VSFX 375 – Blackboard OOP – Exercise 3

DATE DUE: see class notes

Hand in through the Dropbox as per the submission guidelines.

Blackboard:

Write a program in C++ that reads in grades and their corresponding weights and prints out the following information:

1. Grades entered and their weighting.
2. Average for the student.
3. Letter grade for the student.

This program should be written so that you could use it to compute your grade average in any of the classes you take this quarter, regardless of number of assignments or weighting based on those assignments.

The program should be able to be run for multiple students, not just one. This will involve looping. In addition, it would make the most sense to enter the weighting once and then multiple students.

Submission Guidelines:

If using Visual Studios, submit your files as ***F23_V375_E1_LastnameFirstname.zip***
or (dependent on the IDE you have chosen) in a folder ***F23_V375_E1_LastnameFirstname***

DO NOT RENAME YOUR PROJECT DIRECTORY, just your zip file if using Visual Studios. Visual Studios IDE uses the project name internally and your .sln file will not work properly if you rename the project even if your .cpp file is fine.

Geany requires makefiles only when using multiple files (which will be covered later during the quarter).

This assignment will be completed in two phases. Exercise 1 and Exercise 3.
- The first phase (Exercise 1) is to get you familiar with the syntax of C++. This should use loops and arrays or vectors.
- The second phase (Exercise 3) will require you to use **classes and objects** and will not be tackled until after we have covered those topics (and is to be completed after Exercise 2).

Sample data is not given – you are expected to design your own sample data and test your program accordingly. I will be critiquing your design choices as we progress through the class.

Grading Guidelines:

<u>ANY PROGRAM HANDED IN THAT DOES NOT COMPILE WILL RECEIVE A FAILING GRADE!</u>

Design and Debugging (30 points)

- 30 points if the program works exactly as stated in the design specifications given.
- 15 points if the program compiles, does not crash, and displays some of the behavior but has some or minor bugs or omissions.
- 0 points if the program does not compile, or compiles but crashes immediately, crashes part way through, or has major bugs or omissions.

Use of loops (20 points)

- 20 points if the program uses loops where appropriate and with correct syntax
- 10 points if the program has bugs with respect to the loops
- 0 points if the program does not use loops correctly

Use of Arrays/Vectors or Classes (30 points)

- 30 points if data structures are defined properly and are used properly
- 15 points if the program has bugs but uses the correct ideas and concepts in the right places
- 0 points if the program does not include any high level constructs, but simply uses variables.

Coding Style (10 points)

Use of Comments (10 points)