CIS 3309 Lab Assignment 1 - Chapters 1-3

**(Last revised December 28, 2019)**

**WE ARE AWARE THAT THE ANSWERS TO ALL THESE EXERCISES ARE AVAILABLE ALMOST EVERYWHERE.**

**HOWEVER, if you do not do them yourself and you will not learn what you need to learn and you will have more than a little trouble as the course moves forward. As always, it is your choice. But, the wrong choice has consequences.**

Please read Chapters 1-3 and understand the material before Lab 1 begins.



In the first lab we will start to use Visual C# 2017 and learn about the integrated development environment (IDE). We will cover the first 3 chapters. You should not be too intimidated by the quick pace in lab. You should already have an understanding of structured and object-oriented programming from Java. Programming in C# is similar and likely to be easier than Java because of the rich features offered in the IDE. For specifics on how to get started using Visual Studio with C#, be sure to read Chapters 1 and 2.

* Chapter 1 is a tour of the Visual Studio IDE.
* Chapter 2 shows forms and useful components that can be placed on them.
NOTE: If you have a problem with the VS 2017 window that opens when you click on New Project you may need to look in your VS Options window. To do this, pull down the Tools menu and select the Options entry (the last entry in the Tools menu). Open the Projects and Solutions entry on the left side of this window and select General. Among other things: 1) note the location that VS is using to store your projects, and 2) make sure all the options on the right side (bottom) are checked.
* Chapter 3 shows how to place code in form components to perform operations and how to test the code.

The text web page is: <https://www.murach.com/shop/murach-s-c-2015-detail> You can download pdf files of the chapters 2 and 3 if you do not have the text. You can also download the project files for the text at this site. (Once on this web site, click on the **Free Downloads** Button about a third of the way down the page. You probably will want to download the zipped file and then unzip it at your convenience. The projects you will need for the Exercises required in the first four weeks are in the “Exercise starts” directory.)

**Note that once you download the files from the Murach directories, you need to go to your own Downloads directory, find the just downloaded file (it should be highlighted), right click on this file (it should be named “cs15\_allfiles.exe” and then click on “Open the Containing Folder”. In the containing folder find the file “cs15\_allfiles.exe” and double click on it to run it.**

**Your files will be stored in C:\Murach\C# 2015. I would urge you to make your own copy of the *Exercise Starter* directories so that you do not have to modify the originals. This way, if you need to refer to the originals, you will not have to download them again.**

**If you are in the CIS lab, you will want to copy them to your directory on the G drive (and to your flash drive) using the same directory structure provided in the download. It is suggested that you first create your own cis3309 directory and set aside a separate directory for each Lab (such as Labs 1-4, and each Project, such as Bingo and the ATM Project). Your lab assistant will have more to say on this in Lab.**

This lab assignment is to complete the exercises for the first 3 chapters in lab. I suggest you start with Exercise 1-1, then go to Exercise 2-1 and finally do Exercise 3-1, which is to be turned in. (Do not turn in Exercises 1-1 or 2-1 as 3-1 incorporates your work in these exercises. Be sure to do all the steps in each Exercise, as this is the only way you will gain some mastery over the Visual Studio IDE.

***Please be sure to save your work periodically as you proceed and also back it up. You may wish to consider storing it in your Blackboard backpack and zipping it up and storing it on your flash drive.***

***If you are going to zip an application in order to store it, BE SURE TO FIRST CLOSE Visual Studio.***

***If you do store information on your flash drive be sure to copy it to a hard drive on your computer before working with the project.***

**One last note: Unlike some other languages, such as Visual Basic, C# is a case-sensitive language. Thus, unlike VB, your C# editor considers "text" and "Text" to be two different identifiers. Similarly, for "decimal" and "Decimal", or "subtotal" and "subTotal". If you make case sensitive errors, the C# editor will make no effort to correct them.**

**The Remaining Material Does NOT APPLY FOR Spring 2020**

**Exercise 2-1 X: Design a Simple Form**

In this exercise, you’ll design a form that lets the user enter a number grade and then displays the letter grade when the user clicks the Calculate button. You are just designing the form here – that is all.



1. Start a new project named CalculateLetterGrade in the Extra Exercises\Chapter 02\CalculateLetterGrade directory. Be sure to store the solution in its own directory.
2. Add the labels, text boxes, and buttons to the form as shown above. Then, set the properties of these controls as follows:

Default name Property Setting

label1 Text &Number grade:
TextAlign MiddleLeft
TabIndex 0

label2 Text Letter grade:
TextAlign MiddleLeft

textBox1 Name txtNumberGrade
TabIndex 1

textBox2 Name txtLetterGrade
ReadOnly True
TabStop False

button1 Name btnCalculate
Text &Calculate
TabIndex 2

button2 Name btnExit
Text E&xit
TabIndex 3

1. Now, set the properties of the form as follows:

Default name Property Setting

Form1 Text Calculate Letter Grade
AcceptButton btnCalculate
CancelButton btnExit
StartPosition CenterScreen

1. Now at the very top of the Form add a label (lblName) with the text that reads as Enter Your Name: with the text left adjusted. Then, immediately to the right of this label insert a text box (txtName). To the right of the text box insert an OK button (btnOK) with centered text OK.
2. Reset all the tab indices so that lblName has tab value 0, txtName has value 1, btnOK has value 2, and all the other tab indices you have already set have values starting at 3.
3. Use the Form Designer to adjust the size and position of the controls and the size of the form so they look as shown above.
4. Rename the form to frmCalculateGrade. When you’re asked if you want to modify any references to the form, click the Yes button.
5. When all done, disable all 5 controls shown in the form at the top of this exercise. These 5 controls (excluding those you just added in step 4.)
6. Save the project and all of its files. Be sure to do this correctly and do not forget to back it up to a cloud (of your choosing) or to your own personal flash drive.

**Exercise 3-1 X: Code and Test the Calculate Letter Grade Problem**

In this exercise, you’ll add code to the Calculate Letter Grade form that you designed in extra Exercise 2-1X. Then, you’ll build and test the project to be sure it works correctly. Please use the variable names you are given in this specification below.

1. Open your completed version of Exercise 2-1X (as set forth above).
2. Display the form in the Form Designer.
3. Insert code to set the focus on your form to the txtName textbox.
4. Double-click the OK button to generate an event handler for it. In this event handler, write code to check the contents of txtName to be sure it is not blank. If it is blank, display a message box indicating “Name Cannot Be Blank. Re-enter.” And reset focus to the txtName textbox. If this text box is not blank, you can enable the five controls that follow these on the form.

You can look up how to display a message box on pages 190-193 of your Murach text. Just look at the examples if nothing else.

1. Double-click the Calculate button to generate a Click event handler for it. Then, add this statement to the event handler to get the number grade the user enters:

decimal score = Convert.ToDecimal(txtNumberGrade.Text);

1. Add this statement to the event handler to declare and initialize the variable that will hold the letter grade:

string letterGrade = "";

Then, add an if-else structure (See Chapter 5 of the Murach text, page 143 perhaps) that will

 set the letter grade to A if the score is between 90-100 inclusive,

 set the letter grade to B if the score is between 80-89 inclusive,

set the letter grade to C if the score is between 70-79 inclusive,

 set the letter grade to D if the score is between 60-69 inclusive,

 and set the letter grade to an F if the score is between 0 and 59.

If the score is out of range 0-100 set the letter grade to Z.

1. Add this statement to display the letter grade in the Letter Grade text box:

txtLetterGrade.Text = letterGrade;

1. Finally, add this statement to move the focus back to the Number Grade text box:

txtNumberGrade.Focus();

1. Return to the Form Designer, and then double-click the Exit button to generate a Click event handler for it. Then, add this statement to the event handler to close the form:

this.Close();

1. Run the application, enter a number between 0 and 100, and then click the Calculate button. A letter grade should be displayed and the focus should return to the Number Grade text box. Next, enter a different number and press the enter key to display the letter grade for that number. When you’re done, press the Esc key to end the application.