

**Pre Board Examination (2025-26)**  
**Computer Science (083)**  
**Marking Scheme**

Q.No	Answer / Key
1	True – Variable declaration is implicit in Python
2	5 (mode of the list)
3	(a) dict_exam.update(dict_result)
4	(a) True
5	(a) Year . 0. at All the best
6	(a) 14.75
7	(c) 50S
8	(d) Both (a) and (b) → 50@ or 50@70@
9	(b) Statement 4 (strings are immutable)
10	(a) 2-5-8-
11	(d) NOT NULL
12	(b) 6 (original degree 4 + 2 new columns)
13	(b) ALTER
14	(a) 10--20++20
15	(c) count(*)
16	(c) DISTINCT
17	(c) Router
18	(d) TELNET
19	(c) Web browser
20	(a) Both A and R are true and R is the correct explanation for A
21	(a) Both A and R are true and R is the correct explanation for A
22	<p>Dynamic typing in Python refers to the characteristic where the type of a variable is determined at runtime, rather than being explicitly declared at compile time. This means that a variable can hold values of different data types throughout its lifetime in a program. (1 mark for correct definition and 1 mark for example)</p> <p style="text-align: center;">OR</p> <p>Correct difference between mutable and immutable data type (1 mark). Correct example (1 mark)</p>
23	<pre>def prime(): #d in def to be small     n=int(input("Enter number to check :: ") ) #1 closing bracket missing     for i in range (2, n//2):         if n%i==0: # == required             print("Number is not prime \n")             break     else: #indentation level error         print("Number is prime \n")</pre> <p>(1/2 marks for each correct correction)</p>
24	<p>(I) S.upper()                  (II) len(L)                  1 mark for each correct answer</p> <p>OR</p> <p>Output:</p> <ul style="list-style-type: none"> <li>• Knowledge is your power, but enthusiasm pulls your switch</li> <li>• power</li> </ul> <p>1 mark for each correct answer</p>
25	<pre>def count_even(numbers):     count = 0     for num in numbers:         if num % 2 == 0:             count += 1</pre>

	<p>return count  <math>\frac{1}{2}</math> mark each for</p> <ul style="list-style-type: none"> <li>• correct function header</li> <li>• correct looping statement</li> <li>• correct condition check</li> <li>• correct return value</li> </ul> <p>OR</p> <pre>def update_marks(Marks, name, new_mark):     if name in Marks:         Marks[name] = new_mark         print("Marks updated")     else:         print("Student not found")</pre> <p><math>\frac{1}{2}</math> mark each for</p> <ul style="list-style-type: none"> <li>• correct function header</li> <li>• correct looping statement</li> <li>• correct value updation statement</li> <li>• correct else part</li> </ul>
26	<p>OUTPUT  [A, 'D']</p> <p>1 mark for each correct list element</p>
27	<p>i. SHOW TABLES  ii. DROP DATABASE EXAM  1 mark for each correct answer</p> <p>OR</p> <p>1 mark for correct difference and one mark for correct example.</p>
28	<p>SMTP – Simple Mail Transfer Protocol,  VOIP – Voice Over Internet Protocol,  HTTPS – Hypertext Transfer Protocol Secure,  PPP – Point to Point Protocol  1 mark for each correct expansion</p> <p>OR</p> <p>1 mark for correct definition of topology i.e. physical arrangement of devices in a network.  1 mark for correct advantages of star topology i.e. higher fault tolerance as failure of one terminal does not shuts down the entire system.</p>
29	<pre>def COUNTLINES():     with open("TESTFILE.TXT", "r") as file:         data = file.readlines()         for line in data:             if line[0].lower() not in ['a', 'e', 'i', 'o', 'u']:                 print(line)</pre> <p>(1/2 mark for correct function header)  (1/2 mark for correctly opening the file)  (1/2 mark for correctly reading from the file)  (1/2 mark for correct loop)  (1/2 mark for correct use of not in operator)  (1/2 mark for printing result)</p> <p>OR</p> <pre>def ETCCount():     with open("TESTFILE.TXT", "r") as file:         text = file.read()</pre>

	<pre># Split text into words words = text.split() count = 0 for word in words:     if word.endswith('n'):         count += 1 print("Count of words ending with 'n':", count)</pre> <p>(1/2 mark for correct function header)  (1/2 mark for correctly opening the file)  (1/2 mark for correctly reading from the file)  (1/2 mark for correctly using split function)  (1/2 mark for correct loop)  (1/2 mark for correct if statement)</p>						
30	<pre>Univ = [] course=[["MCA", 200000, 3] , ["MBA", 500000, 2] , ["BA", 100000, 3] ] def Push_element(course):     for i in course:         if i[1] &gt; 100000:             Univ.append(i)  def Pop_element():     while Univ:         print(Univ.pop())     else:         print("Underflow")  Push_element(course) Pop_element()</pre> <p>(1/2 mark for stack declaration)  (1/2 mark each for correct function header)  (1/2 mark for correct loop in both functions)  (1/2 mark each for correctly using condition in both functions)</p>						
31	<p>OUTPUT</p> <p>cCMmpuEe\$\$sCIEeCE</p> <p>OR</p> <p>['AI', 'NE', 'RA', 'AL']</p>						
32	<p>I. SELECT Name  FROM Teacher  WHERE Department = 'Computer Sc';</p> <p>II. SELECT *  FROM Teacher  WHERE Gender = 'F' AND Salary &lt; 20000;</p> <p>III. SELECT Gender, AVG(Salary)  FROM Teacher  GROUP BY Gender;</p> <p>IV. SELECT Department, COUNT(*)  FROM Teacher  GROUP BY Department;</p> <p>1 mark for each correct query.</p> <p>OR</p> <p>I.</p> <table> <tr> <td><b>Department</b></td> <td><b>AVG(Salary)</b></td> </tr> <tr> <td>Computer Sc</td> <td>16500.00</td> </tr> <tr> <td>History</td> <td>30000.00</td> </tr> </table>	<b>Department</b>	<b>AVG(Salary)</b>	Computer Sc	16500.00	History	30000.00
<b>Department</b>	<b>AVG(Salary)</b>						
Computer Sc	16500.00						
History	30000.00						

	<p>Mathematics 25000.00</p> <p>II. MAX(Date_of_Join) MIN(Date_of_Join) 2021-09-05            2017-03-24</p> <p>III. Name Saman Samira Shyam</p> <p>IV. <b>Name    Age</b> Randeep 32 Samira 35 Raman 42 Shyam 50</p> <p>1 mark for each correct output</p>
33	<pre>import csv i. def add():     pid = input("Enter Product ID: ")     pname = input("Enter Product Name: ")     price = float(input("Enter Product Price: "))     with open("Product.csv", "a", newline="") as file:         writer = csv.writer(file)         writer.writerow([pid, pname, price])  (1/2 mark for correctly taking user input) (1/2 mark for opening the file in append mode) (1/2 mark for correctly creating the writer object) (1/2 mark for correctly using writerow() of writer object)  ii. def search():     with open("Product.csv", "r") as file:         reader = csv.reader(file)         for row in reader:             pid, pname, price = row[0], row[1], float(row[2])             if price &gt; 20000 and pname[0].lower() in ['a', 'e', 'i', 'o', 'u']:                 print(row)  (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 data)</pre>
34	<p>I. SELECT Firstname, Lastname, Address FROM Employee WHERE City IN ('Paris', 'Upton');</p> <p>II. SELECT * FROM Employee ORDER BY Firstname DESC;</p> <p>III. SELECT E.Firstname, E.Lastname, S.Salary FROM Employee E</p>

	<p>JOIN EmpSalary S ON E.Empid = S.Empid WHERE S.Designation = 'Manager'; IV. UPDATE EmpSalary SET Salary = Salary + 1000 WHERE Designation = 'Clerk';</p> <p>OR SELECT Designation, MAX(Salary), MIN(Salary) FROM EmpSalary GROUP BY Designation;</p> <p>1 mark for each correct query</p>
35	<pre>import mysql.connector conn = mysql.connector.connect(host="localhost",user="admin",password="admin123", database="STOREDB") cursor = conn.cursor() query = "UPDATE INVENTORY SET Price = Price + 100 WHERE ProductName LIKE '%soap%';" cursor.execute(query) conn.commit() print(cursor.rowcount, "record(s) updated successfully.") cursor.close() conn.close()</pre> <p>(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)</p>
36	<pre>import pickle (i) def Create():     with open("PASSENGERS.DAT", "ab") as file:         pnr = input("Enter Passenger Number: ")         pname = input("Enter Passenger Name: ")         brdstn = input("Enter Boarding Station: ")         destn = input("Enter Destination Station: ")         fare = float(input("Enter Fare: "))         record = [pnr, pname, brdstn, destn, fare]         pickle.dump(record, file) (1/2 mark for correctly opening the file in append mode) (1/2 mark for correctly taking user input) (1/2 mark for using dump() method of the pickle module) (ii) def SearchDestn(D):     with open("PASSENGERS.DAT", "rb") as file:         while True:             try:                 record = pickle.load(file)                 if record[3].lower() == D.lower():                     print(record)             except EOFError:                 break (1/2 mark for correctly opening the file in append mode) (1/2 mark for using conditional statement) (iii) def UpdateFare():</pre>

	<pre> records = [] with open("PASSENGERS.DAT", "rb") as file:     while True:         try:             record = pickle.load(file)             record[4] = record[4] * 1.05             records.append(record)         except EOFError:             break with open("PASSENGERS.DAT", "wb") as file:     for rec in records:         pickle.dump(rec, file)         print("Fare updated successfully for all passengers.") </pre> <p>(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)</p>
37	<p>i. The server should be placed at HR Center as it has the maximum number of computers.  1 mark for correct answer</p> <p>ii. 1 mark for correct layout plan using bus/ star topology</p> <p>iii. Switch/Hub should be installed in each block to connect the computers within.  1 mark for correct answer.</p> <p>iv. Optical fibre cable should be used to achieve high speed connection.  1 mark for correct answer</p> <p>v. WAN will be formed as the distance is on the higher side.</p> <p>OR</p> <p>SMTP i.e. Simple Mail Transfer Protocol is used to send e-mails.  1 mark for correct answer</p>