10.4-Abstract Classes and Interfaces

Program

```
using System;

// Define an interface for LibraryMembership
public interface ILibraryMembership
{
    void DisplayMembershipDetails();
}

// Define an abstract class for LibraryMember
public abstract class LibraryMember
{
    public string Name { get; set; }
    public int Age { get; set; }

    public LibraryMember(string name, int age)
    {
        Name = name;
        Age = age;
    }

    // Declare the abstract method
    public abstract void DisplayMembershipDetails();
}

// Create a derived class, PremiumLibraryMember, that inherits from LibraryMember and implements
```

```
ILibraryMembership
public class PremiumLibraryMember : LibraryMember, ILibraryMembership
{
    public PremiumLibraryMember(string name, int age) : base(name, age)
    {
        // Additional initialization for PremiumLibraryMember
    }

    // Implement the abstract method from the base class
    public override void DisplayMembershipDetails()
    {
        Console.WriteLine($"Premium Library Member: {Name}, Age: {Age}");
        Console.WriteLine("Access to premium book collections");
        Console.WriteLine("Extended borrowing periods");
        Console.WriteLine("Priority reservation for new releases");
    }
}

// Create a program class with the main method
class Program
{
    static void Main()
    {
        // Create an instance of PremiumLibraryMember
         PremiumLibraryMember premiumLibraryMember = new PremiumLibraryMember("Alice Smith",
30);
```

```
        // Display the membership details using the interface method
        premiumLibraryMember.DisplayMembershipDetails();

        // Wait for user input before closing the console window
        Console.ReadLine();
    }
}
```

Exercise

Write a C# program that includes the classes, interface, and methods it contains.

Library Membership Interface: Describe the purpose and functionality of the ILibraryMembership interface defined in the program.

Abstract Library Member Class: Explain the role of the LibraryMember abstract class. What properties and constructor does it have? What is the purpose of the abstract method declared within this class?

Premium Library Member Class: Detail the PremiumLibraryMember class, which is derived from LibraryMember and implements ILibraryMembership. What additional features or functionalities does it introduce? How does it implement the abstract method from the base class?

Main Program: Discuss the Program class containing the Main method. What actions are performed within this method? How is the PremiumLibraryMember class instantiated, and how are its membership details displayed?

Execution Flow: Describe the flow of execution when the program is run. What happens when the Main method is called?

User Interaction: Explain how user interaction is facilitated in the program. How does the program prompt the user for input, and how does it display membership details?

Program Output: Provide an example of the output produced when the program runs successfully.

What information is displayed to the user?

Hint

Interface Definition: Review the ILibraryMembership interface declaration. Consider its purpose and what functionality it defines.

Abstract Class Setup: Explore the LibraryMember abstract class. What properties does it have, and what constructor is implemented? Pay attention to the abstract method declaration.

Derived Class Implementation: Examine the PremiumLibraryMember class, which extends LibraryMember and implements ILibraryMembership. Note any additional initialization in its constructor and how it implements the abstract method.

Main Program Logic: Understand the Main method in the Program class. What actions are taken within this method? How is the PremiumLibraryMember instance created, and how is its membership information displayed?

Execution Flow: Consider the flow of execution when the program runs. How does the Main method coordinate the instantiation and display of membership details?

User Interaction: Think about how user interaction is handled in the program. How does it prompt the user for input, and how are membership details presented?

Explanation

This program demonstrates the concept of interface implementation and inheritance through a library membership system.

Firstly, an interface named ILibraryMembership is defined, which declares a method DisplayMembershipDetails(). This interface serves as a contract that any class implementing it must adhere to by providing an implementation for the DisplayMembershipDetails() method.

Next, an abstract class named LibraryMember is created, serving as a blueprint for library members. It contains properties for the member's name and age, along with a constructor to initialize these properties. Additionally, it declares an abstract method DisplayMembershipDetails(), which must be

implemented by its derived classes.

The program then defines a derived class named PremiumLibraryMember, which inherits from LibraryMember and implements ILibraryMembership. This class represents premium library members and extends the functionality provided by the base LibraryMember class. It includes a constructor for additional initialization specific to premium members and provides an implementation for the DisplayMembershipDetails() method as required by the interface.

In the Main method of the Program class, an instance of PremiumLibraryMember named premiumLibraryMember is created, passing in the member's name and age. Then, the DisplayMembershipDetails() method is called on this instance, displaying the membership details specific to premium library members. Finally, the program waits for user input before closing the console window, allowing the user to view the displayed information.