C# Assignment on Polymorphism

Assignment 1: To demonstrate Polymorphism and its Advantages

Create a C# program demonstrating polymorphism by using a base class **Shape** and derived classes **Circle** and **Rectangle**. Each derived class should implement a method **Draw()**. Show how polymorphism helps in calling the correct method based on the object type.

Assignment 2. Method Overloading and its uses

Create a C# program to show method overloading by implementing a **Multiply** method with different parameter types, numbers and order.

Assignment 3. Method Overriding

Write a program demonstrating method overriding by creating a base class Vehicle and a derived class Car that overrides the Drive() method.

Assignment 4. Scenario of Overriding

Consider a scenario of payment processing where different payment methods (e.g., CreditCardPayment, PayPalPayment) override the ProcessPayment() method of a base class Payment. Write a C# program to demonstrate this scenario.

Assignment 5. Polymorphism with Static Data and Methods.

Create a C# program that demonstrates polymorphism using a base class **Employee** and derived classes **Manager** and **Developer**. Include a static field to keep track of the total number of employees and a static method to display the total count. Use method overriding to demonstrate polymorphism, while also explaining the need for static members in this context.

Assignment 6. Polymorphism with Arrays as Properties in a Class

Create a C# program demonstrating polymorphism using a base class **Employee** and derived classes **Manager** and **Developer**. In this program, each employee should have a collection of tasks (stored in an array) assigned to them. Use arrays as properties in the class to handle this data, and demonstrate how polymorphism and arrays work together in the solution.

Assignment 7: Understanding Early Binding and Late Binding in C#

Create a C# program that demonstrates **early binding** (compile-time polymorphism) using **method overloading** and **late binding** (runtime polymorphism) using **method overriding**. This will help illustrate the differences between the two concepts in the context of polymorphism.

Assignment 8. Achieving Runtime Polymorphism with Abstract Classes and Interfaces in C#

Create a C# program that demonstrates how **runtime polymorphism** is achieved using **abstract classes** and **interfaces**. Define an abstract class Shape and an interface IShape, implementing these in derived classes to showcase polymorphism.

Assignment 9. Demonstrating the Need for Multiple Inheritance of Interfaces

Create a C# program that demonstrates the concept of multiple inheritance through interfaces. The program will define two interfaces, IMovable and IDrawable, and implement them in a class Car that showcases how a class can inherit from multiple interfaces.

Assignment 10. Polymorphism in C# with Readonly Property

Create a C# program that demonstrates **polymorphism** with a **readonly property**. Define a base class and derived classes where each class provides specific behavior for a method, while using a readonly property to ensure that certain values cannot be modified after initialization.