

The Course Website

Welcome to CS 61A! We have various resources on the course site which you might find helpful as you go through the course during the semester.

Here's a summary of some of these resources. You can follow along with the navigation bar at the top of the page.

- Calendar:
 - Weekly Schedule:
 - * Weekly lab and discussion schedule.
 - Office Hours:
 - * Weekly office hours schedule where you can get help from course staff. We will have several kinds of office hours available:
 - Advising office hours, for questions that don't pertain directly to course content (e.g. advice on what classes to take or how to go about seeking internship opportunities)
 - In-person office hours, where you can join other students in a larger room to work collaboratively, and join a queue to get help from course staff
 - Online office hours, where you can book a 20-minute with a member of course staff for quick debugging or conceptual questions
- Staff:
 - The list of everyone on course staff, with contact information.
- Resources
 - Request an Extension
 - * Use this Google form to request an extension on an assignment due to extenuating circumstances—you can read more about our extension policy [here](#), but it's good to know that we will approve any extension up to 24 hours on any non-extra credit assignment for any reason, no-questions-asked.
 - Book an Accommodations Appointment
 - * Here you can book an accommodations appointment if you wish to discuss ways we can help make the course work best for you—you can read more about accommodations appointments [here](#), but these will likely be especially of interest to DSP students and students in the process of enrolling in DSP.
 - Topical Resources + Past Exams:
 - * Past exams (grouped by topic or semester) and study guides. Some past exams have video walk-throughs. In these videos, staff members explain how they approached the exam problem as they solve it.
 - PythonTutor:
 - * This allows you to run Python code and visualize the corresponding environment diagram for the execution of the code.
 - Code:
 - * The CS 61A web editor. You can run doctests and use the interpreter. It has support for Python, Scheme, and SQL, as well as Lark (which we won't be using over the summer). It can visualize environment diagrams following the execution of your code for debugging. Moreover, it can visualize trees, lists, and linked lists. Any files you save here get saved on your local storage. You can also edit okpy backups, but you *cannot* run ok tests.
 - Ed:

2 Getting Started

- * The course forum. Feel free to ask questions about the content or logistics here. Make sure to famaliarize yourself with the [policies and guidelines](#) for using Ed!
- Department/Campus Resources:
 - * These are resources, not necessarily specific to CS 61A, that many students find helpful, e.g., campus and department advising, mental health resources, basic needs resources, etc. If you know of a resource that we should add to this page, let us know!
- Syllabus
 - Policies for the course regarding assignments, grades, DSP and accommodations, and other parts of the course.
- Contact
 - Various ways you can contact course staff if you have questions, or concerns.
 - Feedback forms for the instructors, staff, or department..
 - The extension form and accommodation appointments calendar are linked here, in addition to being listed on the course syllabus and listed under the “Resources” dropdown.
 - If a student or staff member makes you feel uncomfortable at any point during the semester, that doesn’t represent everyone, and that incident should be reported. The contact page has the anonymous feedback form and the EECS contact form.

Hot tip: You can also draw on the site for discussion pages! To do so, you can click on the pencil icon on the top left.

Lost on the Moon

Your spaceship has just crashed on the light side of the moon. You were scheduled to rendezvous with a mother ship 200 miles away on the lighted surface of the moon, but the rough landing has ruined your ship and destroyed all the equipment on board except for the 15 items listed below (note: you are able to consume food/water/medicine inside your space suit).

Your crew's survival depends on reaching the mother ship, so you must choose the most critical items available for the 200-mile trip. Your task is to rank the 15 items in terms of their importance for survival. Place a number 1 by the most important item, number 2 by the second most important, and so on, through number 15, the least important.

Item	Your Rank (1)	Group's Rank (2)	NASA's Rank (3)
Box of matches			
Food concentrate			
50 feet of nylon rope			
Parachute silk			
Solar-powered portable heating unit			
Two .45 caliber pistols			
One case of dehydrated milk			
Two 100-pound tanks of oxygen			
Stellar map (of the moon's constellations)			
Self-inflating life raft			
Magnetic compass			
5 gallons of water			
Signal flares			
First-aid kit containing injection needles			
Solar-powered FM receiver-transmitter			

Secrets to Success in CS 61A

CS 61A is definitely a challenge, but we all want you to learn and succeed, so here is a collection of various tips that might help in your journey:

- Find what works for you.
 - There are many different ways to succeed in 61A—we've found that the below advice applies to almost all students, but there are going to be different things that work best for different people, and we recommend exploring different study and work strategies that will work well for you. If you ever want to talk with a TA about your approach, we have [advising office hours](#) for exactly that kind of thing as well!
- Ask questions.
 - If you encounter something you don't know or aren't sure about a concept or problem, *ask* away. We're here to help you learn, and if you ask a question, that tells us where we can help with your understanding of the material. The process of asking questions itself can also be helpful in figuring out for yourself what you would specifically like to ask about, and in therefore pinpointing concepts that you believe you can learn more about.
- Study in groups.
 - Again, this class is not an easy course for most students; you might feel overwhelmed going at it alone. Send a message and reach out to other students in the class; work together on assignments or study together to prepare for exams, as long as you don't violate the course policy on Academic Honesty as stated in the [Syllabus](#).

- When stuck on a problem, try to verbally explain the area in which you are stuck.
 - This doesn't need to require a person who understands how to solve the problem (or even a person—this practice is often referred to as **rubber ducking**, since you can take a rubber duck and consider it your practice audience), because the main goal is for you to clarify your own thoughts and figure out where exactly you're getting stuck with your understanding and code. From there you can focus on that portion to better your understanding.
- If you're still feeling stuck, feel free to make use of the class [Ed Page](#) (our course forum for asking and answering questions) or attend office hours.
- Office hours gives you time with the instructors or staff by themselves, and you will be able to get some (nearly) one-on-one instruction. You are *not* intruding; the instructors and staff are here to help as you learn.
- Do (or at least attempt seriously) all the homework. We do not give many homework problems, but those we do give you may find are challenging, time-consuming, and rewarding.
- Do all the lab exercises. Most of them are designed as more of an introduction to the course material, and may take around the length of a lab section. This is a great time to get acquainted with new material. Feel free to ask the staff members in your lab section or come to office hours if you would like more guidance.
- Do the readings before lecture. They can be helpful in offering an overview of the material covered in lecture and other aspects of the class, as well as being a reference as part of the overall textbook that you can refer to as you go through the course.
- When preparing for the exams, **do past exam questions!**
 - Lecture, lab, and discussion provide a great introduction to the material, but the main way to learn how to solve exam-level problems is to do exam-level problems. You can find past exams (and other resources) under the [Resources](#) tab of the website. Many past exams also have solutions or walkthrough videos where past staff members may walk you through the solution to the exam.