**HW1: Programming Project #1**

Due: April 20th, 23:59

This is the first part of an object-oriented programming project that continues throughout the book. Each part extends the project with new file structures. We begin by introducing two classes of data objects. These objects apply the concepts of the book to produce an information system that maintains and processes information about students and course.

1. Design a class **Student**. Each object represents information about a single student. Members should be included for identifier, name, address, date of first enrollment, and number of credit hours completed. Methods should be included for initialization (constructors), assignment (overloaded “=” operator), and modifying field vales, including a method to increment the number of credit hours.
2. Add methods to class **Student** to read student field values form an input stream and to write the fields of an object to an output stream, nicely formatted. You may also want to be able to prompt a user to enter the field values. Use the C++stream operations to implement these methods. Write a driver program to verify that the class is correctly implemented.
3. Design a class **CourseRegistration**. Each object represents the enrollment of a student in a course. Members should be included for a course identifier, student identifier, number of credit hours, and course grade. Methods should be included as appropriate.
4. Add methods to a class **CourseRegistration** to read course registration field values from an input stream and to write the fields of an object to an output stream, nicely formatted. You may also want to be able to prompt a user to enter the fields values. Use the C++ stream operations to implement these methods. Write a driver program to verify the class is correctly implemented.
5. Create a list of student and course registration information. This information will be used in subsequent exercises to test and evaluate the capabilities of the programming project.

Requirements:

* The program should be compiled and executed in the Linux environments.
* Students compile their programs using g++ and make utility
* Please use separate compile (multiple compile)

What to submit

* Please upload your source codes, Makefile, the report file (soft copy) **to your github repository** with a push command.
* Use an attached file as a report template.