|  |  |
| --- | --- |
|  | **CS1400**  **Lab #04**  **Creating and using a MenuStrip**  **Version 1.0**  **Objectives:**  The objective of this lab is to use Visual Studio to create a Graphical User Interface program that contains and uses a ***MenuStrip***.  **The Interface**  Many GUI projects will include a ***MenuStrip*** with a Menu selection to ***Exit*** the application, and a Menu selection to show an ***AboutBox***. This lab will demonstrate how to create such an application. The ***AboutBox*** displays your name and project information. This simple GUI is shown in Fig. 1 below.    Figure GUI Form with MenuStrip  **Creating the User Interface**   1. Select a folder for this lab assignment. Start Visual Studio and create a new Windows Forms Application. Name your Project Lab\_04. 2. From the Toolbar, drag a MenuStrip onto the Client Area of your Form. The MenuStrip will be pinned to the top, right and left sides of the Client Area. 3. Name the MenuStrip MSpMain. 4. In the Text area on the Menu Strip type ***Exit*** 5. In the Text area below ***Exit*** type ***About*** 6. In the Text area below this, type ***Exit*** again. 7. At this stage your form should look about like Figure 2.     Figure - Completed MenuStrip  **Creating Event Handlers Methods**   1. Double click on the Menu Item ***Exit*** (the second ***Exit*** down (see Red Arrow), not the one on the MenuStrip itself). 2. This will open up the code Editor Window and create a event handler **method** that gets called when a user clicks on ***Exit Menu Item***. 3. Add the following line of code to close the Window, as shown below. The Form's ***Close( )*** method closes the Window. When the Window closes, the program terminates.   /// <summary>  /// Purpose: To respond to the Exit menu click event and close the program.  /// </summary>  /// <param name="sender">Exit Menu</param>  /// <param name="e">Not used</param>  **private** **void** **exitToolStripMenuItem1\_Click**(**object** **sender**, EventArgs **e**)  {  **Close**();  }   1. Double click on the Menu item ***About***. This will add a the event handler method, to handle the event that is generated when a user clicks on ***About Menu Item***. 2. Add the line of code to show a ***MessageBox*** as shown below. The ***MessageBox.Show( )*** method takes two strings, Buttons, Icon enumerated types as arguments, that is displayed in the ***MessageBox***. Of course, you will supply your own name.   /// <summary>  /// Purpose: To respond to the About menu click event and display a MessageBox  /// </summary>  /// <param name="sender">About Menu Click Event</param>  /// <param name="e">EventArgs Object</param>  **private** **void** **aboutToolStripMenuItem\_Click**(**object** **sender**, EventArgs **e**)  {  **string** **aboutMsg** **=** "Dennis A. Fairclough\nCS1400\nLab #04";  **string** **headerMsg** **=** "About Dialog Box";  MessageBox**.Show**(**aboutMsg**,**headerMsg**,MessageBoxButtons**.YesNo**,MessageBoxIcon**.Information**);  }  My ***About*** Box is shown in Figure 3.    Figure - MessageBox Displayed  Now build, test and run your code.  **Documenting your Code**  Before you turn in your lab be sure that you have written a complete Project and method Prolog’s in your source code file (this is done in Form1.cs), and that you have written method prologs for each method you have added to your Form class. For example, the complete documented method to handle the Exit Menu event should look something like this:  /// <summary>  /// Purpose: To respond to the Exit menu click event and close the program.  /// </summary>  /// <param name="sender">Exit Menu</param>  /// <param name="e">Not used</param>  **private** **void** **exitToolStripMenuItem1\_Click**(**object** **sender**, EventArgs **e**)  {    **Submitting Your Assignment**  Place your entire Project folder into a zip file and name the zip file Lab\_04\_your-initials\_v1.0.zip. For example, I would name my file Lab\_04\_DAF\_v1.0.zip. Submit this assignment as Lab #4 on Canvas. |

|  |  |  |
| --- | --- | --- |
|  | **Grading Checklist** |  |
| # | 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 |