

APS101 lab 3 – week 4

This document contains the instructions for the week 4 APS101 lab. The following two rules apply for the rest of the term, and will not be repeated in later lab handouts:

- To earn your lab marks, you must actively participate in the lab. *You don't need to finish in the time allotted, you just need to try hard.*
- The navigator must not touch the keyboard or mouse. For the rest of the term, if the navigator does type when they are not supposed to, the navigator will get a zero for the lab.

1 Objectives

1. Write and use constructors.
2. Work with `static` variables and methods.
3. Practice testing `static` methods (in the Interactions pane).
4. Work with `String` methods.
5. Work with helper methods.

2 Starting up

Sit down with your partner. The rest of these instructions call you two `s1` and `s2`. Pick which one is which. `s1` should log in and start up DrJava, and be the first driver.

3 Writing constructors

In today's lab, you will add functionality to the class `Course`, which represents a course at a university.

Download the class `Course` (right-click on the link and choose "Save target as...") from the Labs page on the course website:

Open `Course.java` in DrJava, read the code, and discuss it with your partner. Write the constructors described below:

Constructor	Description
<code>Course(String, String)</code>	Constructor for the class <code>Course</code> . The first parameter is the name of the course and the second is the name of the instructor. No students are registered for this course.
<code>Course(String, String, String, int)</code>	Constructor for the class <code>Course</code> . The parameters are the name of the course, name of the instructor, students enrolled (see explanation below), and the number of students enrolled.

The students enrolled in the course will be represented by a colon-delimited `String`. For example, "Bogdan Simion:Jin Jin:Astrid Yi".

4 static variables and methods

Switch roles: s2 drives and s1 navigates.

The class `Course` has one `static` variable, which represents the total enrollment for all courses. Did you use this variable in your constructors? You should have assigned it a value in the second constructor. When students are added to this course, the total enrollment needs to increase. If necessary, go back and add that code to your constructor. Now write the following method:

Method	Description
<code>getTotalEnrollment()</code>	A <code>static</code> method. Return the total enrollment for all courses, as an <code>int</code> .

5 Testing

Now you will test what you wrote in the last sections. In the Interactions pane, you should test the constructors, the “getter” methods, and the update enrollment method. Pay special attention to the `static` variable and method. If you are having trouble testing, then ask your TA for help.

Show your TA *how* you tested your code, especially the `static` variable!

6 More methods: working with Strings

Switch roles: s1 drives and s2 navigates.

Here is a list of `String` methods that you will find useful:

- `indexOf(String s)`: return the index of `s` in this `String`. (Hint: Use this to get indices of the colons and spaces.)
- `substring(int i, int j)`: return the substring from index `i` up to `j`. (Hint: Use this to extract the student names).

Write the methods described below:

Method	Description
<code>getFirstEnrolled()</code>	Return the name of the first student enrolled in the course (as a <code>String</code>). (You may assume that there are at least two students enrolled in the course. Why is that important?)
<code>getFirstName(String s)</code>	Return the first name of the given student (as a <code>String</code>). (The parameter will be in the form: “firstName lastName”.)
<code>getLastName(String s)</code>	Return the last name of the given student (as a <code>String</code>). (The parameter will be in the form: “firstName lastName”.)

Compile, test your code, and fix any errors. Show your work to the TA.

7 Helper methods

Switch roles: s2 drives and s1 navigates.

Write the methods described below by calling existing methods (these are called helper methods):

Method	Description
<code>getFirstNameFirstEnrolled()</code>	Return the first name of the first student enrolled in the course (as a String). (You may assume that there are at least two students enrolled in the course.)
<code>getLastNameFirstEnrolled()</code>	Return the last name of the first student enrolled in the course (as a String). (You may assume that there are at least two students enrolled in the course.)

Compile, test your code, and fix any errors. Show your work to the TA.