CSCE 1040 Homeworks 2 and 3 Fall 2015

For this assignment we are going to continue with the simple Grade Book that we started in Homework 1 and also include a makefile for the homework 3 portion described at the end.

For the purposes of this grade book you should still provide a menu with the following options:

1. Add a new course
2. Add a new student
3. Add a student to a course
4. Add grades for a student in a course
5. Print a list of all grades for a student in a course
6. Print a list of all students in a course
7. Compute the average for a student in a course
8. Print a list of all courses
9. Print a list of all students
10. Compute the average for a course
11. Store Grade book (to a disk file)
12. Load Grade book (from a disk file)

Each of these menu items should correspond to a function you will write.

Each student and course is, of course, now represented by a structure.

Modify your grade book so that it uses dynamic memory allocation to eliminate the need for upper bounds on the number of classes, students, students per class and classes per student. You may still use fixed size arrays for the strings representing class names and student names or the C++ String class – it is your choice. The grades for each student and each class should also be dynamically allocated.

Things to consider:

1. How will you know how many classes there are?
2. How will you know how many students there are?

3. How will you know how many students are in a course?
4. How will you know how many grades there are for each student?
5. How will you maintain relationships between students, courses and grades?

Be sure to create a written algorithm, or recipe, for each of these functions and the main program. You will submit this in a Word or PDF document as well.

Be sure to attend class lectures as we will discuss some of the questions listed above!

Program Requirements

Your program will be written in C/C++ not any other computer language.

You will include the steps in your algorithm in your code. You may, of course, paraphrase them if you like.

Your program will be graded based largely upon whether it works correctly on a CSE Department Linux machine.

Your program will also be graded upon your program style. At the very least your program should include:

A consistent indentation style as recommended in the textbook and in class.

Meaningful variable names.

A block header comment section that includes: Your Name and Email Address, and a brief description of the program.

Your program's output should initially display the department and course number, program number, your name, and your email address – as in Homework 1.

Be sure to create appropriate test data and execute tests for proper and improper data on all functions.

You will submit your program source file to the BB Learn website under the **"Homework 2 dropbox.** Make sure you submit your program before the due date and time. You must also submit your design, or recipe file, and a short report about your efforts. In total you will submit 5 or 6 files (1 report, 1 design 2 *.c and 1-2 *.h) You may end up with only 1 .h file which is also fine. Under no circumstances should you have more that a total of 8-10 files including the design and the report PDF files. Your files should all reside in a flat directory structure and only require a single and simple gcc/g++ compile line.

Please be sure and test your program to make sure it is calculating the result properly. You can either do this by hand (calculating some test values on paper to see if they match what your program says), or temporarily display various intermediate values you're calculating in the process and desk check the results to make sure they are correct. The more test cases you try and you get correct answers, the more certain you will be that when the grader uses his own test cases that your program will produce the correct result.

CSCE 1040 Homework 3 Fall 2015

For this assignment we are going to create a makefile that will manage the code and header files created for Homework 2.

The assignment is relatively simple:

1. Create a properly formatted and working Makefile for the program files in Homework 2
2. Use the makefile to properly compile and link the files from homework 2.
3. Make sure the makefie operates correctly when changes are made to only one of the files

<u>Program Requirements</u>

You will not need to create a design file for this assignment. You will still need to create a report discussing the problem, the experience you had in solving it, and evaluating the results.

You will submit your makefile to the **"Homework 3"** drop box, along with the your report. Be sure your makefile is based on the source code and header file names that you submitted in the Homework 2 dropbox, as these will be used to grade this portion of the assignment. Make sure you submit your files before the due date and time. In total you will submit a minimum of 2 files (1 report, 1 makefile)

Please be sure and test your makefile to ensure that it works correctly.