Marking Scheme

Class XII

Computer Science (083)

Time Allowed: 3 hours   **MM: 70**

Ques No	Question and Answers	Distribution of Marks	Total Marks
	SECTION A		
1	False	1 mark for correct answer	1
2	Option b 6,20	1 mark for correct answer	1
3	Option c -244.0	1 mark for correct answer	1
4	PYTHON-is-Fun	1 mark for correct answer	1
5	Option b 8,15	1 mark for correct answer	1
6	Option a PAN	1 mark for correct answer	1
7	Option b `del D1["Red"]`	1 mark for correct answer	1
8	Option b	1 mark for correct answer	1

	ceieP0		
9	Option d Statement 4	1 mark for correct answer	1

10	Option b
	
	     YELLOW*
	     WHITE*
	     BLACK*
	     RED*
		1 mark for correct answer	1

11	Option b Modulator	1 mark for correct answer	1
12	Option c global b	1 mark for correct answer	1
13	True	1 mark for correct answer	1
14	Option c A candidate key that is not a primary key is a foreign key.	1 mark for correct answer	1
15	circuit	1 mark for correct answer	1
16	Option c `seek()`	1 mark for correct answer	1

17	Option d A is false but R is True	1 mark for correct answer	1
18	Option b Both A and R are true but R is not the correct explanation for A	1 mark for correct answer	1

SECTION B

| 19 | (i)

 POP3 – Post Office Protocol 3

 URL – Uniform Resource Locator

 (ii)

 HTML(Hyper text mark Up language)

 • We use pre-defined tags
 • Static web development language – only focuses on how data looks
 • It use for only displaying data, cannot transport data
 • Not case sensistive

 XML (Extensible Markup Language)

 • we can define our own tags and use them
 • Dynamic web development language – as it is used for transporting and storing data
 • Case sensitive | ½ mark for each correct expansion

 1 mark for any one correct difference

 No mark to be awarded if only full form is given | 1+1=2 |
| 20 | ```
def revNumber(num):
    rev = 0
    rem = 0
    while num > 0:
``` | ½ mark for each | 2 |

[3]

| | ```
        rem =num %10
        rev = rev*10 + rem
        num = num//10
    return rev
print(revNumber(1234))
``` | correction made | |
| 21 | ```
PLACES={1:"Delhi",2:"London",3:"Paris",4:"New York",5:"Dubai"}
def countNow(PLACES):
    for place in PLACES.values():
        if len(place)>5:
            print(place.upper())
countNow(PLACES)
```

<div align="center">OR</div>

```
def lenWords(STRING):
    T=()
    L=STRING.split()
    for word in L:
        length=len(word)
        T=T+(length,)
    return T
```
**Note: Any other correct logic may be marked** | ½ mark for correct function header

½ mark for correct loop

½ mark for correct if statement

½ mark for displaying the output


½ mark for correct function header

½ mark for using split()

½ mark for adding to tuple

½ mark for return statement | 2 |

| | | | |
|---|---|---|---|
| 22 | 4*L<br><br>33*4<br><br>21*S<br><br>10*6 | ½ mark for each correct line of output | 2 |
| 23 | (i) L1.insert(2,200)<br><br>(ii) message.endswith('.') | 1 mark for each correct statement | 1+1=2 |
| 24 | SQL Command to add primary key:<br><br>ALTER TABLE Employee ADD EmpId INTEGER PRIMARY KEY;<br><br>As the primary key is added as the last field, the command for inserting data will be:<br><br>INSERT INTO Employee VALUES("Shweta","Production",26900,999);<br>                    OR<br>INSERT INTO Employee(EmpId,Ename,Department,Salary) VALUES(999,"Shweta","Production",26900); | 1 mark for correct ALTER TABLE command<br><br><br>1 mark for correct INSERT command | 2 |
| 25 | 10.0$20<br><br>10.0$2.0### | 1 mark for each correct line of output | 2 |
| <td colspan="4" align="center">**SECTION C**</td> | | | |
| 26 | ND-*34 | ½ mark for each correct character | 3 |
| 27 | | | |

| | |
|---|---|
| (i) | 1 mark for each correct output |

(i)

| COUNT(DISTINCT SPORTS) |
|---|
| 4 |

(ii)

| CNAME | SPORTS |
|---|---|
| AMINA | CHESS |

(iii)

| CNAME | AGE | PAY |
|---|---|---|
| AMRIT | 28 | 1000 |
| VIRAT | 35 | 1050 |

1*3=3

**28**

```
def test():
    fObj1 = open("Alpha.txt","r")
    data = fObj1.readlines()
    for line in data:
        L=line.split()
        if L[0]=="You":
            print(line)
    fObj1.close()
```

OR

1 mark for correctly opening and closing files

½ mark for correctly reading data

1 mark for correct loop and if statement

½ mark for displaying data

3

[6]

| | ```
def vowelCount():
    fObj = open("Alpha.txt","r")
    data = str(fObj.read())
    cnt=0
    for ch in data:
        if ch in "aeiouAEIOU":
            cnt=cnt+1
    print(cnt)
    fObj.close()
```<br><br>**Note: Any other correct logic may be marked** | | 1 mark for correctly opening and closing the files<br><br>½ mark for correctly reading data<br><br>1 mark for correct loop and if statement<br><br>½ mark for displaying the output. | |
| 29 | (i)<br><br>```
UPDATE Personal
SET Salary=Salary*0.5
WHERE Allowance IS NOT NULL;
```<br><br>(ii)<br><br>```
SELECT Name, Salary+Allowance AS
"Total Salary" FROM Personal;
```<br><br>(iii)<br><br>```
DELETE FROM Personal
WHERE Salary>25000
``` | | 1 mark for each correct query | 1*3=3 |

| 30 | ```python
travel = []
def Push_element(NList):
    for L in NList:
        if L[1] != "India" and L[2]<3500:
            travel.append([L[0],L[1]])


def Pop_element():
    while len(travel):
        print(travel.pop())
    else:
        print("Stack Empty")
``` | 1 ½ marks for each function | 3 |

SECTION D

31	a) Bus Topology b) Switch c) Admin block, as it has maximum number of computers. d) Microwave e) Firewall	1 mark for each correct answer	1*5=5
32	(i) r+ mode: • Primary function is reading • File pointer is at beginning of file • if the file does not exist, it results in an error w+ mode:	1 mark for each correct difference (minimum two differences should be given)	2+3=5

- primary function is writing

- if the file does not exist, it creates a new file.

- If the file exists, previous data is overwritten

- File pointer is at the beginning of file

(ii)

```python
def copyData():
    fObj = open("SPORT.DAT", "rb")
    fObj1 = open("BASKET.DAT","wb")
    cnt=0
    try:
        while True:
            data = pickle.load(fObj)
            print(data)
            if data[0] == "Basket Ball":
                pickle.dump(data,fObj1)
                cnt+=1
    except:
        fObj.close()
        fObj1.close()
    return cnt
```

½ mark for correctly opening and closing files

½ mark for correct try and except block

½ mark for correct loop

1 mark for correctly copying data

½ mark for correct return statement

OR

(Only for option ii)

```python
def Searchtype(mtype):
    fObj = open("CINEMA.DAT", "rb")
    try:
        while True:
            data = pickle.load(fObj)
            if data[2] == mtype:
                print("Movie number:",data[0])
                print("Movie Name:",data[1])
                print("Movie Type:",data[2])
    except EOFError:
        fObj.close()
```

½ mark for correctly opening and closing files

½ mark for correct try and except block

½ mark for correct loop

	Note: Any other correct logic may be marked	½ mark for correct if statement 1 mark for correctly displaying data	
33	(i) Domain is a set of values from which an attribute can take value in each row. For example, roll no field can have only integer values and so its domain is a set of integer values (ii)	½ mark for correct definition ½ mark for correct example ½ mark for importing correct module	1+4=5

```python
import mysql.connector as mysql
con1 = mysql.connect(host="localhost",user="root", password="tiger", database="sample2023")
mycursor=con1.cursor()
rno = int(input("Enter Roll Number:: "))
name = input("Enter the name:: ")
DOB = input("Enter date of birth:: ")
fee= float(input("Enter Fee:: "))
query = "INSERT into student values({},'{}','{}',{})".format(rno,name,DOB,fee)
mycursor.execute(query)
con1.commit()
print("Data added successfully")
con1.close()
```

	Note: Any other correct logic may be marked	1 mark for correct connect() ½ mark for correctly accepting the input 1 ½ mark for correctly executing the query ½ mark for correctly using commit()	

SECTION E

34	(i)	1 mark for each correct query	1*4=4
	SELECT PName, BName FROM PRODUCT P, BRAND B WHERE P.BID=B.BID;		
	(ii)		
	DESC PRODUCT;		
	(iii)		
	SELECT BName, AVG(Rating) FROM PRODUCT P, BRAND B WHERE P.BID=B.BID GROUP BY BName HAVING BName='Medimix' OR BName='Dove';		
	(iv)		
	SELECT PName, UPrice, Rating FROM PRODUCT ORDER BY Rating DESC;		
| 35 | ```python
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()
``` | ½ mark for accepting data correctly<br><br>½ mark for opening and closing file<br><br>½ mark for writing headings<br><br>½ mark for writing row | 4 |

| ```python
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()
``` | ½ mark for opening and closing file<br><br>½ mark for reader object<br><br>½ mark for print heading<br><br>½ mark for printing data |