# MARKING SCHEME

## Class: XII Session: 2024-25

## Computer Science (083)

**Time allowed: 3 Hours** **Maximum Marks: 70**

| Q No. | SECTION A (21X1=21) | Marks |
|---|---|---|
| 1. | False<br>*(1 mark for correct answer)* | (1) |
| 2. | (A) #THONPROGRAM<br>*(1 mark for correct answer)* | (1) |
| 3. | (A) not (True) and False<br>*(1 mark for correct answer)* | (1) |
| 4. | (B) ['I', 'ter', 'atio', 'al']<br>*(1 mark for correct answer)* | (1) |
| 5. | ce lo<br>*(1 mark for correct answer)* | (1) |
| 6. | (B) False<br>*(1 mark for correct answer)* | (1) |
| 7. | (B) print(my_dict['apple', 'banana'])<br>*(1 mark for correct answer)* | (1) |
| 8. | (B) Removes the first occurrence of value x from the list<br>*(1 mark for correct answer)* | (1) |
| 9. | (C) 3<br>*(1 mark for correct answer)* | (1) |
| 10. | file.seek(0) ( OR file.seek(0,0) )<br>*(1 mark for correct answer)* | (1) |
| 11. | False<br>*(1 mark for correct answer)* | (1) |
| 12. | (C) 12#15%<br>*(1 mark for correct answer)* | (1) |
| 13. | Alter (or Alter Table)<br>*(1 mark for correct answer)* | (1) |

www.careerindia.com

| | | |
|---|---|---|
| 14. | (A) Details of all products whose names start with 'App'<br>*(1 mark for correct answer)* | (1) |
| 15. | (D) CHAR<br>*(1 mark for correct answer)* | (1) |
| 16. | (B) count()<br>*(1 mark for correct answer)* | (1) |
| 17. | (B) FTP<br>*(1 mark for correct answer)* | (1) |
| 18. | (B) Gateway<br>*(1 mark for correct answer)* | (1) |
| 19. | (B) Packet Switching<br>*(1 mark for correct answer)* | (1) |
| 20. | (C) A is True but R is False.<br>*(1 mark for correct answer)* | (1) |
| 21. | (C) A is True but R is False.<br>*(1 mark for correct answer)* | (1) |

| Q No. | SECTION B (7 X 2 =14) | Marks |
|---|---|---|
| 22. | A mutable object can be updated whereas an immutable object cannot be updated.<br>Mutable object: [1,2] or {1:1,2:2} (Any one)<br>Immutable object: (1,2) or '123' (Any one)<br>*(1 mark for correct difference)*<br>*(½ x 2 = 1 Mark for selecting correct objects)* | (2) |
| 23. | (I) Arithmetic operators: +,-<br>(II) Relational operators: >, >=<br>*(½ x 4 = 2 Marks for each correct operator)* | (2) |
| 24. | (I)<br>A) L1.count(4)<br>OR<br>B) L1.sort()<br>*(1 mark for correct answer)* | (2) |

www.careerindia.com

| | | |
|---|---|---|
| | (II)<br>A) L1.extend(L2)<br>OR<br>B) L2.reverse()<br>*(1 mark for correct answer)* | |
| **25.** | (A), (C)<br>*(½ x 2 = 1 Mark)*<br>Minimum and maximum possible values of the variable b: 1,6<br>*(½ x 2 = 1 Mark)* | (2) |
| **26.** | def swap_first_last(tup**):**<br>if len(tup) < 2:<br>**return tup**<br>new_tup = (tup[-1],) + tup[1:-1] + (tup**[0],)**<br>return new_tup<br><br>result = swap_first_last((1, 2, 3, 4))<br>print("Swapped **tuple:", result)**<br><br>(½ mark each for correcting 4 mistakes) | (2) |
| **27.** | (I)<br>A) UNIQUE<br>**OR**<br>B) NOT NULL<br>*(1 mark for correct answer)*<br>(II)<br>A) ALTER TABLE MOBILE DROP PRIMARY KEY;<br>OR<br>B) ALTER TABLE MOBILE ADD PRIMARY KEY (M_ID);<br>*(1 mark for correct answer)* | (2) |
| **28.** | A) Advantage: Network extension is easy.<br>Disadvantage: Failure of switch/hub results in failure of the network.<br>*(1 mark for correct Advantage)*<br>*(1 mark for correct Disadvantage)*<br>**OR** | (2) |

www.careerindia.com

| | | |
|---|---|---|
| | B) SMTP: Simple Mail Transfer Protocol.<br>SMTP is used for sending e-mails from client to server.<br>*(1 mark for correct expansion)*<br>*(1 mark for correct usage)* | |

| Q No. | SECTION C (3 X 3 = 9) | Marks |
|---|---|---|
| **29.** | (A)<br>def show():<br>f=open("Email.txt",'r')<br>data=f.read()<br>words=data.split()<br>for word in words:<br>if '@cmail' in word:<br>print(word,end=' ')<br>f.close()<br>*(½ mark for correct function header)*<br>*(½ mark for correctly opening the file)*<br>*(½ mark for correctly reading from the file)*<br>*(½ mark for splitting the text into words)*<br>*(1 mark for correctly displaying the desired words)*<br>**OR**<br>(B)<br>def display_long_words():<br>with open("Words.txt", 'r') as file:<br>data=file.read()<br>words=data.split()<br>for word in words:<br>if len(word)>5:<br>print(word,end=' ')<br>*(½ mark for correct function header)*<br>*(½ mark for correctly opening the file)*<br>*(½ mark for correctly reading from the file)*<br>*( ½ mark for splitting the text into words)*<br>*(1 mark for correctly displaying the desired words)* | (3) |

www.careerindia.com

**30.** (A)

(I)

```
def push_book(BooksStack, new_book):
        BooksStack.append(new_book)
```

(II)

```
def pop_book(BooksStack):
    if not BooksStack:
        print("Underflow")
    else:
        return(BookStack.pop())
```

(III)

```
def peep(BooksStack):
    if not BooksStack:
        print("None")
    else:
        print(BookStack[-1])
```

*(3x1 mark for correct function body; No marks for any function header as it was a part of the question)*

**OR**

(B)

```
def push_even_numbers(N):
        EvenNumbers = []
        for num in N:
            if num % 2 == 0:
                EvenNumbers.append(num)
        return EvenNumbers


VALUES = []

for i in range(5):
    VALUES.append(int(input("Enter an integer: ")))

EvenNumbers = push_even_numbers(VALUES)

def pop_even():
    if not EvenNumbers:
        print("Underflow")
    else:
        print(EvenNumbers.pop())

pop_even()
```

(3)

www.careerindia.com

| | | |
|---|---|---|
| | def Disp_even():<br>if not EvenNumbers:<br>print("None")<br>else:<br>print(EvenNumbers[-1])<br>Disp_even()<br>*(1/2 for identifying even numbers)*<br>*(1/2 mark for correctly adding data to stack)*<br>*(1/2 mark for correctly poping data on the stack and 1/2 mark for checking condition)*<br>*(1/2 mark for correctly displaying the data with none)*<br>*(1/2 mark for function call statements)* | |
| 31. | (A) 15@<br>7@<br>9<br><br>OR<br><br>(B) 1 #2 #3#<br>1 #2 #3 #<br>1 #<br><br>*(1 mark for each correct line of output)*<br>*(deduct ½ mark for not printing @/#)* | (3) |

| Q No. | SECTION D (4 X 4 = 16) | Marks |
|---|---|---|
| 32. | (A)<br>(I) select Product, sum(Quantity) from orders group by product having sum(Quantity)>=5;<br>(II) select * from orders order by Price desc;<br>(III) select distinct C_Name from orders;<br>(IV) select sum(price) as total_price from orders where Quantity IS NULL;<br><br>*(4 x 1 mark for each correct query)*<br>**OR**<br>(B)<br>(I)<br><br>C_Name \| Total_Quantity<br>--------- \|----------<br>Jitendra \| 1<br>Mustafa \| 2<br>Dhwani \| 1 | (4) |

www.careerindia.com

(II)

```
O_Id | C_Name | Product        | Quantity  | Price
-----  |--------     |------------      |----------    |-------
1002 | Mustafa  | Smartphone | 2            | 10000
1003 | Dhwani   | Headphone  | 1            | 1500
```

(III)

```
O_Id | C_Name    | Product        | Quantity | Price
-----        |----------      |------------      |----------    |-------
1001       | Jitendra    | Laptop          | 1            | 12000
1002       | Mustafa    | Smartphone | 2            | 10000
1003       | Dhwani     | Headphone  | 1            | 1500
```

(IV)

```
MAX(Price)
-----------
12000
```

*(4 x 1 mark for each correct output)*

33. (I)

```
def show():
    import csv
    f=open("happiness.csv",'r')
    records=csv.reader(f)
    next(records, None) #To skip the Header row
    for i in records:
        if int(i[1])>5000000:
            print(i)
    f.close()
```

*(½ mark for opening in the file in right mode)*
*(½ mark for correctly creating the reader object)*
*(½ mark for correctly checking the condition)*
*(½ mark for correctly displaying the records)*

(4)

(II)

```
def Count_records():
    import csv
    f=open("happiness.csv",'r')
    records=csv.reader(f)
    next(records, None) #To skip the Header row
    count=0
    for i in records:
        count+=1
    print(count)
    f.close()
```

www.careerindia.com

| | | |
|---|---|---|
| | *(½ mark for opening in the file in right mode)*<br>*(½ mark for correctly creating the reader object)*<br>*(½ mark for correct use of counter)*<br>*(½ mark for correctly displaying the counter)*<br>**Note (**for both parts (I) and (II)**):**<br>*(i) Ignore* **import csv** *as it may be considered the part of the complete program, and there is no need to import it in individual functions.*<br>*(ii) Ignore next(records, None) as the file may or may not have the Header Row.* | |
| 34. | (I) Select * from FACULTY natural join COURSES where Salary<12000;<br>Or<br>Select * from FACULTY, COURSES where Salary<12000 and facuty.f_id=courses.f_id;<br>(II) Select * from courses where fees between 20000 and 50000;<br>(III) Update courses set fees=fees+500 where CName like '%Computer%';<br>(IV)<br>(A) Select FName, LName from faculty natural join courses where Came="System Design";<br>Or<br>Select FName, LName from faculty, courses where Came="System Design" and facuty.f_id=courses.f_id;<br>OR<br>(B) Select * from FACULTY, COURSES;<br>*(4x1 mark for each correct query)* | (4) |
| 35. | def AddAndDisplay():<br>import mysql.connector as mycon<br>mydb=mycon.connect(host="localhost",user="root",<br>passwd="Pencil",database="ITEMDB")<br>mycur=mydb.cursor()<br>no=int(input("Enter Item Number: "))<br>nm=input("Enter Item Name: ")<br>pr=float(input("Enter price: "))<br>qty=int(input("Enter qty: "))<br>query="INSERT INTO stationery VALUES ({},'{}',{},{})"<br>query=query.format(no,nm,pr,qty)<br>mycur.execute(query)<br>mydb.commit()<br>mycur.execute("select * from stationery where price>120")<br>for rec in mycur:<br>print(rec) | (4) |

www.careerindia.com

| | | |
|---|---|---|
| | *(½ mark for correctly importing the connector object)*<br>*(½ mark for correctly creating the connection object)*<br>*(½ mark for correctly creating the cursor object)*<br>*(½ mark for correctly inputting the data)*<br>*(½ mark for correct creation of first query)*<br>*(½ mark for correctly executing the first query with commit)*<br>*(½ mark for correctly executing the second query)*<br>*(½ mark for correctly displaying the data)* | |

| Q No. | SECTION E (2 X 5 = 10) | Marks |
|---|---|---|
| 36. | (I)<br>import pickle<br><br>def input_candidates():<br>candidates = []<br>n = int(input("Enter the number of candidates you want to add: "))<br>for i in range(n):<br>candidate_id = int(input("Enter Candidate ID: "))<br>candidate_name = input("Enter Candidate Name: ")<br>designation = input("Enter Designation: ")<br>experience = float(input("Enter Experience (in years): "))<br>candidates.append([candidate_id, candidate_name, designation, experience])<br>return candidates<br>candidates_list = input_candidates()<br><br>def append_candidate_data(candidates):<br>with open('candidates.bin', 'ab') as file:<br>for candidate in candidates:<br>pickle.dump(candidate, file)<br>print("Candidate data appended successfully.")<br><br>append_candidate_data(candidates_list)<br><br>(II)<br>import pickle<br><br>def update_senior_manager():<br>updated_candidates = []<br>with open('candidates.bin', 'rb') as file:<br>while True:<br>candidate = pickle.load(file)<br>if candidate[3] > 10:<br>candidate[2] = 'Senior Manager'<br>updated_candidates.append(candidate)<br><br>with open('candidates.bin', 'wb') as file:<br>for candidate in updated_candidates: | (5) |

www.careerindia.com

```
            pickle.dump(candidate, file)

    print("Candidates updated to Senior Manager where applicable.")

update_senior_manager()
```

(III)

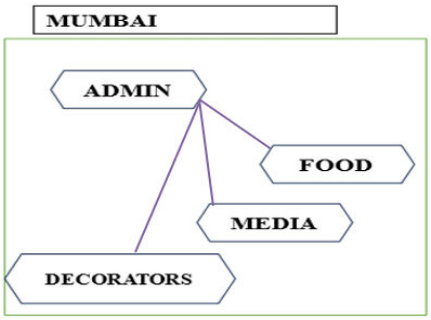
```
import pickle

def display_non_senior_managers():
    with open('candidates.bin', 'rb') as file:
        while True:
                candidate = pickle.load(file)
                if candidate[2] != 'Senior Manager':  # Check if not Senior Manager
                    print(f"Candidate ID: {candidate[0]}")
                    print(f"Candidate Name: {candidate[1]}")
                    print(f"Designation: {candidate[2]}")
                    print(f"Experience: {candidate[3]}")
                    print("--------------------")
display_non_senior_managers()
```

*(1/2 mark of import pickle)*
*(1/2 mark for input)*
*(1/2 mark for opening file in append mode and 1/2 mark for using dump)*
*(1/2 mark for opening file in read mode and 1/2 mark for using load)*
*(1 mark for checking the condition and updating the value)*
*(1 mark for checking the condition and displaying data correctly)*

37. (I) ADMIN Block as it has maximum number of computers. (5)
*(1 mark for correct answer)*

(II) Switch
*(1 mark for correct answer)*

(III)




(or Any other correct layout)

Cable: Coaxial cable
*(½ mark for correct layout + ½ mark for correct table type)*

www.careerindia.com

| | | |
|---|---|---|
| | (IV) There is no requirement of the Repeat as the optical fibre cable used for the network can carry the data to much longer distances than within the campus.<br>*(1 mark for correct answer)*<br><br>(V) (A) a) Video Conferencing<br>OR<br>(B) LAN<br>*(1 mark for correct answer)* | |

www.careerindia.com