

Class XII
Computer Science (083)
Marking Scheme

Time Allowed: 3 hours

MM: 70

<u>Ques No</u>	Question and Answers	Distribution of Marks	Total Marks
<u>SECTION A</u>			
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 <code>del D1["Red"]</code>	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 WHITE* BLACK*	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

	(ii) https (Hyper Text Transfer Protocol Secure) is the protocol that uses SSL (Secure Socket Layer) to encrypt data being transmitted over the Internet. Therefore, https helps in secure browsing while http does not.	1 mark for correct difference.	
20	<pre> def revNumber (num) : rev = 0 rem = 0 while num > 0: rem = num %10 rev = rev*10 + rem num = num//10 return rev print (revNumber (1234)) </pre>	½ mark for each correction made	2
21	<pre> 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) </pre> OR	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()	2

	<pre>def lenWords (STRING) : T=() L=STRING.split() for word in L: length=len(word) T=T+(length,) return T</pre> <u>Note: Any other correct logic may be marked</u>	½ mark for adding to tuple ½ mark for return statement	
22	4*L 33*4 21*S 10*6	½ mark for each correct line of output	2
23	(i) L1.insert(2,200) (ii) message.endswith('.') OR <pre>import statistics print (statistics.mode(studentAge))</pre>	1 mark for each correct statement 1 mark for correct import statement 1 mark for correct command with mode() and print()	1+1=2
24	SQL Command to add primary key: <pre>ALTER TABLE Employee ADD EmpId INTEGER PRIMARY KEY;</pre>	1 mark for correct ALTER TABLE command	2

	As the primary key is added as the last field, the command for inserting data will be: <pre>INSERT INTO Employee VALUES ("Shweta", "Production", 26900, 999);</pre> Alternative answer: <pre>INSERT INTO Employee (EmpId, Ename, Department, Salary) VALUES (999, "Shweta", "Production", 26900);</pre> OR To delete the attribute, category: <pre>ALTER TABLE Sports DROP category;</pre> To add the attribute, TypeSport <pre>ALTER TABLE Sports ADD TypeSport char(10) NOT NULL;</pre>	1 mark for correct INSERT command 1 mark for correct ALTER TABLE command with DROP 1 mark for correct ALTER TABLE command with ADD	
25	10.0\$20 10.0\$2.0###	1 mark for each correct line of output	2
<u>SECTION C</u>			
26	ND-*34	½ mark for each correct character	3
27			
	(i) COUNT(DISTINCT SPORTS)	1 mark for each correct output	1*3=3

	4 (ii) <table><tr><th>CNAME</th><th>SPORTS</th></tr><tr><td>AMINA</td><td>CHESS</td></tr></table> (iii) <table><tr><th>CNAME</th><th>AGE</th><th>PAY</th></tr><tr><td>AMRIT</td><td>28</td><td>1000</td></tr><tr><td>VIRAT</td><td>35</td><td>1050</td></tr></table>	CNAME	SPORTS	AMINA	CHESS	CNAME	AGE	PAY	AMRIT	28	1000	VIRAT	35	1050		
CNAME	SPORTS															
AMINA	CHESS															
CNAME	AGE	PAY														
AMRIT	28	1000														
VIRAT	35	1050														
28	<pre>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()</pre> 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 1 mark for correctly opening and closing the files	3													

	<pre> 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() </pre> <u>Note: Any other correct logic may be marked</u>	½ mark for correctly reading data 1 mark for correct loop and if statement ½ mark for displaying the output.	
29	(i) <pre> UPDATE Personal SET Salary=Salary + Salary*0.5 WHERE Allowance IS NOT NULL; </pre> (ii) <pre> SELECT Name, Salary + Allowance AS "Total Salary" FROM Personal; </pre> (iii) <pre> DELETE FROM Personal WHERE Salary>25000 </pre>	1 mark for each correct query	1*3=3

30	<pre> 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") </pre>	1 ½ marks for each function	3
----	---	-----------------------------	---

SECTION D

31	(i) <pre> SELECT PName, BName FROM PRODUCT P, BRAND B WHERE P.BID=B.BID; </pre> (ii) <pre> DESC PRODUCT; </pre> (iii) <pre> SELECT BName, AVG(Rating) FROM PRODUCT P, BRAND B WHERE P.BID=B.BID GROUP BY BName HAVING BName='Medimix' OR BName='Dove'; </pre> (iv) <pre> SELECT PName, UPrice, Rating FROM PRODUCT ORDER BY Rating DESC; </pre>	1 mark for each correct query	1*4=4
----	---	-------------------------------	-------

32	<pre> 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() </pre>	½ mark for accepting data correctly ½ mark for opening and closing file ½ mark for writing headings ½ mark for writing row ½ mark for opening and closing file ½ mark for reader object ½ mark for print heading ½ mark for printing data	4
----	---	---	---

SECTION E

33	a) <pre> graph TD Admin[Admin] --- Bus subgraph Bus direction LR Engineering[ENGINEERING] Media[MEDIA] Business[BUSINESS] end Bus --- Engineering Bus --- Media Bus --- Business </pre>	1 mark for each correct answer	1*5=5
----	--	--------------------------------	-------

	<pre>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</pre> OR (i) Text files: • Extension is .txt • Data is stored in ASCII format that is human readable • Has EOL character that terminates each line of data stored in the text files Binary Files • Extension is .dat • Data is stored in binary form (0s and 1s), that is not human readable. (ii)	½ mark for correct return statement ½ mark for correctly opening and closing files ½ mark for correct try and except block ½ mark for correct loop ½ mark for correct if statement 1 mark for correctly displaying data	
--	--	---	--

	<pre> 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() </pre> <u>Note: Any other correct logic may be marked</u>		
35	(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) <pre> 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() </pre> <u>Note: Any other correct logic may be marked</u>	½ mark for correct definition ½ mark for correct example ½ mark for importing correct module 1 mark for correct connect() ½ mark for correctly accepting the input 1 ½ mark for correctly	1+4=5

	<u>OR</u> (i) All keys that have the properties to become a primary key are candidate keys. The candidate keys that do not become primary keys are alternate keys. (ii) <pre> import mysql.connector as mysql con1 = mysql.connect(host="localhost",user="root", password="tiger", database="sample2023") mycursor=con1.cursor() query = "SELECT * FROM student where fee>{}".format(5000) mycursor.execute(query) data=mycursor.fetchall() for rec in data: print(rec) con1.close() </pre>	executing the query ½ mark for correctly using commit() 1 mark for correct difference ½ mark for importing correct module 1 mark for correct connect() 1 mark for correctly executing the query ½ mark for correctly using fetchall() 1 mark for correctly	
--	--	--	--

		displaying data	
--	--	--------------------	--