Welcome to CS61A!

What we'll discuss today...

- Introductions
- What is CS?
- About the course
- Demo!

Introductions

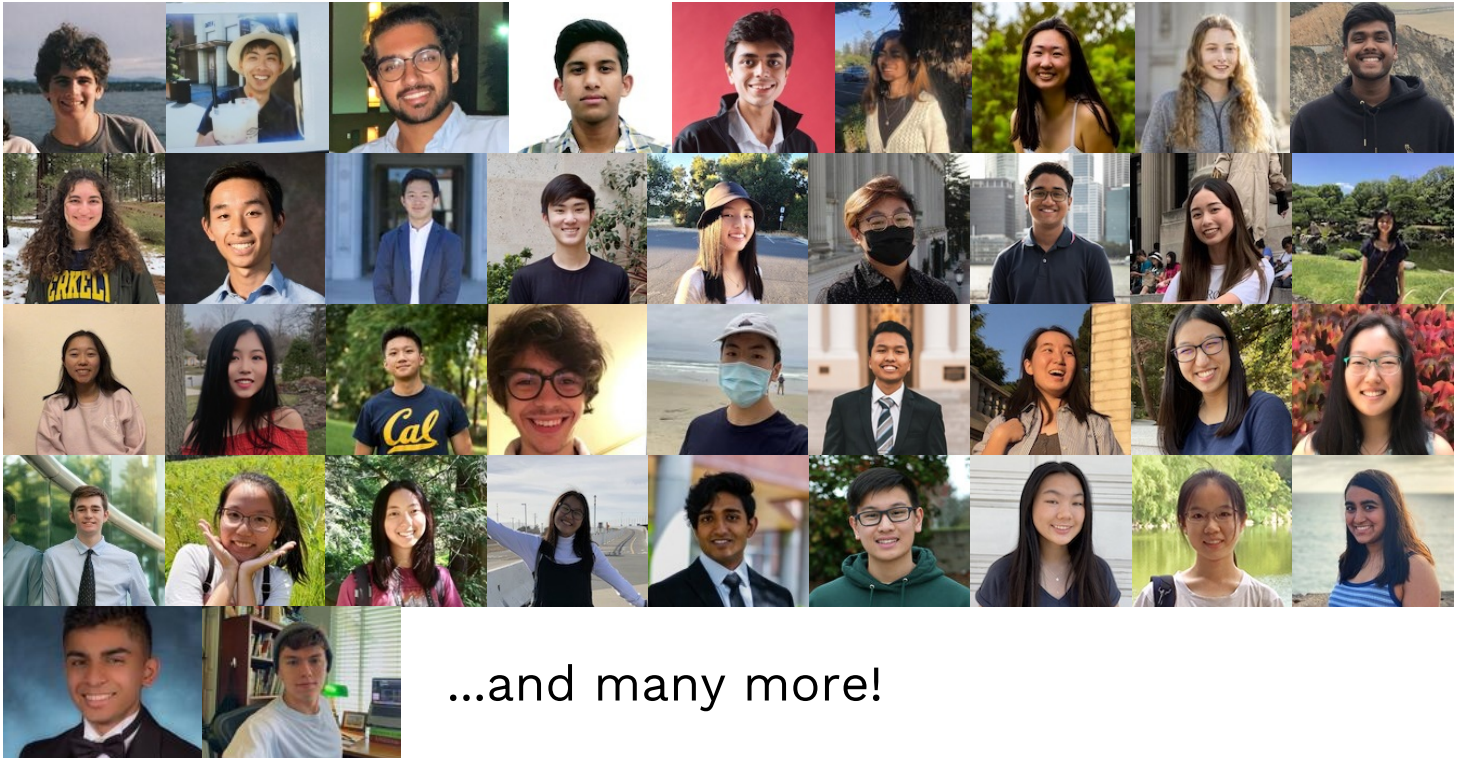
Instructors

John DeNero: Started teaching at UCB in 2014. Created the Python version of CS 61A and co-created Data 8. Associate Dean of the CDSS. Previously worked as a research scientist at Google.

Pamela Fox: Started teaching last spring! Previously created the Khan Academy computing courses, and worked for Coursera, Google, and Woebot.

TAs

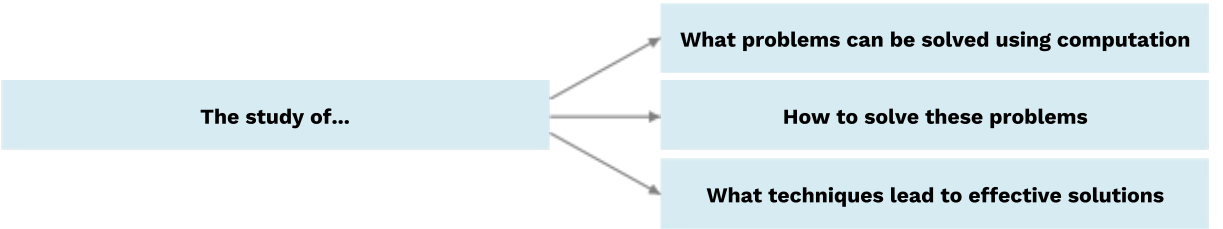
Tutors



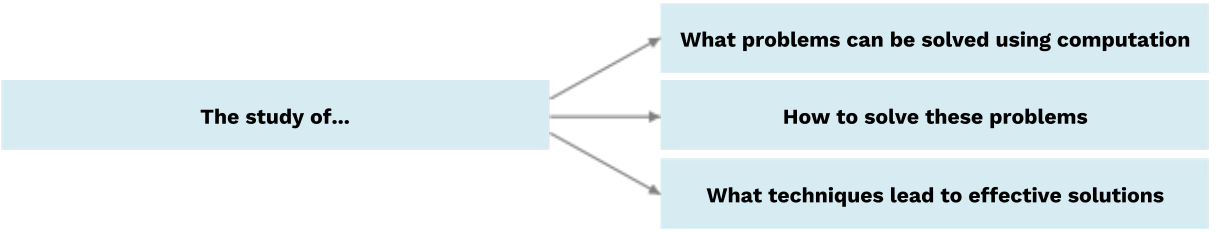
...and many more!

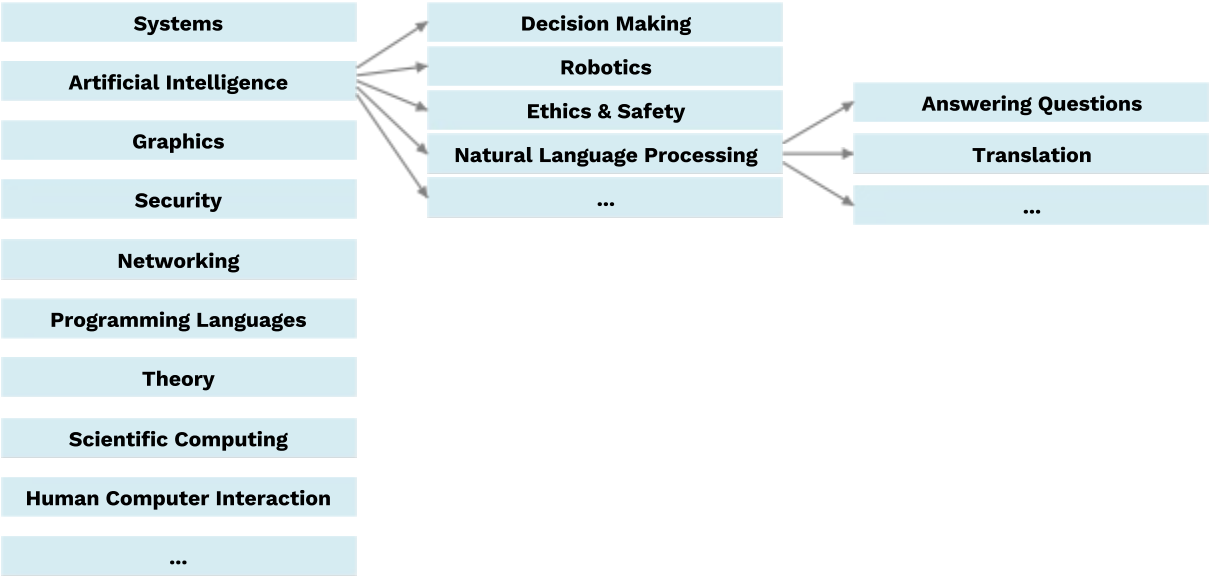
A Very Brief Introduction to Computer Science

What is Computer Science?



What is Computer Science?





About this course

Course topics

- Managing complexity in programs (procedural abstractions, data abstractions, programming paradigms)
- Deep understanding of programming concepts (using Python)
- How computers interpret computer programs
- Different types of languages (Regex, BNF, SQL, Scheme)
- Problem solving techniques (both iterative and recursive approaches)

This course is challenging and often mind-blowing!

Course prerequisites

This is **not** an introductory programming class.

Prerequisites from the official description:

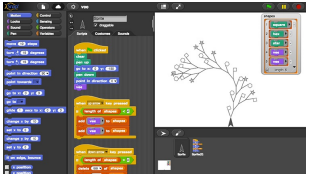
> "MATH 1A (may be taken concurrently); programming experience equivalent to that gained from a score of 3 or above on the Advanced Placement Computer Science A exam."

If you are a data science major, also consider **CS 88**, which goes at a slightly slower pace.

If you do not think you have enough programming experience, consider taking **CS 10** and joining us in the spring.

CS 10: The Beauty and Joy of Computing

- Designed for students without prior coding experience
- Starts off in Snap!, a programming environment created by Berkeley and now used in classrooms globally.



- Introduces higher-order functions and recursion, two of the traditionally challenging CS61A topics.
- Also teaches Python fundamentals.

More info: cs10.org

If you need help enrolling or have any questions, just email us.

Course format

Course components

- Lectures
- Labs
- Discussions
- Homeworks
- Projects
- Exams 😱
- Textbook (composingprograms.org)
- Office hours

Everything is linked from https://cs61a.org

Weekly schedule

	Monday	Tuesday	Wednesday	Thursday	Friday
Morning		Complete Lab, Attempt homework	Complete Lab	Attend Discussion, Finish homework	
2pm	Lecture		Lecture		Lecture
After	Complete Lab		Attend Discussion		Submit Project

Lectures

THREE exciting ways for you to watch lectures:

- John's pre-recorded lectures
  (Typically posted day-of)
- Pamela's "live" Zoom lectures
  (MWF 2:10-3pm)
- Recordings of the Zoom lectures
  (Available around 4:30pm)

Labs & Discussions

Sign up for sections at sections.cs61a.org

You'll have the same TA for both lab and discussion. Community! ❤

Section types:

- Regular
- Remote
- 2x
- CS Scholars

Homeworks & Projects

Homeworks typically due Thursday, projects typically due Friday. Start early, code often!

We will schedule homework and project "parties" so you can be around other students working on them.
You can discuss the assignments at a high-level, but don't copy anyone else's code (unless it's your project partner).

Exams 😱

- Midterm 1: Mon, Sept 13, 8-10pm
- Midterm 2: Mon, Oct 27, 7-9pm
- Final Exam: Thursday, Dec. 16, 3-6pm

All past exams are available on the resources page. Study early, study often!

Office hours

Check out the calendar: cs61a.org/office-hours/

Instructors also have office hours:

- John's will be a recorded Zoom Q&A, Mondays 4-5.
- Pamela's will be non-recorded, Mon Wed 3-4.

Getting help

Post questions on Piazza. If you're debugging assignment code, follow the debugging template.

Check out our contact page for more ways to get in touch.

Course policies

Course policies

Read the syllabus. (There will be a quiz!)

Learning

Community

Course Staff

Collaboration

Asking questions is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner
- Choose a partner from your discussion section

The limits of collaboration

- Please don’t look at someone else's code!
  Exceptions: lab, your project partner, or after you already solved the problem
- Please don't tell other people the answers! You can point them to what is wrong and describe how to fix it, but don't tell them what to type, and don't type for them
- Copying project solutions causes people to fail the course
- We really do catch people who violate the rules, and we're getting better at it.

Demo!

What's next?

- Discussions will meet this week, starting today (sections.cs61a.org)
- Optional Lab 0 to get your computer setup
- Next lecture is on Friday, Zoom you there!