**Create the application described below:**

* Similar to the Chapter 3 project, you will be creating a program to calculate your current grade average. You will use the grades for the assignments completed up to this point  If you have an assignment that has not been graded yet, you can make up a score for that assignment.  (use the Gradebook to see all graded items for the course)
* You will have 4 buttons on your form: Enter Scores, Save Scores to File, Load Grades from File, Exit.
* There are four categories of scores: Tests/Programming Projects/Tutorials/Quizzes.
* Use arrays to store your grades. These need to have class level scope. You will need an 4 arrays (one for each category). Keep in mind that each array will not be the same size, since you have 7 graded tutorials, 8 chapter tests, 6 quizzes, and 7 projects at this point in the term. Pay close attention to how many scores you have in each category.
* Only the points earned should be entered by the user.
* Use constants for the total possible points per category. Example: Tutorials are worth 25 points each and you have completed 7 tutorials. (7\*25=175) But since we will be dropping one score, the maximum possible points for this category will be 150. Your constant for Tutorial Possible Points would have the value of 150 assigned.
* Ignore the Extra Credit and the Intro Post points for this assignment.
* You may assume all quizzes are worth 20 points for this assignment.

**Create a ProcessScores procedure to do the following:**

* The lowest score in each category (Tests/Programming Projects/Tutorials/Quizzes) should be dropped.
* Determine the lowest score in each array. Be sure those scores are not included in the totals.
* Use a listbox for each category of scores.
* Clear out any data previously displayed in the listboxes. Display the scores from each category in each listbox.
* Calculate the total number of points earned based on the user input. Remember to subtract the lowest score in each category from the total points earned.
* Calculate the course average and display as a percentage with 2 decimal places.
* You should have the following output: 4 listboxes displaying each set of grades from each category, total possible points (after dropping one score from each category), total points earned (after dropping one score from each category), and the overall grade percentage. You do not need to display a letter grade.

**Enter Scores Button**

* The user should click the **Enter Scores** button to start entering grades through a loop and an InputBox.  There will need to be a separate loop to input each category of assignments; be sure you put the input data from each loop into the correct array.
* The message of the InputBox should display what gradebook item is being entered. (Don't worry about using the actual name of the assignment. Just have a message such as "Enter score for Test #1", "Enter score for Test # 2", etc up through "Enter score for Test 8").
* This event procedure should load each array with the inputted scores and then call the ProcessScores procedure to display results.

**Save Scores to File Button**

* Create a **Save Scores to File** button to save the entered grades to a text file. If the user clicks this button before data has been entered, display a message stating "*No scores have been entered. Click Enter Scores button to enter your scores*."
* In the **Save Scores to File**event procedure, create a new file with all of the scores. Name the file MyGrades.txt.  If another file exists already, just replace the data with the new data (do not append data to an existing file).
* Using the data from the arrays, write one score per line in the following order: 7 tutorial scores, 7 project scores, 6 quiz scores, and 8 test scores.
* At the end of this event procedure, you will close the file.

**Load Scores from File Button**

* Create an **Load Scores from File**button to display saved grades (from the file) to the form.
* Open MyGrades.txt file.
* If the file doesn't exist, display a message stating, "*No scores have been entered. Click Enter Scores button to enter your scores*."
* Load the scores into each array (overwriting any previous data in the arrays). Remember the data will be on the file in this order:7 tutorial scores, 7 project scores, 6 quiz scores, and 8 test scores (one score per line).
* Call the **ProcessScores**procedure to display output.
* Close the MyGrades.txt file.

**Create an Exit button to exit the program.**

**Instructions:**

1. Add comments at the top of your code file for your name and the date.  Include other comments as appropriate based on the examples in the textbook.
2. Change the text property on the Form to display your first and last name in the title bar of the form (for example, Jane Doe's All COP1000 Grades)
3. Insert the Option Strict On and Option Explicit On statements at the beginning of your code. (You are required to add these two statements to your code even if you have the settings turned on in VB.) See below for how this is done:
4. 'Written by Mary Smith
5. 'January 12, 2015
6. Option Strict On
7. Option Explicit On
8. Public Class Form1
9. .....and then all of your program code.....
10. All controls (user interface objects) on your form must have meaningful names with appropriate prefixes (for example, btnExit for an exit button).
11. All variable names must have the proper prefix and have meaningful names.
12. Submit your All COP1000 Grades project using this online drop box. Zip your Visual Basic project file as AllCOP1000Grades.zip and attach to this drop box by using the buttons at the bottom of this page.
13. Before you submit your program, enter your grades into the InputBoxes, run your program to calculate your current average.
14. Projects cannot be submitted late. Check the online calendar for the exact due date.