

## Lab Attendance

Students typically learn more and have an easier time completing the assignments if they come to lab whenever possible. During lab, in addition to covering the prerequisite knowledge for completing the lab assignment, TAs spend time helping students develop skills such as reading documentation, debugging, and the use of bash shell commands.

Attendance will be taken for all lab sessions. Students are expected to attend lab and will lose points on their lab assignment if absent. If you cannot make your lab section for the week, you might be able to attend another section, if arrangements are made with your TA and the TA of the other section. In general this may not work out, as labs are often full and priority for computers goes to the students who are registered for that section.

## Lab Preparation

In order for lab time to be as productive as possible students should come to lab prepared which means that before a lab starts you should:

- (1) Download and unzip the lab files.
- (2) Carefully read the lab pdf.
- (3) Try some initial steps at coding.
- (4) Be prepared with questions.

Lab assignments get more difficult as the quarter progresses. To take the best advantage of the resources available for help (mentoring hours, TA and Instructor office hours) try to get an early start.

## Lab Structure

- **First five minutes:** Discuss recently graded lab assignments.
- **Next 30-45 minutes:** Discuss lab requirements. Explain concepts, functions, data structures needed to complete lab. Answer general questions about assignment.
- **Next five minutes:** Go over what to submit, grading rubric.
- **Rest of Lab:** Students work on individual assignments. TA is available for individual questions, debugging, grading, etc.

## Grading and Lab Submissions

The last things I will cover in lab (beyond individual help) are submission details (exactly what to submit), grading guidelines (what you will lose or receive points for), and grading methodology (how I will test and evaluate your code). While particular assignments may have some special requirements, and/or grading methodology, there are some grading policies outlined below that hold in general for all lab submissions.

## **Submission Format**

Students may lose some amount of points if the format of their submission is not up to lab specifications. Be careful to:

- (1) Name your submitted files precisely as described in the lab pdf (naming is case sensitive and white space matters).
- (2) Compress your files in the manner prescribed by the lab pdf, e.g, do not submit a .tar.gz file when a .zip file was requested, do not zip a folder of files when the lab prescribes to zip a collection of files, do not compress unless directed to, etc...
- (3) Never include .class files in a submission.

## **Compilation**

You should always try to submit code that compiles. If your code doesn't compile there is a good chance that you will not receive a passing grade for the assignment. When submitted code doesn't compile, TAs spend a few minutes trying to make it work, but they don't always succeed. As a courtesy, if you find you have to submit code that won't compile or which suffers from some runtime error, make a comment of this when you submit, along with your best guess for the source of the error.

## **Coding Standards**

In addition to testing code for correct functionality, submissions will be checked for adherence to the following coding standards. The amount of points which may be lost due to coding standards should not exceed 10 % of the total available points for the assignment.

- (1) Use meaningful names that give the reader a clue as to the purpose of the thing being named.
- (2) Avoid the repeated use of numeric or string literal constants. For any numeric or string literal constants used in your program, initialize them at the onset of the program as a named constant using the **final** keyword.
- (3) Use comments at the start of the program to identify the purpose of the program, the author and the date written.
- (4) Use comments at the start of each method to describe the purpose of the method, the purpose of each parameter to the method, and the return value from the method.
- (5) Use comments at the start of each section of the program to explain what that part of the program does.
- (6) Use consistent indentation, and follow one of the accepted java style conventions for bracket placement.

## **Ops Bad Submission!**

After all the sophisticated reasoning, and negotiation of syntax involved in programming, students often make a mistake during the relatively simple process of submitting an assignment. An older buggy version of the program may be submitted, files may be improperly compressed, or some supporting file may be left out of the submission. These common mistakes will oftentimes make it difficult or impossible for the TA to compile your code, and consequently may have a great negative impact on your grade for the assignment. To ensure you haven't made such a mistake, when submitting complete the following paranoid process:

(1) Submit assignment to Canvas.

## ON A LINUX MACHINE

- (2) In your folder for CS145 make a test folder.
- (3) Download your submission from Canvas to the test folder.
- (4) Place any supporting materials needed to run the code in the test folder as well.
- (5) Compile and run the code from the test folder.

## Late Work

Late work will not be accepted unless some arrangement has been made with your instructor for some extenuating circumstance such as illness or family crisis. If something comes up contact your instructor for approval for submitting late work and they will contact your TA with the details of your submission arrangement.

## Grading Questions

Questions about grading of lab assignments should be directed to your TA. The best times to discuss grading issues are the second hour of lab, or office hours.

## Emailing code :(

The best way to get help with coding is to talk to someone knowledgeable while you are both looking at and manipulating the code on a computer, i.e, talking with your TA, a mentor, or your instructor during the times they have available (office hours, lab times, mentor hours) and you are both in the same place with access to a computer. The department provides a great network of support for help with coding projects, and there are MANY scheduled opportunities for one on one coaching of your coding efforts.

Students sometimes decide to email questions about code if they are in a crunch, and can't make any scheduled times. Keep in mind that this is the worst option for receiving help with a programming assignment. It may take days to receive a response and your correspondent will generally spend less time reviewing your code than they would in person, before crafting a response to an email. If you end up deciding to email code, DO NOT copy and paste a section of code into an email. Attach your file/s, and give the line numbers of the relevant sections in your request for help. Also, give your most cogent explanation of the issue, and your attempts to solve it.

## Collaboration and Academic Honesty

Students are encouraged to discuss lab assignments with other students in the same class. However, there are limits placed on collaboration for individual programming assignments. It is okay to discuss coding strategy, program organization, common bugs and syntax errors, math and coding logic, but sharing code is forbidden.

## Other Operating Systems and Personal Equipment

Your TA will demonstrate and test code using a terminal and a text editor on the Linux operating system installed on the lab computers. If you are using a mac or windows PC to code, make sure your code compiles and runs on the Linux operating system installed on the lab computers. You WILL NOT get credit for your code compiling and running if it does not compile and run on the Linux operating system installed on the lab computers.

If coding on your own equipment, ask your TA how to transfer files to the department network and remotely run code on the lab machines for testing. If you are seeking help during lab or office hours, please put your code on a lab machine so that your TA can best help you with coding and debugging.

### **Support**

**Mentors:** Mon.-Fri. 4-7 CF 162/164

**Aaron's office hours:** Thur. 4:00, CF 162

**Professor Liu's office hours:** MWF 2-3, CF 483