

## CS 205 - Programming for the Sciences

### Spring 2008 - In-class Exercise for 01/10/08

This handout is a tutorial on how to write a program using Visual C# of Visual Studio .NET 2005 in the University of Evansville EECS computer labs. It assumes that the user is logged into ACENET.

#### Setting up Visual Studio

Visual Studio (VS) is launched by doing All Programs -> Programming -> Microsoft Visual Studio 2005. You should maximize the window if it is not full size.

The first time you use VS on a particular machine, it may ask you to choose a default setting. Choose Visual C#.

All Visual C# programs must be part of a project. To create a new (Windows form) program in Visual C#, do the following steps:

1. File -> New -> Project. This gives you the dialog box to create one.
2. In the new project dialog box, make sure that Visual C# is selected as the Project type. Click on WindowsApplication to select it as the Template. Type the name of your project in the name box (for this tutorial type **tutorial**). The location should be somewhere on your network drive space, so that you can access your files from any UE public lab. (The default location is probably okay.) This will also give the solution that same name. If you want to change the location, browse to where you want to put the project. There should be a check by "Create directory for solution". Click OK.
3. The instructor prefers to have the Output, the ToolBox, and the Properties windows pinned open. This can be done using View -> Output, View -> ToolBox, and View -> Properties Window.

#### Writing a C# Windows Application program

Generally, a program is written to solve a particular problem. Problem statements may be simple or complex. For today's tutorial, the problem statement is: Write a program to read in an integer and print out its square.

Windows Application programs consist of two parts: a form design that determines what the application will look like when it is run, and a set of handler methods written in Visual C# programming language code that are executed when actions like mouse clicks or key presses are performed. These are explained below.

#### Form Design

The form design is presented as the GUI will appear. There is actual code behind the form, but very rarely does it need to be seen by programmers. All Visual C# programs start with a main form named Form1.

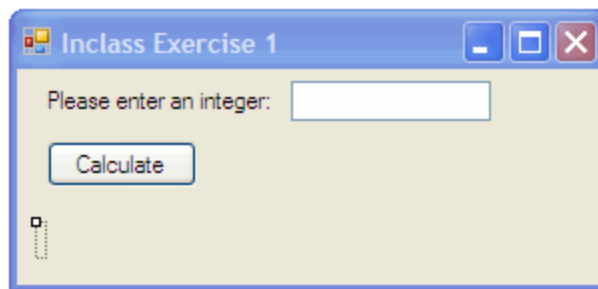
The form design is created by selecting and dropping *controls* from the ToolBox onto Form1. Each item has a set of *properties* that may be set in the Properties Window for that item. For example, almost every item has Color, Font, and Background properties. The Properties Window changes to the appropriate

property list each time a different item is selected in the form design. Try changing the Text property of Form1 "Tutorial Program". This will change the text of the titlebar of the resulting application window.

For many of the programs in this course, we will use just three controls, Label, Textbox, and Button, in the following way:

- Label: the Text property is the text displayed in the label. Labels are used as a prompt for an input data or to display results.
- Textbox: the Text property is the text shown in the box. Textboxes allow users to input data from keyboard.
- Button: the Text property is the text displayed as the label of the button. Buttons causes an action to be perform when clicked.

Create a Windows form that looks like the following:



This is done as follows:

1. Set the Text property of Form1 to "Inclass Exercise 1".
2. Click on Label in the Toolbox, then click on Form1. This will place a Label control on the form. Click and drag it to place it in its final location on the form. Visual C# gives each control a default name and default Text. Set the Text property to "Please enter an integer:".
3. Click on Textbox in the Toolbox, then click on Form1. Click and drag the control so that it is to the right of the label. Note that alignment handles are shown when controls line up with other controls on the form to make it easier to place controls neatly. Because we are going to use the name of this control in our program code, the (Name) property should be set to something more meaningful than textbox1. Set the (Name) property to **anInteger**.
4. Click on Button in the Toolbox, then click on Form1. Click and drag the control to be under the label and textbox. Set the Text property to "Calculate" and the (Name) property to **calculate**.
5. Click on Label in the Toolbox, then click on Form1. Delete the text in the Text property and set the (Name) property to **nsquared**.

### Handler methods

The handler methods that are executed when an action like a mouse-click occurs are written in the Visual C# programming language. Each GUI control has a default handler whose code shell is created when a double-click is performed on it in the form design. For the tutorial program, we need only one handler that is run when the Button is clicked.

Double-click on the Button. A method shell for the handler is created called `calculate_Click`. We fill in the code to compute the results of the program. For this tutorial, that would be to take the integer input by the user and displaying the square of that number. Here is the method shell filled in:

```
private void calculate_Click(object sender, EventArgs e)
{
    // This part is the added code
    int n, result;
    n = int.Parse(anInteger.Text);
    result = n * n;
    nSquared.Text = "The square of the integer is: " +
        result.ToString();
}
```

Type this code into the method shell as shown. Note that this editor automatically indents the code and uses different colors for some of the constructs in C#. E.g., the comments are in green and the reserve words are in blue.

When you are done typing in the program, save it. (File-> Save or click the floppy icon on the toolbar.) For longer programs, you want to do this fairly frequently, and always before you try to compile your program.

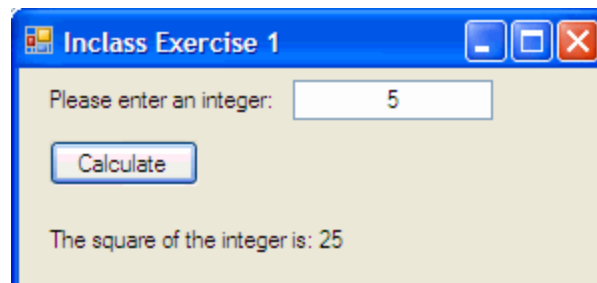
### Building a program

To build the program, do Build -> Build Solution. This will compile and link all of the files in the project into an executable program. (In this tutorial, the executable file `tutorial.exe` would be formed.) Progress from the build can be observed in the Output window.

If you have any syntax errors, the Output window will contain a list of errors and the line numbers. (You may have to scroll up to see them.) Double-clicking on an error message should cause the cursor in the source edit window to be put at that line. Correct any errors and do Build -> Build Solution again.

### Running a program

The integrated environment allows us to run the program from within VS, do Debug -> Start Without Debugging. The system will execute your program. A Windows Application is launched and waits for you to type in an integer in the textbox and click the Calculate button. For example if you enter the integer 5 and click the Calculate button, the result might look like:



Note that if you do not type in a valid sequence of digits in the textbox (including no input at all), the program will crash. We will talk about how to handle this type of error later in the course.

If you change the source file and try to run the program again, the VS system will detect this and automatically rebuilds the program. The system keeps track of what has changed and only does enough to bring everything up-to-date. Note that the IDE will not recompile a program when there is a running instance of the previous version.

Do the calculation a few times using different values for input. Make sure the program computes the correct results. Exit the program by clicking on the Windows Close button.

### **Assignment**

(5 points) Modify the tutorial program in the following ways:

1. Add a Label named **nCubed** to Form1 under the **nSquared** label
2. Add code to the **calculate\_Click** method to compute the number n raised to the third power
3. Add code to the **calculate\_Click** method to set the Text property of **nCubed** to display the appropriate result

Compile and test your program until it runs correctly. Make sure your name is in a comment at the beginning of the file. When you have completed this exercise, print out the file and turn it in to the instructor.

### **Notes about the Express Edition of Visual C# 2005.**

For the most part, the Express Edition works the same way as the VS edition. The main difference is that in the New Project dialog box, you only have to choose the Template (since it already knows you are writing a C# program) and you are only asked for the Project name. Then when you try to save the project, as Save Project dialog box is presented where you can name the project and browse to a location.