Course: EGDF20
Module: EGE202 Application Programming

Practical 5a: Calculator Application: Effective GUI Development and Event Handling

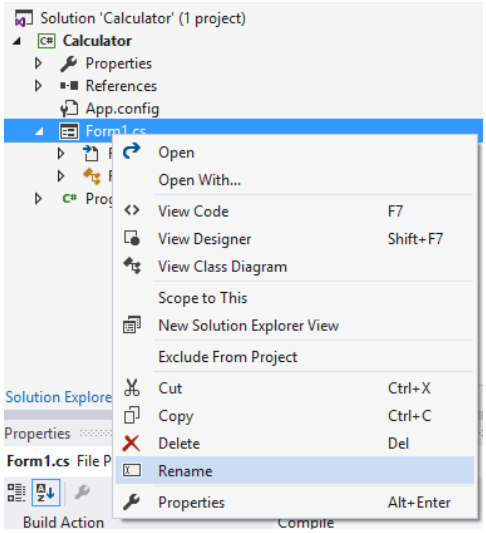
Objectives: At the end of this lab, the student should be able to describe some of the core elements and operations involved in developing a GUI software application. It will highlight some of advance techniques in event handling and understand how to develop and effective GUI.

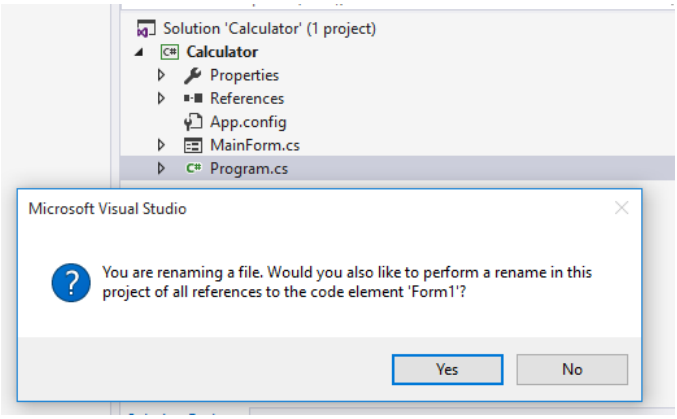
Exercise 1 – Develop the GUI and Event Handling for Basic Calculator Program

Part 1: Creating a Full Numeric Number Pad

1. Under the ***File*** menu, click ***New Project*** or use the ***New Project*** button to create a new project. Alternatively, use the ***Create New Project*** link in the ***Get Started*** *popup* dialog.
2. From the pop-up dialog, select **"C#"** for the ***Language filter***, **"Windows"** for ***the Platform filter*** and **"Desktop"** for the ***Project type filter***.
3. Then choose ***Windows Forms App (.Net Framework)*** and click the ***Next*** button.
4. Type the name of your new project as ***Calculator*** and keep the Solution name the same as Project name.
5. Set the Location to put the project in your own created folder.
6. **Do not** tick on the check-box of [☐ **Place solution and project in the same directory**].
7. Click the **Create** button to start your project.

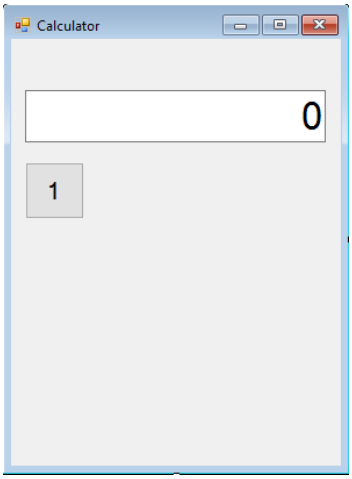
8. In the ***Properties*** window of the ***Form*** control, change the ***TopMost*** property of the ***Form1*** to 'True'.
9. The default ***Form*** file name and class name is always ***Form1***. Let's change and rename it to ***MainForm.***

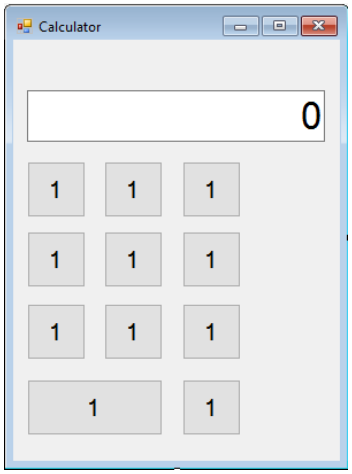


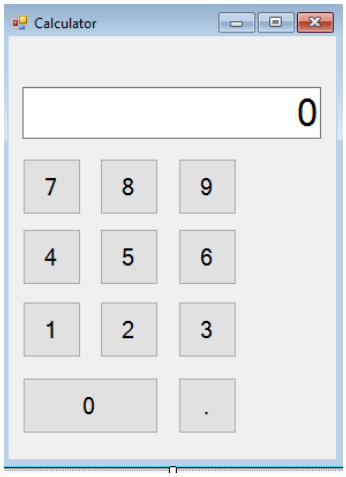


10. **Right click on "Form1.cs"** ***Solution Explorer*** window and select the Rename option to change from ***Form1*** to ***MainForm***.
11. When prompted to rename all references to code element "Form1", **choose 'Yes'.**
12. Double click on "*MainForm.cs*" ***Solution Explorer*** window to launch the ***Form Designer*** tab.
13. Change the ***Text*** property of the ***Form*** from '*Form1'* to '*Calculator'* and the ***Name*** property from 'Form1' to 'frmMain'
14. From the ***Toolbar***, drag in 1 *Button* and 1 *TextBox* control into the ***frmMain*** window area. Modify the properties based on the table below and resize ***frmMain*** accordingly

{Name} From	{Name} To	{Text}	{Font->Size}	{TextAlign}	{FormBorderStyle}
Form1	frmMain	Calculator			FixedSingle
textBox1	txtResults	0	25	Right	
button1	btn1	1	15		







15. Next in the Form Designer, select the Button '1' then copy the button using <Ctrl + C> key. Subsequently paste <Ctrl + V> 10 times and rearrange them according to the figure above.
16. Finally rename the ***Text*** and ***Name*** properties of the buttons such that **'0'** correspond to **'btn0'**, **'1'** correspond to **'btn1'**, **'2'** correspond to **'btn2'** and so on. The button **'.'** should be named **'btnDot'**
17. Build and run your application by hitting <F5> or <Ctrl + F5> key. You should see an application with the UI shown in the diagram above.

No	Actions	Observation / Explanation/Action
1	Try to resize the application Window. Which Control and Property that produces this effect?	
2	Place your cursor on the TextBox, can you edit the value using keyboard?	

	How can you disable this?	

18. Stop the application. At the ***Form Designer***, double click on button '9' to create a *btn9_Click(…)* event handler. Add the following codes.

```
private void btn9_Click(object sender, EventArgs e)
{
    string temp = txtResults.Text;
    temp += '9';
    txtResults.Text = temp;
}
```

19. Build and run your application.

No	Actions	Observation
1	Click button '9'. Did you see any unexpected results?	

2	Coding Task to Fix Problem in Step 1:
	
	`if (temp == "0")`
	`  temp = "";`
	`temp += '9';`	

20. With minor code modifications, repeat step 15 & 16 for button **'0'** to button **'8'** and repeat step 15 only for button **'.'**.
21. Build and test your application.

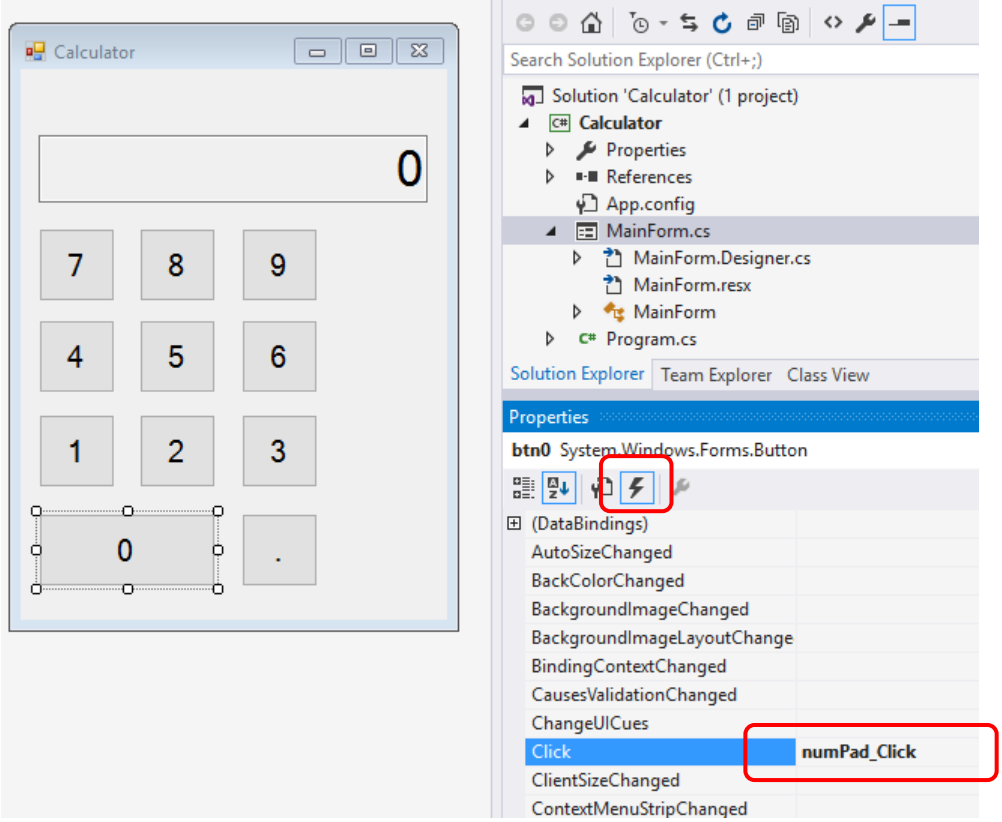
No	Actions	Observation
1	Click button '.' twice Did you see any unexpected results?	

2	Coding Task:
	
	`private void btnDot_Click(object sender, EventArgs e)`
	`{`
	`  string temp = txtResults.Text;`
	`  if (!temp.Contains('.'))`
	`  {`
	`  //Add in your codes`
	`  }`
	`}`	

	Use the codes above to fix the observation in part 1

Part 2: Optimizing Event Handling

1. Using the same solution/project from Part1, open ***Form Designer*** and select button **'0'**. At the **properties panel** click and select the ***event*** button. Next to the Click event, type in *numPad_Click* and press ENTER key.



2. From here the code editor will create and display *numPad_Click (…)* event handler.

From Part 1, we created 10 event handlers for Button '0' to '9' and Button '.' Now we are going to use ONLY 1 event handler for all these 10 buttons.

3. Modify *numPad_Click (…)* event handler with the following codes:

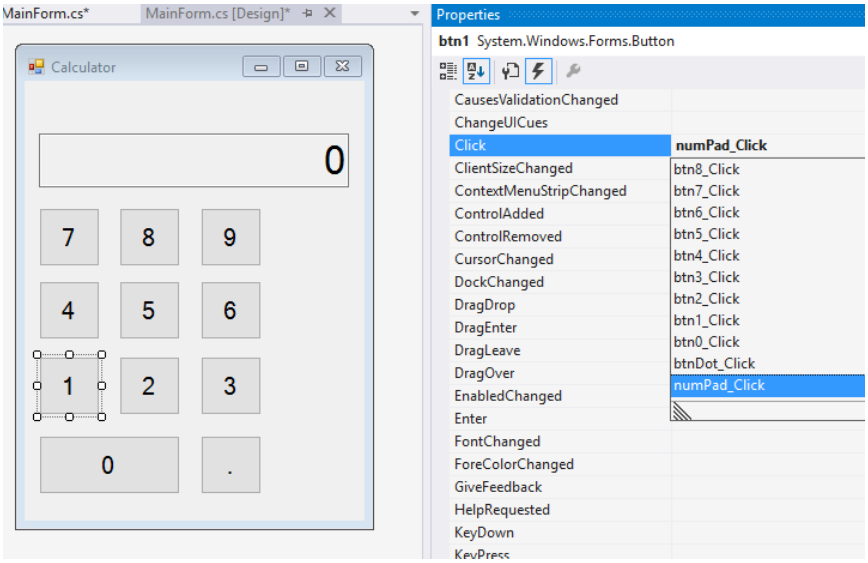
```
private void numPad_Click(object sender, EventArgs e)
{
    Button btn = (Button)sender;
    string num = btn.Text;

    string temp = txtResults.Text;

    if (temp == "0")
        temp = "";
    temp += num;
    txtResults.Text = temp;
```

```
}
```

4. Build and test the calculator application. Ensure that button **'0'** is functionally working.
5. Next modify the ***Click*** event handler for button **'1'** to button **'9'** and button **'.'** to use *numPad_Click (…)*.
6. Select button **'1'**, then at the **properties panel** click and select the ***event***  button. Choose *numPad_Click (…)* as the event handler.



7. Repeat step 6 for button **'2'** to button **'9'** and button **'.'**
8. Build and test your application. Notice what happens when you press button **'.'** more than one time.
9. Modify *numPad_Click (…)* event handler with the following codes:

```
private void numPad_Click(object sender, EventArgs e)
{
    Button btn = (Button)sender;
    string num = btn.Text;

    string temp = txtResults.Text;
    switch (num)
    {
        case ".":
            if (!temp.Contains('.'))
            {
                temp += '.';
            }
            break;
        default:
            if (temp == "0")
                temp = "";
            temp += num;
            break;
    }
    txtResults.Text = temp;
}
```

10. Build and test your application.