Syllabus & Course Policies

Overview

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

1. 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.
2. CS 61B deals with the more advanced engineering aspects of software, such as constructing and analyzing large programs.
3. 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 other languages in the latter half of the course, including the Scheme programming language.

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. If you have limited prior experience and you find it challenging to complete all of the required coursework in the first three weeks, please 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 a preparatory class first.

Preparatory Classes

If you want to build programming experience before taking CS 61A, we recommend that you first take a class that introduces you to programming. The most appropriate class within the Berkeley CS department is CS 10, described below, but you may also find similar classes at Berkeley extension or in online courses. Feel free to contact course staff (/contact/) if you are not sure what's best.

CS 10

CS 10: The Beauty and Joy of Computing (http://cs10.org) is an introductory computer science course which is similar to CS 61A but moves at a friendlier pace. CS 10 covers variables, functions, recursion, algorithmic complexity, object-oriented programming, and many other relevant CS 61A topics, with the overall content overlap being about 50%. CS 10 starts the semester in Snap!, a block-based programming language which allows students to focus on conceptual understanding without worrying about unfamiliar syntax. After the midterm, the course transitions into Python (the primary language 61A uses), applying the same concepts you already learned to the new language, as well as introducing new concepts more relevant to Python. CS 10 also covers big ideas and social implications that go beyond programming, showing you the beauty and joy of computing.

Alternative Classes

Data 8 and CS 88

If you are majoring in data science, the pathway of Data 8 plus CS 88 is the recommended way for you to learn programming, since it is designed for data science students and will introduce you to many of the core concepts.

Data 8: The Foundations of Data Science (http://data8.org/) 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.

CS 88: Computational Structures in Data Science (https://cs88-website.github.io/) 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, lost sections, 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

Lectures will be held live over Zoom 2:10pm-3pm PDT, starting on Wednesday 1/20. For students who cannot attend live, recordings will be posted to the website afterwards.

This course moves fast, and lecture is tightly coordinated with section. Please attend or watch each lecture before attending lab and discussion orientation.

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 at times TBD. All lab orientations 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. The best way to find other students to collaborate with

is to attend a lab party, described below. We will release a form on Piazza to help you find partners to work with!

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 at times TBD. 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 6-10 students each week. Each 50-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.

If you do not attend orientation, you will likely have a tough time keeping up in tutorial, as your tutor will assume that you have already understood part of the worksheet in orientation.

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 join the office hours queue at oh.cs61a.org (https://oh.cs61a.org). Each week's schedule of office hours (when we are helping students on the queue) appears in the weekly office hours schedule (/office-hours/).

We will try hard to make sure that enough staff are available to meet office hours demand, 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 timely help with a staff member.

Study Group

In addition to the office hours queue, you can collaborate with other students working on the same assignment using the Study Group tool (https://oh.cs61a.org/party). If the whole group you form wants help from a staff member, 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 on Monday morning and due Tuesday at 11:59 PM, with the exception of Lab 0, which will be due on the same day as Lab 1.

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. **There is no partial credit on labs.**

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 Policy

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:

1. Fill out a form declaring which question you want to recover by Friday 11:59PM, the day after the homework is due.
2. Join the recovery session for that question the following week. 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.

You are allowed and encouraged to pair program (/articles/pair-programming) with a partner. Make sure to alternate roles so that both of you understand the complete results. We recommend finding a partner in your tutorial group, but course staff will also provide a partner matching form at various points in the semester. It is your responsibility to contact and collaborate with your partner; we cannot guarantee that your partnership works out.

You may also work alone on all projects, although partners are recommended.

Projects are graded on both correctness and composition (/articles/composition).

Exams

Each exam has a scheduled time, but students in time zones for which the exam would start between 8pm and 6am (inclusive) or with conflicting **midterm** exams may request an alternate exam time. Details for this request will be shared before the exam date.

Midterm 1 will be held 5-7pm PST Wednesday, 2/17.

Midterm 2 will be held 5-7pm PST Friday, 3/19.

The final exam will be held 11:30am-2:30pm Tuesday, 5/11.

> We will not be providing alternates for those with conflicting final exams unless it is due to a DSP accommodation. Please review the Spring 2021 Final Exam Groups (https://registrar.berkeley.edu/scheduling/academic-scheduling/final-exam-guide-schedules) before the add/drop deadline.

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

Exam Proctoring

All exams will be proctored through Zoom (https://berkeley.zoom.us).

1. Before the exam: Download Zoom on your smartphone and determine how you will set up your phone to proctor your work. Please reach out privately on Piazza if you are having setup difficulties. Example setups are here: https://cs161.org/sample-setups

On the night before the exam, make sure you've received an email with a Zoom meeting link.

1. On the day of the exam: Join the Zoom meeting from your smartphone and position your smartphone so that we can clearly see all screens you are using and both of your hands. Mute yourself.
2. Technical issues: Don't worry if your video feed disconnects briefly during the exam. If you encounter significant technical problems, don't worry about video proctoring and focus on finishing the exam.

If you need to use the bathroom, just leave the video feed on while you're away.

Resources

Textbook

The online textbook for the course is Composing Programs (http://composingprograms.com/), which was created specifically for this course, based on the classic textbook Structure and Interpretation of Computer Programs (https://mitpress.mit.edu/sites/default/files/sicp/index.html). 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.

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 final or last part of the course, only for students who have completed the majority of the coursework, and only if 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. Lab 00 is not graded, and Labs 04 and 08 are midterm review labs that everyone will receive points for. There is no Lab 09.

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 0 attendance does not count for points, but we suggest that you attend to get to know your tutor and the people in the section.

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 througout 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 course staff (/contact/) and your partner as early as possible. Depending on the circumstance, we may grant extensions.

- **Labs**: We rarely accept late lab submissions.
- **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.

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 (http://www.piazza.com/berkeley/spring2021/cs61a), 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. There is an option to make a Private post, if you would prefer. If you need to share sensitive information that you would only like the Head TAs and instructors to see, you should email us (/contact/).

Academic Honesty

The minimum penalty for any students caught collaborating on exams will be negative points on that exam. 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 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 contact course staff (/contact). 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 (https://goo.gl/forms/nL2yOj1Z81HcQYDi2). When we find an online solution, we 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 projects with code written only by you and your partner. 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!