EECS 211 – Spring Quarter, 2010
Program 1

Due Monday, April 12th, 2010 at 11:59PM

At http://www.eecs.northwestern.edu/~iraicu/teaching/EECS211/code/program1.cpp, you will find a very simple program that is supposed to find the average of a set of exam scores. That program has many flaws, among which are that there are no checks for invalid scores or for division by zero. In addition, a user can enter only one set of scores; if the user wants to average several sets, the program has to be restarted again for each set.

You are to augment the basic program by adding the following features:

- check for scores outside the range 0-100
- check for no valid scores before calculating the average
- prompt the user for another set of scores after the average has been printed

In addition, you should change the way that the program determines when all scores for one set have been entered; your program should ask the user after each score has been entered whether or not there is to be another score for that set. Finally, the output from the program should include words in addition to the computed average.

Write the code to implement the new averaging program. Submit this part to the Blackboard drop box for Assignment 1.

YOUR CODE MUST INCLUDE COMMENTS AND MUST START WITH A COMMENT IDENTIFYING YOU AND THE ASSIGNMENT AND THE DATE.

You must have a Makefile for your program 1. Please submit your source code and Makefile in a zip archive file on Blackboard using the Assignment 1 interface ("View/Complete Assignment link").

Suggestion:

It is rarely a good idea to try to make several changes at once to an existing program. It is much better to identify small individual changes that you can make in succession and then to do one change at a time. You make the first small change and test it and fix any mistakes. When that change is ok, move on to the second change, and so on. In this assignment there are several independent steps that are each quite easy.

1. Add text output to the existing program – prompt for input, informative output.
2. Change the way the program terminates one set of scores.
3. Add validity tests and check for division by 0.
4. Add the outer loop that prompts for additional sets of scores

Do item 1 until you are satisfied with the way the output looks. Then do item 2, etc.