



Object Oriented Programming

Lab Manual 08



Introduction

After a week of rigorous coding, Welcome back!

You have learned all about the **Classes, Association, Aggregation, and Composition** in the previous lab manuals. Let's move on to the next, new, and interesting concepts.

Students, In Object-Oriented Programming, the Class is a combination of data members and member functions. In this Lab, we will learn about the **Inheritance** in our program to achieve the object's oriented philosophy.

What is Inheritance?

Consider you have a class **Student** with the following attributes.

<table><tr><th>Student</th></tr><tr><td><code>name</code> <code>session</code> <code>isDayScholar</code> <code>EntryTestMarks</code> <code>HSMarks</code> <code>RoomNumber</code> <code>isFridgeAvailable</code> <code>isInternetAvailable</code> <code>isBusCardIssued</code> <code>PickUpPoint</code> <code>BusNo</code> <code>PickupDistance</code></td></tr><tr><td><code>getHostelFee()</code> <code>getBusFees()</code> <code>calculateMerit()</code></td></tr></table>	Student	<code>name</code> <code>session</code> <code>isDayScholar</code> <code>EntryTestMarks</code> <code>HSMarks</code> <code>RoomNumber</code> <code>isFridgeAvailable</code> <code>isInternetAvailable</code> <code>isBusCardIssued</code> <code>PickUpPoint</code> <code>BusNo</code> <code>PickupDistance</code>	<code>getHostelFee()</code> <code>getBusFees()</code> <code>calculateMerit()</code>	<p>The Problem with this approach is that</p> <ul style="list-style-type: none">Some functions and data are not interrelated, such as the highlighted attributes and functions are only related to hostelite students but not to the day scholars.
Student				
<code>name</code> <code>session</code> <code>isDayScholar</code> <code>EntryTestMarks</code> <code>HSMarks</code> <code>RoomNumber</code> <code>isFridgeAvailable</code> <code>isInternetAvailable</code> <code>isBusCardIssued</code> <code>PickUpPoint</code> <code>BusNo</code> <code>PickupDistance</code>				
<code>getHostelFee()</code> <code>getBusFees()</code> <code>calculateMerit()</code>				

To overcome this problem, we can incorporate the **inheritance** concepts that would allow us to **reuse** a previously implemented class.

In this way, you can implement three classes namely, and **reuse** the **student class** in the **remaining two classes**.

- Hostelite
- DayScholar
- Student



Object Oriented Programming

Lab Manual 08



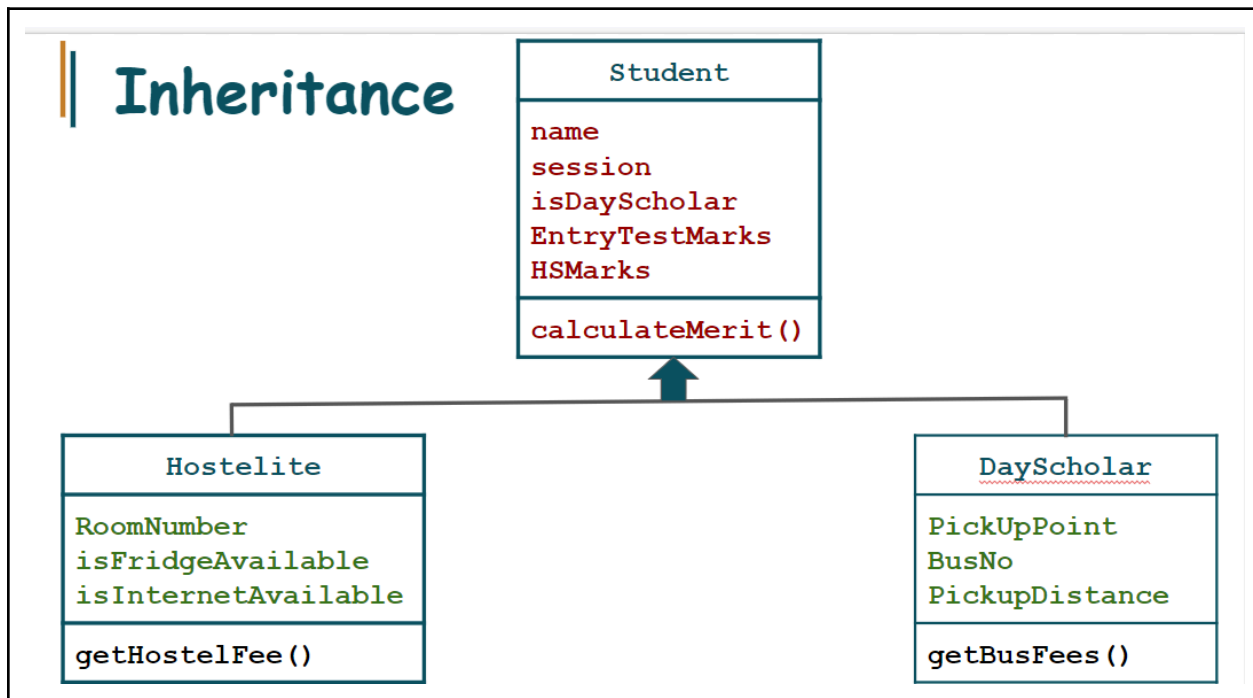
We create **three** classes:

Student
<code>name</code> <code>session</code> <code>isDayScholar</code> <code>EntryTestMarks</code> <code>HSMarks</code>
<code>calculateMerit()</code>

Hostelite
<code>RoomNumber</code> <code>isFridgeAvailable</code> <code>isInternetAvailable</code>
<code>getHostelFee()</code>

DayScholar
<code>PickUpPoint</code> <code>BusNo</code> <code>PickupDistance</code>
<code>getBusFees()</code>

Introduce **Inheritance** to implement the required functionality.



Initially, Attempt on your own.

Don't Worry. There is a solution on the next page.



Object Oriented Programming

Lab Manual 08



Solution:

Sr #	Code	Description
1	<pre>class Student { public string name; public string session; public bool isDayScholar; public int EntryTestMarks; public int HSMarks; public double calculateMerit() { double merit = 0.0; // Code to calculate merit return merit; } }</pre>	<ul style="list-style-type: none">● Student Class<ul style="list-style-type: none">○ It is a Parent Class
2	<pre>class Hostelite : Student { public int RoomNumber; public bool isFridgeAvailable; public bool isInternetAvailable; public int getHostelFee() { int fee = 0; // Code to calculate fee return fee; } }</pre>	<ul style="list-style-type: none">● Hostelite Class that is inheriting Student Class<ul style="list-style-type: none">○ It is Child Class
3	<pre>class DayScholar : Student { public string pickupPoint; public int busNo; public int pickupDistance; public int getBusFee() { int fee = 0; // Code to calculate fee return fee; } }</pre>	<ul style="list-style-type: none">● DayScholar Class that is inheriting Student Class<ul style="list-style-type: none">○ It is a Child Class



Object Oriented Programming

Lab Manual 08



4	<pre>static void Main(string[] args) { DayScholar std = new DayScholar(); std.name = "Ahmad"; std.busNo = 1; Console.WriteLine(std.name + " is Allocated Bus " + std.busNo); Console.ReadKey(); }</pre> <p>For Hostelite Class</p> <pre>static void Main(string[] args) { Hostelite std = new Hostelite(); std.name = "Ahmad"; std.RoomNumber = 12; Console.WriteLine(std.name + " is Allocated Room " + std.RoomNumber); Console.ReadKey(); }</pre>	<ul style="list-style-type: none">• Main Driver Program
---	---	--

Congratulations !!!!! You have successfully learned how to implement inheritance in your code.

Now, Let's Attempt the challenges that are listed on the next page.



Object Oriented Programming

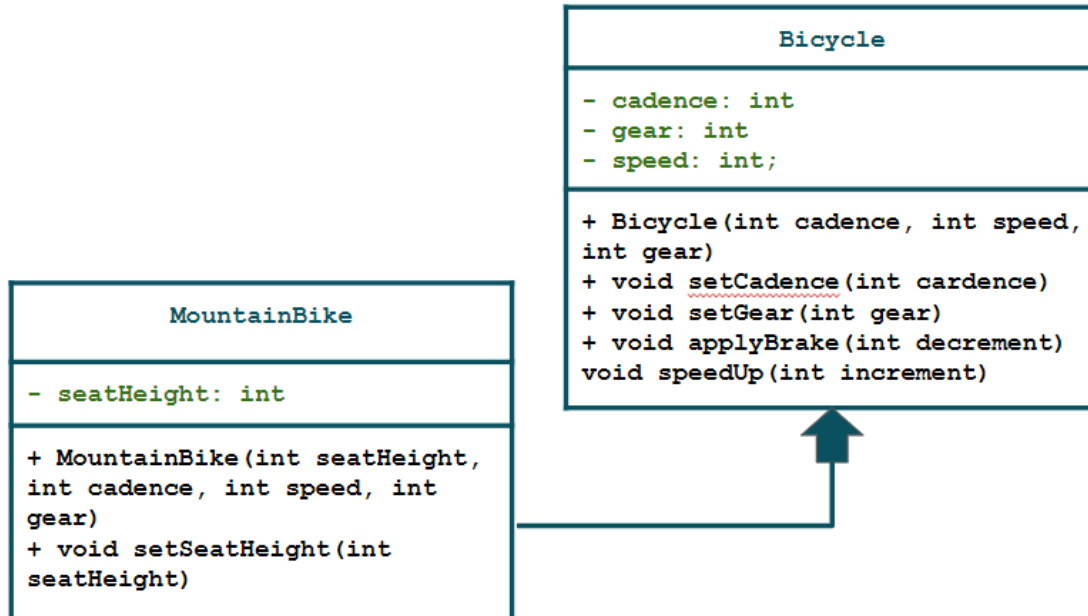
Lab Manual 08



Challenge 01:

Consider the following diagram and implement the C# code for this problem.

Implement the Following Classes



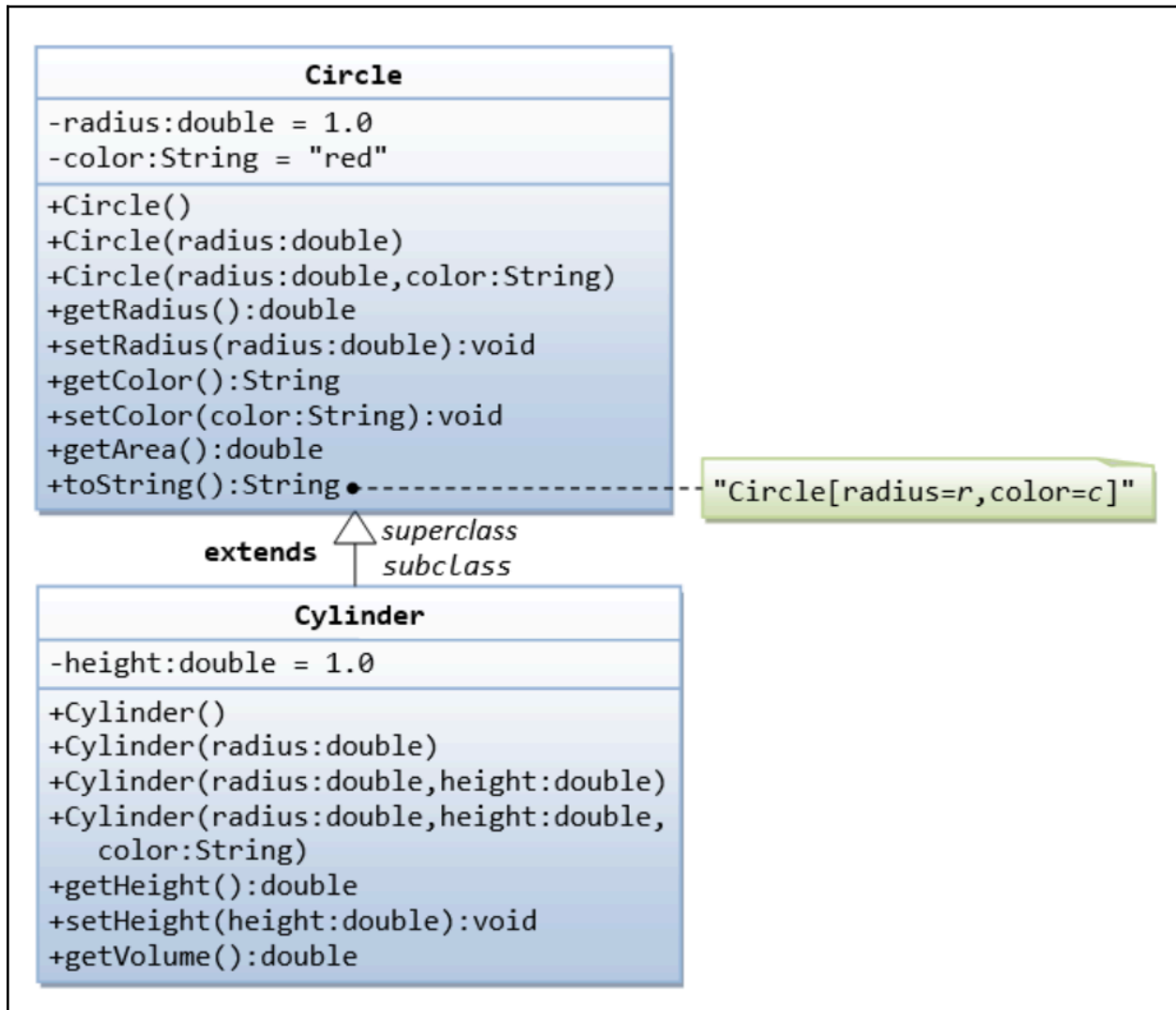
Challenge 02:

Consider the following diagram and implement the C# code for this problem.



Object Oriented Programming

Lab Manual 08



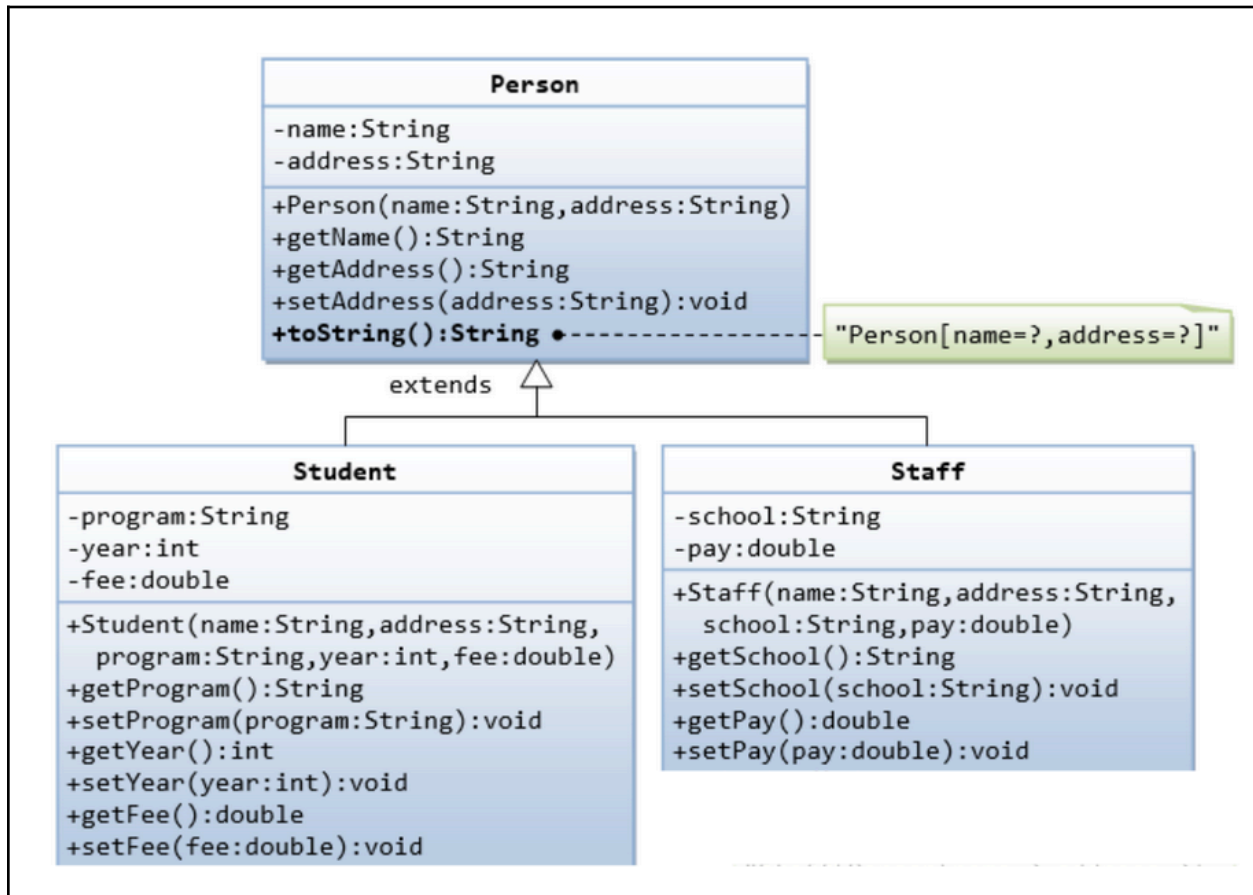
Challenge 03:

Consider the following diagram and implement the C# code for this problem.



Object Oriented Programming

Lab Manual 08



Challenge 04:

You've been hired by a financial institution to rebuild their ATM software system. One of the main challenges they're facing is accommodating the diverse needs of their customers, who have varying types of accounts with different privileges and restrictions (e.g. different types of profits, different tax implementation). Your task is to design a flexible system that can handle these complexities efficiently. How would you approach designing the software CRC to manage the different account types and their associated functionalities in the ATM system, ensuring that it can easily adapt to future changes in account types and requirements? Additionally, could you provide a code snippet showcasing how you would implement the handling of at least two distinct account types within the system?

Good Luck and Best Wishes !!



Object Oriented Programming

Lab Manual 08



Happy Coding ahead :)