

Course Syllabus

Instructor

Name: **Prof. Blank**
E-mail: blank@caltech.edu
Office: ANB 115

Course Website

<https://cs1.caltech.codes>
Visit early. Visit often.

Instructor

Name: **Prof. Vanier**
E-mail: mvaniem@cms.caltech.edu
Office: ANB 110

Lecture

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

Course Philosophy

We have structured the course around our four core values:

CS and programming are for everyone!

We have taken care to design and implement CS 1 to be accessible to students interested in any option, **regardless of prior experience**. We firmly believe that every student can gain valuable, lifelong skills from this course! We will only consider the course successful if we are able to serve you in whatever discipline you end up working in. Inclusivity, of all kinds, is one of our core values, and we strive to ensure that everyone feels included and welcome in all aspects of the course.

Our projects cover diverse disciplines!

As CS 1 is a part of core, we aim to provide points of connection to as many other fields as possible. To accomplish this, we've designed weekly projects to be "themed" around different disciplines (physics, chemistry, engineering, math, etc.). We're aiming to have at least one project that feels relevant and cool to every student. (If we miss on this, please please let us know so we can add/edit for the future.)

CS 1 material is sequenced to set you up for success!

The course is separated into three modules: (0) Reading and Modifying Python code, (1) Using Python to Get Things Done, and (2) Another Programming Language and Preparation for Future CS Courses. Here's more details, in case you're curious:

- In Module 0, we will imagine that you're working in an existing codebase (perhaps, for research) and help you navigate how to work with code in this context. We start here, because, realistically, most code you write won't be from scratch.
- In Module 1, we will work on helping you create a mental model of what Python is doing; so, you can effectively debug your own code and write small programs from scratch.
- In Module 2, we will switch to the Java programming language to compare and contrast with Python. This isn't only because we want to prepare you for future CS courses! Lots of research, in many fields, is not done in Python! Languages like C++, C, Rust, and Julia all require some understanding of features that learning Java sets you up for!

CS 1 is not an easy course, but we want you to succeed!

Our goal is to help you succeed. Please ask for help if you need it. Productively struggling is good, but struggling in silence is needless and painful. If you are stuck on any single issue for more than a half-hour, ask for help at office hours. We allocate a ton of TA and Professor time (more than fifty people-hours) to office hours because we want to provide as much help as possible, and we hope you will find them as helpful as we have in the past!

Late Policy

We've designed a late policy that is very intentional in providing flexibility while also making sure students do not fall so far behind that they might not be able to pass. Please see below for details.

- **Flexibility:** If you turn in an assignment after the due date, but BEFORE we start grading that assignment, we will grade the last submitted version as of when we start grading. Note that this is not a guaranteed extension, because if we grade earlier than you thought we would, we will grade whatever submission was made at the beginning of grading. Typically, we will be grading on Wednesday afternoons.
- **Extensions:** If you are requesting an extension due to significant circumstances (e.g., a family emergency, a long term illness, etc.), please **email the dean's office**, not us. We will definitely work something out, but the deans must be involved. Otherwise, the following apply:
 - All extensions must be requested at least 24 hours before the due date.
 - Send an email to cs001@caltech.edu with the following information:
 - (1) Your name, access username, and UID
 - (2) How long of an extension you are asking for (in hours from the due date)
 - (3) A list of all extensions you have previously received (including assignment and length) in this course
 - (4) Please don't include any personal privileged information. We would like to respect your personal privacy.

Course staff will respond with a final decision either allowing or denying the extension within approximately 24 hours.

- **Accountability:** If your assignment was submitted more than 0 seconds after the deadline, you must sign up for a code review that week to receive any credit on the corresponding assignment. We are not able to award any credit on the assignment otherwise. **Note that code reviews are available to all students, but they are not (usually) required except in this circumstance.**

Grading Scheme

To pass, you must meet the following criteria:

- Earn at least 75% on in the code quality category
- Earn a minimum of 60% on the midterm
- Earn a non-zero score on every project
- Earn a total average of at least 75% cumulatively on the projects, midterm, and final
- Pass at least 65% of the diagnostics (we will drop your worst two)

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

$$0.2 \times (\text{Code Quality Average}) + 0.7 \times (\text{Project Correctness Average}) + 0.1 \times (\text{Diagnostic Average})$$

For the Project Correctness Average and Diagnostic Average, we will use a normal points-based average. For the Code Quality Average, we will use a “standards-based grading” strategy. Each assignment will have the opportunity to show you understand several particular style components. As long as you demonstrate a style component by the end of the quarter, you get credit for it. This means, you can get it wrong without penalty several times, as long as you eventually get it correct!