SAHODAYA PRE BOARD EXAMINATION (2025-26)

CLASS – XII

Sub.: COMPUTER SCIENCE (083)

MARKING SCHEME (SET – 1)

	SECTION – A	
1.	False (1 mark for correct answer)	**[1]**
2.	(C) Annu.lmeet.tthecmpus (1 mark for correct answer)	**[1]**
3.	(C) True and False or not (False) (1 mark for correct answer)	**[1]**
4.	PRIMARY KEY (1 mark for correct answer)	**[1]**
5.	(D) 5 times (1 mark for correct answer)	**[1]**
6.	10$- 7$- (1 mark for correct answer)	**[1]**
7.	7.0 (1 mark for correct answer)	**[1]**
8.	SELECT department, COUNT(*) FROM employees GROUP BY department HAVING COUNT(*) > 5; (1 mark for correct answer)	**[1]**
9.	(D) A#Hello*C (1 mark for correct answer)	**[1]**
10.	(D) Not Specified (1 mark for correct answer)	**[1]**
11.	(B) 950@ (1 mark for correct answer)	**[1]**
12.	(B) 50#5 (1 mark for correct answer)	**[1]**
13.	(C) CHECK (1 mark for correct answer)	**[1]**

<table>
<tr><td>14.</td><td>(B) (‘Ye’, ‘a’, ‘r 2022 at All the best’)
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td>15.</td><td>(C) 3
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td>16.</td><td>(B) DROP TABLE
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td>17.</td><td>(A) TELNET
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td>18.</td><td>(C) Switch
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td>19.</td><td>(A) eXtensible Markup Language
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td>20.</td><td>(A) Both A and R are True and R is the correct explanation for A.
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td>21.</td><td>(A) Both A and R are True and R is the correct explanation for A
(1 mark for correct answer)</td><td>[1]</td></tr>
<tr><td></td><td>SECTION – B</td><td></td></tr>
<tr><td>22.</td><td>sort function arranges the list elements in-place (modifies the original list) whereas sorted() function returns a new sorted list and leaves the original iterable unchanged.
Ex: python lst = [3,1,2]
lst.sort()
print(lst)
# Output: [1,2,3]
python lst = [3,1,2]
new_lst = sorted(lst)
print(new_lst) # Output: [1,2,3]
print(lst) # Output: [3,1,2]
(1 mark for correct difference)
(1/2 mark for each correct example)
(Any other correct answers may be considered)

OR
(B)
<table>
<tr><td>split()</td><td>partition()</td></tr>
<tr><td>It separates the string from the every occurrence of the substring.</td><td>It separates the string from the first occurrence of the substring.</td></tr>
<tr><td>It returns in the form of list.</td><td>It returns in the form of tuple.</td></tr>
</table></td><td>[2]</td></tr>
</table>

	Example: s = 'computer is a machine' print(s.split('i')) output: ['computer ', 's a mach', 'ne'] Example: s = 'computer is a machine' print(s.partition('i')) output : ('computer ', 'i', 's a machine') (1 mark for correct difference) (1/2 mark for each correct example)	

23.	def sumeven(N):
	c=0
	for i in range(1,N+1):
	s=0
	while i>0:
	d=i%10
	s+=d
	i=i//10
	if s%2==0:
	c+=1
	print(c)
	
	(½ marks for each correction.)
		[2]

24.	(A) (I) L2.append(100) / L2.insert(len(L2)-1,100) (II) del L1[3] / L1.pop(3) **OR** (B) {'p': {'val': 9}, 'q': {'val': 9}} {'nums': [0, 1, 2]} (1 mark for each correct answer)	**[2]**

25.	def remove_key(student_data, roll_no):
	if roll_no in student_data:
	del student_data[roll_no]
	else:
	print("Key not found")
	OR
	def remove_all_occurrences(L, n):
	if n not in L:
	print("Data unavailable")
	else:
	while n in L:
	L.remove(n)
	print(L)
	(1/2 mark for function header)
	(1½ marks for the correct/similar logic)
		[2]

26.	{'Q1': 135000, 'Q2': 100500, 'Q3': 125000, 'Q4': 92500} None **(1½ marks for the 1st line)**	**[2]**

(½ mark for 2nd line)

27. **[2]**

(A) (I) ALTER TABLE Teacher ADD Salary float ;
(II) SELECT SUM(Salary) FROM Teacher ;
(1 mark for each correct answer)

OR

(B)

ALTER	UPDATE
Alter command by default initializes values of all the tuples as NULL.	UPDATE command sets specified values in the command to the tuples.
It operates on the attributes of a relation.	It operates on the attribute value of a specific tuple in a relation.
Ex:- ALTER table student add column marks int;	Ex:- UPDATE student set marks=20 where rno=2;

(2 marks for the correct answer)

28. **[2]**

A. Define the following terms:

I. Wi-Fi card is a hardware component (also called a wireless network adapter) that allows a computer or device to connect to a wireless network (Wi-Fi). It enables devices to communicate over a wireless LAN without using physical cables.

II. Ethernet card, also called a Network Interface Card (NIC), is a hardware component that allows a computer to connect to a wired network using Ethernet cables. It facilitates communication and data transfer over a LAN.

(1 mark for each correct definition)

OR

(B)(I) URL- Uniform Resource Locator
VoIP- Voice over Internet Protocol

(1/2 mark for each correct expansion.)

(II) SMTP (Simple Mail Internet Protocol) is a protocol used for sending Emails.
(1 mark for writing the correct answer)

SECTION – C

29. **[3]**

```
def CountEU():
    f = open("IMP.TXT", "r")
    text = f.read()
    count = 0
    for ch in text:
        if ch in 'EeUu':
            count += 1
    return (count)
    f.close()
```

½ marks for opening the file with the correct mode
1 mark for using read()
1 mark for condition
½ mark for return

OR

	def Count():
	file = open('quotes.txt' , "r")
	content = file.readlines() # read the entire file
	count = 0
	for line in content:
	if line[-2]=='.' or line[-2]=='!' or line[-2]=='?':
	count = count + 1
	print("Total punctuation marks:", count)
	file.close()
	# Function call
	Count( )
	(1/2 mark for correct function header)
	(1/2 mark for correctly opening the file)
	(1/2 mark for correctly reading from the file)
	(1 mark for correctly displaying the count)
	(1/2 mark for correctly calling the function)	

30.	(I)
	R={"GEETA":78,"SEETA":92,"GARIMA":60,"MANISHA":85, "GIRIJA":77}
	Student_Exam = []
	def PUSH(R):
	for i in R :
	if i[0] == 'G':
	Student_Exam.append(R[i])
	
	(II)
	def POP():
	while True :
	if Student_Exam == []:
	print("UNDERFLOW")
	break
	else :
	print(Student_Exam.pop())
	
	(1 ½ marks for each correct function definition)
		[3]

31.	(A) b8*r6x **(3 marks for the correct output)** **(1.5 marks for first three or last three consecutive letters)** **OR** (B) New Delhi Beijing Washington DCOk LondonOk *(1 mark for New Delhi Beijing* *1 mark for Washington DCOk* *1 mark for LondonOk)*	**[3]**
	SECTION – D	
32.	(A) (i) Select * from WATCHES where Watch_Name like '%Time'; (ii) Select Watch_Name, Price from WATCHES where Price between 5000 and 15000; (iii) Select sum(Qty_Store) where Type= 'Unisex'; (iv) Select * from WATCHES order by Price desc;	**[4]**

(4 x 1 mark for each correct query)

OR

(B)

(i)

Type	Total_Price
Unisex	17000
Ladies	15000
Gents	45000

(ii)

Watchid	Watch_Name	Price	Type	Qty_Store
W005	GoldenTime	25000	Gents	100

(iii)

MAX(Price)	MIN(Price)
25000	7000

(iv) DISTINCT Type

Unisex

Ladies

Gents

(4 x 1 mark for each correct output)

33. **[4]**

```
import csv
def CREATE():
    f=open("SALES.CSV",'w',newline='')
    ew=csv.writer(f)
    for i in range(1,11):
        print("\nEnter the details of the product:",i)
        P_no=int(input("Enter Product no. "))
        P_name=input("Enter Product Name: ")
        Price=int(input("Enter Price: "))
        ew.writerow([P_no, P_name, Price])
    f.close()
def DISPLAY():
    f=open("SALES.CSV",'r')
    records=csv.reader(f)
    for rec in records:
        if(int(rec[2])>5000):
            print(rec)
    f.close()
CREATE()
DISPLAY()
```

Marking Process of CREATE() function:

(1/2 mark for correctly taking user input)
(1/2 mark for opening the file in appropriate mode)
(1/2 mark for correctly creating the writer object)
(1/2 mark for correctly using writerow() of writer object)

	Marking Process of DISPLAY() function: (1⁄2 mark for opening in the file in right mode) (1⁄2 mark for correctly creating the reader object) (1⁄2 mark for correctly checking the condition) (1⁄2 mark for correctly displaying the records) **Note (for both parts (I) and (II)):** **Ignore import csv as it may be considered the part of the complete program.**	
34.	I) SELECT DO.DName, DE.Department, DE.Charges FROM DOCTOR DO, DEPT DE WHERE DO.DId=DE.DId; II) SELECT SUM(Charges) FROM DEPT GROUP BY Department; III) UPDATE DEPT SET Charges=Charges+Charges*0.1 WHERE Department='Neurology'; IV) (A) SELECT * FROM DOCTOR WHERE Age BETWEEN 40 and 50 and Gender='M'; **OR** (B) SELECT Gender, Count(*) FROM DOCTOR GROUP BY Gender; *(4x1 mark for each correct query)*	**[4]**
35.	import mysql.connector con=mysql.connector.connect(host='localhost',user='admin_user', password='Hockey2025',database='SportsDB') cursor=con.cursor() update_query="update Hockey set Award=Award+4500 where H_code=115" cursor.execute(update_query) con.commit() print("Data updated successfully.") cursor.close() con.close() (1⁄2 mark for correctly importing the connector object) (1⁄2 mark for correctly creating the connection object) (1⁄2 mark for correctly creating the cursor object) (1 mark for correct creation of update query) (1 mark for correctly executing the query with commit) (1⁄2 mark for correctly closing the connection)	**[4]**
	SECTION – E	
36.	I. import pickle def add_customer(): Cust_id = int(input("Enter Customer ID: ")) Cust_Name = input("Enter Customer Name: ") Item = input("Enter Item: ") Amount = float(input("Enter Amount: ")) record = [Cust_id, Cust_Name, Item, Amount] with open("Customer.dat", "ab") as f: pickle.dump(record, f) print("Record added successfully!") **(1/2 mark for correctly defining the function header)** **(1/2 mark for correctly opening the file in append mode)**	**[5]**

(1/2 mark for correctly taking user input)
(1/2 mark for using dump() method of the pickle module)

II.
```
def update_Amount():
    f=open("Customer.dat", "rb+")
    try:
        while True:
            pos = f.tell()
            record = pickle.load(f)
            if record[0] == 200:
                record[3] = record[3] + (15/100)* record[3]
                f.seek(pos)
                pickle.dump(record, f)
    except EOFError:
        f.close()
    print("Record updated successfully!")
```

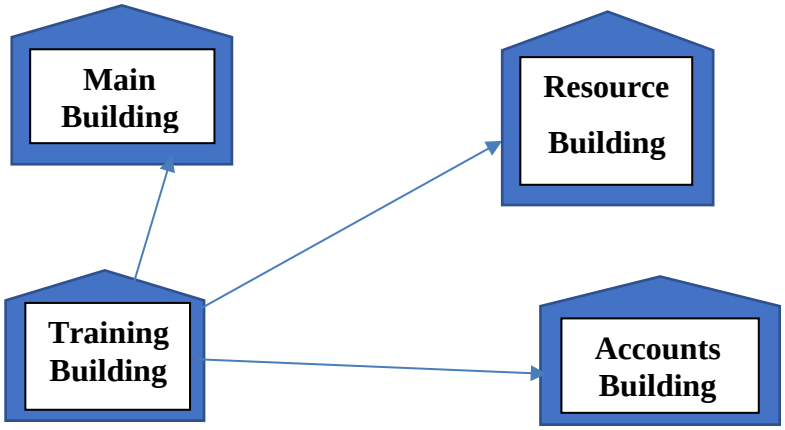
(1/2 mark for correctly defining the function header)
(1/2 mark for correctly opening the file)
(1 mark for using load() with while loop and try-except block)
(1 mark for checking the condition and updating the value)

Note (for both parts (I) and (II)):

(i) Ignore import pickle as it may be considered the part of the complete program.

37. **[5]**

I. BUS/STAR topology.



II. Training Building as it contains maximum number of computers.
III. Repeater- is not needed as per the above topology.
Hub/switch- to be placed in every building.
IV. Optical fibre.
V. (A) Firewall.
OR
(B) LAN
(5 x 1 mark for each correct part)

* * * * *