

Course Syllabus

Information At-A-Glance

Instructor	
Name:	Adam Blank
E-mail:	blank@caltech.edu
Office:	ANB 115
Office Hours:	Mon: 7:00pm – 9:00pm Thu: 7:00pm – 9:00pm Fri: 5:30pm – 7:00pm Or by private meeting .

Course Website
https://debuggi.ng Visit early. Visit often.

Lecture
BCK Auditorium on MWF 02:00 PM – 02:55 PM

Course Overview

This course is intended as a continuation of CS 1. We focus on abstraction in programming—both in design of programs and data storage. We discuss and implement fundamental data structures and algorithms via a series of labs and projects. We will grade on correctness and efficiency of our programs. This course uses Java as an implementation language, but we do not expect any prior Java programming experience.

Assessments

This course will consist of 9 labs, 8 programming projects, and 1 final exam.

Labs

Our definition of lab is significantly closer to a chemistry or physics lab than it is to the CS 1 definition of lab. During labs, we will practice pair programming, testing, debugging, and implementing skills in a supervised setting. Labs are intended to be a bridge between lecture and the projects; as such, we expect the labs to significantly decrease the amount of time necessary to complete the projects. If you do not finish the lab during class, you and your partner may *collaboratively* finish the lab outside of class. We consider it a violation of the honor code for one student to finish the lab on their own outside of class rather than working with their partner. Final submissions for lab assignments will be due at *11:30 PM on Thursdays*. Due to all of these features of lab, *to receive any credit, you must attend your assigned lab section*. In exceptional circumstances, students might be allowed to get credit for the lab without attending at the discretion of the instructor. We will drop your lowest lab grade.

Projects

The projects are the heart and soul of this course. We prefer the term *project* to *set* because all the individual parts of the assignment will come together to create a single finished product that we hope you will be proud of. The projects will be exclusively programming assignments. They will be auto-graded via sets of unit and integration tests we have designed to catch most of the common mistakes. Because this is an introductory course, we will provide you with the source code of our tests for all projects. In addition to passing the tests, we have several static analysis tools which test basic stylistic components of your code. To receive any credit, you must be passing these tools. Projects will be due at *11:30 PM on Mondays*.

Final Exam

The final exam will have a written component and a programming component. It will be cumulative and untimed, and it is intended to test all aspects of the material (implementing and using data structures, developing and using algorithms, etc.). We expect the final to be *extremely difficult* in order to cover all of this material adequately.

Grading Policy

Labs and programming projects will be graded on an A/B/C/D/F scale (where A = 97, B = 85, C = 70, D = 60, F = 0). Not all assessments will have every grade available due to the nature of the automated tests. Let E be your score on the final, and L be the average of your highest 7 labs and P be the average of all 9 projects, all equally weighted.

- If $0.3L + 0.7P \geq 0.85$, you are exempt from the final, and your grade in the course will be $0.3L + 0.7P$.
- Otherwise, your score will be $0.7(0.3L + 0.7P) + 0.3E$.

We expect the pass threshold for the course will be set at 67%, but we reserve the right to change this.

Late Policy

In this course, you have the opportunity to *earn* up to 10 “late tokens”. Each late token will allow you to submit a project or lab up to 24 hours late; tokens are indivisible and you may not go into “token debt”. You may not use more than two late tokens per project/lab.

Attending a lecture will net you $\frac{1}{3}$ of a late token. You may not use partial tokens, however.

You do not need to use tokens for serious medical (physical or mental) or emotional circumstances; in such situations, contact the instructor to work out a plan for completing the work in a reasonable time frame.

Getting Help

Please don't be afraid to ask for help if you don't understand something. Adam holds *at least three* office hours a week, and they get lonely and bored if you don't show up! They also show up early to lecture and are happy to answer any questions you might have before or after lecture.

At office hours, you can ask for clarification on a lecture (or for a *repetition* of the lecture!). You can ask for help with a frustrating part of the homework. You can even show up just to tell us you're frustrated and vent.

Here's some first steps on how to get help:

- Create a ticket on our ticketing system
- Ask someone on course staff questions before/after lecture, before/after lab, etc.
- Come to office hours

Collaboration & Academic Integrity

See our “collaboration table” on the website. We reserve the right to modify or clarify this policy as needed. Notably, you may not, under any circumstances, look at another student's/group's code.