

## Policies

### Communication:

There will be several routes of communication for this course:

- The main mode of electronic communication between students and staff, as well as amongst students, will be through  [Piazza](#). It is intended for general questions about the course, clarifications about assignments, student questions to each other, discussions about material, and so on. We strongly encourage students to participate in discussion, ask, and answer questions through this site. The course staff will monitor discussions closely.
- If you need to contact the course staff privately, you should email cs188 AT berkeley.edu. You may of course contact the professors or GSIs directly, but the staff list will produce the fastest response.

### Prerequisites:

- **CS 61A or 61B:** Prior computer programming experience is expected (see below)
- **CS 70 or Math 55:** Facility with basic concepts of propositional logic and probability are expected (see below)

*CS61A AND CS61B AND CS70 is the recommended background.*

Course programming assignments will be in Python. We do not assume that students have previous experience with the language, but we do expect you to learn the basics very rapidly. Project 0 is designed to teach you the basics of Python, but if you want to get a head start here is a good tutorial: [ACM Python Tutorial](#)

## Assignments

This class includes **five programming projects** and **eleven homework assignments (with each homework assignment having an electronic and a written component)**.

**Collaboration:** Project 0 is to be completed alone. Projects 1 through 5 can be completed alone or in teams of two. If done in a team of two, the person who submits needs to tag the other team member through Gradescope. However, it is important that the submission reflects the understanding of both team members. Homework is to be submitted individually, but may be discussed in groups. If discussed in a group, acknowledge your collaborators in the submission per standard academic practice.

**Project slip days:** Programming projects must be turned in electronically by the listed due date and time. You will have a total of **5 slip days** for these projects, up to **2 of which can be used for each project**. Note that slip days are counted at the granularity of days, rounded up to the nearest day. For example, for a project due at 4pm on Friday, any submission from Friday 4pm - Saturday 4pm will use up one slip day, any submission from Saturday 4pm - Sunday 4pm will use up two slip days, and any submissions after Sunday 4pm will receive no credit. The other homework assignments do not have late days!

**Project grading:** Projects will by default be graded automatically for correctness, though we will review projects individually as necessary to ensure that they receive the credit they deserve. Projects can be submitted as often as you like; we strongly encourage you to keep working until you get a full score.

**Electronic component of HW:** Electronic homework (hosted on Gradescope) is meant to reinforce and give practice with concepts covered in class. They will be automatically graded for correctness, and you can submit as many times as you like up to the deadline; again we encourage you to work until you have fully solved the homework.

**Written component of HW:** Written homework (submitted into Gradescope) is meant to make you think beyond strict repetition of what is covered in class and is a sample of the kind of assessment material you will see on exams. Points will be given for completion/effort rather than correctness. In addition, the following homework will contain a retrospective self-assessment in which you compare your solution to the staff solution (also graded on completion). You are welcome to discuss any stage of the written homework process with others provided you acknowledge your collaborators.

**Homeworks have no slip days.**

**Ethics:** Submissions should acknowledge all collaborators and sources consulted. All code and written responses should be original. We trust you all to submit your own work, but to protect the integrity of the course from anyone who doesn't want to play by the rules, we will actively be checking for code plagiarism (both from current classmates and previous semesters). We are not lenient about cheating; we sympathize with [Kris Pister's policy](#).

## Grading

Overall grades will be determined from:

- Programming Assignments (25%)
- Homework Assignments (15%)
- Midterm 1 (10%)
- Midterm 2 (15%)
- Final exam (35%)

Grades are on the following fixed scale:

A	[85 -- 100]%
A-	[80 -- 85]%
B+	[75 -- 80]%
B	[70 -- 75]%
B-	[65 -- 70]%
C+	[60 -- 65]%
C	[55 -- 60]%
C-	[50 -- 55]%
D+	[45 -- 50]%
D	[40 -- 45]%
D-	[35 -- 40]%
F	[0 -- 35]%

The instructors may adjust grades upward based on class participation, extra credit, etc. The grade of A+ will be awarded at the instructors' discretion based on exceptional performance.

**Regrade Policy:** If you believe an error has been made in the grading of one of your exams or assignments, you may resubmit it for a regrade. Regrades for cases where we mis-applied a rubric in an individual case are much more likely to be successful than regrades that argue about relative point values within the rubric, as the rubric is applied to the entire class. Because we will examine your entire submission in detail, your grade can go up or down as a result of a regrade request.

---

## **Enrollment**

Here are [the policies that govern admission into classes](#), and here are some answers to [frequently asked questions about admission](#). The course staff does not control enrollment!