

CS 112 Programming II

Lab Session 4: Object-Oriented Thinking (Part 2)

Objectives

In your previous lab session you have used three classes: **Student**, **Professor** and **Course**. You have created an object from the class **Course** and used the setters and getters provided with this class. By the end of the lab you were asked to reflect on your use of the class and think of ways as to improve the **Course** class implementation. In other words, what operations you may need to invoke for the **Course** objects (e.g. printing all students in the Course) that can be written as methods in the **Course** class?

In today's lab session you will be adding more methods to the **Course** class and you will be given the chance to add your own improvements. Before we get started, the first 45 minutes of this lab will be dedicated to go through your solutions for the previous lab exercises.

Are you sitting right? Always check your posture!



Source: <http://www.straightenmeup.com/sitting-sleeping-and-standing-sss/>

Exercise 1



You have seen that we have added some code in the (testing.CourseTest.java) file as to print out the details of the two **Course** objects that we have created. That is, whenever you need to print any course details you would need to write some code to accomplish that task. *Do not you think that there is a better way to accomplish that task without having to repeat the printing code every time we need to print an object?* How about improving your class implementation by adding a method or a group of methods that will print any **Course** object in different levels of details?

Here are a few suggested methods along with their description:

Course	
:	
+ printCourseSummary():void	This method prints a summary of a course including its name, code and professor.
+ printCourseStudents():void	This method prints the list of students registered in the course (i.e. in the studentList array)
+ printCourseDetails():void	This method prints all details of the course (summary plus list of students).



Write the code of these methods in the **Course** class and test them all with your created objects in the (testing.CourseTest.java) file (i.e., replace your printing code with calls to these methods).

Exercise 2



Write another method named **searchStudent** which takes the student ID as a parameter and returns *true* if the student is found and *false* if not found. In your testing class, prompt the user to enter an id and search for it using the method **searchStudent** and print that the student is found in the course if it returns true otherwise print that the student is not found.

Hint: use equals() method to compare the entered id with the Student object's id.

Homework



Your class **Course** offers more features now, but still there is room for improvement. It is your turn now to think of other methods that can be added to the class to facilitate the use of it.

Bring your work with you to be demonstrated at the beginning of the lab. Expect questions on your written code for which your answers will be used for your overall lab work assessment.