

Syllabus & Course Policies

Overview

The CS 61A series is an introduction to computer science, with particular emphasis on software and on machines from a programmer's point of view.

- CS 61A concentrates on the idea of abstraction, allowing the programmer to think in terms appropriate to the problem rather than in low-level operations dictated by the computer hardware.
- CS 61B deals with the more advanced engineering aspects of software, such as constructing and analyzing large programs.
- CS 61C focuses on machines and how they execute programs.

In CS 61A, we are interested in teaching you about programming, not about how to use one particular programming language. We consider a series of techniques for controlling program complexity, such as functional programming, data abstraction, and object-oriented programming.

CS 61A primarily uses the Python 3 programming language. Python is a popular language in both industry and academia. It is also particularly well-suited to the task of exploring the topics taught in this course. It is an open-source language developed by a large volunteer community that prides itself on the diversity of its contributors. We will also use two other languages in the latter half of the course: the Scheme programming language and the Structured Query Language (SQL).

Mastery of a particular programming language is a very useful side effect of CS 61A. However, our goal is not to dictate what language you use in your future endeavors. Instead, our hope is that once you have learned the concepts involved in programming, you will find that picking up a new programming language is but a few days' work.

A complete list of lecture topics, readings, and assignments appears in the [daily schedule](#).

Prerequisites

Math 1A is listed as a corequisite for CS 61A. (That is, it may be taken concurrently.) Math 10A or Math 16A are also fine. It is possible to take CS 61A without knowing or learning calculus; all of the old calculus-based examples have been removed over the years. However, taking calculus is a great way to brush up on the arithmetic and algebra that appear regularly in CS 61A.

There are no formal programming-related prerequisites for CS 61A, but it's not the right first course for all students. Many CS 61A students have had significant prior programming experience, including prior coursework. Some students take the course without any prior programming experience, but they typically must work substantially harder to master the material, perhaps simply because they have less practice working with programs. If you have limited prior experience and you find it challenging to complete all of the required coursework in the first three weeks, you should seriously consider taking another course first. You'll likely have a better experience taking 61A later, and you won't fall behind in any meaningful way by taking one of the alternatives below prior to taking 61A.

Alternatives

If you want to build programming experience before taking CS 61A, we recommend that you take one of these courses first. You can always take CS 61A in a future semester.

CS 10

[CS 10: The Beauty and Joy of Computing](#) provides a bird's-eye view of the field of computer science. The course teaches students how to program using Snap (based on Scratch), one of the friendliest programming languages ever invented, as well as Python, the same language used in 61A. But the course is far more than just learning to program! You'll also learn about some big ideas of computing, such as abstraction, design, recursion, concurrency, simulations, and the limits of computation. You'll see beautiful applications of computing that have changed the world, as well as talk about the history of computing and where it will go in the future.

Data 8 and CS 88

[Data 8: The Foundations of Data Science](#) is an introduction to data science designed to be accessible and useful for all Berkeley students. This course was built for students without prior programming experience. It teaches students to program in Python 3, but covers a much smaller subset of the language than CS 61A. Most of the course focuses on data processing and statistical techniques that are central to using computers to answer questions about the world. The overlap between Data 8 and CS 61A is small (perhaps 25%), but the programming skill you will acquire in Data 8 will help you maintain the faster pace of CS 61A.

[CS 88: Computational Structures in Data Science](#) is an introduction to programming and computing that has more than 50% concept overlap with CS 61A. It is designed for students interested in data science who want to expand their knowledge of programming and program structures beyond what is covered in Data 8. Students who complete CS 88 can either proceed directly to CS 61B or subsequently take CS 61A, a path that offers a substantial amount of review because of the high topic overlap between the courses.

Course Format

The course includes many events and opportunities for learning: lecture, lab, discussion, tutorials, office hours, parties, and exam prep. We understand that everyone learns differently, so not all of these events are required. However, it is recommended that you try everything out to figure out what combination of these events works best for you.

Lecture

The first lecture will be held live over Zoom 2:10pm–3pm PDT on Wednesday 8/26. Subsequent lectures will be released as playlists of short videos posted each Sunday, Tuesday, and Thursday evening. Playlists will typically contain 3–8 videos and 30–45 minutes of video content.

The course instructors will hold a question-and-answer session for each lecture the morning after the videos are posted: 9:10am Monday, 8:10am Wednesday, and 9:10am Friday. These Q&A sessions will be recorded, but students will be removed from the recordings before the recordings are posted. The Q&A recordings will be added to the lecture playlist around noon.

This course moves fast, and lecture is tightly coordinated with section. Please watch each lecture before attending section on Monday and Wednesday. If you want to watch the recorded Q&A and attend a 2pm section, 12pm–2pm is an ideal window of time to watch lecture.

Section

Lab

The goal of each week's lab is to provide you with some programming practice. Lab assignments are similar to homework assignments, but the problems are more straightforward, and you're welcome to work through lab assignments with other students.

To get you started, each week's lab will begin with a short online orientation from one of the course GSIs. These are held Monday at 2:10pm, 6:10pm, 7:10pm, and 9:10pm. These will be recorded (but attend one live so you can ask questions).

Once you've attended a lab orientation, complete the lab assignment that day. There are several ways to get help and work with other students on lab assignments, which will be described during the lab orientation. You're encouraged to work with other students on lab. You'll learn more that way.

Discussion & Tutorials

Discussion sections provide problem-solving guidance and practice. Each discussion section has two parts: an orientation and a tutorial.

The 50-minute orientation sessions are held Wednesday at 2:10pm, 6:10pm, 7:10pm, and 9:10pm. These will be recorded (but attend one live so you can ask questions). Please watch an orientation before attending your weekly tutorial.

Small-group tutorials offer problem solving with the same group of around 4–6 students each week. Each 25-minute tutorial will focus on particular questions from the week's discussion worksheet. You can ask any questions you want during the tutorial, but please come prepared by attending an orientation first.

Office Hours

Our virtual office hours are one-on-one meetings with a course tutor in which you can ask questions and get help. You can make an appointment at oh.cs61a.org/appointments or join the office hours queue at oh.cs61a.org. Each week's schedule of office hours (when we are taking appointments and helping students on the queue) appears in the [weekly office hours schedule](#).

We will try hard to make sure that enough staff are available to meet the demand for one-on-one appointments each day, but there will likely be some time this semester when demand exceeds supply. Please be patient as we try to support all the students in the course. Try a Piazza private post if you can't find a timely appointment with a course tutor.

Join Us

In addition to one-on-one appointments with tutors, you can collaborate with other students working on the same assignment using the Join Us tool (official link TBA). If the whole group you form wants help from a course tutor, you can request help any time that the office hours queue is available.

Parties

For students who prefer a more collaborative environment in which they can discuss problems with other students, we will offer Zoom-based lab, homework, and project parties each week. Students and staff will work together to solve problems. Try one out and see if you like it.

Assignments

Each week, there will be problems assigned for you to work on, most of which will involve writing and analyzing programs. These assignments come in three categories: lab exercises, homework assignments, and projects.

Labs

Lab exercises are designed to introduce a new topic. All labs are released and due on Monday, with the exception of Lab 2, which is released and due on Friday 9/4.

Lab exercises are scored on correct completion. To receive credit, you must complete all of the problems that are not marked as optional and pass all tests.

Homework

Weekly homework assignments let you apply the concepts learned in lecture and section to more challenging problems. They will usually be released on Friday night and be due the following Thursday night.

Collaboration

You are encouraged to discuss homework questions with other students, as long as you write your own code and submit your own work. Finding a study group is a great idea. The purpose of homework is for you to learn the course material, not to prove that you already know it. Therefore, you can expect to receive substantial assistance from the course staff. You're welcome to help others once you solve a problem.

You are not allowed to copy solutions or share your solution to a question with other students who haven't completed the question already. You are not allowed to use solutions that you find on the internet. **If you are stuck on a problem, come get help instead of copying the answer from someone else or the Internet; you'll still get credit and won't be flagged for cheating.**

Partial Credit

Homework is scored out of 2 points, and every incorrect solution costs you a point. Your lowest homework score will be dropped from your point total for the semester.

Homework Recovery Policy

You can recover one question per homework by going through the homework recovery process:

- Fill out a form declaring which question you want to recover by Friday 11:59PM, the day after the homework is due.
- Join the recovery session for that question the following Monday. These sessions are not recorded; you must attend live to receive recovery credit.

Recovery sessions describe how to approach and solve a past homework problem. You are welcome to attend these even if you don't need the recovery point.

Projects

Projects are larger assignments intended to combine ideas from the course in interesting ways. Some parts of the projects can be completed in pairs. When working in pairs, you should work together to ensure that both of you understand the complete results. We recommend finding a project partner in your tutorial group. You may also work alone on all projects, although partners are recommended for the paired projects.

Projects are graded on both correctness and [composition](#).

Exams

Each exam has a scheduled alternate time, but students in time zones for which the exam would start before 7am or after 10pm may request an alternate exam time. Details for this request will be shared before the exam date.

Midterm 1 will be held 7pm–9pm PT Monday, 9/14.

Midterm 2 will be held 7pm–9pm PT Wednesday, 10/28.

The final exam will be held 3pm–6pm Thursday, 12/17.

If you have a direct conflict with another final exam, we will allow you to take the final 7pm–10pm Thursday, 12/17. We will distribute a form after the add/drop deadline to resolve conflicts.

Exams will be web-based multiple-choice and fill-in-the-blank, in a similar format to the Spring 2020 final exam.

Exam Proctoring

All exams will be proctored by video, either using [Zoom](#), [Loom](#), or another recording technology of your choosing, such as a cell phone. We will not watch you live as you take the exam. Instead, you will submit a video recording of your screen and your head. You are welcome to use a virtual background. You do not need to record your surroundings or your hands. Your recording should include sound. You will need to display a photo ID to the camera. Instructions for how to record your exam video with minimal fuss will be provided.

If you wish to request an exemption from this exam proctoring policy for any reason, please fill out this [Proctoring Exemption Request](#) form by Thursday, 9/10.

We will only use these videos for the purpose of investigating academic integrity cases.

If for some reason your recording fails during an exam, please continue to take the exam. We may still score your exam even without a video recording.

Resources

Textbook

The online textbook for the course is [Composing Programs](#), which was created specifically for this course, based on the classic textbook [Structure and Interpretation of Computer Programs](#). Readings for each lecture appear in the course schedule. We recommend that you complete the readings before attending lecture.

Grading

Your course grade is computed using a point system with a total of 300 points.

- Midterm 1, worth 40 points.
- Midterm 2, worth 50 points.
- The final exam, worth 75 points.
- Four projects, worth 100 points.
- Homework, worth 18 points.
- Lab, worth 11 points.
- Tutorials, worth 6 points.

There are a handful extra credit points throughout the semester, perhaps around 10, that are available to everyone.

Each letter grade for the course corresponds to a range of scores:

A+	≥ 300	A	≥ 285	A-	≥ 270
B+	≥ 250	B	≥ 225	B-	≥ 205
C+	≥ 190	C	≥ 180	C-	≥ 175
D+	≥ 170	D	≥ 165	D-	≥ 160

Your final score will be rounded to the nearest integer before being converted to a letter grade. 0.5 rounds up to 1, but 0.49 rounds down to 0.

There is no curve; your grade will depend only on how well you do, and not on how well everyone else does. Score thresholds are based on how students performed in previous semesters. Unlike some previous semesters you may have heard about, these thresholds **will not be adjusted** based on student performance. You could all get A's. You could all get D's.

These are the exact thresholds that will be used at the end of the course to assign grades. In a typical semester, about 60% of students taking the course for a letter grade will receive a B+ or higher.

Incomplete grades will be granted only for medical or personal emergencies that cause you to miss the work or last part of the course, only for students who have completed the majority of the coursework, and only if you work up to the point of the emergency has been satisfactory.

Your lowest homework score will be dropped.

Each lab that you complete is worth 1 point, and you can receive a maximum of 11 lab points. There are going to be 13 lab assignments, so you can skip two and still get full credit.

Each tutorial you attend is worth 1 point, excluding the tutorial just after the first lecture. You can receive a maximum of 6 tutorial points. There are going to be around 12 tutorials, so you can skip many, but you should really continue to attend tutorial throughout the semester because they are useful.

Tutorial Participation

Attending more than 6 tutorials will contribute to recovery points on midterms. Attending at least 10 tutorials will give you the maximum amount of midterm recovery.

We calculate your midterm recovery using the following logic, where [attendance](#) is the number of weeks that you attend tutorial.

```
def exam_recovery(your_exam_score, attendance, max_exam_score, cap=10):
    half_score = max_exam_score / 2
    max_recovery = max(0, (half_score - your_exam_score) / 2)
    recovery_ratio = min(attendance, cap) / cap
    return max_recovery * recovery_ratio
```

According to this formula, if you receive more than half the available points on each midterm, then you don't recover any points. If you score just below half the points, you will recover a few points. If you score far below half the points, you will recover many points. The more weeks you attend tutorial, the more exam points will be recovered.

The purpose of this policy is to ensure that all students who continue to invest time in the course throughout the semester are able to pass.

There are no recovery points available on the final exam.

Late Policy

If you cannot turn in an assignment on time, contact cs61a@berkeley.edu and partner as early as possible. Depending on the circumstance, we may grant extensions.

- Labs:** We rarely accept late lab submissions. There is no partial credit.
- Homework:** We rarely accept late homework submissions.
- Projects:** Submissions within 24 hours after the deadline will receive 75% of the earned score. Submissions that are 24 hours or more after the deadline will receive 0 points.

Citizenship

For exceptionally rude or disrespectful behavior toward the course staff or other students, your final grade will be lowered by up to a full letter grade (e.g., from an A- to a B-) at the discretion of the course instructors. You don't need to be concerned about this policy if you treat other human beings with even a bare minimum of respect and consideration and do not engage in behavior that is actively harmful to others.

Learning Cooperatively

With the obvious exception of exams, we encourage you to discuss course activities with your friends and classmates as you are working on them. You will definitely learn more in this class if you work with others than if you do not. Ask questions, answer questions, and share ideas liberally.

You are also welcome to code collaboratively with others in your lab and solve lab problems in small groups.

Learning cooperatively is different from sharing answers. You shouldn't be showing your code to other students or looking at others' code, except:

- During lab, you can share all you want as long as you're all learning.
- For a project that allows partners, you can share anything with your partner.
- If you've finished a problem already, you can look at others' code to help them finish.

If you are helping another student, don't just tell them the answer; they will learn very little and run into trouble on exams. Instead, try to guide them toward discovering the solution on their own. Problem solving practice is the key to progress in computer science.

Since you're working collaboratively, keep your project partner informed. If some medical or personal emergency takes you away from the course for an extended period, or if you decide to drop the course for any reason, please don't just disappear silently! You should inform your project partner, so that nobody is depending on you to do something you can't finish.

Online Forum

If you have any questions, please post them to [Piazza](#), the course discussion forum. Piazza allows you to learn from questions your fellow students have asked. We encourage you to answer each others' questions!

Piazza is the best and most reliable way to contact the course staff. You are also welcome to email cs61a+fa20@berkeley.edu or denero@berkeley.edu.

Academic Honesty

Any students caught collaborating on exams will receive an F in the course. Please don't be one of these students. Assignment cooperation has a limit, and in CS 61A that limit is reading others' homework or project solution to a problem before you solve that problem on your own. You are free to discuss the problems with others to help you beforehand, but you must write your own solutions. You may share code with your project partner.

If you are unsure if what you are doing is cheating, please clarify with the instructor or email cs61a@berkeley.edu. The following is a list of things you should NOT do. This list is not exhaustive, but covers most of the big offenses:

- Do not copy code from any student who is not your partner.
- Do not allow any student other than your partner to copy code from you.
- Do not copy solutions from online sources such as Stack Overflow, Pastebin, and public repositories on GitHub.
- Do not post your solutions publicly during or after the semester.

If you find a solution online, please [submit a link to that solution](#) *anonymously*. When we find an online solution, we will ask the author to remove it. We also record the solution and use it to check for copying. By reporting online solutions, you help keep the course fair for everyone.

In summary, we expect you to hand in your own work, take your own tests, and complete your own projects. The assignments and evaluations are structured to help you learn, which is why you are here.

Rather than copying someone else's work, ask for help. You are not alone in this course! The entire staff is here to help you succeed. If you invest the time to learn the material and complete the projects, you won't need to copy any answers.

A Parting Thought

Grades and penalties aren't the purpose of this course. We really just want you to learn. The entire staff is very excited to be teaching CS 61A this semester and we're looking forward to meeting such a large and enthusiastic group of students. We want all of you to be successful here. Welcome to CS 61A!

Overview	Prerequisites	Alternatives	CS 10	Data 8 and CS 88
Course Format	Lecture	Section	Lab	Discussion & Tutorials
Office Hours	Join Us	Parties	Assignments	Labs
Homework	Collaboration	Partial Credit	Homework Recovery Policy	Projects
Textbook	Tutorial Participation	Late Policy	Citizenship	Exams
Exam Proctoring	Resources	Resources	Textbook	Grading
Grading	Staff	Resources	Staff	Resources
Staff	Resources	Staff	Resources	Staff

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources

Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources
Weekly Schedule	Office Hours	Staff	Resources	Staff	Resources