

CS 12 - Assignment 8

Class Inheritance and Virtual Functions

Due Friday, June 12

In this assignment you will be using class inheritance and virtual functions. You will be using your code from assignment 7 and from some labs. Be sure to include all necessary information at the top of each file and use good commenting and style throughout. You must use linux to do this assignment and you must divide your program up into separate files and use a makefile.

By the end of this project you should have the following classes:

Course class Maintains the course name, day, time, maximum number of students, and the current number of students enrolled.

Lab Course class Maintains the information for a course that requires a lab, such as the course name, day, time, maximum number of students, current number of students enrolled, lab day, and lab time.

Schedule class Maintains a list of *course* pointers to depict what courses a student has enrolled in.

SSN class Maintains each of the three integer parts to a Social Security number.

Student class Maintains information about a student, their name, gpa, *SSN*, and *schedule*.

Your *lab course* class should contain all the data members that the *course* class contains as well as the day of the lab (assume that all labs are only once a week for 3 hours) and the starting time of the lab. Each course only contains one lab section (assume that this is a very small elite University and thus course enrollments are kept small). You should have the following function members:

Set Set all data members of the lab course class.

Print Print all data members of the lab course class.

Change Allow the user to change the lecture days, lecture starting time, lab day, and lab starting time.

Enroll Allow a student to enroll this course which contains a lab (the lecture and lab are not separate, enrolling in the lecture automatically enrolls you in the lab).

Drop Allow a student to drop this course.

Think about what class constructs you can use to implement this while rewriting as little code as possible. (Hint, look at the title of this assignment).

Your *student* class should contain the student's full name, their gpa, their *SSN*, and the student's *schedule* of courses. I suggest that you get the rest of the program working correctly before you add in this part to your program. I also suggest that you think about how you wish to implement these parts before beginning. You may decide on the function members to have in this class. For example, you may want to have a function that allows you to set the info, a function that allows you to print the info, and a function that allows you to manipulate the student's schedule. Again, there are many different ways to implement this class. Be sure to use good programming and try and rewrite code as little as possible.

You may want to go back to using the print function rather than the overloaded insertion operator in some classes. You should overload operators where possible to make your code look nice, however your code should not become more complex just to try and use them. Feel free to use the overloaded insertion operator in some classes and not in others. For example, virtual functions work better with regular functions rather than operators.

In your main you should be able to maintain 10 courses. The user should be able to decide at run-time whether to set up a regular *course* or a *lab course*. Thus you should use virtual functions. Your main should also have an array of 5 *students*. You should allow the user to manipulate all parts of these arrays and classes. Using menus and submenus is helpful.

Use good error checking as often as possible in this program. For example, think about the following questions.

- Do you allow the student to enroll in a course that has not been set up?
- Can a course be set to start at an invalid time (a negative number or an invalid hour such as 100)? What about the other course data members?
- Can the student enroll in the same course twice?
- Can a student drop a course they never enrolled in?
- Do you make sure you do not go out of bounds on any array?

You may obtain extra credit on this assignment by adding special features to your code. The better your program is the more credit you will receive (obviously up to a certain limit, making this a huge program that UCR's Registrar's

office could use will still not be able to make up for bombing the final or missing another assignment). Be sure that the required features are working properly, before you begin to add features to your program.

Special features may include additional user functions or data members with extra functions, making the program very user friendly (more than the required error checking), and showing off your CS 12 knowledge by using dynamic allocation, overloading (function and/or operator), inheritance, templates... When showing off your CS 12 knowledge, be sure that the constructs are necessary and following good programming practices, not just thrown in for no reason (that would lower your grade instead).