# BCIT

**Comp 3951 Technical Programming 1**

**Technical Programming Option**

# Option Head Mirela Gutica

**Winter 2017**

Mark: \_\_\_\_\_\_\_\_ /100

Lab 2 Calculator C#

This is a peer or individual **project**. Submit only one project per peer group. Indicate your partner. The assignment should be handed-in no later than 8: 30am Tuesday January 24, 2017**. No late assignments will be accepted**.

Readings:

<http://msdn.microsoft.com/en-us/library/system.windows.forms.sendkeys(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/library/aa984308(v=vs.71).aspx>

<http://msdn.microsoft.com/en-us/library/7k989cfy(v=vs.80).aspx>

<http://msdn.microsoft.com/en-us/library/vstudio/w0x726c2(v=vs.100).aspx>

<http://msdn.microsoft.com/en-us/library/system.clscompliantattribute(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/library/k5532s8a.aspx>

**Requirements**:

\_\_\_\_\_\_\_\_\_\_/100

1. Implement a calculator similar to a scientific calculator, with the following requirements:
   1. The application is a control box (no minimization/maximization or full screen options).
   2. Use containers (group boxes and panels)
   3. Create your icon (use Visual Studio or the editor of your choice to create the icon)
   4. “On” button starts the calculator.
   5. The calculator performs minimum four operations: addition, subtraction, multiplication and division.
   6. If you enter a number by mistake, you can use **CE** to clear only the current operand. It does not clear previous calculations. **C** deletes all. Backspace deletes the last digit.
   7. Implement multiple operations executed from left to right (i.e. 2 – 5 + 7 =…).
   8. Implement the keyboard functionality (see the Form’s KeyPress event)
   9. Submit one solutions; however be familiar with both following approaches: To activate a button event you will use:
      1. **the SendKeys class** to get the keys and the operators’ values
      2. **the Sender object and the same handler for several keys**.
   10. Create a picture box that loads an image or paint a rectangle.
   11. Add the image as a resource.
   12. Identify what exceptions are needed and add them to your code.
2. Test your calculator on a different resolution
3. **Use good programming style:**
   1. Meaningful naming: e.g., buttonStart, formCalculator, etc.
   2. Comments //
   3. Bloc descriptors for classes and procedures. Use /// (XML comments) at the beginning of a class, method, etc. The class block descriptors should contain:

Purpose:

Input:

Output:

Author:

Date:

Updated by:

Date:

**Based on: (here you have to give the url or the name of the book that you used to solve this problem or if you use their code examples)**

* 1. Mark the assemblies as CLS-Compliant.
  2. Do not repeat code; use functions instead.
  3. Do not leave unused code in your final submission.

**Bonus**:

\_\_\_\_\_\_\_\_\_\_/25

1. Implement memory functions to solve brackets and order of operations.
2. For some calculations, you may want to remember one calculation to use later. You can add a calculation into memory by clicking the **M+** button. Then you recall it later with the **MR** button. You clear the memory with the **MC** button.
3. Note that **M+** will add whatever you have on the display to whatever is already in the memory, so it is good to click **MC** before adding with **M+**, unless you have recently started the calculator.
4. The MS button stores the currently displayed result in memory. In the case when you display only the most recent operand, MS and M+ serve the same function.

Example: You want to add 1 + 2 and then multiply that by the sum of 9 - 2

Click **C** to clear all.

Add **1 + 2** and click =

Click **MC** to clear the memory, then click **M+** to add the value to memory.

Click **CE** to clear previous addition.

Then enter **9 - 2 =**

Click **\*** to multiply by another number.

Click **MR** to recall what is in memory. Then click **=**

You should get a total of 21.

