

CSCE 1040 Homework 2 Fall 2017

For this assignment we are going to revisit the simple Grade Book and upgrade it”.

For the purposes of this grade book you should first 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.

For this program you will create a class to represent each of the basic entities, as well as a class to represent each of the collections. For the collections you may use a vector as the data type, but you will still need to provide wrapper functions to implement the application specific portions of the assignment.

Student should have an ID, a name, and a classification

Course should have an ID, a name, a location and a meeting time

Enrollment should have a ID, a student ID, a course ID and a list of up to 10 grades, an average grade and a letter grade based on a standard 90/80/70 grading rubric with rounding

Grades will be whole numbers only (no fractional part). Averages will be saved and displayed to the hundredths position.

As indicated in the menu you will need to store and load using a disk file so that the data is retained.

- There are no restrictions on the number of courses or the number of students
- A course cannot have more than 48 students enrolled
- A student cannot take more than 5 classes at a time

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

Program Requirements

Things to consider:

- What type of data structure(s) will I use?
- How will I know where the end of the grade list is for each student?
- How can I associate text with the course number and student number?
- How will I build the reports?

You should use C++ classes and vectors to contain your collections. You may use fixed arrays for course grades but you need to be able to mark where the last grade occurs.

Your program will be written in C++ Style syntax only, 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

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

Each class you create should have a separate header file (.h) and code file (.cpp) and main will be in a separate code file (.cpp) since there will be 6 classes plus main that means you should have a total of 7 .cpp files, 6 .h files plus two pdf files for your algorithm and report. Also create a makefile called "Hw2make.txt" to properly compile and link your executable program include this with your other files. This means a total of 16 files.

You will submit your program source file to the BB Learn website under the **"Homework 2"** drop box. 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. Your report should be a "post-mortem" on the assignment. What was hard? What was easy? Do you need to review some things? Were you able to complete the assignment in time? Etc.

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.