|  |  |
| --- | --- |
|  | **CS1400 Lab #5**  **Fundamentals of Programming**  **Creating and Using TextBoxes**  **Version 1.1**  **Objectives:**  The objective of this lab is to write a Graphical User Interface Application that uses ***TextBoxes*** to obtain 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 data-type string.  **The User Interface**  The best way to start this lab is to duplicate what you did for Lab #4 and name this new Project folder Lab\_05.  **Changing the Interface**   1. Double click on the .**sln** file in this new folder to open up the project. 2. From the ToolBox Menu, drag two **TextBoxes**, two **Labels**, and a **Button** onto the Client Area of the Form. Your interface should look something like Figure 1 below:   http://debryro.tc.uvu.edu/1400/labs/lab05/form.png  Figure 1: Lab 5 Form and Controls  **Creating Event Handler Methods**  The objective of this program is to have the user type string containg integers into the top TextBox. When you press the Button, your program will read the number from the top TextBox and display the value + 5 back in the second TextBox. To create the event handler that will do this:   1. Double click on the Button. 2. This will open up the code Editor Window and create a method to handle the event that will get generated when a user clicks on the Button. You will add the code below. This code below will 3. Get the string that the user typed into the first TextBox. 4. Convert it into an integer (int), add five (5) to this value. 5. Convert the integer back into a string. 6. Display the string in the second 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. I have named my TextBoxes ***TxtIn*** and ***TxtOut***.  /// <summary>  /// Purpose: Btn Compute to get the input as a string, convert  /// the string to a int add 5 to it and display it with  /// four or more digits in the output.  /// </summary>  /// <param name="sender">BtnCompute</param>  /// <param name="e">Not used</param>  **private** **void** **BtnCompute\_Click**(**object** **sender**, EventArgs **e**)  {  **int** **iValue** **=** 0;  **string** **sValue** **=** **TxtInput.Text**;  **int.TryParse**(**sValue**, **out** **iValue**);  **iValue** **+=** **INC\_VAL**;  **TxtOutput.Text** **=** **string.Format**(**"{0:D4}"**, **iValue**);  }  Now build, test, debug and refactor your code until it meets the required specifications.  **Submitting Your Assignment**  Place your entire 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\_DAF\_V1.0.zip. Submit this assignment as Lab #5 on Canvas. |

|  |  |  |
| --- | --- | --- |
|  | **Grading Checklist** | C (correct)  X (incorrect) |
| # | Program | Submission |
| 1 | Meets & works to specifications | 6 points |
| 2 | Error Free, elegant & efficient | 4 points |
| 3 | Pseudo-Code | -3 points |
| 4 | Style Guide | -1 points |
| 6 | Magic Numbers | -2 points |
| 7 | Project Prolog | -1 points |
| 8 | Method Prologs | -1 points |
| 9 | Zip Filename | Lab = 0 points |
| 10 | Lab & Project Names | Lab = 0 points |
| 11 | Zip File is invalid or will not unzip | Lab = 0 points |
|  | Total Points | 0-10 |