

CSCE 1040 Homework 1 Fall 2015

For this assignment we are going to build a simple Grade Book using a versatile data structure.

For the purposes of this grade book you should 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 will be represented by a structure.

The student structure should contain a numeric ID number and a text-based name as well as any other info you need to complete the representation of a student. Similarly the course structure will contain a numeric course number and a text-based name for the course plus any other required information. **You may add other fields to each structure as needed and even create other structures if needed.**

1. Be sure to include the description of the structures you will be using along with the descriptions of the fields they contain as part of your design document.

2. store the declaration of the structures in a header file

3. store the prototypes for all the menu functions in a different header file

4. Store the definitions of the menu functions in a separate C file

5. link all of them together during compiling

So for this assignment you will have at least 1 header files (.h) and 2 code files (.c)

We are using static declarations of arrays, etc. No dynamic memory allocation is allowed (yet!).

Grades will be whole numbers only (no fractional part)

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

Here are some limiting values to help you in defining your data:

- Maximum Number of students (total) 100
- Max number of courses 25
- Max number of courses per student 4
- Max number of grades per student per course 10
- Max number of students per course 20

Be sure to create a written algorithm, or recipe, for each of these functions and the main program. You will submit this in a 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 will I relate the students to the course and vice-versa?
- How will I build the reports?

Your program will be written in C, not in C++ or any other computer language. You may use C or C++ style I/O (e.g. cin and cout instead of printf and scanf) for console and file I/O. You may also use the String class instead of C String arrays. You may not use the STL or any other object-based libraries (Except String). If you are unsure then please ask.

You will include the steps in your algorithm in your code as comments. 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. Be sure to compile and test it there, even if you do the majority of your development on another platform.

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

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 Moodle website under the **"Homework 1"** 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. In total you will submit at least 6 files (1 report, 1 design, 2 *.c or .cpp and 1 *.h)

Please be sure and test your program to make sure it is calculating the results 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/her own test cases that your program will produce the correct results.