

CS 12 - Assignment 7

Complex Classes and Operator Overloading

Due Thursday, June 4

In this assignment you will be using your class from assignment 6. You will also be doing operator overloading. You must use linux to do this assignment and you must divide your program up into separate files and use a makefile. Be sure to include all necessary information at the top of each of your files and use good commenting and style throughout.

You will no longer use your print function in your *course* class. You will need to overload the insertion operator to print out the course information.

Write a class to maintain information about a student's schedule of University courses. Use your previously written *course* member functions wherever possible. You should have the following data members:

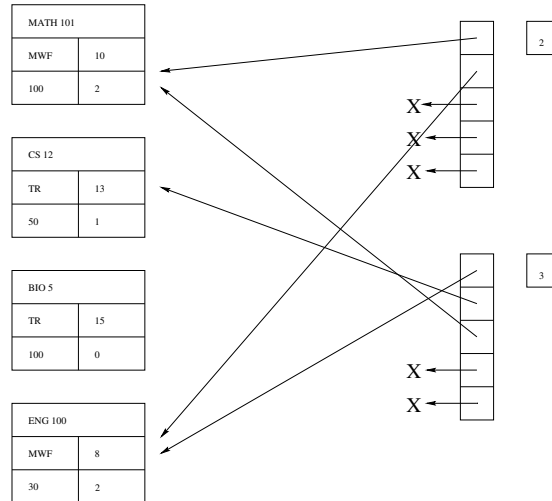
Number of courses Maintain a count of the number of courses the student is enrolled in. Each time they enroll in a course, the counter should be incremented. No student is allowed to take more than 5 courses at one time (the load is too heavy otherwise).

Courses You should have an array of *course* POINTERS. Rather than storing the information for each course in each student's schedule, you will use pointers to simply point to the course they have enrolled in. This saves space as well as making it unnecessary to make sure that each student always maintains the current information for each course (if a course changes day or time). There will be one list of courses that is maintained (outside of this class). The diagram shows what will be happening in memory. On the left is the list of courses. On the right is two student's schedules. The "X"s represent NULL pointers.

Your class should also have the following function members:

Enroll in a course This function should take one parameter, a pointer to the course the student desires to enroll in. Be sure not to allow the student to enroll in more than 5 courses. You should use the *course* enroll function to make sure that the course does not get overenrolled. Be sure to update any pointers and the number of courses the student is currently enrolled in as necessary.

Drop a course You decide how to implement this function. Use the *course* drop function. Be sure to update any pointers and the number of courses the student is currently enrolled in as necessary.



Print Print the information for each course the student is enrolled in. You should overload the insertion operator to do this. Be sure everything is nicely formatted.

Be sure to include a constructor(s) or destructor if necessary.

Modify your main program of assignment 6. You should maintain a list of 10 courses (in assignment 6 main). You should also maintain a list of 10 students schedules. You may use the array index to specify a particular *student* or *course*. You should allow for the following options. Each number in the diagram specifies another menu.

- ```

1 Deal with Course List
 2 set up course
 2 manipulate course
 3 Reset all info
 3 Change day/time
 3 Print course info
 3 Return to course menu
 2 print all courses
 2 return to main menu
1 Deal with a Student Schedule
 4 Enroll in course
 4 Drop course
 4 Print schedule
 4 return to main menu

```

Future assignments will build off of this assignment, so be sure to do a good job. Be sure to ask if you have any questions.