CIS 3309 Lab Assignment 2 – Chapters 4-5

**(Last revised September 2, 2020)**

Please read Chapters 4 and 5, and understand the material before Lab 2 begins. (Read over also, Chapter 9)

Chapters 4 and 5 build on what we learned in Lab Assignment 1, which you need to have done by now. We will cover the instantiation and use of numeric and string data types and coding control structures.

* Chapter 4 covers the use of numeric and string variables.
* Chapter 5 covers control structures.

The lab assignment is to complete Chapter 4 Exercise 4.2, and Chapter 5 Exercise 5.2 as shown below (these are modifications of the exercises in the Murach text). Again, you are to do the modified versions of these exercises as outlined below.

NOTE: In these and subsequent text exercises, you will be asked to either modify some lines of code and/or replace some lines. I strongly urge you to **first comment out** any line you are about to change or about to replace (use the // comment form) and then enter the modified or replacement line you need. Modifying or deleting original, working lines of code prior to being certain your code does what you want is not a good idea.

**Reminder: 25% of your grade is based on**

**1. proper and consistent naming of variables, and class instances (objects)**

**2. meaningful comments (not over done nor “under done”) ; used primarily to explain the purpose of your software components (classes and functions).**

**The purpose of any classes you create should be clearly described, and the purpose of each function should also be documented. An additional 25% is for clean, well organized code, including well thought out class and function structures. If your program works but has sloppy code, bad variable names, and no comments, you will only get 50% of the maximum grade. This is a good time to start developing and practicing good habits.**

## Exercise 4.2: Enhance Invoice Total Application

## Validation of input data is one of the most important things you can do in any project you develop. When you have completed the tasks for this exercise as outlined in the text, add the following capabilities to your project.

## Be sure to statically disable both the Enter Subtotal label and the Enter Subtotal textbox. If you do not know what this means, ask.

## Add a label, a text box, and an OK button (see Exercise 3.1 X if you need a refresher) to your form to enable the user to enter his or her name and click OK.

## In the handler code for the OK button, first check (as you did in Exercise 3.1 X) to ensure that the name is not blank. If the name is blank generate a message as requested in Exercise 3.1 X, reset focus and have the user try again to enter a non-blank name.

## Once a non-blank name has been entered, dynamically (in your own code) enable the Enter Subtotal label and the Enter Subtotal text box.

## Now in your handler code for btnCalculate, insert try-catch code (see p. 193 for an example) to catch an invalid decimal value entry and ask the user to re-enter his or her data.

## Test your project and if the code you have added does not catch a negative input value, make sure you correct this problem and retest your project.

## 

## Exercise 5.2: Develop the Future Value Application

After completing the Text Exercise Exercise 5.2),

1. If you have not already created a form-load event handler, do so now and have it call the void method clearForm (see 3. below).

2. Add a button at the top left of the form with Text property "Enter Data". The button click event for this button should enable the four labels and four text boxes disabled at form load time and set the Text fields of the text boxes to "". It should also disable the "Calculate" button. Then set the form focus to the Monthly Investment text box.

3. Write a function clearForm to be called by the form load event handler, which sets all four text boxes on the form to "", and then disables all all four data labels and all four text boxes and the "Calculate" button.

4. Move the "Calculate" button " above the label and text box for Future Value.

5. For the last step of the button Calculate click event handler, insert the line of code

MessageBox.Show ("If you want another calculation press Enter Data button.", "Re-run!");

6. Double click on the Number of Years textbox to create the text\_changed even handler for this control. Insert one statement into this handler – a statement to enable btnCalculate.

1. Test your program two or three time to be sure it works. Then answer do the following:

List three possible advantages of your new form design over the original form as specified in the Text Exercise 5.2