghav is trying to write an object obj1 = (1,2,3,4,5) on a binary file "test.bin". **[1]**
Consider the following code written by him.
import pickle
obj1 = (1,2,3,4,5) myfile = open("test.bin".'wb')
pickle.________ #Statement 1
myfile.close()
Identify the missing code in Statement 1.

a) dump(myfile.obj1) b) write(obj1.myfile)

c) dump(obj1.myfile) d) load(myfile.obj1)

8. ________ operation is required when you want to create your records into a database table. **[1]**

a) Insert b) Read

c) Update d) Delete

9. Consider the table with structure as: **[1]**
Student(ID, name, dept name, tot_cred)
Which attribute will form the primary key?

a) Dept b) ID

c) Total credits d) Name

10. What is the use of mkdir() method? **[1]**

11. State true or false: **[1]**

Default parameters cannot be skipped in the function call.

12. The insertion operation in the stack is called ________. **[1]**

a) push b) insert

c) top d) pop

13. What is data redundancy? What are the problems associated with it? **[1]**

14. The ________ translates internet domain and hostnames to IP address. **[1]**

a) internet relay chat b) routing information protocol

c) network time protocol d) domain name system

15. Which two lines of code are valid strings in Python? **[1]**

A. This is a string

B. 'This is a string'

C. (This is a string)

D. "This is a string"

a) A, C b) B, D

c) A, D d) C, D

16. Which of the following clauses in SQL is most appropriate to use to select matching tuples in a specific range of values? **[1]**

a) IS b) BETWEEN

c) IN d) LIKE

17. In computer, process of superimposing a low frequency signal over a high frequency signal is called **[1]**

a) frequency modulation b) amplitude modulation

c) modulation d) demodulation

18. An eight wired connector that is used to connect computers on a local area network? **[1]**

a) RJ-45 b) Ethernet Card

c) Switch d) Modem

19. **Assertion (A):** When the comparison operator is directly applied to a series object, it returns a filtered result containing the value that returns true. **[1]**
Reason (R): Applying comparison operator on series works in vectorized way.

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.

20. **Assertion (A):** Python provides the remove() method which is used to remove the file pointer. **[1]**
Reason (R): The mkdir() method is used to create the directories in the current working directory.

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.

21. **Assertion (A):** The enumerate() function takes a collection (e.g. a tuple) and returns it as an enumerate object. **[1]**
Reason (R): The enumerate() adds a counter as the key of the enumerated object.

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.

Section B

22. What is collision in a network? How does it impact the performance of a network? **[2]**

23. Write the code to create a table Product in database Inventory with following fields: **[2]**

Fields	**Datatype**
PID	varchar(5)

PName	char(30)
Price	float
Rank	varchar(2)

24. In the below given code fragments, indicate the data type of each bold part by choosing the correct type of data from the following type. **[2]**

a. int

b. float

c. bool

d. str

e. function

f. list of int

g. list of str

i. L = inputline.split( )
while L != ( ) :
print(L)
L = L[1 :]

ii. L = ['Hiya', 'Zoya', 'Preet']
print(L[0] + L[1])

OR

Underline the run-time error in the following program.

```
a = int(input("Enter a : "))
b = 0
c = a/b
```

25. What are data types? What are the main objectives of datatypes? **[2]**

26. Find the errors in following code and write the correct code. **[2]**

```
if v < 5:
for j in range(v):
print "ABC"
else:
print "XYZ"
```

i. Underline the corrections

ii. Write the reason!error next to it in comment form.

OR

Write the suitable method's name for the below conditions.

i. Adds an element in the end of list.

ii. Returns the index of first occurrence.

iii. Adds contents of list2 to the end of list1.

27. What is the output produced by following code? **[2]**

```
obj = open("New.txt", "w")
obj.write("A poem by Paramhansa Yogananda")
obj.write("Better than Heaven or Arcadia")
obj.write("I love thee, O my India !")
obj.write("And thy love I shall give")
obj.write("To every brother nation that lives.")
obj.close()
obj1 = open("New.txt", "r")
s1 = obj1.read(48)
print(sl)
obj1.close()
```

OR

Read the code given below and answer the question:

```
fh = open ("main, txt", "w")
fh.write("Bye")
fh.close()
```

If the file contains "GOOD" before execution, what will be the contents of the file after execution of this code?

28. Predict the output of the following code: **[2]**

```
def first (s, i):
```

```
return s * second(i)
def second(n):
return n * 3
print(first('*', 2))
```

Section C

29. What is the difference between a local variable and a global variable? Also, give a suitable Python code to illustrate both. **[3]**

OR

Trace the flow of execution for following programs:

i.
```
1 def power(b., p):
2 r = b ** p
3 return r
4
5 def cpower(a):
6 a = a + 2
7 a = power(a, 0.5)
8 return a
9
10 n = 5
11 result = cpower(n)
12 print (result)
```

ii.
```
1. def increment(x)
2. x = x + 1
3.
4. # main program
5. x = 3
6. print(x)
7. increment(x)
8. print(x)
```

iii.
```
1. def increment(x):
2. z = 45
3. x = x + 1
4. return x
5.
6. # main
```

```
7. y = 3
8. print(y)
9. y = increment(y)
10. print(y)
11. q = 77
12. print(q)
13. increment(q)
14. print(q)
15. print(x)
16. print(z)
```

30. Give a suitable example of a table with sample data and illustrate Primary and Alternate Keys in it. **[3]**

OR

Write the output of the SQL queries (a) and (b) based on the following two tables FLIGHT and PASSENGER belonging to the same database:

Table: FLIGHT

FNO	DEPART	ARRIVE	FARE
F101	DELHI	CHENNAI	4500
F102	DELHI	BENGALURU	4000
F103	MUMBAI	CHENNAI	5500
F104	DELHI	MUMBAI	4500
F105	MUMBAI	BENGALURU	5000

Table: FLIGHT

PNO	NAME	FLIGHTDATE	FNO
P1	PRAKASH	2021-12-25	F101
P2	NOOR	2021-11-20	F103
P3	HARMEET	2020-12-10	NULL
P$	ANNIE	2019-12-20	F105

a. SELECT NAME, DEPART FROM FLIGHT
   NATURAL JOIN PASSENGER;

b. SELECT NAME, FARE
   FROM PASSENGER P, FLIGHT F

WHERE F.FNO = P.FNO AND F.DEPART = 'MUMBAI';

31. Write definition of a method OddSum(NUMBERS) to add those values in the list of NUMBERS, which are not even. **[3]**

OR

What are different types of arguments/parameters that a function can have?

Section D

32. A linear stack called status contains the following information: **[4]**
Phone number of Employee
Name of Employee
Write the following methods to perform given operations on the stack status:

i. **Push_element ()** To Push an object containing Phone number of Employee and Name of Employee into the stack.

ii. **Pop_element ()** To Pop an object from the stack and to release the memory.

OR

Write a program that depends upon the user's choice, either pushes or pops an element in a stack.

33. Write a program with method countand() to count the word 'and' or 'And as an independent word in a text file "status.txt". e.g. if the content of the file "status, txt" is as follows: **[4]**
Welcome to your one-step solutions for all your study, practice and assessment needs for various competitive & recruitment examinations and school segment. We have been working tirelessly for over a decade to make sure that you have best in class study resources because you deserve SUCCESS AND NOTHING LESS...
Then the output of the program should be:
Count of _and_ in file is/are: 3

34. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the table. **[4]**

TABLE: ACCOUNT

ANO	**ANAME**	**ADDRESS**
101	Nirja Singh	Bangalore
102	Rohan Gupta	Chennai

103	Ali Reza	Hyderabad
104	Rishabh Jain	Chennai
105	Simian Kaur	Chandigarh

TABLE: TRANSACT

TRNO	**ANO**	**AMOUNT**	**TYPE**	**DOT**
T001	101	2500	Withdraw	2017-12-21
T002	103	3000	Deposit	2017-06-01
T003	102	2000	Withdraw	2017-06-12
T004	103	1000	Deposit	2017-10-22
T005	101	12000	Deposit	2017-11-06

i. To display details of all transactions of TYPE Deposit from Table TRANSACT

ii. To display the ANO and AMOUNT of all Deposits and Withdrawals done in the month of October 2017 from table TRANSACT

iii. To display the last date of transaction (DOT) from the table TRANSACT for the Accounts having ANO as 103.

iv. To display all ANO, ANAME and DOT of those persons from tables ACCOUNT and TRANSACT who have done transactions less than or equal to 3000.

v. SELECT ANO, ANAME FROM ACCOUNT WHERE ADDRESS NOT IN (CHENNAI BANGALORE);

vi. SELECT DISTINCT ANO FROM TRANSACT

vii. SELECT ANO. COUNT (*), MIN (AMOUNT) FROM TRANSACT GROUP BY ANO HAVING COUNT (*) > 1

viii. SELECT COUNT (*), SUM (AMOUNT) FROM TRANSACT WHERE DOT < = '2017-06-01'

OR

Give output for following SQL queries as per given table(s) :

Table : Books

Book_Id	Book_Name	Author_Name	Publishers	Price	Type	Qty.
F0001	The Tears	William Hopkins	First publ.	750	Fiction	10

F0002	Thunderbolts	: Anna Roberts i	Fist Publ.	700	Fiction	5
T0001	My first C++	Brian and Brooke	EPB	250	Text	10
T0002	C++ Brainworks	A.W. Rossaine	TDH	325	Text	5
C0001	Fast cook	Lata Kapoor	EPB	350	Cookery	8

Table : Issued

Book_Id	Quantity_Issued
F0001	3
T0001	1
C0001	5

i. SELECT COUNT (DISTINCT Publishers) FROM Books.

ii. SELECT SUM(Price) FROM Books WHERE Quantity > 5.

iii. SELECT Book_Name, Author_Name FROM Books WHERE Price < 500.

iv. SELECT COUNT (*) FROM Books.

35. Consider the table product and client with structures as follows: **[4]**

Product	client
ProductName	ClientName
Manufacturer	City
Price	P_ID

Write Python codes for the following:

i. To display the details of those clients whose city is Delhi.

ii. To display the details of products whose price is in range of 50 to 100 (Both values included)

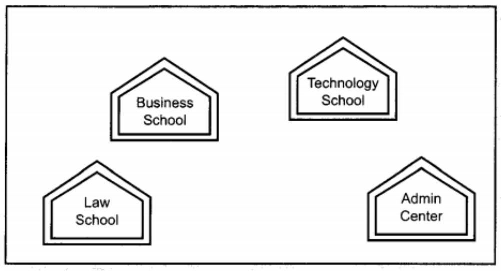
iii. To display the Client Name, City from table Client.

iv. To display the price of all the items by the manufacturer ABC.

v. To display the product name and 4 items its price from the product table.

Section E

36. Great Studies University is setting up its Academic schools at Sunder Nagar and planning to set up a network. The university has 3 academic schools and one administration centre as shown in the diagram below: **[5]**



Center to center distances between various buildings is as follows:

Law School to Business School	60 m
Law School to Technology School	90 m
Law School to Admin Center	115 m
Business School to Technology School	40 m
Business School to Admin Center	45 m
Technology School to Admin Center	25 m

Number of Computers in each of the Schools/Center is as follows:

Law School	25
Technology School	50
Admin Center	125
Business School	35

i. Suggest the most suitable place (i.e., Schools/Center) to install the server of this university with a suitable reason.

ii. Suggest the most efficient connecting medium for connecting these Schools/center for wired connectivity.

iii. Which device you will suggest to be placed/installed in each of these Schools! center to efficiently connect all the computers within these Schools/center?

iv. The university is planning to connect its admission office in the closest big city, which is more than 350 km from the university. Which type of network out of LAN, MAN or WAN will be formed? Justify your answer.

37. Write SQL commands for the queries (i) to (iv) and output for (v) to (viii) based on the tables 'Watches' and Sale given below. **[5]**

Watches

Watchid	**WatchName**	**Price**	**Type**	**Qty_Store**
W001	High Time	10000	Unisex	100
W002	Life Time	15000	Ladies	150
W003	Wave	20000	Gents	200
W004	High Fashion	7000	Unisex	250
W005	Golden Time	25000	Gents	100

Sale

Watchid	**Qty_Sold**	**Quarter**
W001	10	1
W003	5	1
W002	20	2
W003	10	2
W001	15	3
W002	20	3
W005	10	3
W003	15	4

i. TO DISPLAY ALL THE DETAILS OF THOSE WATCHES WHOSE NAME ENDS WITH TIME.

ii. TO DISPLAY WATCH'S NAME AND PRICE OF THOSE WATCHES WHICH HAVE PRICE RANGE IN BETWEEN 5000-15000.

iii. TO DISPLAY TOTAL QUANTITY IN-STORE OF UNISEX TYPE WATCHES.

iv. TO DISPLAY WATCH NAME AND THEIR QUANTITY SOLD IN the FIRST QUARTER.

v. SELECT MAX (PRICE), MIN(QTY_STORE) FROM WATCHES;

vi. SELECT QUARTER, SUM(QTY_SOLD) FROM SALE GROUP BY QUARTER;

vii. SELECT WATCHNAME, PRICE, TYPE FROM WATCHES W, SALE S WHERE W. WATCHID!= S.WATCHID;

viii. SELECT WATCHNAME, QTYSTORE, SUM (QTYSOLD), QTY_STORE - SUM (QTY_SOLD) "STOCK" FROM WATCHES W, SALE S WHERE W. WATCHID = S.WATCHID GROUP BY S.WATCHID;

OR

Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables:

DVD

DCODE	**DTITLE**	**DTYPE**
F101	Henry Martin	Folk
Cl 02	Dhrupad	Classical
C101	The Planets	Classical
F102	Universal Soldier	Folk
R102	A day in the life	Rock

MEMBER

MID	**NAME**	**DCODE**	**ISSUEDATE**
101	AGAM SINGH	R102	2017-11-30
103	ARTH JOSEPH	F102	2016-12-13
102	NISHA HANS	C101	2017-07-24

i. To display all details from the table MEMBER in descending order of ISSUEDATE.

ii. To display the DCODE and DTITLE of all Folk Type DVDs from the table DVD.

iii. To display the Dtype and number of DVDs in each DTYPE from the table DVD.

iv. To display all NAME and ISSUEDATE of those members from the table MEMBER who have DVDs issued (i.e., ISSUEDATE) in the year 2017.

v. SELECT MIN (ISSUEDATE) FROM MEMBER;

vi. SELECT DISTINCT DTYPE FROM DVD;

vii. SELECT D.DCODE. NAME, DTITLE: FROM DVD D, MEMBER M WHERE D.DCODE=M.DCODE;

viii. SELECT DTITLE FROM DVD WHERE DTYPE NOT IN ("Folk", "Classical");

Solution

SAMPLE QUESTION PAPER - 4

Computer Science (083)

Class XII (2024-25)

Section A

1.

   (b) False

   Explanation:

   False

2.

   (d) Relations

   Explanation:

   Fields are the column of the relations

3.

   (c) All of these

   Explanation:

   The data dictionary is structured in tables and views just like other database data.

4.

   (b) None

   Explanation:

   When no value is returned by function in Python, it returns **None** object by default.

5. Price*Qty, class, 4thCol cannot be used for naming variables/functions in a Python program:

6. **(a)** accepted

   Explanation:

   A checksum is a value used to verify the integrity of a file or a data transfer. In other words, it is a sum that checks the validity of data. Checksums are typically used to compare two sets of data to make sure they are the same.

7.

   (c) dump(obj1.myfile)

   Explanation:

   The pickle.dump(obj1, myfile) line serializes the tuple obj1 and writes it to the file.

8. **(a)** Insert

   Explanation:

   Insert

9.

   (b) ID

Explanation:

ID, A primary key is a key that is unique for each record.

10. mkdir() method of the OS module is used to create a directory in the current directory. We pass an argument to this method which contains the name of the directory to be created. mkdir() method raise FileExistsError if the directory to be created already exists.
11.

    (b) False

    Explanation:

    We can skip default parameters in the function call. They will take the default value assigned to them in the function definition.
12. **(a)** push

    Explanation:

    The insertion operation in the stack is called push operation.
13. Data redundancy means having multiple copies of the same data in the database. It leads to problems like wastage of space and data inconsistency.
14.

    (d) domain name system

    Explanation:

    The domain names systems matches the name of website to ip addresses of the website.
15.

    (b) B, D

    Explanation:

    B, D, In Python string is the collection of the characters surrounded by single quotes, double quotes, or triple quotes.
16.

    (b) BETWEEN

    Explanation:

    BETWEEN
17. **(a)** frequency modulation

    Explanation:

    frequency modulation is process of superimposing a low frequency signal over a high frequency signal.
18. **(a)** RJ-45

    Explanation:

    Registered Jack 45 is an eight wired connector that is used to connect computers on a local area network(LAN), especially Ethernet.

19.

(d) A is false but R is true.

Explanation:

Applying comparison operator on series works in vectorized way i.e applies this check on each element and then return true/ false for each element.

20.

(d) A is false but R is true.

Explanation:

The os module provides the remove() method which is used to remove the specified file, not the file pointer. The mkdir() method is used to create the directories in the current working directory.

21.

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

Explanation:

The enumerate() is a buit-in function available with the Python that takes a collection (e.g. a tuple) and returns it as an enumerated object. The enumerate() function adds a counter as the key of the enumerated object.

Example :

```
students = ['X', 'Y' ,'Z']
students_list = enumerate(students)
print(list(students_list))
```

will print [(0, 'X'), (1, 'Y'), (2, 'Z')]

Section B

22. In a computer network, collision is a specific condition that occurs when two or more nodes on a network transmit data at the same time. For example, if two computers on an Ethernet network send data at the same moment, the data will "collide" and not finish transmitting. In case of a collision, the data gets garbled and cannot be read. Also, it may hamper the overall performance of the network as collisions often lead to more retransmissions which clog the network and deteriorate the overall performance of the network.

23.
```
import mysql.connector
mycon = mysql.connector.connect(host = "localhost", user = "system",passwd = "hello",da
cur = mycon.cursor()
db = cur.execute("CREATE TABLE Production(PID varchar (5) Primary key, PName char
mycon.close()
```

24. i. List

ii. String

OR

```
a = int(input("Enter a : "))
b = 0
c = a/b # Division by zero
```

Here, a/b will generate the run-time error.

25. Data types are the classification of data items. Data types represent a kind of value which determines what operations can be performed on that data. Some common data types are Integer, Float, Varchar, Char, String, etc.

    Main objectives of datatypes are:
    i. Optimum usage of storage space
    ii. Represent all possible values
    iii. Improve data integrity

26.
```
v = 3 # v must be defined before being used
if v < 5:
for j in range(v):
print('ABC") # () missing for print()
else: # wrong indentation; else clause can either be for if
# or for for loop
print ( "XYZ” ) # () missing for print()
```

OR

i. append()
ii. index()
iii. extend()

27. The output produced by above code will be:

    A poem by Paramhansa Yogananda Better than Heaven

OR

The content in the file will be "Bye" string only because when an existing file is opened in write mode ("w"), the existing data in the file is truncated. So, "GOOD" string is truncated when the file is opened in write mode.

28. Output of the code is:

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

first('*', 2) calls function first(), which calls second(2) function, then it returns 6 to first() function. Therefore, prints * 6 times as output.

Section C

29. The differences between a local variable and a global variable are as given below :

Local Variable	**Global Variable**
1. It is a variable which is declared within a function or within a block	1. It is a variable which is declared outside all the functions
2. It is accessible only within a function/block in which it is declared	2. It is accessible throughout the program
3.Local variables are created when the function has started execution and are lost when the function terminates.	3.Global variable is created as execution starts and is lost when the program ends.

For example, in the following code, x, xCubed are global variables and n and cn are local variables.

```
def cube(n):
cn = n * n * n
return cn
x = 10
xCubed = cube(x)
print(x, "cubed is", xCubed)
```

OR

i. 1 → 5 → 10 → 11 → 5 → 6 → 7 → 1 → 2 → 3 → 7 → 8 → 11 → 12

ii. 1 → 5 → 6 → 7 → 1 → 2 → 8
[Control did not return to function call statement (7) as no value is being returned by increment()]

iii. 1 → 7 → 8 → 9 → 1 → 2 → 3 → 4 → 9 → 10 → 11 → 12 → 13 → 1 → 2 → 3 → 4 → 14 → 15 → 16
[Control did not return to function call statement (13) as its result is not being stored anywhere.]

30. **Primary Key.** It is the set of one or more attributes that can uniquely identify tuples within a relation..

Alternate Key. It is a candidate key which is not the primary key.

Example:

Table : Class 11

AdmNo	**RollNo**	**Name**	**Marks**
1011	1	Rahat	85
1083	2	Irfan	75
2011	3	Maya	63
1000	4	Shaun	60

999	5	Sukhi	92
1200	6	Zoya	86

In the above table, columns AdmNo and RollNo have unique values for each row, so both are candidates to become primary keys. Hence both of these columns are Candidate keys. Out of these two, we can assign one as Primary key and the other one will become Alternate key.

Candidate keys: AdmNo, RollNo

Primary key: RollNo

Alternate key: AdmNo

OR

a. **SELECT NAME, DEPART FROM FLIGHT**
NATURAL JOIN PASSENGER;

NAME	DEPART
PRAKASH	DELHI
NOOR	MUMBAI
ANNIE	MUMBAI

b. **SELECT NAME, FARE**
FROM PASSENGER P, FLIGHT F
WHERE F.FNO = P.FNO AND F.DEPART = 'MUMBAI';

NAME	FARE
NOOR	5500
ANNIE	5000

31.
```
def OddSum(NUMBERS) :
odd_sum = 0
for num in range (len(NUMBERS)):
if (NUMBERS[num] % 2 != 0:
odd_sum = odd_sum + NUMBERS [num]
print odd_sum
```

OR

A function can have following types of arguments/parameters:

i. **Positional (regular) arguments:** These are the passed argument values in the actual arguments, which are copied to formal arguments by their position in the function call. That is, 1st argument value is given to the 1st parameter, 2nd argument's value is given to the 2nd parameter and so on, e.g., in the following code, argument 5 will be given to the parameter a and argument 7 will be given to the parameter b.

```
def add(a, b):
    return a + b
add(5, 7)
```

ii. **Default Arguments:** These are the default values defined in the function definition for a parameter. Python uses these defaults if corresponding actual arguments are not passed in the function call, e.g., in the following code, the third argument is passed and hence its default value 3 will be taken by the function for the third parameter c:

```
def add(a, b, c = 3):
    return a + b + c
add(5, 7)
```

iii. **Keyword Arguments:** Also called named arguments, the Keyword arguments are passed by it's name instead of their position as opposed to positional arguments in the function call and the position of arguments is irrelevant when calling a function, e.g.,

```
def add(a, b):
    return a + b
add(b = 5, a = 7)
```

Section D

32.
```
def Push_element(Status, Top):
    phone_no = int(input("Enter phone number:"))
    emp_name = input("Enter employee name:")
    St = (phone_no, emp_name)
    Status.append(St)
    Top = Top + 1
    return Top
def Pop_element(Status, Top):
    Slen = len(Status)
    if (Slen <= 0):
        print("Status is empty")
    else:
        phone_no, emp_name = Status.Pop()
        Top = Top-1
        print("Phone number %s and name %s deleted"%(phone_no, emp_name))
    return Top
```

OR

push and pop operation into the stack:-

```
MAX_SIZE = 1000
stack = [0 for i in range(MAX_SIZE)] -
top = 0

def push():
global stack, top
x = int( input ("Enter element to push into stack: " ))
if top >= MAX_SIZE:
print("Cannot push. Stack is full. Overflow!")
else:
stack[top] = x
top += 1

def pop():
global stack, top
if top == 0:
print("Cannot pop. Stack is empty. Underflow!")
else:
top -= 1
def printStack():
print(stack[:top])

# __main__
while True:
print("Please choose operation")
print("1. Push")
print("2. Pop")
print("3. Print")
print("4. Exit")
choice = int(input("Please enter 1/2/3 : " ))
if choice == 4:
break
elif choice == 3:
printStack()
elif choice == 2:
pop( )
elif choice == 1:
```

```
push()
else:
print("Please give a correct input")
```
33.
```
import os
def countand():
    if os.path.isfile("status.txt"):
        fob=open("status.txt", 'r')
        c=0
        while True:
            Str=fob.readline()
            if not Str:
                break
            Str=Str.rstrip().lower().split()
            for i in range(len(Str)):
                if(Str[i]=='and' or Str[i]== 'AND'):
                    c=c+1
        print("Count of_and_in file is/are: ", c)
    else:
        print("File does not exist")
countand()
```

34. i. SELECT * FROM TRANSACT WHERE TYPE = 'Deposit' ;

ii. SELECT ANO , AMOUNT FROM TRANSACT WHERE DOT > = ' 2017-10-01 'AND DOT < = '2017-10-31'

iii. SELECT DOT FROM TRANSACT WHERE ANO = 103 ORDER BY DOT DESC LIMIT =1;

iv. SELECT A.ANO, ANAME, DOT FROM ACCOUNT A, TRANSACT T WHERE A.ANO = T.ANO AND AMOUNT <= 3000;

v.

ANO	ANAME
103	Ali Reza
105	Simran Kaur

vi. DISTINCT ANO

101

102

103

vii.

ANO	COUNT(*)	MIN(AMOUNT)

101	2	2500
103	2	1000

viii.

COUNT (*)	SUM(AMOUNT)
2	5000

OR

i. COUNT (DISTINCT Publishers)

3

ii. SUM(Price)

1350

iii.

Book_Name	Author_Name
My first C++	Brian and Brooke
C++Brainworks	A.W. Rossaine
Fast cook	Lata Kapoor

iv. COUNT (*)

5

35. i.
```
import MySQLdb
db= MySQLdb.connect('localhost', 'Admin','Admin@pwd', 'cosmetics')
cursor=db.cursor()
sql= """Select*from client where city=%s"""
city_query= ('Delhi')
try:
cursor.execute (sql, city_query)
results = cursor.fetchall()
print "Client ID, Name City Product"
For row in results:
CID = row[0]
CName = row[1]
CCity = row[2]
CProd = row[3]
prints "%s %s %s %s" % \ (CID, CName, CCity, CProd)
except
print "Error: unable to fetch data"
cursor.close()
db. close ()
```

ii. import MySQLdb
```
db=MySQLdb.connect('localhost', 'Admin', 'Admin@pwd', 'cosmetics')
cursor= db.cursor()
sql= ""select * from Product where Price BETWEEN %F to %F"""
value = (50,100)
try:
cursor.execute(sql, value)
results = cursor.fetchall()()
print "ID Product Name Manufacturer Price"
For row in results:
PID = row[0]
PName = row[1]
PManu = row [2]
PPrice = row [3]
print " %s %s %s %s", % \ (PID, PName, PManu, PPrice)
print (cursor.rowcount; "Records Found")
except:
print ( "Error unable to Fetch data")
cursor.close()
db.close()
```
iii. import MySQLdb
```
db = MySQLdb.connect('localhost', 'Admin', 'Admin@pwd', 'cosmetics')
cursor= db.cursor()
sql= """select ClientName, City From Client"""
try:
cursor.execute(sql)
results = cursor.fetchall()
print "ClientName","City"
For row in results:
print "%s %s", % \ (row[0], row[1])
print (cursor.rowcount, "Records Found")
except:
print ("Error unable to Fetch data")
cursor. close()
db.close()
```
iv. import MySQLdb
```
db = MySQLdb.connect('localhost', Admin', Admin@pwd', 'cosmetics')
```

```
cursor= db. cursor (buffered= TRUE)
sql=""" select price From Product where
Manufacturer = %s"""
query_value= ('ABC',)
try:
cursor.execute(sql, query_value)
results — cursor.fetchall()
print "Price of Items by %s", % \ (query_value)
For row in results:
print rowT[0]
print (cursor.rowcount, "Items listed")
except:
print "Error unable to Fetch data"
cursor.close()
db.close()
```

v. import MySQLdb

```
db = MySQLDb.connect(Tocalhost', 'Admin', 'Admin@pwd', 'cosmetics')
cursor = db.cursor()
sql = """Select ProductName, Price *4 From Product"""
try:
cursor.execute(sql)
results = cursor. fetchall()
print "ProductName","Price"
For row in results:
print "%s %s" % \(row[0], row[1])
print (cursor.rowcount "Records found")
except:
print ("Unable to fetch data")
cursor. close()
db.close()
```

Section E

36. i. The most suitable place to install the server is Admin Centre because it has maximum number of computers. (using 80-20 rule).
   ii. Fibre optic cable
   iii. Switch
   iv. WAN because LAN and MAN cannot span more than 100 km.
37. i. SELECT * FROM WATCHES WHERE WATCHNAME LIKE '%TIME';

ii. SELECT WATCHNAME, PRICE FROM WATCH WHERE PRICE BETWEEN 5000 AND 15000;

iii. SELECT SUM (QTY STORE) FROM WATCHES WHERE TYPE LIKE 'Unisex';

iv. SELECT WATCHNAME, QTY SOLD FROM WATCHES W, SALE S WHERE W.WATCHID = S.WATCHID AND QUARTER = 1;

v.

max (price)	min(qty_store)
25000	100

vi.

quarter	sum(qty_sold)
1	15
2	30
3	45
4	15

vii.

watchname	price	type
HighFashion	7000	Unisex

viii.

watchname	qty_store	qty_sold	Stock
HighTime	100	25	75
Wave	200	30	170
LifeTime	150	40	110
Golden time	100	10	90

OR

i. SELECT * FROM MEMBER ORDER BY ISSUEDATE DESC

ii. SELECT DCODE, DTITLE FROM DVD WHERE DTYPE = 'Folk'

iii. SELECT DTYPE, COUNT(*) FROM DVD GROUP BY DTYPE

iv. SELECT NAME, ISSUEDATE FROM MEMBER, WHERE ISSUEDATE LIKE '2017%'

v. MIN (ISSUEDATE) 2016-12-13

vi. DISTINCT (DTYPE)
Folk
Classical
Rock

vii.

DCODE	name	DTITLE
R102	AGAM SINGH	A day in the life
F102	ARTH JOSEPH	Universal Soldier

C101	NISHA HANS	The Planets

viii. DTITLE

A day in the life