Programming lab (

March

Date: February 18th 2024

Course Instructor(s)

Mr. Muhammad Ali

Total Time: 2 Hours

Total Marks: 20

Total Questions: 03

Semester: SP-2024

Campus: Karachi

Dept: CS/CY/SE

Student Name Roll No Section Student Signature

Question # 1: **[LLO- 2]** **[5 x1 = 5 Marks]**

Q1. Design and implement a **Superman** class with the following specifications: once Superman's health and power are set upon creation, they should remain same and cannot be modified thereafter.

1. The **Superman** class should have two private member variables: **int health** and **int power**.
2. Provide a constructor that initializes the **health** and **power** of Superman.
3. Implement a method named **displaySupermanInfo()** to display the health and power of Superman.
4. Attempt to include a method named **setHealth(int newHealth)** and **setPower(int newPower)** to modify the health and power, respectively. However, due to the requirement that both health and power should remain same after creation, attach the output of compilation in the doc file.
5. In the **main()** function, create an instance of the **Superman** class with given health and power values, display its information, and attempt to modify the health and power. Attach the output of the code when attempting to modify these attributes.

Question # 2: **[LLO- 3]** **[7 Marks]**

Q2 Suppose you're developing a household management system in C++ to organize various household items. As part of this system, you need to implement a function that checks the presence o a "has" relation between two arrays representing items in different rooms. A "has" relation between two arrays means that all items of one room are present in another room, possibly in a different order. Design and implement a class named room function named **hasRelation()** within the class having following specifications:

1. The function should take two arguments: two arrays of objects **(room1** and **room2)** representing items in different rooms. [1]
2. Use pointers to iterate through the elements of both arrays. [1]

3. Check if all items of **room1** are present in **room2.** [1]

4. If all items of **room1** are found in **room2,** return **true**; otherwise, return **false.** [1]

In the **main()** function:

1. Create two arrays of objects: **room1** and **room2**, representing items in different rooms.[1 Mark]

2. Call the **hasRelation()** function with **room1** and **room2** as arguments. [1 Mark]

3. Output whether a "has" relation exists between the items in the two rooms. [1 Mark]

Question # 3: [LLO- 3] [4 x 2 = 8 Marks]

You are tasked with designing a software system specifically tailored for managing operations within the court system in Karachi, including personnel, courtrooms, and legal proceedings. As part of this system, you need to create a class using multiple inheritance to model the entities involved in the Karachi court system.

Consider the following entities:

1. **Victim class:** Represents a victim. Victim have attributes such as name, victim history and role e.g. Guilty or not guilty. [1 Mark]

2. **Lawyer:** Represents a legal professional practicing within the Karachi court system. Lawyers have attributes such as license number, specialization, years of practice and victim evidence. [1 Mark]

3. **Judge:** Represents a judge presiding over legal proceedings within the Karachi court. Judges have attributes such as judicial ID, years of experience, and specialization. [1 Mark]

4. **Admin class:** Represents a physical courtroom within the Karachi court where legal proceedings take place. Courtrooms have attributes such as room number, capacity, and availability schedule. Additionally, include display administrator function. Hint: use multiple inheritance here in such way that when admin function call it shows judge decision based on lawyer evidence and victim history [1 Mark]

Note: Propose a set of sample methods or functionalities that could be implemented in the derived classes to demonstrate the use of **multiple inheritance** in the court management system. These functionalities should reflect the interactions between victim, lawyer, judge, and the administrator class in the court system. [4 Marks]