# CIS2275 C++ Programming Part II NAME:

**Program 5 Simple Calculator on a Windows Form 50 points Due: Tuesday, March 8, 2016**

**Objective: Write a simple class and implement a Windows Form application**

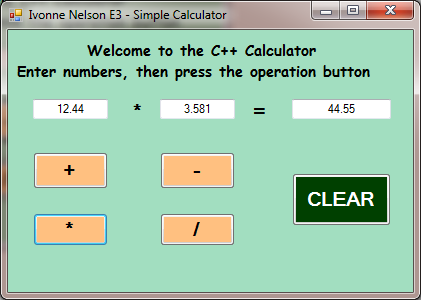
**Turn In Requirements:**

1. **5 pts Name your Visual C++ 2015 project LastnameP5, such as NelsonP5.**
2. **5 pts Upload your project to Visual Studio Online.**
3. **5 pts If you upload to Blackboard, print out the \*.h and \*.cpp files, staple this page to the front of your printed source code when you turn it in for grading.**
4. **5 pts If you upload to Blackboard, remove BOTH debug folders and the .sdf file from your project before uploading it to Blackboard.**

**Program Requirements:**

1. **3 pts Write your name, email address and file name at the top of your source code in a comment.**
2. **5 pts Your main function should have cout statements that write “header” information to the screen. The header info includes your name, course and program information, as well as a 1-2 line description of the program.**
3. **5 pts. Use good C++ programming style and formatting for your program. Use appropriate comments to explain what you are doing.**

This is our introduction to Windows forms. We will learn about the visual side of VS 2015 by programming a calculator.



We will build this form together in the lab.

We will learn to convert the .NET String is retrieved from the textboxes into primitives that we can use.

We will then use our native C++ class to “power” the calculator on the form.

We will also learn to format our output on the form so that our results will be displayed with two decimal places of precision.

You have already written the SimpleCalc class. Here are the modifications to be made: Instead of returning the formatted string, you will return a double, the result. You will set the result into the appropriate textbox on the form.

At the top of the form1.h, add #include "SimpleCalc.h". Create an instance of SimpleCalc on the form just above the declaration of public ref class MyForm : etc.

For the form, you will have 3 text boxes, two for the user’s numerical input and one to display the answer (or result). You will have 4 buttons, one for each type of operation. There will be an additional button, Clear. In the event handler for the Clear button, clear the text boxes and operation label so your form is reset. You will have labels which will present the title and describe the program input required from the user, and 2 labels which will appear to make an “equation” on the form. Put your name and Program title in the Text property of Form1.