Centennial College

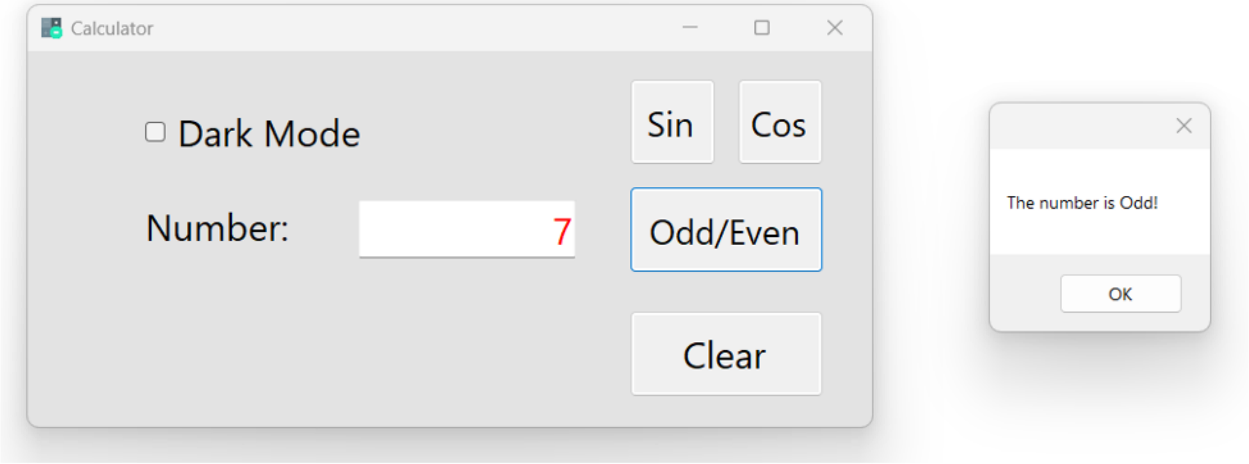
COMP 123 – Programming II

Final Exam (Part B)

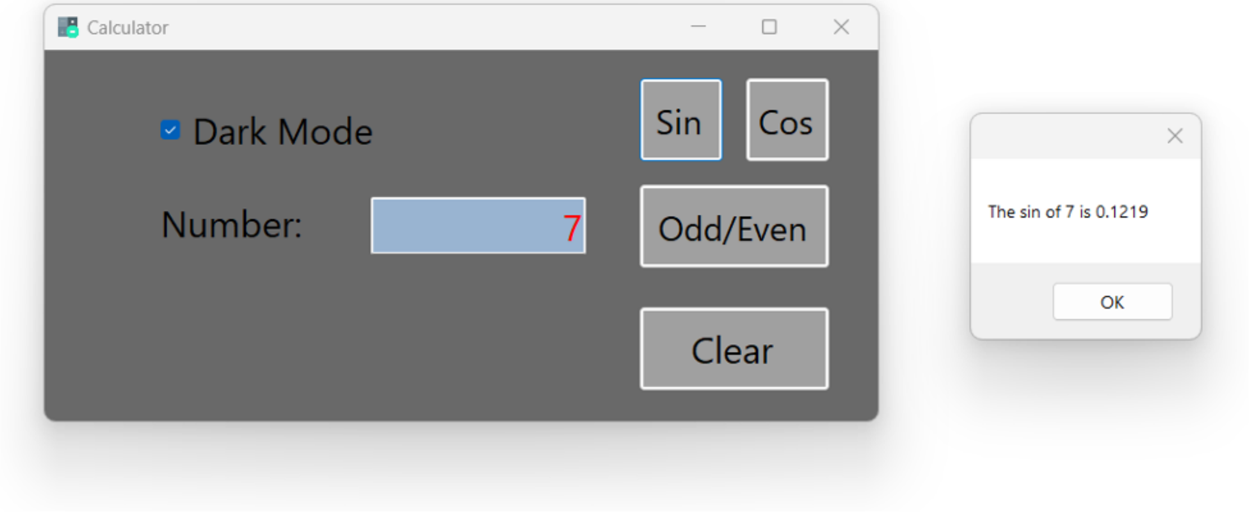
Question 01: GUI

Now, design and implement a simplified calculator using C# Windows Form Application project, based on the following steps:

1) Change the form name to Calculator.
2) Change the icon of the form to something like calculator.
3) Create a checkbox to switch between Light Mode and Dark Mode as it is displayed in the picture.
4) The form size should be 700 x 350.
5) Add labels, text boxes and buttons as shown in the figure.
    a. Clear button should change the text of number textbox to Zero.
6) Change the text colour of the text box to red.
7) Change the size of Sin and Cos buttons to 70 x 70 and the font size = 16. (The size of other Buttons is based on your decision.)
8) Add the needed code to make the buttons work properly. The result of calculating Sin, Cos, Odd/Even should be displayed in a Message Box as you can see in the following screenshots.



The number is Odd!



The sin of 7 is 0.1219

Question 02: Delegate-based Text Processing

You are tasked with developing a text processing application using delegates. The application should allow users to perform various text processing operations on an input text, including converting to uppercase, reversing, and removing spaces. Create a class structure to represent text processing operations and the functionality of the application.

1. **Define the `TextOperation` Delegate:**

  - Create a new namespace named `DelegateTextProcessor`.

  - Inside the namespace, define a delegate named `TextOperation` that takes a single `string` parameter and returns a `string`.

  - This delegate will be used to represent different text processing operations.

2. **Define the `TextProcessor` Class:**

  - Inside the `DelegateTextProcessor` namespace, create a class named `TextProcessor`.

  - Declare static methods within the class to perform the following text processing operations:

   - `Uppercase`: Converts the input text to uppercase using the `ToUpper` method.

  - `Reverse`: Reverses the characters in the input text using the `ToCharArray` and `Array.Reverse` methods.

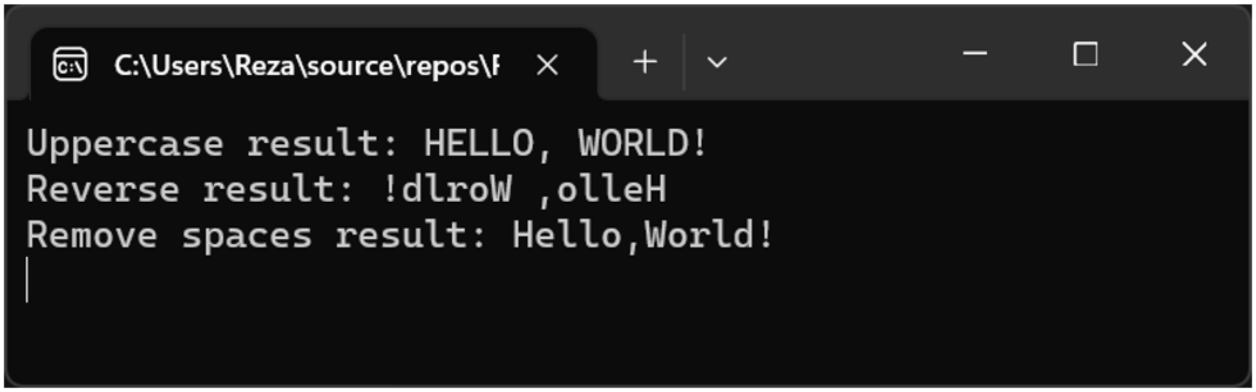
  - `RemoveSpaces`: Removes spaces from the input text using the `Replace` method.

- Each method should take a `string` parameter and return a `string`.

3. Use the above code in the Main method. The Main is given to you separately in a text file to save your time. You should get the following output.

Note: The steps outlined above guide you through creating the `TextOperation` delegate, the `TextProcessor` class with text processing methods, and the `Program` class with the `Main` method to demonstrate the delegate-based text processing application.

Output:



```
Uppercase result: HELLO, WORLD!
Reverse result: !dlroW ,olleH
Remove spaces result: Hello,World!
```