Test 1 15%

IC 140 – OOP NET1 Date: June 23, 2021

Dr. Mohannad AlMousa

Calculator

Create an application that allows the user to perform the following arithmetic calculations (+, -, *, /, and %) between two numbers, then display the history in a list, and allow to store the history to a file.

A. Your interface should contain at minimum the following **controls**: (20%)

1- Read only text box control named ***txtMain*** for the main calculation area.

2- Read only text box control ***txtResult*** for the calculation result.

3- 10 button controls for each of the numbers 0-9

4- 5 button controls for the arithmetic operations (+, -, *, /, and %).

5- A total, clear, and submit buttons,

6- A List box control to hold the history.

7- A groupbox control contains the history listbox, the submit button, and the following radio button if you choose to do them.

The following two (8 and 9) are Bonus1

8- Two radio buttons to save and load data to and from text file.

9- Open file dialog, and save file dialog to handle loading and saving files.

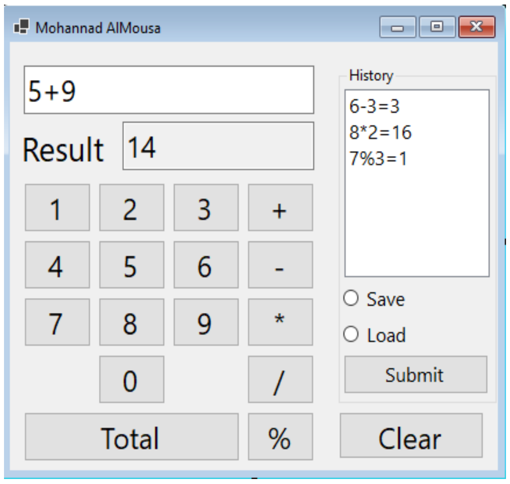

Figure 1

You can use figure 1 as a reference for the above controls, and feel free to rearrange the controls based on what you see fit. However, do not eliminate any control.

<u>Note:</u>

For easer implementation, work with numbers as string until you need to perform the arithmetic operation then parse them to integers.

Your application must handle the following functionalities (15%):

1. If any of the Buttons 0-9 is clicked, you need to display the corresponding number in the main area.
2. If any of the arithmetic buttons is clicked, you need to display the corresponding operation in the main area and apply that calculation between the two entered numbers.
3. Once the user clicks the total button, the results of the operation should be displayed in the results read only textbox (***txtResult***) and the history list box will also add the complete operation with the result at the bottom of the list.
4. User may save the operation's history once they click the save button, more details are in the submit button click event below.
5. If the user clicks Clear, the main area, result, and history listbox controls should be cleared.
6. (**Bonus1**) Users may select the *Save* or *Load* radio buttons and click Submit button to save the operations history to a file or load the history from a file, respectively.

At minimum, you need to implement the following **<u>Class member variables</u>** (10%):

1- String Number1Str: a string type member variable to store the first number entered by the user.

2- String Number2Str: a string type member variable to store the second number entered by the user.

3- Int Result: an integer member variable to store the result of the performed operation.

4- String operation: a string type member variable to store the selected operation.

5- You may add any other member variable if needed to use (i.e. flags,…).

Test 1 15%

IC 140 – OOP NET1 Date: June 23, 2021
Dr. Mohannad AlMousa

At minimum, you need to implement the following **class member method** (25%):

1- `void SetNumber(string num)`: this method expects a string parameter, and sets the Number1Str or Number2Str member variables. First it confirms if the operation member variable has been set or not, if not then you need to set Number1Str, otherwise set Number2Str.

2- `bool SetOperation(string op)`: this method expect a string parameter and returns true only if the operation is set correctly. This method should use the ***if().. else if()*** statement as follows:

   a. If Number1Str is not set: display an error "`Please Select the first number first, then select the operation.`" and return false.
   b. Else if Number2Str is set already: display and error "`Invalid Selection, Clear current Calculation first and start over.`" and return false.
   c. Else if the operation had already been set, use the following line of code to erase the previous operation from the main display text box named ***txtMain*** and then set the new operation string into the operation member variable.

      ```
      txtMain.Text = txtMain.Text[0..^1];
      ```
   d. Else, set the operation member variable.

3- `void UpdateGUI(string val):` this method expects a string parameter and appends it to the text value in the ***txtMain*** control.

4- `int Calculate(int Number1, int Number2):`This method receives two integer parameters and returns the results of the pre-set operation for the two parameters. You must use a **switch** statement on the operation member variable and return the result of the operation.

At minimum, you need to implement the following **events** (25%):

1- Numbers buttons click event: each number button need to be linked to a button click event to capture the clicked number and set Number1Str and/or Number2Str by calling the ***SetNumber*** method. The event must then call the ***UpdateGUI*** method and pass the clicked number to be displayed on the screen.

2- Arithmetic buttons click event: each arithmetic operation button need to be linked to a button click event to capture the clicked operation and set the operation by calling the ***SetOperation*** method. Then this event will update the interface by calling the ***UpdateGUI*** `with the` clicked operation **only If** the ***SetOperation*** method return true.

3- The total button click event: This event should confirm that the two numbers and the operation are entered before it converts the numbers to integer values. Once the numbers are converted to integer this method should call the ***Calculate*** method passing the integer values of the Number1Str and Number2Str to perform the calculation. The result of the ***Calculate*** method call should be stored in the ***Result*** member variable, then the **txtResult** control must be updated with the result value. Finally, the List box control must add the operation to its list.

4- The Clear button click even: this event must reset your member variables to null (Number1Str, Number2Str, and Operation) and Result to 0. Finally, clears the **txtMain** and **txtResult** text box controls.

5- The Submit button click event: this event will read the history list box control items and write the content to a local text file named "**CalculationHistory.txt**" the file should be saved to the "**C:\Test1Result"** folder, if you don't have the folder you need to create it manually, once the data is saved successfully you need to clear the list box operations history. (**Bonus1**: allow the user to save and load the file using the SaveFileDialog and OpenFileDialog controls.

   *Note: **Bonus1** is worth 7% towards this test mark.*

Submission instructions (5%):

Include your name as the title of the form.

Compress the entire project into a zip file.

Rename the zib file with "your name.zip"

Upload the zip file to the test area in Blackboard.

Email submissions are not accepted.