Introduction to CS 51 Lab

Welcome to the CS 51 lab! These lab sessions are a way to help you master the topics covered in class through hands-on practice with the Java programming language. Lab starts at 1:15 pm and goes till 4 pm each week unless otherwise specified. If you’ve finished the assignment and submitted it, you may leave early.

There are student TAs available to help you during the lab sessions. (Note: the professor will go back and forth between both lab sections). If you are stuck on a programming problem, spend some time thinking about and trying to solve the problem on your own before asking for help. This is the fastest way to develop your CS skills and is particularly important if you plan to take more CS courses! If you still can’t solve it, please raise your hand for help.

Submitting Assignments

1. Export your project from Eclipse
You can find instructions for exporting your project from Eclipse on the course webpage. Exporting will create a new folder on your desktop that has the same name as the project. If you correctly exported your assignment, this folder should contain both a src folder and a bin folder.

2. Rename the directory
Rename the folder on your desktop by clicking on it and pressing return. The naming convention for all assignments is as follows:

   LabXX_LastNameFirstName

where “XX” is the number of the lab (e.g. “01” – note the leading zero) and “LastNameFirstName” should be replaced with your last and first name. There should be no spaces in the directory name. IMPORTANT: If you do not follow this naming convention, it is likely that your assignment will not be found and will not be graded!

3. Drag folder
Double click on the “cs051” folder on your Desktop. Inside, you should see a folder called “dropbox”. Drag the folder you just renamed into the dropbox folder. The computer may warn you that you won’t be able to look inside this folder. That’s fine. Click “OK”

Academic Honesty Policy

The academic honesty policy for the CS 51 lab programming assignments is covered specifically by Section 1.1 of the CS department’s Collaboration and Academic Honest Policy:

For programming assignments, students may normally discuss general approaches to assignments, and they may give or receive “consulting” help for specific problems with software or computer programs. A student may look at another student’s work only when help is requested. In that situation the student takes on the role of mentor, and the interaction must be limited to the immediate problem. Two students sitting side-by-side and working through a program step-by-step will certainly produce work that will be considered illegal collaboration.”
The academic honesty policy for the test programs is covered specifically in Section 1.3:

As stated in the Pomona College Student Handbook, “Students neither give nor receive assistance with examinations.” Each examination will have a clear statement of what resources are permitted. Any use of material beyond those limits is not allowed. Take-home examinations will have time limits and similarly explicit rules; they are subject to the same policies. During examinations, students may ask the instructor questions of clarification. The instructor will decide how complete an answer can be given. Some courses have “test programs” which are programming assignments that are to be treated in the same way as take-home examinations.

You are responsible for reading and understanding the Collaboration and Academic Honesty Policy in its entirety (http://www.cs.pomona.edu/academichonesty). If you have any questions regarding the academic honesty policy, please come speak with one of the professors.

---

**Designs**

Some assignments will require you to create a design of your program and bring it with you to lab. A design is similar to an architect’s blueprints. Before construction starts, the architect must create blueprints that show how the building will look and operate, and how it should be built. In the same way, before you write a single line of code, you will create a “blueprint” for your program using pseudocode and comments – this is the design. Here are some considerations when writing your design:

- What classes will you need? How will these classes communicate with one another?
- What parameters will the constructors of these classes take?
- What instance variables, constants, and methods does each class need?

You should also include JavaDoc comments for each instance variable (describing the purpose of the variable) as well as for each method (describing the what not the how).

---

**Pair programming**

Pair programming is when two programmers code together at the same computer. Both people switch off between being the *driver* and the *observer*. The driver types the code while the observer reviews the code, makes suggestions, and generates ideas. Here are some guidelines for pair programming:

- The observer is an active participant, i.e. being the observer is not free time for you to check your cell phone!
- When you’re the observer, resist the urge to grab the keyboard and mouse from the driver.
- Switch roles every 15-20 minutes
- Put both of your names at the top of every Java file
- Submit only one assignment using one person’s first and last name