CS 1400 Lab 5: Creating and Using TextBoxes

Objectives:

The objective of this lab is to write a Graphical User Interface program that uses Text Boxes to get input from the user and to display output to the user. It is important to note that the Text Property of a TextBox is a Reference-Type of String.

The Interface

The best way to start this lab is to make a copy of the complete project folder that you created for Lab #4. Name this new project folder Lab05. Do not change any of the file names inside of your new folder.

Changing the Interface

1. Double click on the .sln file in this new folder to open up the project.
2. From the the Toolbox, drag two TextBoxes and two Labels onto the Client Area of the form. Change the names on your TextBoxes to give them memorable names. For example, *InTextBox* is much more useful than the default name *textbox1* that Visual Studio creates. Your interface should look something like figure 1 below:

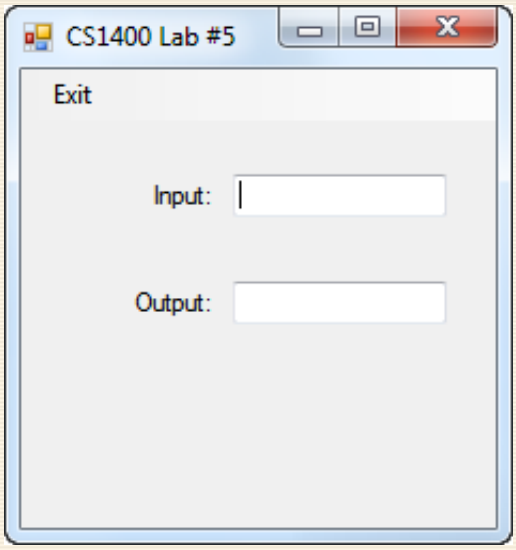

Figure 1: Lab 5 Form and Controls

Creating An Event Handler

The objective of this program is double the value that the user enters into the top TextBox, and display the result in the bottom TextBox. When the user enters a value into the TextBox and presses the tab key, you program should double this value and display the result in the bottom TextBox. When the cursor leaves a TextBox, we say that the TextBox loses **Focus**. To have your TextBox generate an event when it loses focus, follow these instructions (see Figure 2):

1. Select the TextBox that you want to generate the "loses focus" event.
2. In the Properties window, click on the Lightning Bolt at the top of the Window. When you click the Lighting Bolt, you will see a long list of events that can be generated for the Text Box.
3. In the group labeled **Focus** you should see an event called "Leave".
4. Double click on this event name.
5. The skeleton code for an event handler for the "Leave" event will be generated.
6. Now fill in the code that you want executed when this event is generated.

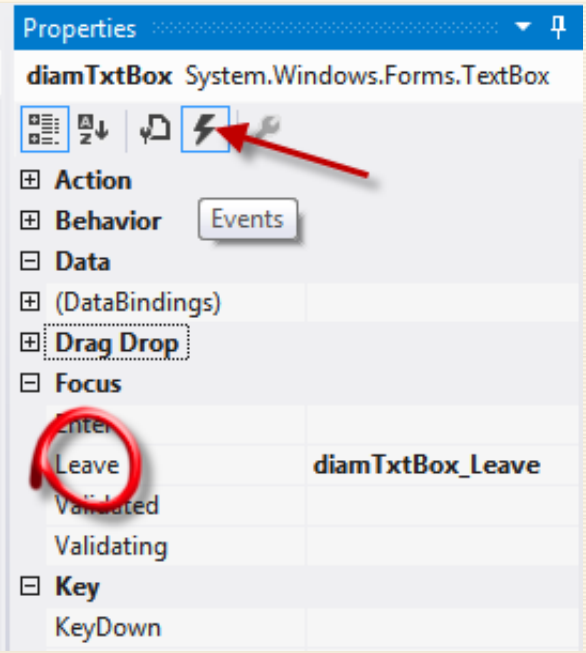

Figure 2: Picking the Leave Event

The code that you enter into your event handler will:

- Get the string that the user typed into the top TextBox.
- Convert it into an integer (int).
- Double it.
- Convert the integer back into a string.

- Display the string in the bottom TextBox.

All of this is required because the data that is shown in a TextBox is always a **string** and when we do arithmetic, we need to use a number. For this example I have named my TextBoxes `inTxtBox` and `outTxtBox`.

```
// The inTxtBox_Leave Method
// Purpose: Get a value from the user and display it
back again
// Parameters: The sending object, and the event
arguments
// Returns: none
private void inTxtBox_Leave(object sender, EventArgs e)
{
    const int DOUBLE = 2;
    int num = int.Parse(inTxtBox.Text);
    int doubledNum = num * DOUBLE;
    string outStr = string.Format("{0:D}", doubledNum);
    outTxtBox.Text = outStr;
}
```

Questions

When you submit your lab to Canvas, answer these two questions by adding a comment to your submission.

Question #1: Given a variable, *cash* that contains a currency amount like $4.56, what would the statements look like that would properly format and output that variable in a TextBox?

Question #2: Suppose that you wanted this program to halve the number input by the user. Write the line of code that would do that.

Now build and test your code. Make sure that your code is properly documented and meets all of the style guidlines. When it runs to your satisfaction submit your lab to Canvas.

Submitting Your Lab

Place your complete project folder into a zip file and name the zip file lab_05_your-initials_V1.0.zip. For example, I would name my file lab_05_RKD_V1.0.zip. Submit this assignment as Lab #5 on Canvas.

Grading Guidelines

Description	Points possible

Assignment meets grading guidelines: o Source code files contain a declaration that you did not copy any code, except that provided. o Assignment has been properly submitted to Canvas o Code meets style guidelines	2
Your user interface looks like the example, and works as directed.	3
You have correctly answered the two questions.	2
Total	7