Welcome to CS 61A!

Lecture will begin at 2:10pm.

(Until then, you just get to watch us type.)

Looking for something to do?
Try reading the "Syllabus" link at the top of cs61a.org

Questions?
There is an Ed thread for this lecture

Welcome to CS 61A

Your Instructors

John DeNero
denero@berkeley.edu

CS 61A instructor many times

Teaching Professor in EECS and help with
the Data Science undergraduate program

Research focused on human-computer
interaction with text generation AI

Office hours start next week (Jan 24):

• 2-3:30 Tuesdays in 101B Warren Hall

Hany Farid
hfarid@berkeley.edu

Intro CS instructor many times

Professor in both the School of
Information and EECS

Research focused on digital forensics,
forensic science, misinformation, image
analysis, and human perception

Office hours:

• TBD

61A Course Staff

https://cs61a.org/TAs/   https://cs61a.org/tutors/ (coming soon)

About the Course

Parts of the Course

Lecture: Videos/recordings posted to cs61a.org

Lab: Practice ideas from lecture on a computer (often in pairs)

Discussion: Practice ideas from lecture on paper (often in groups)

Assignments: Weekly homework + 4 large projects

Staff Office Hours: Get 1-on-1 help with assignments & work with peers

Online textbook: http://composingprograms.com

Monday-Wednesday: Attend lab and complete the lab assignment

Wednesday-Friday: Attend discussion

Watch (or go to) lecture before you show up to lab/discussion!

Sunday/Monday/Tuesday: Start on the homework or project

Wednesday/Thursday: Finish the homework or project

Friday: Finish projects you didn't finish on Thursday

?

Ed: All staff (private posts) and students (public posts)

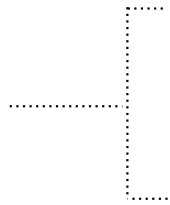**cs61a@berkeley.edu:** Head TAs and instructors

denero@berkeley.edu or **hfarid@berkeley.edu:** Might be slow

cs61a.org: Self-service answers to many questions

cs61a.org/contact/: Even more ways to reach the course staff

An Introduction to Computer Science

What is Computer Science?

The study of ⋯⋯⋯⋯⋯
- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

A course about managing complexity

  Mastering abstraction

An introduction to programming

  Full understanding of Python fundamentals

  Combining multiple ideas in large projects

  How computers interpret programming languages

  Different kinds of programming languages

Should you take CS 61A?

According to the Syllabus

There is no formal programming-related prerequisite for CS 61A, but...

- Taking the course without any prior programming experience is typically very challenging.

- Most CS 61A students have had significant prior programming experience.

- Students who take the course without prior programming experience typically must work substantially harder to master the material and tend to receive lower final grades in the course.

Students who take the course later often get more out of it due to increased understanding.

# CS 10: The Beauty and Joy of Computing

Designed for students without prior experience

A programming environment created by Berkeley, now used in courses around the world and online

An introduction to fundamentals (& Python) that sets students up for success in CS 61A

If you might switch to CS 10, start attending ASAP and enroll soon before it fills!

More info: http://cs10.org/
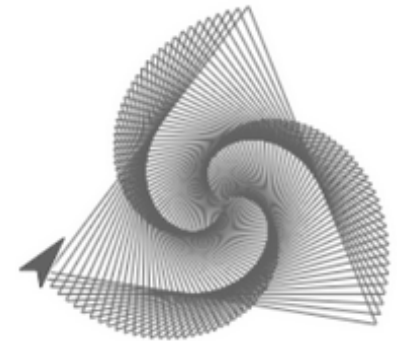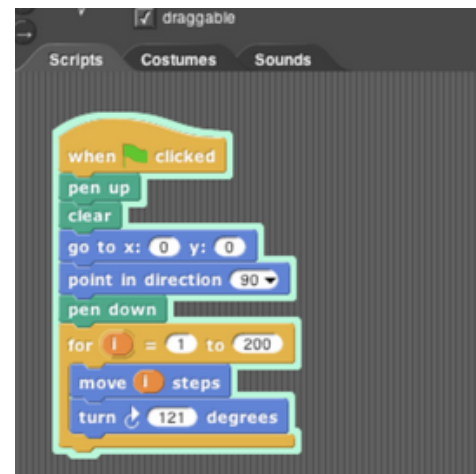
Data C88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

- Two programming projects (instead of four) that are adapted from CS 61A projects

- Everything you need to know to continue on to CS 61B

- Omits the unit on how programs run other programs

For students taking Data 8 (Foundations of Data Science) or who took it already

We're investigating expansion options

Course Policies

Learning

Community

Course Staff

Details...

https://cs61a.org/articles/about/

Getting Help

If you're struggling, let us know.

If you need more time, ask for it.

If you need special accommodations, make an appointment.

Collaboration

Working together 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

What constitutes academic misconduct?

- *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 or show them a related example

- Copying project solutions causes people to fail the course

(Switch)

75% of students report cheating on at least one exam per semester
(~20% are probably lying)

75% of students report cheating on at least one exam per semester
(~20% are probably lying)

50% of faculty ignore cheating in their class
(not us!)

75% of students report cheating on at least one exam per semester
(~20% are probably lying)

50% of faculty ignore cheating in their class
(not us!)

every semester dozens of students fail this course because of cheating

cheating is disrespectful

cheating is disrespectful

cheating destroys trust

cheating is disrespectful

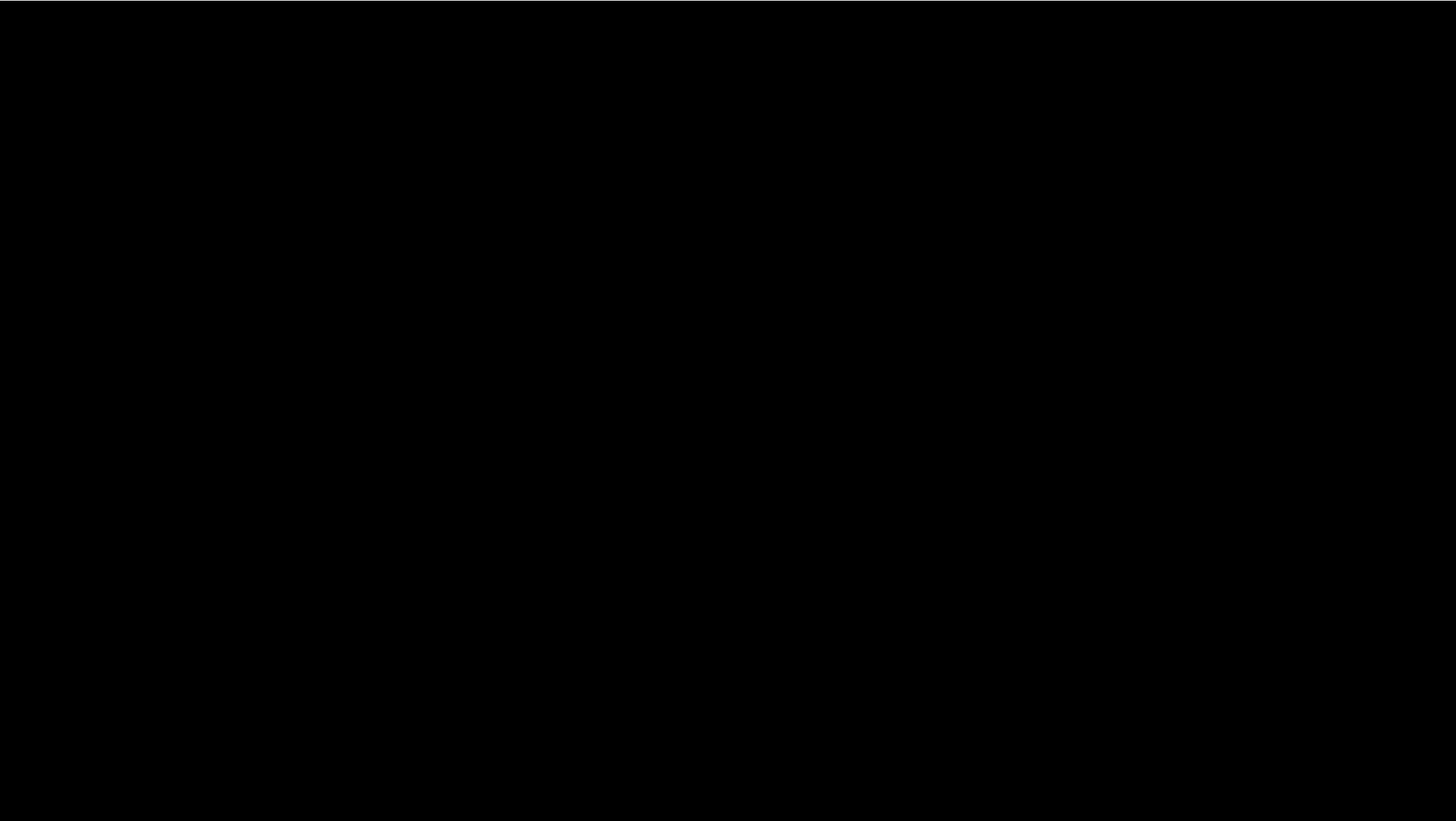cheating destroys trust

cheating is a trap

cheating is disrespectful

cheating destroys trust

cheating is a trap

cheating completely misses the point

it is just wrong

computer science is (much) more than coding

```python
1  # load some libraries
2  import cv2
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import matplotlib.image as image
6  from PIL import Image
7  from numpy import linalg as la
```

```python
1  # load and display image
2  im = image.imread('license.png')
3  plt.imshow(im)
4  plt.axis('off')
5  plt.show()
```


abstraction

```
1   # specify world coordinates of rectangular license plate
2   X = np.array([100, 400, 400, 100])
3   Y = np.array([100, 100, 200, 200])
4
5   # specify image coordinates of distorted license plate
6   u = np.array([129.6, 169.5, 183.7, 142.9]);
7   v = np.array([216.5, 218.9, 86.4, 91.1]);
8
9   # display image with image coordinates
10  im = image.imread('license.png')
11  plt.scatter( u, v, s=100, facecolors='none', edgecolors='y')
12  plt.imshow(im)
13  plt.axis('off')
14  plt.show()
```


data structures

data visualization

function + form

```python
1   # estimate homography
2   A = np.zeros((8,9))
3   for i in range(0,4):
4       A[2*i,:]   = [0, 0, 0, -X[i], -Y[i], -1, v[i]*X[i], v[i]*Y[i], v[i] ]
5       A[2*i+1,:] = [X[i], Y[i], 1, 0, 0, 0, -u[i]*X[i], -u[i]*Y[i], -u[i] ]
6
7   At    = A.transpose()
8   L,V   = la.eig( At@A )
9   h     = V[:,-1] # minimial eigenvalue eigenvector (assumes that eigenvalues
10  H     = np.reshape(h,(3,3))
11  H     = la.inv(H)
12
13  # warp source image based on homography
14  im_warp = cv2.warpPerspective(im, H, (2*im.shape[1],im.shape[0]))
15
16  # display rectified image
17  plt.imshow(im_warp)
18  plt.gca().invert_yaxis()
19  plt.axis('off')
20  plt.show()
```


math

numerical methods

solving problems

ethical computing

Course Climate

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

From the EECS department mission:

"Diversity, equity, and inclusion are core values in the Department of Electrical Engineering and Computer Sciences. Our excellence can only be fully realized by faculty, students, and staff who share our commitment to these values."

All faculty and staff members are *mandated reporters*. If we ever receive a report of harassment, we must report to the Office for the Prevention of Harassment & Discrimination.

- CS61A Anonymous feedback form: If you want to stay anonymous but make us aware of something happening in the course.
- EECS Student Climate & Incident Reporting Form: Informs the EECS department of any issues. You can also contact Susanne Kauer (skauer@berkeley.edu) directly.

The Best Approach to CS 61A

Help each other understand concepts in the class, whether in section, on Ed, or in study groups, without expectation of anything in return.

Be great project partners by listening to what your partner suggests and helping them understand the work you've done together.

Recognize that we're all valuable members of the CS community!

Programming

(Demo)