PRG650-NBBL **LAB ASSIGNMENT # 2**

Marks: 3 Due: Week 4, May 29

Question No.1

Develop a Visual C# .NET calculator solution. This solution will have a class library component and one form application component contained within the same solution.

The Calculator Class

Create a class library project named **LibCalculator** that contains a class named **Calculator**. It is up to you to decide which properties, variables, or methods to use but you may use the specifications given below to start your project. The **Calculator** class may have the following specifications:

Private field	Description
`displayString`	A string that stores the stores the value that's currently displayed by the calculator.
`isNewValue`	A Boolean that controls numeric entry (to check if displayString has a value).
`hasDecimal`	A Boolean that tracks whether the display string contains a decimal point.
`operand1`	A decimal that stores the value of the first operand.
`operand2`	A decimal that stores the value of the second operand.
`op`	An Operation type that stores a member of the Operation enumeration.
`Operation`	An enumeration with these constants: Add, Subtract, Multiply, Divide, SquareRoot, Reciprocal, and None.
Constructor	**Description**
`()`	Creates a Calculator object with default values. The default value for the op field is Operation.None. The default value for the isNewValue field is true and hasDecimal is false.
Property	**Description**
`DisplayString`	Gets the value of the displayString field, or "0" if it's null.
`displayValue`	Gets the value of the DisplayString property, converted to a decimal type. A private property that's only used in the class.
Method	**Description**
`Append(string)`	Adds the string value passed to it to the end of the displayString field.
`RemoveLast()`	Removes the last character from the displayString field.

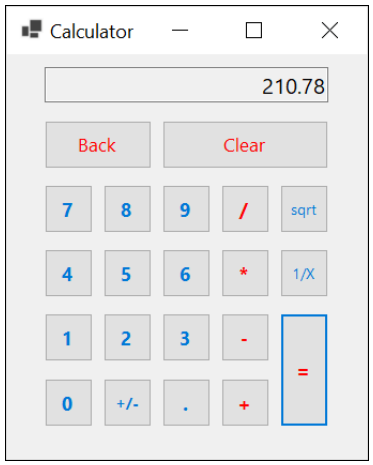
Method	Description
`ToggleSign()`	Switches the display value from positive to negative or vice versa.
`AddDecimalPoint()`	Adds a decimal point to the displayString field, and ensures there can only be one decimal point.
`Add()`	Sets the op field to Operation.Add, the operand1 field to the value of the displayValue property, the isNewValue field to true, and the hasDecimal field to false.
`Subtract()`	Sets the op field to Operation.Subtract, the operand1 field to the value of the displayValue property, the isNewValue field to true, and the hasDecimal field to false.
`Multiply()`	Sets the op field to Operation.Multiply, the operand1 field to the value of the displayValue property, the isNewValue field to true, and the hasDecimal field to false.
`Divide()`	Sets the op field to Operation.Divide, the operand1 field to the value of the displayValue property, the isNewValue field to true, and the hasDecimal field to false.
`Equals()`	Sets the operand2 field to the value of the displayValue property. Then, performs the operation specified by the op field on the operand1 and operand2 fields, and stores the result in the displayString field. Sets the isNewValue field to true, and the hasDecimal field to false.
`SquareRoot()`	Sets the op field to Operation.SquareRoot, the operand1 field to the value of the displayValue property, the isNewValue field to true, and the hasDecimal field to false. Calculates the square root of the operand1 value and stores it in the displayString field.
`Reciprocal()`	Sets the op field to Operation.Reciprocal, the operand1 field to the value of the displayValue property, the isNewValue field to true, and the hasDecimal field to false. Calculates the reciprocal of the operand1 value and stores it in the currentValue field.
`Clear()`	Sets the private fields to their default values.

The Windows Application

Create a Windows Forms application. The form window for the application appears as shown below. Use proper names for the project, the form, and the controls and give your form a title.

Operation

- To perform an addition, subtraction, multiplication, or division operation, the user clicks the first number, followed by the appropriate operator key (+, -, *, /), followed by the second number and the equals key (=).
- To perform an addition, subtraction, multiplication, or division operation on the result of a previous operation, the user clicks another operator key, followed by another number and the equals key.
- To perform a square root or reciprocal operation, or to change the sign of a number, the user clicks the number followed by the appropriate operator key (sqrt, 1/X, +/-).
- To perform a square root or reciprocal operation on the result of a previous operation, the user clicks the appropriate operator key.
- Each time the user clicks a number key, the number is displayed in the text box at the top of the form. This text box also displays the result of an operation when the user clicks the sqrt, 1/X, +/-, or = key.
- To erase the last digit entered, the user clicks the Back key.
- To clear all the values entered, the user clicks the Clear key.


Calculator
210.78
Back
Clear
7
8
9
/
sqrt
4
5
6
*
1/X
1
2
3
-
=
0
+/-
.
+