Kendriya Vidyalaya Sangathan, Jaipur Region
PRACTICE PAPER -4 (MARKING SCHEME)

Class: XII **Subject**: Computer Science (083)
Maximum Marks: 70 **Period**: 3 Hours

Instructions:

- This question paper contains 37 questions.
- All questions are compulsory. However, internal choices have been provided in some questions. Attempt only one of the choices in such questions
- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A consists of 21 questions (1 to 21). Each question carries 1 Mark.
- Section B consists of 7 questions (22 to 28). Each question carries 2 Marks.
- Section C consists of 3 questions (29 to 31). Each question carries 3 Marks.
- Section D consists of 4 questions (32 to 35). Each question carries 4 Marks.
- Section E consists of 2 questions (36 to 37). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.
- In the case of MCQ, the text of the correct answer should also be written.

Q.	Section-A (21 x 1 = 21 Marks)	Mark
1	State True/False: The python expression 3.2+2 is evaluated as 3*2+2 (**False**)	1

2	Observe and find output of the following python code:
	str1 = "Python"
	str2 = "Programming"
	print(str1[:2] + str2[-4:])
	a. Pyming b. Pythming c. Pyogramming d. Pythonming
		1

3	Evaluate the following expression and find correct value of y: y = 3 * 4 + 2**3 // 2 - (7 % 3) **Ans. 15**	1
4	What will be the output of following python code: s = 'Programming' print(s.split("m")) **a. ['Progra', '', 'ing']** b. ['Progra', 'ing'] c. ['Progra', 'm', 'ing'] d. ['Progra', 'ming']	1

5	What will be the output of following python code:
	s = “yoBananaBoy”
	print(s[: :-1])
	Ans. yoBananaBoy
		1

6	What will be the output of following python code:
	t1 = 1,2,3
	t2 = (1,2,3)
	print(t1 is t2)
	a. True b. 1 c. False d. 0
		1

7	What would the following code print:
	fruits = {'apple': 5, 'banana': 3, 'orange': 2, 'grape': 4}
	print(fruits.get('mango'))
	a. 0 **b. None** c. Error d. 5
		1

8	Consider the given list L: L = list('All is well in ') What python code should be written for inserting the word 'Havana' at the end of the list as separate characters? a. L.extend('Havana') b. L.append(list('Havana')) **c. both a and b** d. None of the above	1

9	What will be the output of following python code:
	l2= [1,2,3,'[4,5]']
	print(type(l2[-1]))
	a. error b. <class 'list'> **c. <class 'string>** d. <class 'NoneType'>
		1

10	Suppose the content of a text file xyz.txt is as follows:
	"The best way to predict the future is to create it."
	What will be the output of the following python code?
	f = open("xyz.txt")
	f.seek(18)
	s = f.read(7)
	print(s)
	f.close()
	a. Predict b. The best way to
	c. predict the d. to predict the future
		1

11	In Python exception handling, the finally block is executed regardless of whether an exception occurs or not. (**True**/False)	1
12	def func(a, b, c=3, d=4): pass Identify the keyword and positional arguments in the function given above: **a) a and b are positional arguments; c and d are keyword arguments** b) a, b, c, and d are all positional arguments c) a, b, c, and d are all keyword arguments d) a, b, and c are positional arguments; d is a keyword argument	1
13	What is the output of following SQL statement? SELECT Department, COUNT(*) FROM employees WHERE Salary > 50000 GROUP BY Department; a. The total number of employees in each department **b. The departments with employees earning over 50,000 and the count of such employees in each department** c. The departments with average salary over 50,000 and their total number of employees d. The number of departments with employees earning over 50,000	1
14	Consider a table named 'Products' with columns 'product_id', 'product_name', and 'category'. Which of the following SQL queries will retrieve all products that are not in the categories 'Electronics' or 'Furniture'? **a. SELECT product_id, product_name FROM Products** **WHERE category NOT IN ('Electronics', 'Furniture');** b. SELECT product_id, product_name FROM Products WHERE category NOT IN 'Electronics', 'Furniture'; c. SELECT product_id, product_name FROM Products WHERE category != 'Electronics' AND != 'Furniture'; d. SELECT product_id, product_name FROM Products WHERE category NOT LIKE ('Electronics', 'Furniture');	1
15	In MySQL, which command does not change the cardinality of a relation? **a. ALTER** b. INSERT c. DELETE d. None of these	1

16	Sita is creating a table for her project. She wants that a particular column always has a unique value. Which constraint should she use? a. DISTINCT **b. UNIQUE** c. NOT NULL d. DEFAULT	1
17	Which of the following is a network protocol? a. Firewall **b. HTTP** c. Modem d. Switch	1
18	The Router in a network primarily functions as a __________. a. Converter **b. Traffic director** c. Amplifier d. Modulato	1
19	Write the full form of the following: (i) FTP: File Transfer Protocol (ii) DNS: Domain Name Server	1
	Q20 and Q21 are Assertion(A) and Reason(R) based questions. Mark the correct choice as: (A) Both A and R are true and R is the correct explanation for A (B) Both A and R are true and R is not the correct explanation for A (C) A is True but R is False (D) A is False but R is True	
20	**Assertion**: Assertion: In Python, a function can return multiple values. **Reason:** Python functions can return tuples, which can be unpacked into multiple variables. **Ans. (A) Both A and R are true and R is the correct explanation for A**	1
21	**Assertion**: The FOREIGN KEY constraint is used to establish links between tables. **Reason:** A FOREIGN KEY in one table points to a FOREIGN KEY in another table. **Ans. (C) A is True but R is False**	1

Q	Section-B (7 x 2=14 Marks)	Mark
22	Mark the valid and invalid identifiers in Python from the below options: myVariable : **Valid** 1st_try: **Invalid** For: **Invalid** _total_sum: **Valid**	2

23 Rohan is writing a Python program to determine if a year is a leap year. He has written a program, but it's not working correctly. Help him rewrite the code, underlining the changes. 2

```
year = input("Enter a year: ")  #Input need to be converted into integer
if year % 100 == 0:
   if year % 400 ==0:
      print(year, "is a century and leap year")
   else:
      print(year,"is a century year but not leap year")
else if year%4==0:                #elif instead of else if
      print(year, "is a leap year")
   else:
      print(year, "is not a leap year")
```

Correct Code:

```
year = int(input("Enter a year: "))  #used int() funtion
if year % 100 == 0:
   if year % 400 ==0:
      print(year, "is a century and leap year")
   else:
      print(year,"is a century year but not leap year")
elif year%4==0:                #elif instead of else if
      print(year, "is a leap year")
   else:
      print(year, "is not a leap year")
```

24 (A) Write a Python program to find the largest and smallest numbers from a list. Assume the list is given as [55, 12, 98, 34, 76, 1, 88]. 2

Ans.

```
numbers = [55, 12, 98, 34, 76, 1, 88]
largest_number = max(numbers)
smallest_number = min(numbers)
print("Largest number:", largest_number)
print("Smallest number:", smallest_number)
```

OR

(B) Write a Python program to check if a string is a palindrome (reads the same backward as forward). The string should be entered by the user.

Ans.

```
def is_palindrome(text):
   processed_text = text[::-1]
   return processed_text

user_input = input("Enter a string: ")
if is_palindrome(user_input):
   print(f"'{user_input}' is a palindrome.")
else:
   print(f"'{user_input}' is not a palindrome.")
```

25	Identify the correct output(s) of the following code from the choices i to iv. Write all incorrect choices. Also specify the maximum values that can be assigned to each of the variables start and end.
	import random
	numbers = [10, 20, 30, 40, 50, 60]
	start = random.randint(0, 2)
	end = random.randint(3, 4)
	for i in range(start, end):
	    print(numbers[i], end=", ")
	(i) 10, 20, 30, (ii) 30, 40, (iii) 20, 30, 40, **(iv) 40, 50,**
	The maximum value for start is 2, and the maximum value for end is 4.
		2

26	Define the following in context of MYSQL: **(i) Tuple**: A single row in a MySQL table. **(ii) Cardinality**: The number of rows (tuples) in a table.	2
27	(A) Write difference between DISTINCT and GROUP BY clause of SQL. **DISTINCT** removes duplicate rows from the result set; **GROUP BY** groups rows with the same values in specified columns for aggregate functions. OR (B) What is the difference in INT and FLOAT? **INT** stores integers; **FLOAT** stores floating-point numbers (numbers with decimal places).	2
28	Differentiate between Coaxial Cable and Optical Fiber. **Coaxial cable** uses electrical signals transmitted through a copper core; **Optical fiber** uses light signals transmitted through a glass or plastic core, offering higher bandwidth and less signal degradation. OR What is the use of the following devices? (i) Modem (ii) Repeater (i) A **modem** modulates and demodulates signals to allow communication between digital devices and analog lines (like telephone lines). (ii) A **repeater** amplifies and retransmits signals to extend the range of a network.	2

Q	**Section-C (3 x 3 = 9 Marks)**	**Mark**

29 (A) Write a Python function count_vowels() that reads text from a file named "input.txt" and counts the number of vowels (a, e, i, o, u) in the file. The function should return the vowel count. 3

```
def count_vowels(filepath):
    try:
        with open(filepath, 'r') as f:
            text = f.read().lower()  # Read and convert to lowercase
            for i in text:
                      if i in "aeiou":
                                vowel_count +=1
            return vowel_count
    except FileNotFoundError:
        return -1
```

OR

(B) Write a Python function longest_word() that reads text from a file "words.txt" and returns the longest word in the file.

```
def longest_word(filepath):
    l=[]
    try:
        with open(filepath, 'r') as f:
            words = f.read().split()
            if not words: #Handle empty file
                return None
            else:
                for i in words:
                      l.append(len(i))
            val=max(l)
            pos=l.find(val)
            return words[pos]
    except FileNotFoundError:
        return None
```

30 (A) A website uses a stack to manage recently viewed products. Each product is represented as a tuple: (product_id, product_name, price). Write the following Python functions to manage this RecentlyViewed stack: 3

(I) add_product(RecentlyViewed, new_product): This function adds a new product tuple to the top of the RecentlyViewed stack.

(II) remove_product(RecentlyViewed): This function removes and returns the most recently viewed product from the stack. If the stack is empty, it should print "No products recently viewed."

(III) show_latest_product(RecentlyViewed): This function displays the most recently viewed product without removing it. If the stack is empty, it should print "No products recently viewed."

Ans.

```
def add_product(RecentlyViewed, new_product):
    RecentlyViewed.append(new_product)

def remove_product(RecentlyViewed):
    if RecentlyViewed:
        return RecentlyViewed.pop()
    else:
        print("No products recently viewed.")

def show_latest_product(RecentlyViewed):
    if RecentlyViewed:
        print(RecentlyViewed[-1])
    else:
        print("No products recently viewed.")
```

OR

(B)

A hospital is managing patient data using a stack-based system. Patient records are initially stored in a list. Each record is a tuple containing (patient_id, age, priority_level). Priority levels are integers, with higher numbers representing higher priority.

(I) Create a list named **patients** containing the following patient records:
(101, 65, 2), (102, 32, 4), (103, 78, 1), (104, 45, 3), (105, 52, 5), (106, 28, 2)

(II) Write the definition of a user-defined function push_high_priority(patients, priority_threshold). It should push only those patient records with a priority level greater than or equal to the priority_threshold onto a stack called **high_priority_patients**.

(III) Write a function get_high_priority() to display all elements of the high_priority_patients stack while deleting them one by one. If the stack is empty, the function should display No high-priority patients.

Ans.

```
# (I) Create the list of patient records
patients = [(101, 65, 2), (102, 32, 4), (103, 78, 1), (104, 45, 3),
                (105, 52, 5), (106, 28, 2)]

# Initialize the high_priority_patients stack
high_priority_patients = []

# (II) Function to push high priority patients to stack
def push_high_priority(patients, priority_threshold):
    for patient in patients:
        if patient[2] >= priority_threshold:
            high_priority_patients.append(patient)

# (III) Function to get and display high priority patients
def get_high_priority():
```

31 Observe the table Students and write query for (i) to (iii): 3

Table: Faculty

F_ID	FName	LName	Department	Gender	Hire_Date	Salary
102	Ibomcha	Thounaojam	Exam	M	10/02/2020	75000
103	Shantanu	Fernandes	Exam	M	11/01/2015	120000
104	Tashi	Dorjey	ICT	F	14/03/2023	50000
105	Bhanwar	Singh	ICT	M	13/12/2019	80000
106	Kanta	Kumari	HOD	F	11/01/2024	140000

(A)

(i) Display Gender wise number of faculties who earn more than 85000:
SELECT Gender, COUNT(*) as Count FROM Faculty
WHERE Salary > 85000 GROUP BY Gender;

(ii) Display all data separated by Department and in decreasing order of Salary:
SELECT * FROM Faculty ORDER BY Department, Salary DESC;

(iii) Display FName and F_ID of faculties from ICT department:
SELECT FName, F_ID FROM Faculty WHERE Department = 'ICT';

OR

(B)

(i) Display Gender wise average salary of those faculties with average salary more than 90000:
SELECT Gender, AVG(Salary) as AvgSalary FROM Faculty
GROUP BY Gender HAVING AVG(Salary) > 90000;

(ii) Display FName and F_ID of faculties having the string 'ta' in the Fname:
SELECT FName, F_ID FROM Faculty WHERE FName LIKE '%ta%';

(iii) Change data of table to award 5% annual increment in salary:
UPDATE Faculty SET Salary = Salary * 1.05;

Q	**Section-D (4 x 4 = 16 Marks)**	**Mark**

32 (A) Explain the difference between the 'a' and 'x' file opening modes in Python. 4

- **'a' (append) mode:** Opens the file for appending. If the file exists, it appends new data to the end. If the file doesn't exist, it creates a new file.
- **'x' (exclusive creation) mode:** Opens the file for exclusive creation. If the file already exists, it raises a FileExistsError. If the file doesn't exist, it creates a new file.

(B) Observe the following code and predict the output of (i), (ii), and (iii):

```
def process_data(data):
    try:
        value = int(data)
        if value > 100:
            print("Value is greater than 100.")
        else:
            print("Value is not greater than 100.")
    except ValueError:
        print("Invalid input: Not an integer.")
    finally:
        print("Data processing complete.")
```

(B) Predicted outputs:

(i) process_data(150)

Output:

```
Value is greater than 100.
Data processing complete.
```

(ii) process_data("abc")

Output:

```
Invalid input: Not an integer.
Data processing complete.
```

(iii) process_data(50)

Output:

```
Value is not greater than 100.
Data processing complete.
```

33 A librarian is managing book inventory using a CSV file named `Inventory.csv`. The file structure is: `[BookID, Title, Author, Available]` where `BookID` is an integer, `Title` and `Author` are strings, and `Available` is an integer representing the number of copies available. 4

The librarian needs to write the following functions:

- add_book(): This function accepts new book details from the user and adds them to `Inventory.csv`. The file should be created with column headers if it doesn't exist.
- check_availability(book_id): This function takes a `book_id` as input and returns the number of copies available for that book. If the book is not found, it should return -1.

Ans.

```
import csv
import os

def add_book():
    file_exists = os.path.isfile('Inventory.csv')
    with open('Inventory.csv', 'a', newline='') as file:
        writer = csv.writer(file)
        if not file_exists:
            writer.writerow(['BookID', 'Title', 'Author', 'Available'])
        book_id = input("Enter BookID: ")
        title = input("Enter Title: ")
        author = input("Enter Author: ")
        available = input("Enter number of copies available: ")
        writer.writerow([book_id, title, author, available])
    print("Book added successfully!")

def check_availability(book_id):
    try:
        with open('Inventory.csv', 'r') as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row['BookID'] == str(book_id):
                    return int(row['Available'])
    except FileNotFoundError:
        print("Inventory file not found.")
    return -1
```

34 Give output of the following queries as per given table(s): 4

WORKER

WID	WNAME	JOB	SALARY	DNO
1001	RAHUL SHARMA	CLERK	15000	D03
1002	MUKESH VYAS	ELECTRICIAN	11000	D01
1003	SURESH	FITTER	9000	D02
1004	ANKUR	GUARD	8000	D01

DEPT

DNO	DNAME	LOC	MANAGER
D01	PRODUCTION	GROUND FLOOR	D K JAIN
D02	ACCOUNTS	1ST FLOOR	S ARORA
D03	SECURITY	1ST FLOOR	R K SINGH

Ans.

(i) SELECT DISTINCT JOB FROM WORKER;

```
JOB
----
CLERK
ELECTRICIAN
FITTER
GUARD
```

(ii) SELECT DNAME, LOC FROM DEPT WHERE DNO IN (SELECT DNO FROM WORKER WHERE SALARY > 10000);

```
DNAME            LOC
-----------      ------------
PRODUCTION   GROUND FLOOR
SECURITY        1ST FLOOR
```

```
(iii) SELECT W.WNAME, D.MANAGER
      FROM WORKER AS W, DEPT AS D
      WHERE W.DNO = D.DNO;
```

```
WNAME                MANAGER
-------------         ----------
RAHUL SHARMA   R K SINGH
MUKESH VYAS      D K JAIN
SURESH                S ARORA
ANKUR                  D K JAIN
```

(iv) SELECT WNAME FROM WORKER WHERE WNAME LIKE 'R%';

```
WNAME
-------------
RAHUL SHARMA
```

35 A table named Products in a database named Inventory stores information about products. The table has the following columns: ProductID (integer, primary key), ProductName (string), Price (float), and Quantity (integer). Assume the database username is 'admin' and the password is 'secure123'.
Write a Python code that prompts the user to enter a ProductID and updates the Quantity of that product by adding 10 to the existing quantity. Handle any potential errors (e.g., the product ID not existing in the table). 4

Ans.

```
import mysql.connector
from mysql.connector import Error

def update_product_quantity(product_id):
    try:
        connection = mysql.connector.connect(
            host="localhost",
            database="Inventory",
            user="admin",
            password="secure123" )

        if connection.is_connected():
            cursor = connection.cursor()
            cursor.execute("SELECT Quantity FROM Products
                                  WHERE ProductID = %s", (product_id,))
            result = cursor.fetchone()
            if result:
                current_quantity = result[0]
                new_quantity = current_quantity + 10
                update_query = "UPDATE Products SET Quantity = %s
                                     WHERE ProductID = %s"
                cursor.execute(update_query, (new_quantity, product_id))
                connection.commit()
                print(f"updated successfully. New quantity: {new_quantity}")
            else:
                print("Product not found.")

    except Error as e:
        print(f"Error: {e}")

    finally:
        if connection.is_connected():
            cursor.close()
            connection.close()
```

Q	Section-E (2 x 5 = 10 Marks)	Mark

36 Simran is developing a Python program to manage customer orders for an online store. Order data (order_id, customer_name, order_date, total_amount) is stored in a binary file named "Orders.dat". Each order is represented as a tuple. Help Simran complete the following tasks: 5

Ans.
import pickle

(i) Write a function `add_order()` to input order details from the user (order_id, customer_name, order_date, total_amount) and store them in "Orders.dat". The program should allow adding multiple orders until the user chooses to stop.

```
def add_order():
    orders = []
    try:
        with open("Orders.dat", "rb") as file:
            orders = pickle.load(file)
    except FileNotFoundError:
        pass

    while True:
        order_id = input("Enter order ID: ")
        customer_name = input("Enter customer name: ")
        order_date = input("Enter order date (YYYY-MM-DD): ")
        total_amount = float(input("Enter total amount: "))

        order = (order_id, customer_name, order_date, total_amount)
        orders.append(order)

        if input("Add another order? (y/n): ").lower() != 'y':
            break

    with open("Orders.dat", "wb") as file:
        pickle.dump(orders, file)
```

(ii) Write a function `update_order_amount()` to modify the `total_amount` for orders placed. The function should increase the `total_amount` of each qualifying order by 10%.

```
def update_order_amount():
    try:
        with open("Orders.dat", "rb") as file:
            orders = pickle.load(file)
        updated_orders = []
        for order in orders:
                new_amount = order[3] * 1.10
                updated_order = (order[0], order[1], order[2], new_amount)
                updated_orders.append(updated_order)

        with open("Orders.dat", "wb") as file:
            pickle.dump(updated_orders, file)

        print("Orders updated successfully.")
    except FileNotFoundError:
        print("Orders file not found.")
```

(iii) Write a function `count_high_value_orders()` to count and display the number of orders with a `total_amount` greater than 1000.

```
def count_high_value_orders():
    try:
        with open("Orders.dat", "rb") as file:
```

37 Kendriya Vidyalaya No 1 Jaipur is setting up the network between its Different Wings of school campus. There are 4 wings named as – SENIOR(S), JUNIOR(J), ADMIN(A) and HOSTEL(H). 5

Distance between various wings are given below:

Wing A to Wing S	80m
Wing A to Wing J	200m
Wing A to Wing H	400m
Wing S to Wing J	70m
Wing S to Wing H	120m
Wing J to Wing H	450m

Number of Computers installed at various wings are as follows:

Wing	No. of Computers
Wing A	20
Wing S	150
Wing J	50
Wing H	25

(i) Suggest a most suitable cable layout for the above connections.

```
            A
            |
            | (80m)
            |
J--------S----------R----------H
(70m)      (120m)
```

* ***R*** is the repeater

(ii) Suggest the most appropriate topology of the connection between the wings: **Star Topology**

(iii) The company wants internet accessibility in all the wings. What type of network (LAN/MAN/WAN) will be created if we connect all buildings? : **LAN**

(iv) Suggest the placement of the following devices with justification:

(A) Repeater: **Between S and H (the distance is more than 100)**

(B) Firewall: **in Wing S as it is also suitable for installing a server**

(v) (A) Which building will host the server of the company: **Wing S**

OR

(B) Suggest a device to be used for accessing the internet. **Modem**