CSIS 112: Lab 7—Grade Book II

*[Courtesy of Dr. Terry Metzgar and Adapted from Starting Out with C++, by Tony Gaddis, 4th edition, Scott/Jones Inc, pg 1030.*

*]*

**Instructions**:

Design a class called *StudentInfo*. It should have member variables for the following information:

Student Name

Student ID (a numeric data type)

Major (e.g., Computer Science, Engineering, etc)

Write appropriate member functions to store and retrieve information in the member variables above. The constructor should dynamically allocate memory to hold the student name. The destructor should release the memory.

Design another class called *StudentGradeInfo*. This class should be derived from the *StudentInfo* class. It should have member variables for the following information:

Test grades (The array or vector should hold up to 5 grades)

Write appropriate member functions to store and retrieve information in the member variables above. I would suggest a member function to allow the addition of one grade at a time to Test grades.

The == operator should be overloaded to allow a comparison of two student ID numbers to see if they are equivalent. The << operator should be overloaded to output, neatly formatted, the information associated with a student, including the **computed** test average.

To demonstrate the classes above, create a class to maintain a *GradeBook*. The class should allow information on up to 3 students to be stored. Your *GradeBook* class should support operations to Add a student record to the book (i.e., the book entries do not have to be stored in alphabetical order), to Sort by Student ID number, and to display all student records (all info associated with a student) in the *GradeBook*.

You should write a main program that creates a grade book and presents a menu to the user that allows them to select either Add (A), or List (L), or Quit (Q). *Add* should allow the user to enter a student record (name, ID, major, and grades) and add it to the *Gradebook* list **if the ID does not already exist in *GradeBook***. Do not accept input test scores less than 0 or greater than 100 (i.e., no extra credit). *List* should list, sorted by ID number, all student records currently in the grade book. You should be able to add and list repeatedly until you select Q to quit.

Use good coding style and principles for all code and input/output formatting. All data in a class must be private. Put each class declaration in its own header file and its implementation in a separate .cpp file. Submit all the .h and .cpp files you create, along with your executable.

**Deliverables**:

* Complete the programming assignment described above and submit your completed assignment in accordance with the lab submission policies.