Common Pre-Board Examination Chandigarh Region 2020-21

Class: XII   **Sub: COMPUTER SCIENCE**

Max. Marks: 70   **Time: 3 HRS**

Marking Scheme

General Instructions:

1. This question paper contains two parts A and B. Each part is compulsory.

2. Both Part A and Part B have choices.

3. Part-A has 2 sections:

   a. Section – I is short answer questions, to be answered in one word or one line.

   b. Section – II has two case studies questions. Each case study has 4 case-based subparts. An   examinee is to attempt any 4 out of the 5 subparts.

4. Part - B is Descriptive Paper.

5. Part- B has three sections

   a. Section-I is short answer questions of 2 marks each in which two question have internal   options.

   b. Section-II is long answer questions of 3 marks each in which two questions have internal   options.

   c. Section-III is very long answer questions of 5 marks each in which one question has internal   option.

6. All programming questions are to be answered using Python Language only.

Question No.	Part A	Marks
	Section-I Select the most appropriate option out of the options given for each question. Attempt any 15 questions from question no 1 to 21.	
1	(i) // **(1 mark for correct answer)**	1
2	(i) math(ii) random **(1/ 2 mark for each module)**	1
3	(ii) fp.tell( )	1

	(1 mark for each correct type)	
4	(i) Day={1:'monday',2:'tuesday',3:'wednesday'} (1markforcorrectanswer)	1
5	Swap(num1=100,num2=200) 1markforcorrectanswer)	1
6	True	1
7	Using the keyword global	1
8	LIFO	1
9	mputer S	1
10	i) [10,20,30] ii)[10,20,30,60]	1
11	Bandwidth	1
12	a) Simple Mail Transfer Protocol b) Global System for Mobile Communication	1
13	Wired transmission medium – Optical fiber cable Type of network – MAN.	1
14	C. A system designed to prevent unauthorized access.	1
15	d) gateway	1
16	b) describe	1
17	view	1
18	ii)distinct	1
19	ii)having	1
20	iii) rowcount	1
21	True	1
	SECTION - II **Both the Case study based questions are compulsory. Attempt any 4 sub parts from each question. Each question carries 1 mark**	
22	i) id ii) Ans. select avg(sal) from employee;	1 1

	iii) Ans. select designation, count(*) from employee group by designation;	1
	iv) Ans. select designation, count(*), sum(sal) from employee group by designation having count(*)>1;	1
	v) Degree : 4 Cardinality : 6	1
	1 mark for each correct answer (ANY FOUR)	

23	1 mark for each correct answer (ANY FOUR)
	 (a) Line 1 : csv
	 (b) Line 2 : a
	 (c) Line 3 : reader
	 (d) Line 4 : close()
	 (e) Line 5 : Arjun 123@456
	          Arunima aru@nima
	          Frieda myname@FRD
		1
		 1
		 1
		 1
		 1

	Part - B	
	Section - I	
24	(i) True (ii) 17 1 Mark for each correct answer	2
25	2 marks for correct Answer.	2
26	2 marks for correct Answer	2
27	Code Division Multiple Access Hyper Text Transfer Protocol Extensible Markup Language Uniform Resource Locator (½ Marks for each correct answer)	2

28	To=30
	 for K in range(0,To):
	      if K%4==0:
		2

	print(K*4) <u>else</u>: print(K+3) 2 marks for correct error detection	
29	Possible outputs : ii) , iii) randint will generate an integer between 2 to 4 which is then raised to power 2, so possible outcomes can be 4,9 or 16	2
30	2 marks for correct answer	2
31	2 marks for correct answer	2
32	2 marks for correct answer	2
33	exam$$*CBSE*COM 2 marks for correct output	2
	Section - II	

34	def sum(numbers):
	
	 total = 0
	
	 for x in numbers:
	
	 total += x
	
	 return total
	
	 (2 Marks for Logic 1 mark for function definition)
		3

35	def TwoCharWord():
	
	 f = open('poem.txt') count = 0
	
	 for line in f:
	
	 words = line.split()
	
	 for w in words:
	
	 if len(w)==2:
	
	 print(w,end=' ')
	
	 (2 Marks for Logic 1 mark for function definition)
	
	 or
		3

	``
def COUNT():
    f = open('REPEATED.txt')
    count = 0
    for line in f:
        words = line.split()
        for w in words:
            if w.lower()=='catholic' or w.lower()=='mother':
                count+=1
print('Count of Catholic,mother is',count)
```

(2 Marks for Logic 1 mark for function definition) | |
| 36 | (i)
DISTINCT TID
101
103
102
104
105
(1 mark for correct Answer)


(ii)
TIDCOUNT(*)MIN(FEES)
101 2 12000
(1 mark for correct Answer)


(iii)
COUNT(*)SUM(FEES)
4 65000
(1 mark for correct Answer)

| 3 |
| 37 | defMakePush(Package): | 3 |

```
        a=int(input("enterpackagetitle:"))
    Package.append(a)


defMakePop(Package):
    if(Package==[]):

            print("Stackempty")
    else:

            print("Deletedelement:",Package.pop())
```

(½markforMakePush() header)

(½markfor addingvalueinlist)

(½markforMakePop() header)

(½markforcheckingemptystack and displaying"Stackempty")

(½markfordisplayingthevaluetobedeleted)(½markfor deletingvaluefromlist)

3 marks for correct answer.

	Section - III	
38	(i) 1 Mark for correct Layout. (ii) Research Lab (1 Mark) (iii) 1 Mark for correct Justification. (iv) Antivirus/ Firewall (1 Mark for Correct Answer) (v) 1 Mark for correct Justification.	5

| 39 | (i) SELECT M_Company, M_Name, M_Price FROM MobileMasterORDER BY M_Mf_Date DESC;
(½ mark for correct SELECT)
(½ mark for correct ORDER BY)

(ii) SELECT * FROM MobileMaster WHERE M_Name LIKE "S%" or M_Name LIKE "%a";
(½ mark for correct SELECT)
(½ mark for correct WHERE clause)

(iii) SELECT M_Supplier, M_Qty FROM MobileStock WHERE M_Id <>"MB003";
(½ mark for correct SELECT)
(½ mark for correct WHERE clause)

(iv) SELECT M_Company FROM MobileMaster WHERE M_PriceBETWEEN 3000AND 5000;
(½ mark for correct SELECT)
(½ mark for correct BETWEEN clause)

(v) SELECT M_Id, SUM(M_Qty) FROM MobileStock GROUP BY M_Id;
(½ mark for correct SELECT)
(½ mark for correct Group By) | 5 |
| 40 | 1. 2.5 marks for first part
½ mark for import
½ mark for opening a file
1 marks for input and making object
½ for dump command
2. 2.5 marks for 2 part
½ mark for import
½ mark for opening a file
½ marks for try and except or any other loop | 5 |

½ for load command

½ mark for display

or

```
import pickle
def search():
    f=open("emp.dat","rb")
    while True:
        try:
            d=pickle.load(f)
            if(d['sal']>=25000 and d['sal']<=30000):
                print(d)
        except EOFError:
            break
    f.close()
```

½ mark for import

½ mark for function

½   mark for opening a file

2 marks for load and matching with if

½ mark for closing a file

END OF THE MARKING SCHEME