CSCE 1040 Homework 5 Fall 2015

For this assignment we are going to implement the Smallsville Library Management System that we designed in Homework 4

For the purposes of this assignment you will need to create a user interface menu that matches your design as well as implement each of the entities as classes in C++. Not we will be using C++ and you will need to use the g++ version of the compiler.

You ***may*** use any of the C++ STL classes for this assignment.

You may need to modify your design from Homework 4 based on grader comments and class discussion. You will need to turn this updated design in as well as use it for the basis of your program

Be sure to attend class lectures as we will discuss many of the topics you will need to complete the assignment!

Program Requirements

Your program will be written in C++ not any 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 5"** drop box. Make sure you submit your program before the due date and time. You must also submit your updated design, or recipe file, and a short report about your efforts. In total you will submit 6 files (1 report, 1 design 2 *.cpp and 2 *.h)

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.