CS 2: Introduction to Programming Methods — Winter 2025

Course Syllabus

Information At-A-Glance

Instructor	
Name:	**Adam Blank**
E-mail:	blank@caltech.edu
Office:	ANB 115
Office Hours:	TBD Or by private meeting.

Course Website

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

Lecture

BCK Institute 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 labs, programming projects, and two (in person) quizzes.

Labs

Our definition of lab is surprisingly close to a chemistry or physics 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*.

Quizzes

There will be two written in-person quizzes (one in week 4, and the other in week 8). **Approximately 60% of the questions on these quizzes will be pulled directly from in-class exercises.** These quizzes might involve code, but they will not involve writing entire functions from scratch on paper. The goal of the quizzes is mostly to assess non-programming skills (e.g., code analysis or choosing the right data structure for a problem) that cannot be tested on the projects.

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.

To calculate your shadow grade, we will use the following formula:

$$0.2 \times (\text{Lab Average}) + 0.4 \times (\text{Project Average}) + 0.4 \times (\text{Quiz Average})$$

To pass the course, you must (1) get $\geq 60\%$ on every project, lab, and quiz, and (2) get an overall average of at least 75% according to the formula above.

Late Policy

All extensions will be handled dynamically by the following website:

https://extensions.caltech.codes/

Please do not e-mail us asking for an extension. You must use the website form. There are limits and restrictions on extensions which the form outlines. Please read these **before** you need an extension!

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:

- Ask someone on course staff questions before/after lecture, before/after lab, etc.
- Come to office hours
- E-mail cs002@caltech.edu

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 or write pseudocode with another group.