Task#01

Step 1: Class Creation

Create a class named "Calculator" that represents a calculator. The class should have the following methods:

- ➢ Add: This method should accept two integer parameters and return their sum.
- ➢ Add: This method should accept two double parameters and return their sum.
- ➢ Multiply: This method should accept two integer parameters and return their product.
- ➢ Multiply: This method should accept two double parameters and return their product.

Step 2: Object Creation

Instantiate an object of the Calculator class in the main program.

Step 3: Method Overloading

Implement the methods in the Calculator class according to their respective functionality. Make sure to use method overloading by having methods with the same name but different parameter types.

Step 4: Program Execution

In the main program, perform the following tasks:

- ✓ Call the Add method with two integers and display the result.
- ✓ Call the Add method with two doubles and display the result.
- ✓ Call the Multiply method with two integers and display the result.
- ✓ Call the Multiply method with two doubles and display the result.

Step 5: Run and Test

Compile and run the program. Verify that the calculator methods are correctly performing addition and multiplication operations for both integers and doubles.

Task#02

Step 1: Class Creation

Create a class named "BankAccount" that represents a bank account. The class should have the following properties:

- AccountNumber (string): A unique identifier for the account.
- AccountHolderName (string): The name of the account holder.
- Balance (decimal): The current balance in the account.

Step 2: Default Constructor

Implement a default constructor for the BankAccount class. The default constructor should set the AccountNumber to a default value (e.g., "00000000"), AccountHolderName to an empty string, and Balance to 0.0.

Step 3: Parameterized Constructor

Implement a parameterized constructor for the BankAccount class. The constructor should accept parameters for AccountNumber, AccountHolderName, and Balance. It should initialize the corresponding properties with the provided values.

Step 4: Display Account Information

Implement a method in the BankAccount class called "DisplayAccountInfo" that displays the account information. The method should print the AccountNumber, AccountHolderName, and Balance.

Step 5: Program Execution

In the main program, create a method named "Main" and perform the following tasks:

Instantiate an object of the BankAccount class using the default constructor.

Instantiate another object of the BankAccount class using the parameterized constructor, providing values for AccountNumber, AccountHolderName, and Balance.

Call the "DisplayAccountInfo" method for both objects to display their account information.

Step 6: Run and Test

Compile and run the program. Ensure that the program executes without any errors and displays the correct account information for both objects.