

Lecture 1: Welcome to CS88!



-





CS88 Team – Gerald and Michael

- Michael Ball
 - ball@Berkeley.edu – You're best off by using Ed! ☺
 - 784 Soda Hall / [Berkeley.zoom.us](https://berkeley.zoom.us) / my apartment
 - <http://michaelball.co> – I don't update this much...
 - » It was great procrastination when I was a CS student.
 - Office hours: tentatively Monday afternoon.
- Things I do:
 - Intro CS Research (Tools, curriculum)
 - Training TAs
 - Building Educational Software (Gradescope)
 - Tools for web accessibility





Head Teaching Assistant



Anjali Gurajapu [she/her]

Office Hours: TBD

agura@berkeley.edu

Hi! I'm Anjali, and I'm a senior studying Chemistry and Data Science. In my (limited) free time, I like drawing, baking, and jigsaw puzzles. Looking forward to meeting you all, and reach out anytime! :)

CS88 Team

Teaching Assistants

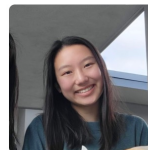


Amit Sant [he/him]

Office Hours: TBD

amitsant2000@berkeley.edu

Hello! My name is Amit, and I am a senior studying computer science. I am super excited to be teaching this course during my last semester as an undergrad, and I hope that we can make it something special. In my free time, I like to learn more about history, city planning, and transportation, though to be honest I just obsess whatever wikipedia rabbit hole I get sucked down for the week. I also love playing League of Legends, and I am a big fan of JoJo's Bizarre Adventure.



Christine Zhang [she/her]

Office Hours: TBD

cyuzhang@berkeley.edu

Hi! I'm Christine and I'm a 2nd year EECS major and physics minor. I love playing badminton, cooking, eating Chipotle, and occasionally watching NFL. I can beat expert minesweeper in under 3 minutes. Reach out to me anytime for anything! I'd love to meet you :)



Ethan Yoo [he/him]

Office Hours: TBD

ethanyoo7912@berkeley.edu

Hello! My name is Ethan and I am a second-year Applied Mathematics and Data Science major. My favorite activities are going out with friends and exercising (soccer, tennis, and hiking). Hope I can assist you all, and look forward to meeting you!



Hetal Shah [she/her]

Office Hours: TBD

hetal.shah@berkeley.edu

Hello! I am a senior studying CS. I am from Redondo Beach (it is mentioned in the Beach Boys song Surfin' USA), and I love the beach a lot. I am currently rereading all of the Ranger's Apprentice books, and my favorite grocery store is Berkeley Bowl. I can't wait to meet all of you! Feel free to ask me questions about anything!

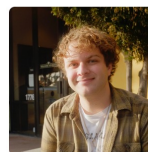


Joanna Yoo [she/her]

Office Hours: TBD

joannayoo@berkeley.edu

Hi! I'm a senior studying data science from Seattle! Excited to meet you all :)



Karim Kaylani [he/him]

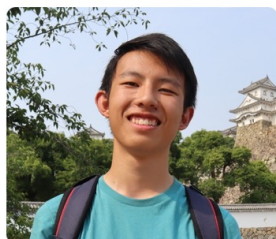
Office Hours: TBD

karimkaylani@berkeley.edu

Hi everyone! My name is Karim and I'm a 3rd year CS major from Southern California. I'm super passionate about all things music whether it's playing guitar, collecting vinyls, or going to shows. I also love taking film photos, hiking, wordle, and ghibli movies. I'm very excited to be your TA this semester and meet you all, never hesitate to reach out to me about anything anytime! :)



CS88 Team



Lukas Chang [he/him]

Office Hours: TBD

lukasc@berkeley.edu

Hi everyone I'm Lukas, a 4th year CS major from the south bay area. This is my fourth semester TAing for CS88 and I'm excited to meet you all! A little about me—in my free time I love making/listening to music, thrifting, and watching anime. I hope I can share my love for CS with you all!



Michelle Chen [she/her]

Office Hours: TBD

michelle.chenn@berkeley.edu

hi! im michelle a sophomore studying cs+econ. some things i love doing: eating, traveling, photography/videography, baking, arts and crafts, exploring new things & ofc meeting my students. feel free to reach out :)



Mingxiao Wei [she/her]

Office Hours: TBD

mingxiaowei@berkeley.edu

Hi there! I'm Mingxiao, a junior majoring in CS/MCB major from China. I've been on course staff for cs61a for 5 semesters and this is my first time TAing for cs88. I enjoy cooking, strolling around, and eating poke. Super excited to meet y'all (•̀•́) /



Sebastian Zhao [he/him]

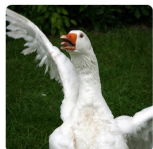
Office Hours: TBD

sebbyzhao@berkeley.edu

Hiya! I'm a CS and DS double major from Erie, PA (a little town under the lake). I like making plated desserts, Chopin etudes, and impressionist art. Talk to me about anything, I love conversations about random stuff!

CS88 Team

Tutors

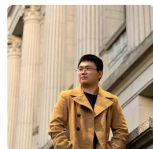


Aymeric Barrier [he/him]

Office Hours: TBD

aymeric.barrier@berkeley.edu

"Hi everyone! I'm Aymeric and I'm a fourth-year checked-out senior double majoring in CS and Political Science! My main goal in life is to curl up and read in bed, but I'm usually too busy for that. I'm also involved with CSM on campus, go check them out! Come talk to be about anything, I promise I don't bite (and I love being distracted from my work), especially if you're trying to figure out your major or what to do at Cal. Fun fact(s), I only discovered computer science my second year at Berkeley, I came in as a History major then thought I'd do Econ, and I was a data science major for two whole months!"



John Teng [he/him]

Office Hours: TBD

johnnteng9@berkeley.edu

Hi, I'm John, a second year CS major from Pennsylvania. I like playing video games, soccer, and working out. Looking forward to this semester!



Ramya Chitturi [she/her]

Office Hours: TBD

ramya.chitturi@berkeley.edu

"Hi! I'm Ramya, a sophomore majoring in CS. I enjoy sci-fi books, trivia, crosswords, rock music, museums, civic technology, and more! Excited to get to know you this semester :)"

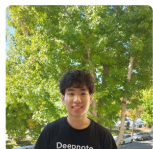


Rebecca Dang [she/her]

Office Hours: TBD

rdang@berkeley.edu

Hey there! I'm a 2nd year EECS major from San Jose, CA. This is my first time as a tutor and I'm super excited to meet you! Besides being on course staff, at Berkeley I'm involved in CSM and Codebase. In my free time, I like to play guitar (and maybe record a few covers if I'm feeling brave) and fangirl about the latest book/movie/TV show/song I consumed. If you ever have any questions about 88, classes, clubs, or professional development, feel free to reach out :D



Sean Yang [he/him]

Office Hours: TBD

sean_yang@berkeley.edu

Hi, I'm Sean, a 2nd year Data Science Major from LA. I enjoy playing games, taking walks, and talking to new people. Looking forward to meeting you all!





In The News

Madison Square Garden Uses Facial Recognition to Ban Its Owner's Enemies

MSG Entertainment, the owner of the arena and Radio City Music Hall, has put lawyers who represent people suing it on an “exclusion list” to keep them out of concerts and sporting

By Kashmir Hill and Corey Kilgannon

NY Times

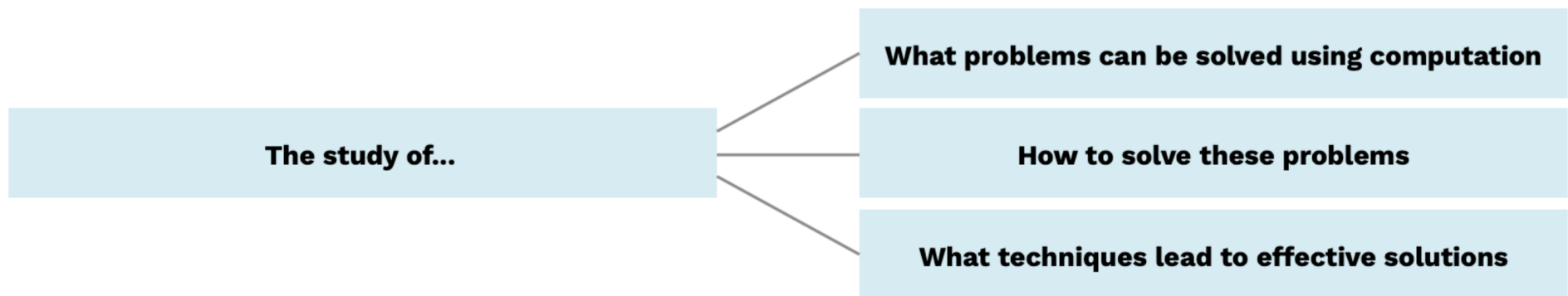
Published Dec. 22, 2022

Over Thanksgiving weekend, Kelly Conlon, 44, a personal injury lawyer from Bergen County, N.J., was chaperoning her 9-year-old daughter's Girl Scout troop on a trip into Manhattan to see the “Christmas Spectacular” at Radio City Music Hall.

Before she could even glimpse the Rockettes, however, security guards pulled Ms. Conlon aside and her New York jaunt took an Orwellian turn.

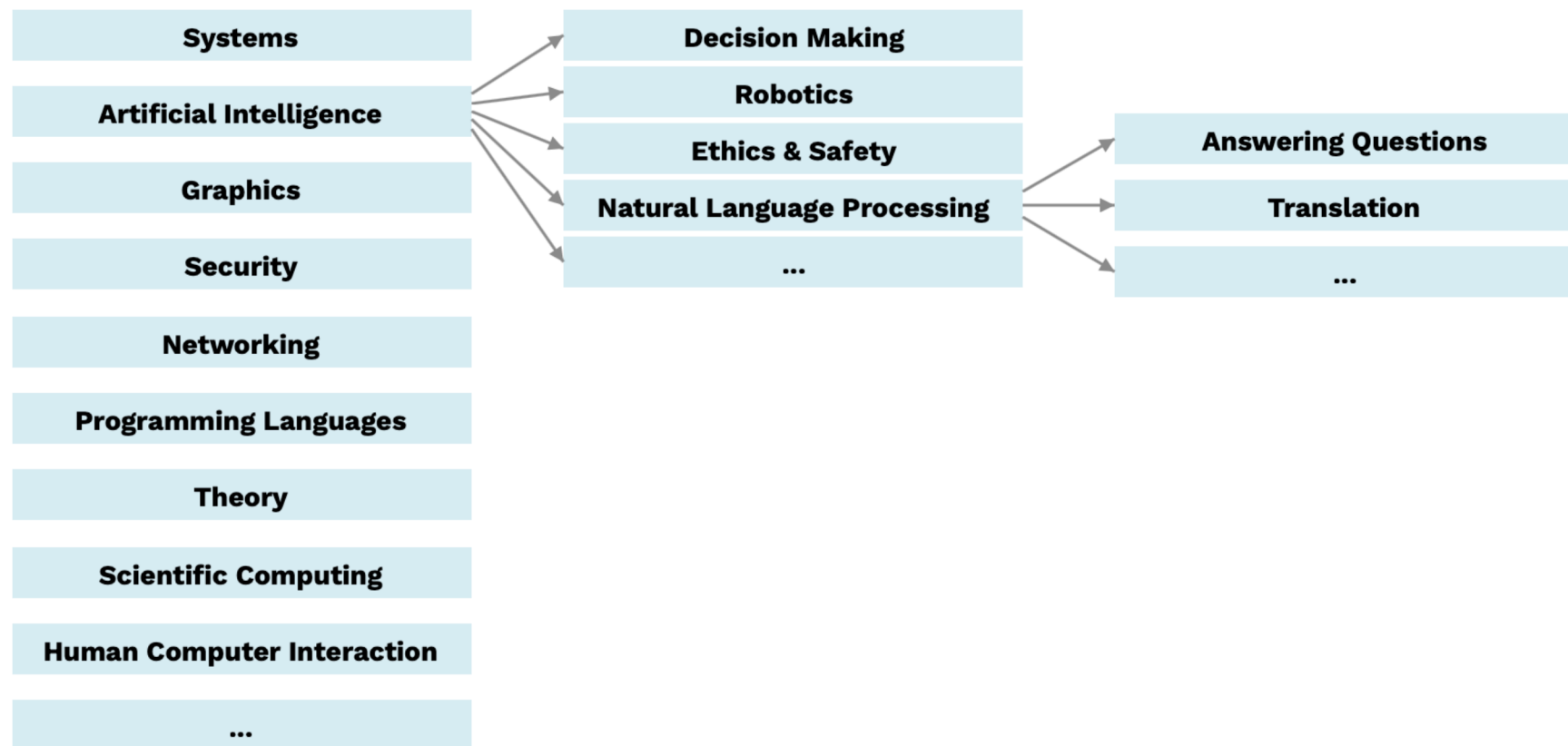


Computer Science

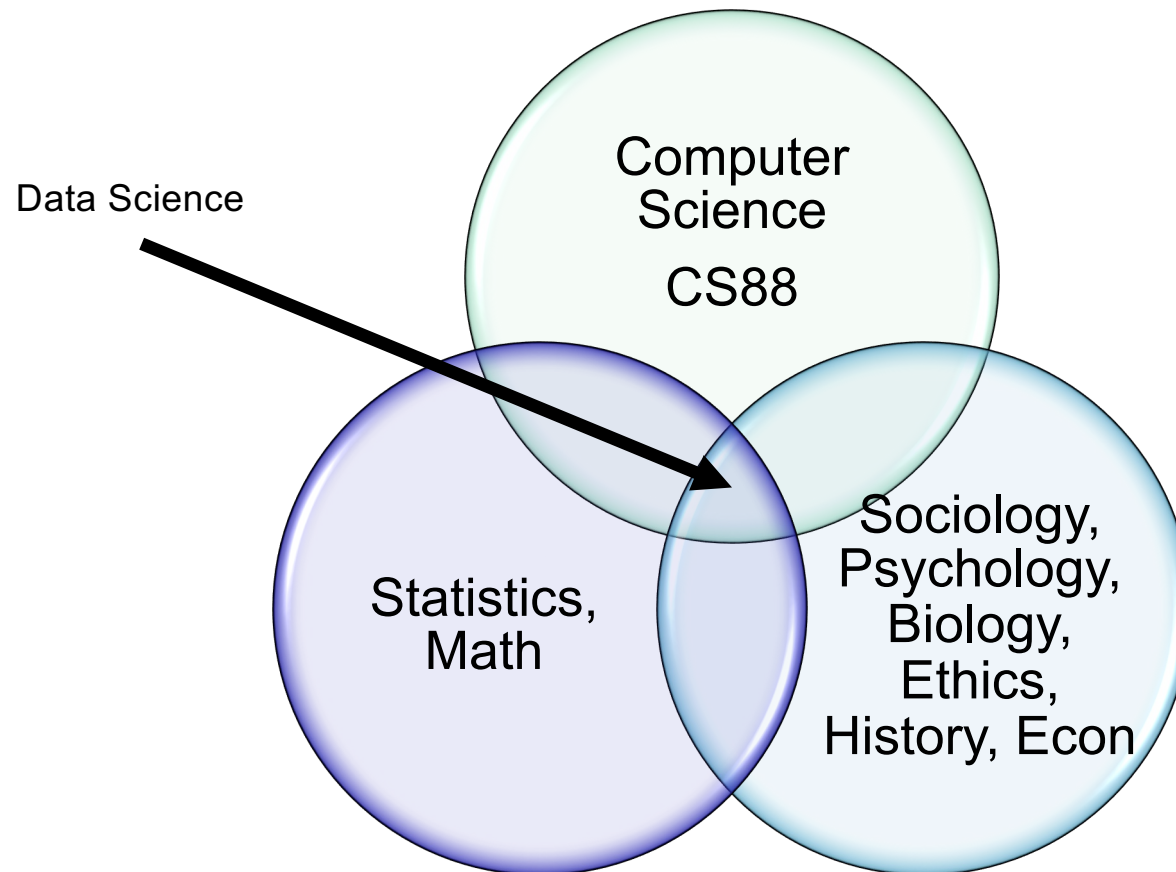




Computer Science, Some Ideas...Definitely Not Exhaustive!



Computer Science & Data Science (One View)



CS88 – Computational Structures in Data Science



- Deeper understanding of the computing concepts introduced in DATA8
 - Hands-on experience => Foundational Concept
 - How would you create what you use in DATA 8?
- Extend your understanding of the structure of computation
 - What is involved in interpreting the code you write ?
 - Deeper CS Concepts: Recursion, Objects, Classes, Higher-order Functions, Declarative programming, ...
 - Managing complexity in creating larger software systems through composition
- Create complete (and fun) applications
- In a data-centric approach



Course Culture

- Learning
- Community
 - Collaboration
 - Peer Instruction
- Respect
- A supportive course staff & environment
 - Lots of outside community, CS Mentors, HKN, others.



So...COVID... Still

- Lectures are in person, but attendance isn't required
- Lecture recordings will be posted afterwards
- Labs primarily in-person, but will have an online section (TBD)
- OH mostly in person, but will have some online, and some online review sessions



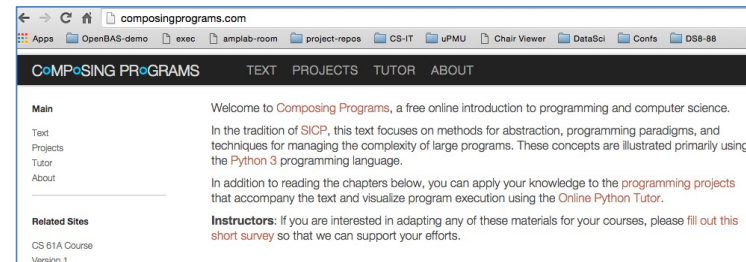
Collaboration

- Asking questions, discussing topics, helping each other is always encouraged!
 - When you're working with a partner, you are expected to share in the work.
- Collaboration has limits
 - Please don't read someone else's code
 - » except for LABS if you have already turned in the assignment, or a TA/staff member is present.
 - You can help others, but not give the solutions.
- We have a very particular set of skills and we will use them.



Course Structure

- 2 lectures, 1 lab each week
- Lecture introduces concepts (quickly!), answers why questions.
- Lab provides concrete detail hands-on
- Homework (12) cements your understanding
- Projects (2) put your understanding to work in building complete applications
 - Maps
 - Ants vs Some Bees



- Readings: <http://composingprograms.com>
 - Same as cs61a



Class Format

- Mon and Weds Lectures:
 - Each lecture has a series of short self-check questions
 - Lectures go quickly
- Labs are paced throughout the week. See the Ed post to pick a time.
- Labs are HANDS ON – get help as you're trying the lab.



Class Format: Assignments

- Lecture Quizzes, 1 point, max 20.
 - 1 per lecture, due in 1 week. (Partial credit after)
 - <https://go.c88c.org/1>
- Lab Work: 4 points, 12 labs, 1 drop
 - Start them during lab! You can probably finish some labs in 2 hours. Will be Python + some interactive questions.
- Homework: 8 points, 12 HW, 1 drop
 - Start early!
- Projects: 100 points between 2 projects
 - Start early! "Checkpoint" assignments
 - Maps & Ants



Class Format: Assignments

- Slip Days: 9 total
 - Use up to 3 on any assignment
 - We apply them in the order that's most beneficial!
 - » i.e. use them on projects if you need!
 - Can be used for homework, labs, projects, but not project checkpoints.
- Slip Days take care of nearly all, but not all special circumstances!
- We will release an exceptions / extensions request form.
- What if you go over slip days?
 - 25% deduction for each day over. Mathematically you can still earn 25% if you turn in something 3 days late.

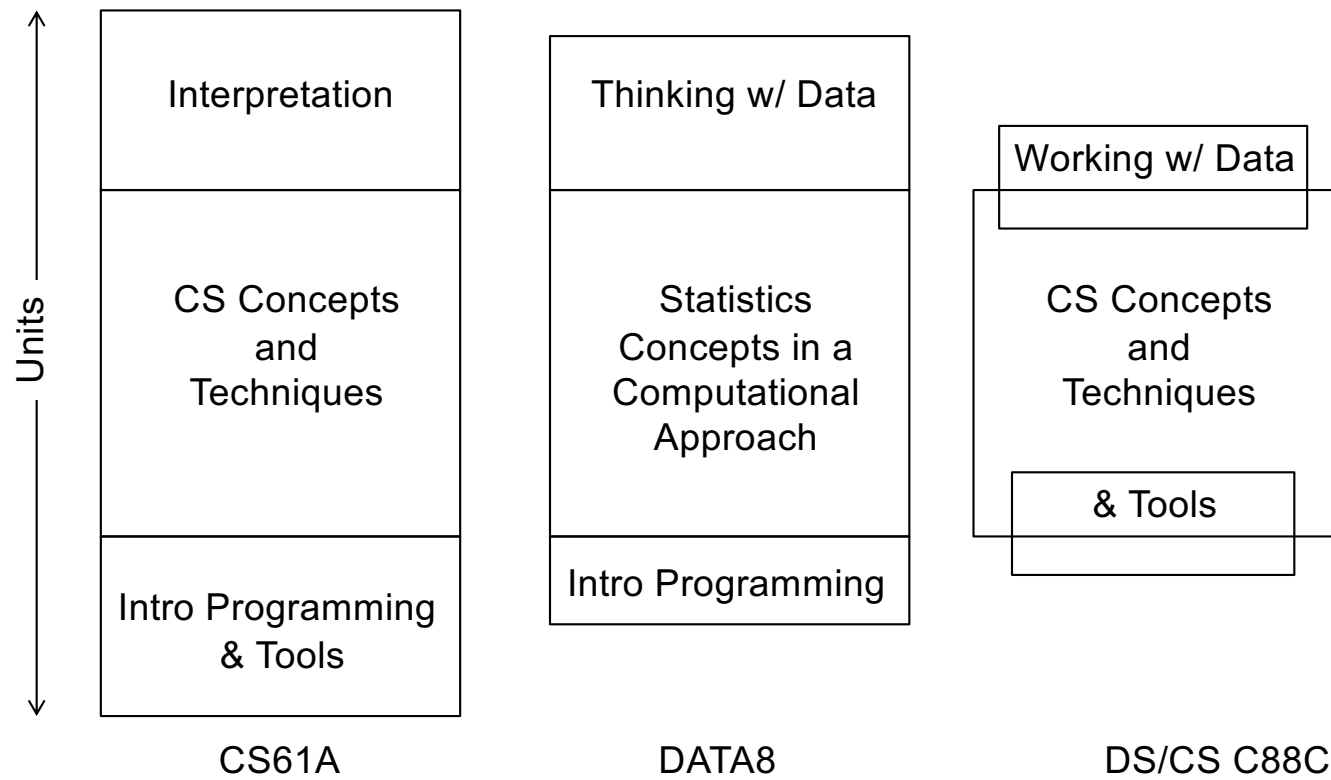


Class Format: Exams

- 1 midterm and 1 final exam, in person
 - remote, backup options
 - Remote exams proctored via Zoom
- Midterm 2 hours, mid-March
- Exam will be during the slot assigned by campus.
 - Tues 11:30am
- 8 *handwritten* cheat sheets double-sided.



How does C88C relate to CS61A ?



Opportunities for students



DATA 8	CS88
--------	------

DS Minor

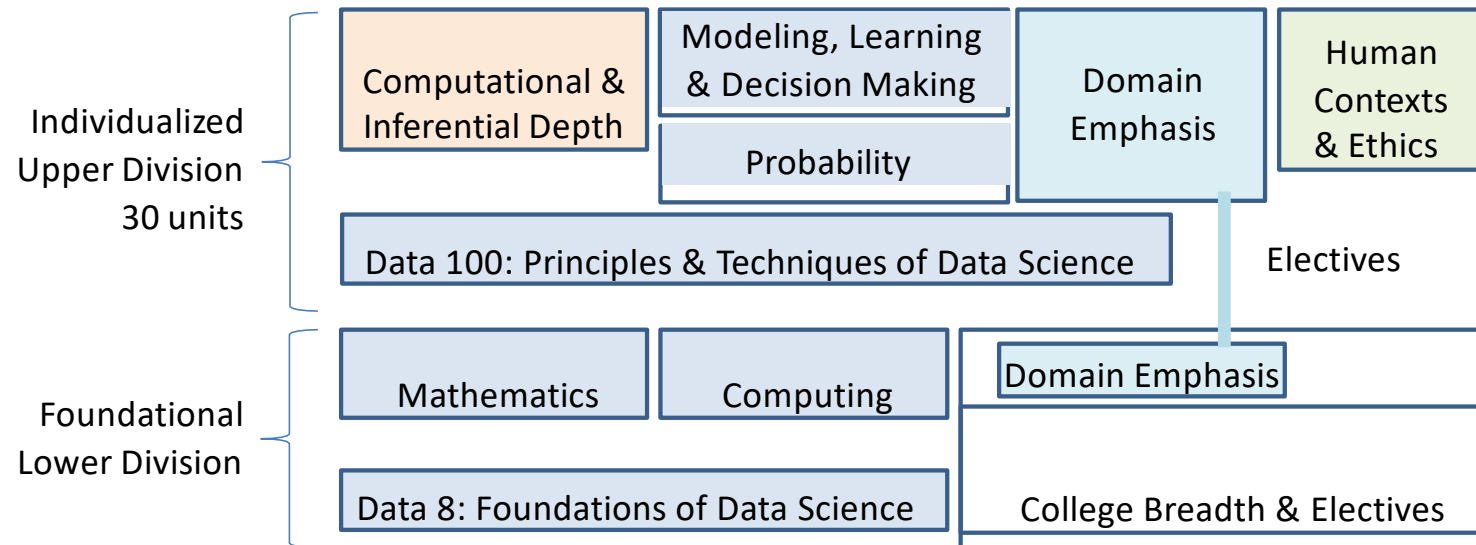
DATA 8	CSv88	CS61B
--------	-------	-------

CS minor & DS Major

DATA 8	CS88	CS47A
	CS61A	

CS major:
CS47A path is not recommended, but there if you change your mind.

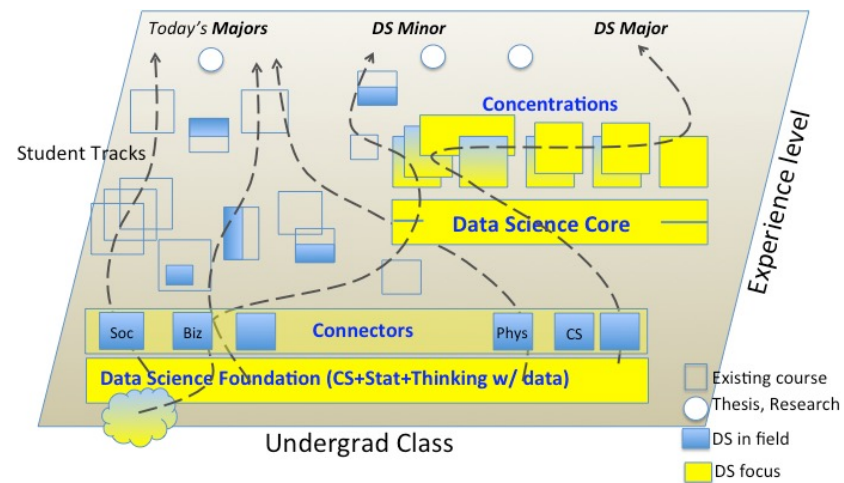
The Data Science Major





Data 8 – Foundations of Data Science

- Computational Thinking + Inferential Thinking in the context of working with real world data
- Introduce you to several computational concepts in a simple data-centered setting
 - Authoring computational documents
 - Tables
 - A LOT of statistics





Ed For Class Discussion: Try it!

New Thread

COURSES

Berkeley CS Sandbox8

CS302

CS 61BL3277

CS 883

CS 1693

CATEGORIES

General

Lectures

Social

Labs

Homework

Projects

Exams

Search

Filter

Pinned

Lab 0 Section Times

Labs Brian M INSTRUCTOR 2h

Fall 2020 Zoom Links

Lectures Michael Ball INSTRUCTOR 13h

Welcome

General Michael Ball INSTRUCTOR 2d 1 8

This Week

Two Finals?

Exams Anonymous 32m 2

Final exam timing conflict

Exams Meghana Kumar 4h 1 (1 new)

recording for live lecture and lab

General Anonymous 15h 1

CS88

General Dat Le 18h 2 (1 new)

Lecture Zoom Link?

Lectures Anonymous 20h 1

Test Staff Post

General Michael Ball INSTRUCTOR 20h 8

1 unit GSI course?

General Anonymous 21h 2

Lab times

Welcome

M

Michael Ball INSTRUCTOR

2 days ago in General

UNPIN

STAR

WATCHING

881 VIEWS

Hi everyone,

8

Welcome to CS88 Fall 2020!!

We're just getting things setup, so you'll find some stuff is less than perfect. Please bear with us! (Bad pun intended. If you're allergic to bad puns I might recommend another course. No hard feelings.)

A Short List Week 1 Tasks:

Please attend any lab section this week! We will be sending out a welcome survey as well as form to sign up for permanent section times. Labs in CS88 are part lab, part discussion. They're a time to meet peers and your TA. They are challenging, but hopefully interesting and engaging. There's plenty of times to get questions answered!

Please checkout this short welcome video and let us know how you're feeling about the course.

Welcome to CS88

Computational Structures in Data Science

Fall 2020

Berkeley

UNIVERSITY OF CALIFORNIA



Where will we work?

- Your laptop
 - Using an editor and a terminal
- okpy – okpy.org
- c88c.org
- gradescope.com – Lecture Self-Checks
- Ed Discssion: edstem.org
 - Can write and run (!!) python in you own posts!



Your Tasks

- Lecture 1 Quiz On Gradescope
- Watch Ed for info about section signup.
- Attend Lab o:
 - * Wed 4-6pm, Hildebrand B51
 - * Th 10-12, Hearst Mining 310
 - * Th 12 - 2pm, Hearst Field Annex B
 - * Th 12 - 2pm, Etch 3111

Welcome, and Good luck!

Questions?





UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science



Abstraction



Abstraction

- **Detail removal**
"The act of leaving out of consideration one or more properties of a complex object so as to attend to others."
- **Generalization**
"The process of formulating general concepts by abstracting common properties of instances"
- Technical terms: Compression, Quantization, Clustering, Unsupervised Learning



Henri Matisse "Naked Blue IV"

Experiment

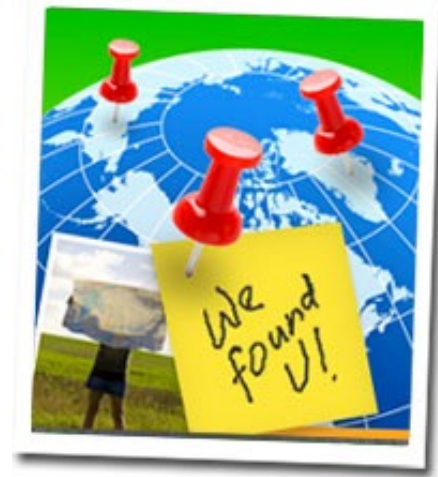




Where are you from?

Possible Answers:


- Planet Earth
- Europe
- California
- The Bay Area
- San Mateo
- 1947 Center Street, Berkeley, CA
- 37.8693° N, 122.2696° W



All correct but different levels of abstraction!



Abstraction gone wrong!




I Can Stalk U


Raising awareness about inadvertent information sharing

[Home](#) [How](#) [Why](#) [About Us](#) [Contact Us](#)


What are people *really* saying in their tweets?




[denisluque](#): I am currently nearby <http://maps.google.com/?q=-23.6193333333,-46.5506666667>
1 minute ago • [Map Location](#) • [View Tweet](#) • [View Picture](#) • [Reply to denisluque](#)




[nikosofficiel](#): I am currently nearby <http://maps.google.com/?q=48.8699833333,2.3282833333>
5 minutes ago • [Map Location](#) • [View Tweet](#) • [View Picture](#) • [Reply to nikosofficiel](#)



[dilmanarede](#): I am currently nearby <http://maps.google.com/?q=-15.7878333333,-47.8291666667>
7 minutes ago • [Map Location](#) • [View Tweet](#) • [View Picture](#) • [Reply to dilmanarede](#)



[downtownvan](#): I am currently nearby <http://maps.google.com/?q=49.2833333333,-123.1198333333>
10 minutes ago • [Map Location](#) • [View Tweet](#) • [View Picture](#) • [Reply to downtownvan](#)



[MommaGooseBC](#): I am currently nearby 15745 Weaver Lake Rd Maple Grove MN

Links

- Mayhemic Labs
- PaulDotCom
- SANS ISC
- Electronic Frontier Foundation
- Center for Democracy & Technology

How did you find me?

Did you know that a lot of smart phones encode the location of where pictures are taken? Anyone who has a copy can access this information

Detail Removal (in Data Science)

- You'll want to look at only the interesting data, leave out the details, zoom in/out...
- Abstraction is the idea that you focus on the essence, the cleanest way to map the messy real world to one you can build
- Experts are often brought in to know what to remove and what to keep!



The London Underground 1928 Map & the 1933 map by Harry Beck.



The Power of Abstraction, Everywhere!

- Examples:

- **Functions (e.g., $\sin x$)**
- **Hiring contractors**
- **Application Programming Interfaces (APIs)**
- **Technology (e.g., cars)**

- Amazing things are built when these layer

- **And the abstraction layers are getting deeper by the day!**

*We only need to worry about the interface, or specification, or contract
NOT how (or by whom) it's built*

Above the abstraction line

Abstraction Barrier (Interface)
(the interface, or specification, or contract)

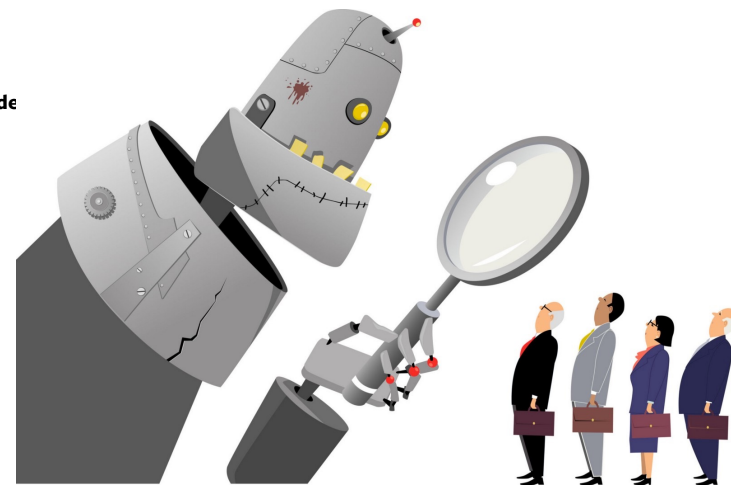
Below the abstraction line

This is where / how / when / by whom it is actually built, which is done according to the interface, specification, or contract.



Abstraction: Pitfalls

- **Abstraction is not universal without loss of information (mathematically provable). This means, in the end, the complexity can only be “moved around”**
- **Abstraction makes us forget how things actually work and can therefore hide bias. Example: AI and hiring decisions.**
- **Abstraction makes things special and that creates dependencies. Dependence**

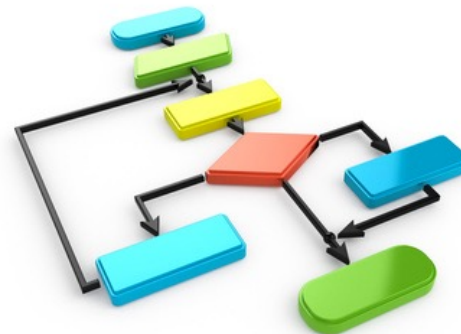




Algorithm

- An algorithm (pronounced AL-go-rith-um) is a procedure or formula to solve a problem.
- An algorithm is a sequence of instructions to change the state of a system. For example: A computer's memory, your brain (math), or the ingredients to prepare food (cooking recipe).

Think Data 8: Change or retrieve the content of a table.





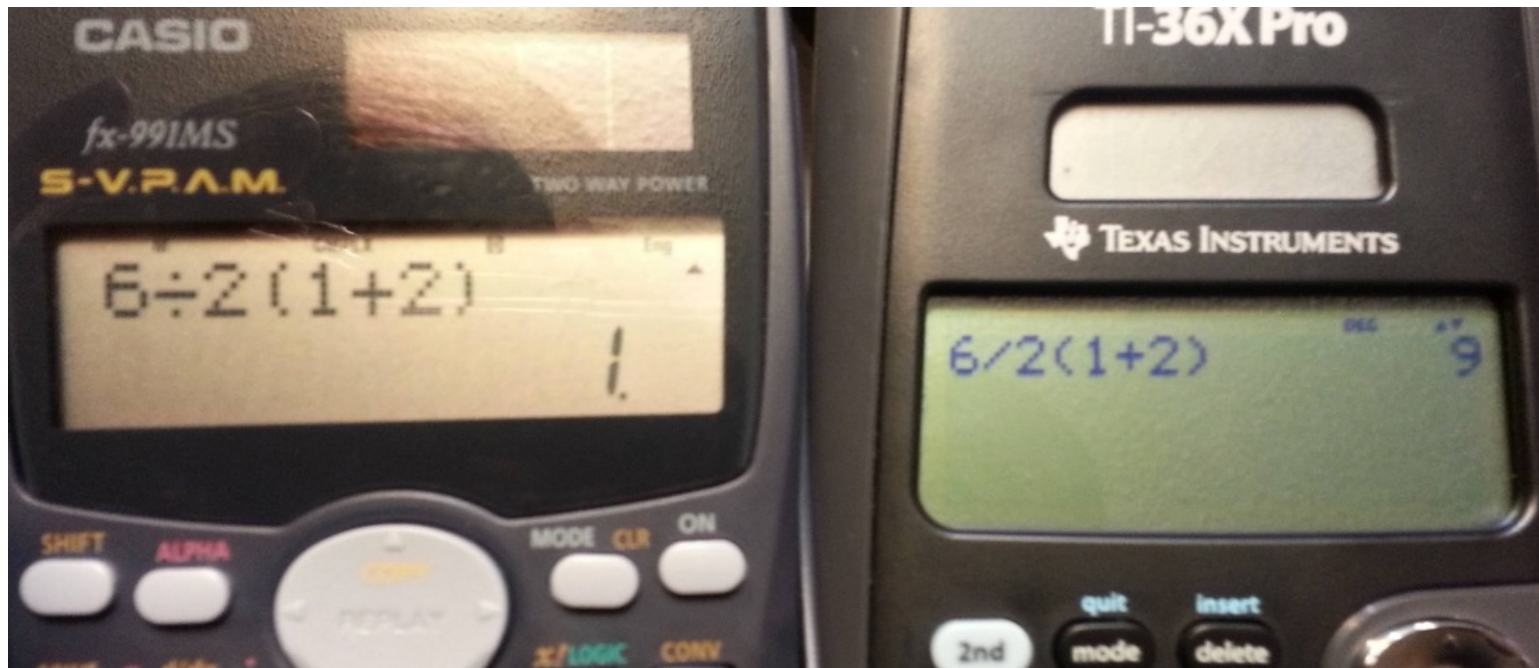
Algorithm: Properties

- An algorithm is a description that can be expressed within a finite amount of space and time.
- Executing the algorithm may take infinite space and/or time, e.g. “calculate all prime numbers”.
- In CS and math, we prefer to use well-defined formal languages for defining an algorithm.

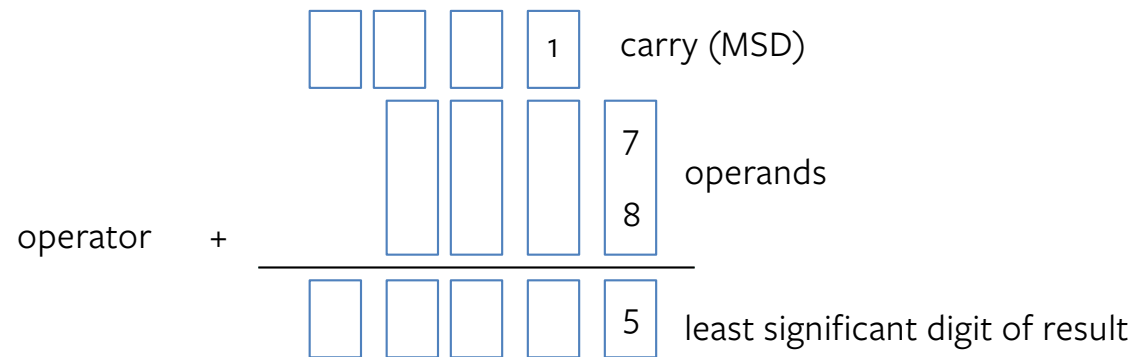
$$6 \div 2(1+2) = ?$$

1 or 9

Algorithm: Well-Definition



Algorithms Early In Life (1st Grade)





More Terminology (Intuitive)

Code

A sequence of symbols used for communication between systems (brains, computers, brain-to-computer)

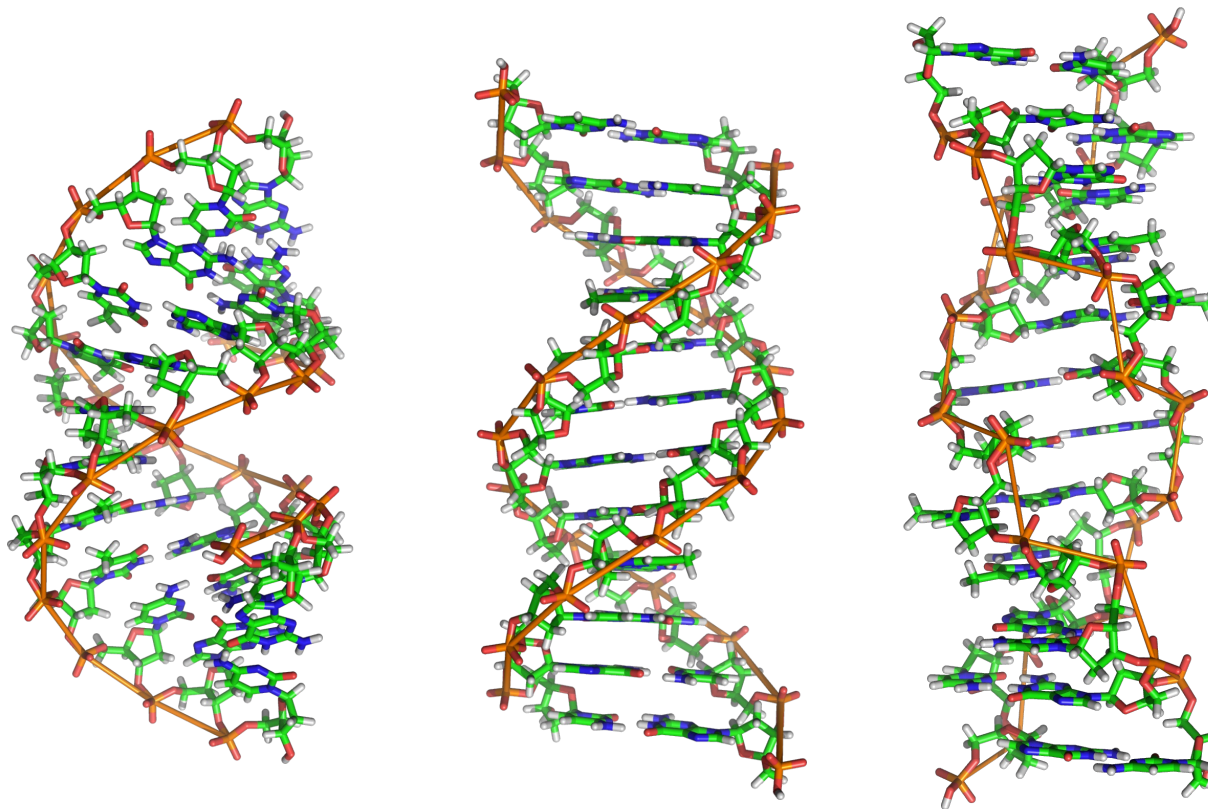
Data

Observations

Information

Reduction of uncertainty in a model (measured in bits)

Data or Code?





Data or Code?

```
00000000 10000000 01000001 10000000 00010000 00000000 10000001
01000001 10000001 00010000 00000000 10000002 01000001 10000002
00010000 00000000 10000003 01000001 10000003 00010000 00000000
10022133 01000001 10022133 00010000 00000000 10000000 01000001
20000000 00010000 00000000 10000001 01000100 20000001 00010000
00000000 10000001 01000100 10000000 00010000 00000000 10031212
01000001 10031212 00010000 00000000 10031212 01000100 10031213
00010000 00000000 10000002 01001001 10000001 00010000 00000000
10000001 01001001 10000001 00010000 00000000 10000101 01001001
10000001 00010000 00000000 10011111 01001001 10011111 00010000
00000000 10100220 01001001 10011111 00010000 00000000 10000001
```



Data or Code?

Here is some information!

Integer

00000000	10000000	01000001	10000000	00010000	00000000	10000001
01000001	10000001	00010000	00000000	10000002	01000001	10000002
00010000	00000000	10000003	01000001	10000003	00010000	00000000
10022133	01000001	10022133	00010000	00000000	10000000	01000001
20000000	00010000	00000000	10000001	01000100	20000001	00010000
00000000	10000001	01000100	10000000	00010000	00000000	10031212
01000001	10031212	00010000	00000000	10031212	01000100	10031213
00010000	00000000	10000002	01001001	10000001	00010000	00000000
10000001	01001001	10000001	00010000	00000000	10000101	01001001
10000001	00010000	00000000	10011111	01001001	10011111	00010000
00000000	10000220	01001001	10011111	00010000	00000000	10000001

Instruction

String



Data or Code? Abstraction!

Human-readable code
(programming language)

```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print ' %s [label="%s" % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s];' % ast[1]  
        else:  
            print "]"  
    else:  
        print "[";  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
        print ' %s -> {' % nodename,  
        for name in children:  
            print '%s' % name,
```

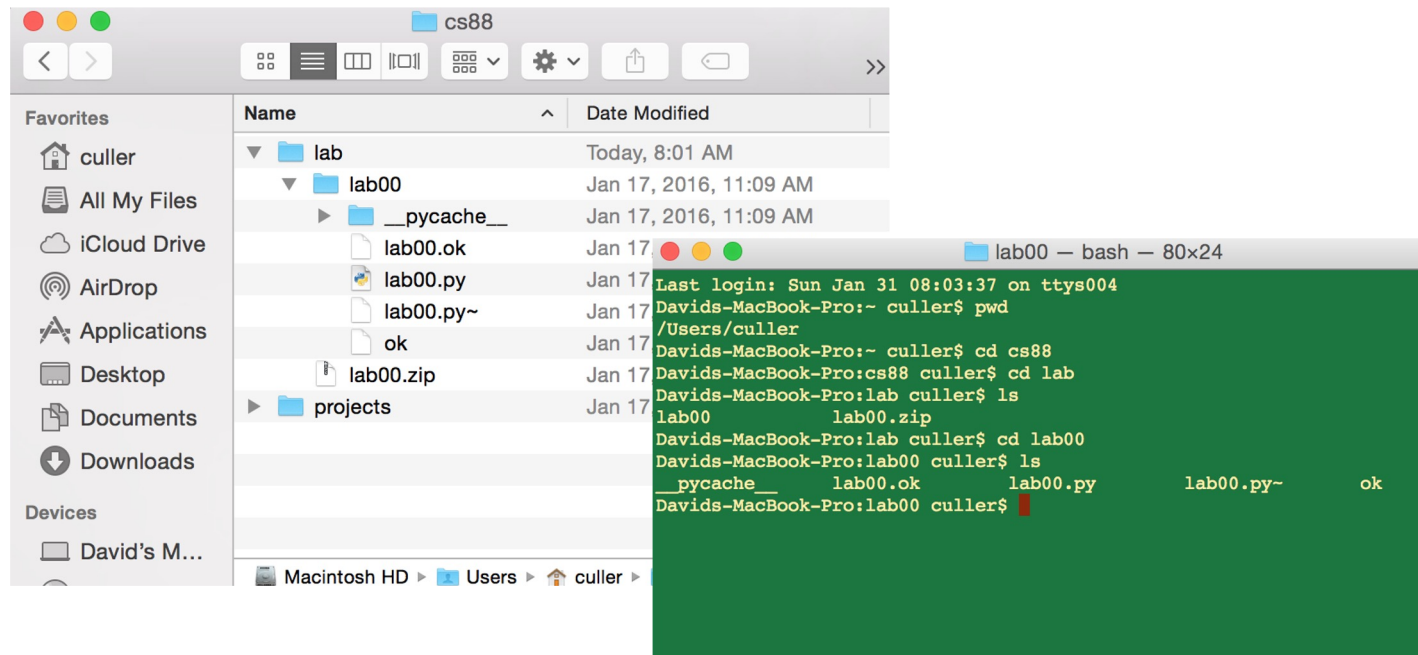
Machine-executable
instructions (byte code)

```
011100111100010011011000010001110100010011111011000111  
1011101100000011111101111001100001111111111110111110  
111111111100111101000110101001100011100010010111001000  
1111000110101011001111011011110010011110111111111111  
11001111111001100000000001101111010010110011111101111  
111111110000011100011100011111001110000000110101111110  
000011101001110010011111011111000011111001100110001011  
100111110000110001100110101111001111100010111010111111  
10010011111111001110111100011111100011011111000111110  
110111101110101111011100111111100111111001111000100111  
1111100010010111100011110001111111111111111111110111  
1110111111110000111000001011110011111110000000111001100  
101000001110011111101111111111110000000110001000011000  
1110011101101110111111111001011111011101111000000111111  
1100110011000100001000111111110001111100100000100001000  
00001111110111001001110000111111011111111111111000100111  
100001100110010111001000110001001101111000011000111111  
0011110011111100111111001111001110011011011111110010111111  
1110011111110111100010011111110111111100111111110000  
0101101101110110111111110100110101010111111101000010
```

Compiler or Interpreter
Here: Python



Code or GUI: More Abstraction!



- Big Idea: Layers of Abstraction
 - The GUI look and feel is built out of files, directories, system code, etc.



Review:

- Abstraction:
 - Detail Removal or Generalizations
- Code:
 - Is an abstraction!
 - Can be instructions or information

Computer Science is the study of abstraction