

CMSC 132 Week 1 Lab #1

A. Notes for TAs

1. If a student does not have a laptop, pair him/her up with one who does.
2. These notes are for you (do not display them nor make them available online).
3. In the future, make sure that you access the lab material days before the lab (do not wait until the day before as CS servers may be down). Sorry these instructions came to you so late!
4. If you see any errors in the material provided, let me know immediately.

B. Intro

1. Tell students your name and a little bit about yourself and about the structure of the course. If they ask a question about something you don't know, tell them to ask again in lecture.
2. Please ENCOURAGE students to bring laptops to the lab. Please tell them that if they have their own laptop, it would be a big help for them to bring it to the lab sessions -- we will have frequent coding exercises that students will work on during the lab.
3. Give students 5 minutes so they can introduce each other.

C. Eclipse

1. Ask students that have not used Eclipse before to pair up with a student who has.
2. **Project Layouts**
 - a. Explain the two approaches to create projects in Eclipse after reaching the New Java Project window through **File**→**New**→**Project**→**Java Project**
 - i. With a Project layout where you use the project folder as root for sources and class files (first option in **Project layout** section).
 - ii. With a Project layout with separate source and output folders.
 - b. Show students how we can create a project using both of the options mentioned in a. and indicate we will use the second choice.
 - c. Show students how to configure the default for project creation so that separate source and output folders are created for new projects.
 - d. Create a simple class and show how to run it.

D. Review

I have distributed a couple of examples to everyone's CVS repositories in a project called 132Fall2016Lab01Examples. You should check these out in Eclipse on your own machine and then use that to project the examples onto the screen at the front of the room. Also explain to students that they can "check out" these examples themselves from their repositories at any time.

Go over the classes associated with the packages example1 and example2. This is a review of concepts students should already know.

Please go over the code carefully, and then run each example. Be sure to talk about the following items specifically:

Example 1:

- What is the StringBuffer class?
- Instance variables vs. static variables
- Private
- Using "this" as a reference to the current object
- Using "this" to have a constructor call a different constructor in the same class
- What are the differences between the "reference copy", "shallow copy", and "deep copy"?

[CONTINUED ON NEXT PAGE...]

Example 2:

- In the Student class:
 - Javadoc and the Javadoc annotations (like @param and @return)
 - Equals method (Note: I did NOT show them the "self-comparison check" at the beginning in 131 last semester. Explain that this is an optimization just in the case that the current object and the parameter really are the SAME OBJECT (not just two objects that have the same state).)

- In Roster class:
 - Final
 - Syntax for "symbolic constants" (all capital letters, like "SCHOOL")
 - How arrays work
 - ++ and += operators
 - Syntax for instantiating an array with literals (at the end)
 - break