www.Learnpython4cbse.com

XII - Computer Science (083)

Marking Scheme

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

Ques No	Question and Answers	Distribution of Marks	Total Marks
	SECTION A		
1	True	1 mark for correct answer	1
2	Option d delete	1 mark for correct answer	1
3	Option b 18	1 mark for correct answer	1
4	Option d ('BHASA', ' ', 'SANGAM@75')	1 mark for correct answer	1
5	Option b 15,50	1 mark for correct answer	1
6	Option a PAN	1 mark for correct answer	1
7	Option a `r g b`	1 mark for correct answer	1
8	Option b 2@tr	1 mark for correct answer	1

9	Option b Statement 4	1 mark for correct answer	1
10	Option b Wait#Stop#	1 mark for correct answer	1
11	Option b SMTP	1 mark for correct answer	1
12	Option a 21 7	1 mark for correct answer	1
13	True	1 mark for correct answer	1
14	Option b It is case sensitive	1 mark for correct answer	1
15	Packet	1 mark for correct answer	1
16	Option c `seek()`	1 mark for correct answer	1
17	Option a Both A and R are true but R is the correct explanation for A	1 mark for correct answer	1

[2]

18	Option a Both A and R are true but R is the correct explanation for A	1 mark for correct answer	1
	SECTION B		
19	(i) SMTP – Simple Mail Transfer Protocol IMAP – Internet Message Access Protocol (ii) Active hubs amplify the incoming electric signal, whereas passive hubs do not amplify the electric signal. (Any other valid difference may be considered) **OR** (i) A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network. (ii) Hub is an electronic device that connects several nodes to form a network and redirect the received information to all the nodes in a broadcast mode. Whereas Switch is an intelligent device that connects several nodes to form a network and redirect the received information only to the intended node(s). (Any other valid difference may be considered)	½ mark for each correct expansion 1 mark for any one correct difference 1 mark for correct definition 1 mark for any one correct difference	1+1=2
20	`def table ():` `  n=int (input ("Enter number which table U need: "))` `  for i in range (1,11):` `  print ("able of Enter no=",i*n)` `table ()`	½ mark for each correction made	2

21	SUBJECT={1:"Hindi",2:"Physics",3:"Chemistry",4:"CS",5:"MATH"} def countMy (SUBJECT): for S in SUBJECT.values(): if len(S)>5: print(S.upper()) countMy() OR def lenLines (STRING): t=() L=STRING.split() for line in L: length=len(line) 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	(22, 44, 66)	1½ mark for each correct digit ½ mark for parenthesis	2

23	(i) L1.insert(1,100) (ii) S1.isdigit() OR pop() function removes the lastvalue and returns the same. >>>L=[10,20,30,20] >>> L.pop () 20 The *remove()* method removes thefirst matching value from the list. >>>L.remove (20) [10, 30, 20]	1 mark for each correct statement 1 mark for correct difference and 1 mark for suitable example	1+1=2
24	SQL Command to add primary key: select * from student where fee IS NULL **OR** DDL : CREATE, ALTER DROP DML: INSERT UPDATE DELETE	2 mark for correct Command 1 mark for each correct DDL & DML Categorized commands	2
25	-22 # 756 # -9 # 230 #	½ mark for each correct number and ½ mark for each correct # symbol	2
	SECTION C		
26	['DelhiDelhi', 'JaipurJaipur', 'AgraAgra', 'SuratSurat', 'MumbaiMumbai', 'BhopalBhopal']	½ mark for each correct output	3

27	(a) **Item Name** White lotus Comfort Zone Wood Comfort (b) **Dateofstock** 13/12/2001 22/02/2002 20/02/2003 (c) **Type** **Sum(Price)** Double Bed 80000 Baby Cot 30500 Office Table 43000 Sofa 57500 Dining Table 11500	1 mark for each correct output.	1*3=3
28	def SHOWWORD () : c=0 file=open('STORY.TXT,'r') line = file.read() word = line.split() for w in word: if len(w)<5: print(w) file.close() OR def count H( ): f = open ("para.txt" , "r") lines =0 L=f. readlines () for i in L: if i [0]== 'H': lines +=1 print ("No. of lines are: " , lines)	(½ Mark for opening the file) (½ Mark for reading line and/or splitting) (½ Mark for checking condition) (½ Mark for printing word)	3
29	(i) UPDATE EMP SET Salary=Salary + Salary*0.10 WHERE Allowance IS NOT NULL; (ii) SELECT Name, Salary + Allowance AS "Total Salary" FROM EMP; (iii) DELETE FROM EMP WHERE Salary>40000;	1 mark for each correct query	1*3=3

30

```
N=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38]
def PUSHEl(S,N):
    S.append(N)
def POPEl(S):
    if S!=[]:
        return S.pop()
    else:
        return None
ST=[]
for k in N:
    if k%4==0:
        PUSHEl(ST,k)
while True:
   if ST!=[]:
       print(POPEl(ST),end=" ")
   else:
       break
```

1½ marks for each Push and Pop operation

3

SECTION D

31

(i)

3

(ii)

1

1

2

(iii)

Dname	Pname
PARESH	Lal singh
MANISH	Arjun
AKASH	Narender
KUMAR	Mehul
PARESH	Naveen
MANISH	Amit

(iv)

Manish

1 mark for each correct output

1*4=4

[7]

32	 `import csv` `def createcsv():` `  f=open("result.csv","w", newline="")` `  w=csv.writer(f)` `  w.writerow([1,'Anil',40,34,90,""])` `  w.writerow([2,'Sohan',78,34,90,""])` `  w.writerow([3,'Kamal',40,45,9,""])` `  f.close()` `import csv` `def copycsv():` `  f=open("result.csv","r")` `  f1=open("final.csv","w",newline="")` `  w1=csv.writer(f1)` `  r=csv.reader(f)` `  for x in r:` `  x[5]=int(x[2])+int(x[3])+int(x[4])` `  w1.writerow(x)` `f.close()` `f1.close()`	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	(i) M/s Computer Solutions should install its server in finance block as it is having maximum number of computers. (ii) Any suitable layout (iii) Satellite Link. (iv) Switch. (v) LAN	1 Mark of each correct answer	1*5=5

34	(i) **rb+** Opens a file for both reading and writing in binary format. (+) the file pointer will be at the beginning of the file. **wb+** Opens a file for both reading and writing in binary format. Overwrites the existing file If the file exists. If the file does not exist, creates a new file for reading or writing. (ii) ``` def Readfile(): s=open("Employee.dat" , "rb+") try: while True: r=pickle.load(s) if r[2]>=20000 and r[2]<=30000: print(r) except: print("end of file") ``` OR (i) In pickle module, dump () method is used to convert (pickling) Python objects for writing data in a binary file Whereas the load () function is used to read data from a binary file or file object. (ii) ``` import pickle as p L=[] with open('emp.dat','rb') as f: L=p.load(f) for r in L: if r[2]>5000: print("name=",r[0]) print("designation=",r[1]) print("salary=",r[2]) ``` Note: Any other correct logic may be marked	1 mark for <u>each correct</u> difference ½ mark for correctly opening and closing files ½ mark for correct loop ½ mark for correct split 1 mark for correctly reading / writing data ½ mark for printing data	2+3=5

[9]

35	(i) A table can only have one primary key, but it can have multiple candidate key in a database. (any suitable example) (ii) import mysql.connector mydb=mysql.connector.connect(host="localhost",user="root",passwd="admin",database="SCHOOL") mycursor=mydb.cursor() while 1: ch=int(input("enter -1 to exit / any other no to insert record into student table")) if ch==-1: break eno=int(input("Enter Employee no")) ename=input("Enter Employee Name") edept=input("Enter dept name") sal=int(input("Enter salary")) mycursor.execute("insert into EMP values ('"+str(eno)+"','"+ ename+"','" +edept + "','"+str(sal)+"')") mydb.commit() for x in mycursor: print(x) **OR** **(i)** **Degree:** The total number of attributes which in the relation is called the degree of the relation. **Cardinality:** Total number of rows present in the Table. (any suitable example) (ii) import mysql.connector mydb=mysql.connector.connect(host="localhost",user="root",passwd="admin",database="SCHOOL") mycursor=mydb.cursor() mycursor.execute("alter table emp add (bonus int(3))") mycursor.execute("desc emp") for x in mycursor: print(x) Note: Any other correct logic may be marked	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 displaying data	1+4=5

[10]