CS 121 Week 8 Worksheet Solutions - Classes & Arrays

Syntax Work: (you may want to use your notes, last week's worksheet, or online as a reference)

In this exercise there are two files to implement: s*tudent.h* and *student-main.cpp:*

- *student.h* will consist of the Student class blueprints (prototypes) and implementation.
    - **NOTE:** Keep the prototyping and implementation separate.
- *student-main.cpp* will consist of testing the *Student* class you implement. Tests include:
    - Creating a Student with no arguments, with arguments, and from an existing Student
    - Create a list of 5 Student (which represents a small classroom)
    - Give unique and complete information to each student of the list
    - Print each student's (in the list) information
    - Print the letter grade of each student (in the list).

Answer the three questions below, then fill the following pages with your code work.

Preliminary Questions:

For the following questions, assume you were given the task to set up a database program to keep track of a classroom of students. To complete this task, you decided to use C++, to take an object-oriented approach, and figured to represent students as objects in your program.

1. What kind of attributes (member variables) would a Student consist of? Think about what is essential to signify a Student and (for now) keep things simple (around 3 to 5 attributes).

   - int id; // for identification of the student

   - float grade; // percentage grade of student

   - string name; // full name of student

2. What sort of methods would a Student have? Include any getters, setters, constructor(s), other helpful methods, etc. and have at least 10 methods to implement. For help, refer to the prompt at the start of this page and discuss with others.

   - getID(), getGrade(), getName() - setID(int i), setGrade(float g), setName(string n)

   - Student(), Student(int I, float g, string n), Student(const Student& s) // from the test part

   - printSelf(), getLetterGrade() // these were from the test part too

3. Other than a Student class, would any other classes be necessary for this task?

   Not necessarily, but you could make a Classroom class to manage a list of Student objects.

My own work is in *student.h* and *student-main.cpp* for the remaining coding portions.

NOTE: Your work **WILL** vary (unless you chose the exact same Student attributes as me to work with).