Assignment Question on Overloading and Overriding in C#

Method Overloading

Q**1:** Write a C# program to create a class **MathOperations** with overloaded methods `Multiply`. The first method should multiply two integers, the second should multiply three integers, and the third should multiply two doubles.

Q2: Implement a Calculator class with overloaded methods Subtract. One method should subtract two integers, another should subtract three integers, and a third should subtract two doubles. How does the compiler differentiate between these methods?

Q3: Create a class Printer with overloaded methods Print. The first method should print an integer, the second should print a string, and the third should print an array of integers. How does method overloading help in this scenario?

Q4: Design a Converter class with overloaded methods ConvertToString. One method should convert an int to a string, another should convert a double to a string, and another should convert a DateTime to a string. Implement these methods.

Method Overriding

Q1: Create a base **class Animal** with a **virtual method MakeSound**. Derive **classes Dog and Cat** from Animal and **override the MakeSound method** to print different sounds for each animal.

Q2: Suppose you have a base **class Shape** with a **virtual method Area.** Create two derived classes **Rectangle and Circle**. Implement the Area method in both derived classes and overload it in the Rectangle class to allow calculating the area with different sets of parameters.

Q3: Write a base class **Employee** with a virtual method **CalculateBonus**. Create two derived classes, **Manager and Developer**, that override the **CalculateBonus** method to calculate bonuses differently based on their roles.

Q4: Create a **Vehicle** class with a virtual method **StartEngine**. Then, create two derived classes **Car and Motorcycle**, each overriding the **StartEngine method** to start their engines in a unique way. Demonstrate how polymorphism works in this scenario.

Q5: How would you use **the base keyword** to call a base class method that has been overridden in a derived class? Implement a class **Bird** with a method Fly**,** and a derived class **Penguin** that overrides Fly. Show how to call the base class method from the derived class.