



CS88 Team – Gerald and Michael

- Michael Ball
 - ball@Berkeley.edu – You're best off by using Ed! 😊
 - 625 Soda Hall / 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 Wednesday early afternoon.
- Things I do:
 - Intro CS Research (Tools, curriculum)
 - Training TAs
 - Building Educational Software (Gradescope)
 - Tools for web accessibility





Head Teaching Assistants



Matt Au [he/him]

Office Hours: Tuesday 5pm - 7pm,
Thursday 5pm - 6pm

mattau@berkeley.edu

Hi! I'm Matt, a 3rd-year CS major from Southern California. I've recently been listening to CLOUDS by NF, and am waiting on "I'm Not Going Anywhere" to drop. Let me know if you have any music recommendations :D



Shreya Kannan [she/her]

Office Hours: Tuesday 11am - 12pm

shreyakannan@berkeley.edu

Hi everyone, I'm super thrilled to meet y'all this semester! In no particular order, I prefer musicals over concerts, cold over warm weather, dancing over singing, and sushi over tacos. Feel free to talk to me about great restaurants, cool study spots, and of course, CS 88 :)

CS88 Team

Teaching Assistants



Anjali Gurajapu [she/her]

Office Hours: Wednesday 4pm - 5pm

agura@berkeley.edu

Hi! I'm Anjali, and I'm a third year studying Chemistry and Data Science. In my free time, I've been drawing, baking, and trying to befriend the stray cat that sleeps on my porch. I'm excited to be a TA this semester, and looking forward to meeting you all! :)



Chi Tsai [she/her]

Office Hours: Tuesday 12pm - 1pm

chitsai@berkeley.edu

Hi! I'm Chi, and I'm from the LA area. I enjoy cooking and trying new food, playing minecraft, planning upcoming vacations, and watching Brooklyn 99 (Terry loves yogurt). Come talk to me about anything, happy to meet you!

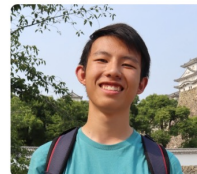


Jessica Lin [she/him]

Office Hours: Monday 10am - 11am

linjessica@berkeley.edu

Hi friends! I'm a second year CS major from Southern California. I enjoy dancing, reading, doing crosswords, and making random peace signs. Feel free to reach out to me for anything (:



Lukas Chang [he/him]

Office Hours: Monday 11am - 12pm

lukasc@berkeley.edu

Hi everyone I'm Lukas, a 3rd year CS major from the south bay area. This is my third 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!

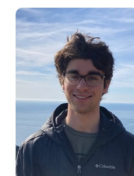


Minnie Chen [she/her]

Office Hours: Tuesday 6pm - 7pm

chn_minnie_22@berkeley.edu

Hi! I'm Minnie, a 4th year Civil Engineering major and minoring in EECS and Sustainability. I like to cook when I'm not too lazy and enjoy active sports when I'm not too tired or lazy :D I'm trying to get back into reading so please let me know if you have any book recs! I also recently studied abroad last semester, so feel free to ask me about that. Excited to meet everyone!



Tommy Joseph [he/him]

Office Hours: Thursday 4pm - 5pm

tommy11jo@berkeley.edu

Hi I'm a third-year CS major from Southern California, and this is my third time TAing for CS 88. Right now, some things I like are bread, blogs, and tv shows (highly recommend The Wire and Silicon Valley). Feel free to reach out about anything

CS88 Team

Tutors



Amit Sant [he/him]

Office Hours: Wednesday 3pm - 4pm,
Friday 12pm - 1pm

amitsant2000@berkeley.edu

Hello, my name is Amit, and I am a third year CS major at UC Berkeley. My hobbies include gapping League of Legends esports, chess, anime, and osu! Aside from that I love to code, teach, and work on stuff related to making our planet somewhat more livable.



Hetal Shah [she/her]

Office Hours: Friday 11am - 12pm

hetal.shah@berkeley.edu

Hello! I am a 3rd year CS major from Southern California. I love the beach and the sun. In my free time, I like working out, reading, hiking, and traveling. Feel free to reach to me!



Kevin Gu [he/him]

Office Hours: Thursday 6pm - 7pm,
Friday 10am - 11am

kevinjgu@berkeley.edu

Hi everyone! I'm Kevin, a 5th year Master of Information and Data Science student in the UC Berkeley School of Information. I'm so excited to be a tutor and am looking forward to meeting all of you during office hours! Outside of academics, I love to listen to music (mainly classical, but I can appreciate any genre), take walks, travel (unfortunately not during Covid, but in general), and learn languages (just not enough time to do it, but am willing to practice speaking French), so feel free to reach out to me regarding any of these. :D



In The News

AI Turned a Rembrandt Masterpiece into 5.6 Terabytes of Data

Popular Science

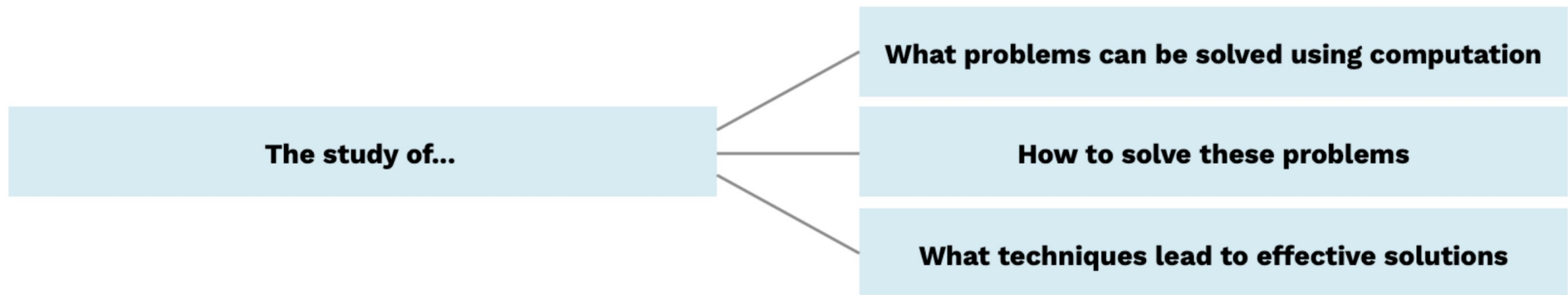
Purbita Saha

January 11, 2022

The Rijksmuseum in the Netherlands has posted an ultra-high-resolution image of Rembrandt's "The Night Watch," constructed from 8,439 photos taken with a 100-megapixel HD camera that were digitized, color-corrected, and stitched together by algorithms. The Rijksmuseum said it is the largest digital image of an art piece ever created, at 717 billion pixels and 5.6 terabytes of data. Visitors to the museum's Website can zoom into every 0.0005-millimeter square at fine resolution. The digital version of "The Night Watch" was created following a two-year restoration process in which artificial intelligence was used to restore missing elements from the original painting.

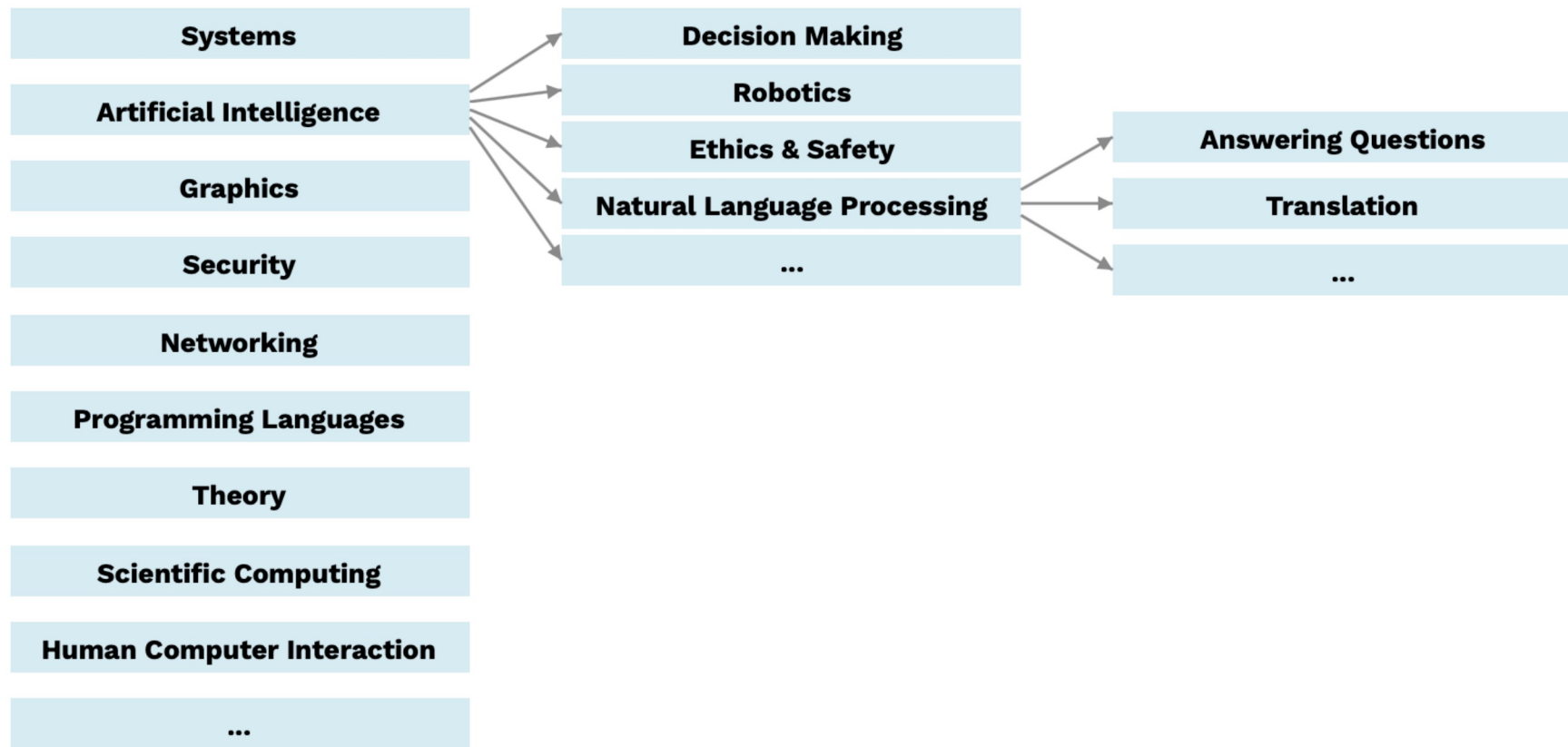


Computer Science

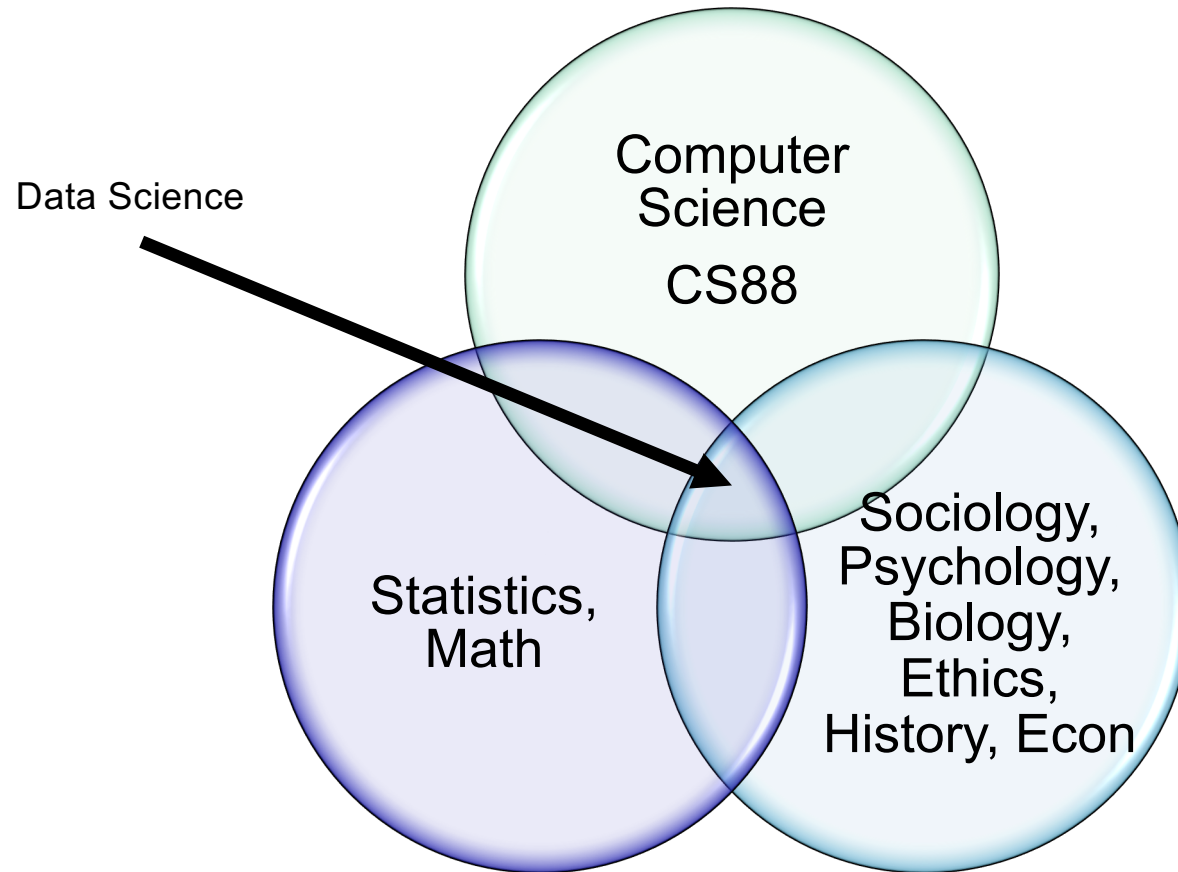




Computer Science, Some Ideas...Definitely Not Exhaustive!



Computer Science & Data Science (One View)



CS88 – Computational Structures in Data Science



- Deeper understanding of the computing concepts introduced in c8
 - Hands-on experience => Foundational Concept
 - How would you create what you use in c8 ?
- 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...

- Lots of things are up in the air!
- Entirely remote for 2 weeks, then we will see
 - 245 Li Ka Shing after that.
- Lectures will always be recorded.
- Hybrid teaching is an experiment of sorts. I may play with the format.



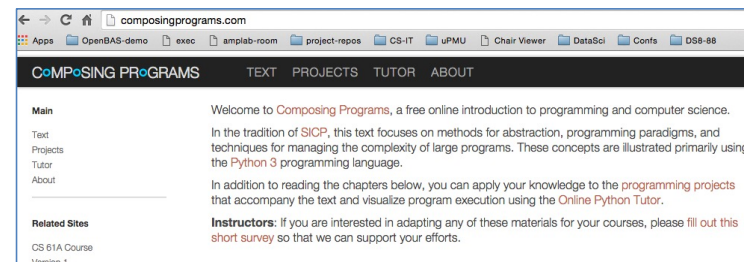
Collaboration

- **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 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:
 - Some lectures will be pre-recorded, class time will be largely demo/Q&A
 - Each lecture has a series of short self-check questions
- 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)
- 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!



Class Format: Assignments

- **Projects: 100 points between 2 projects**
 - Start early! "Checkpoint" 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!**
- **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.

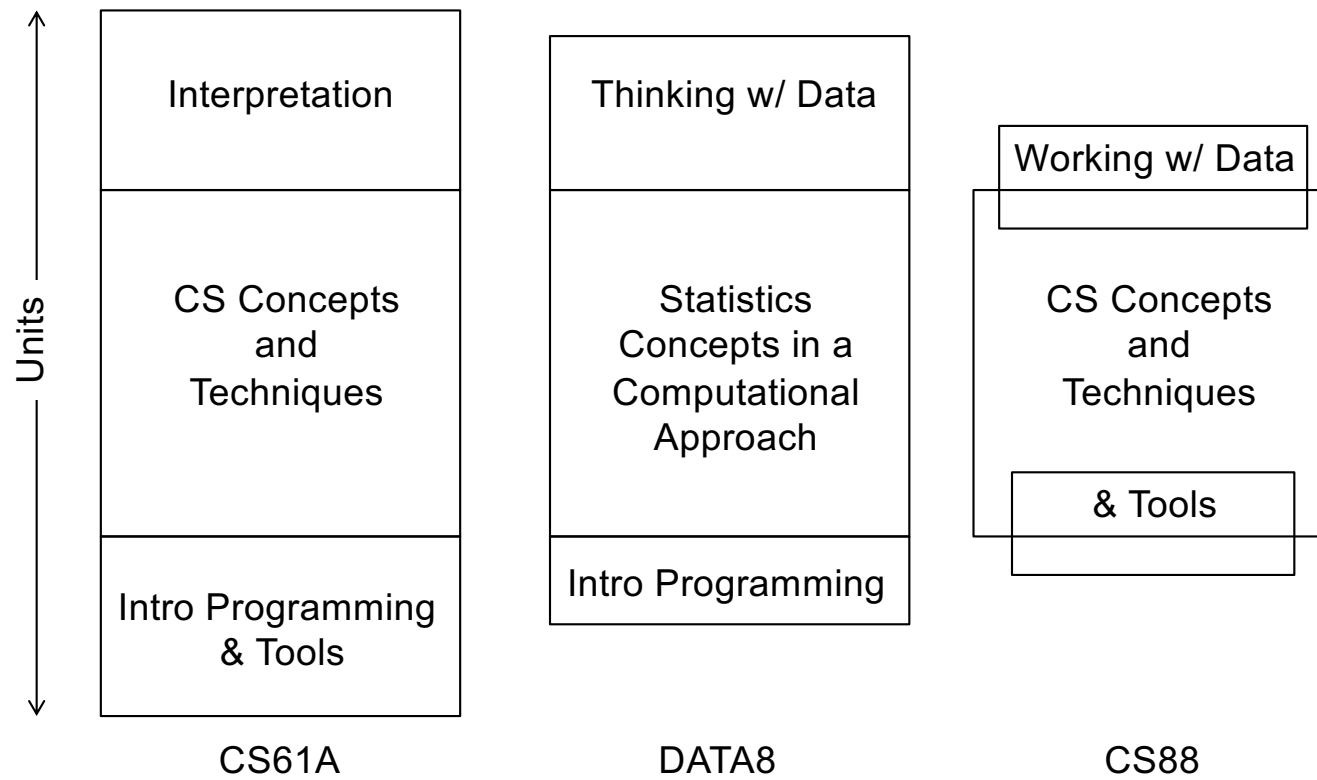


Class Format: Exams

- 1 midterm and 1 final exam, in person or remote
- Midterm 2 hours, mid-March
- Exam will be during the slot assigned by campus.
- We will be proctoring via Zoom. Exact policies coming soon, but, essentially, you'll record yourself at home.
- Unlimited *handwritten* cheat sheets



How does CS88 relate to CS61A ?



Opportunities for students



C8

C8 CS88

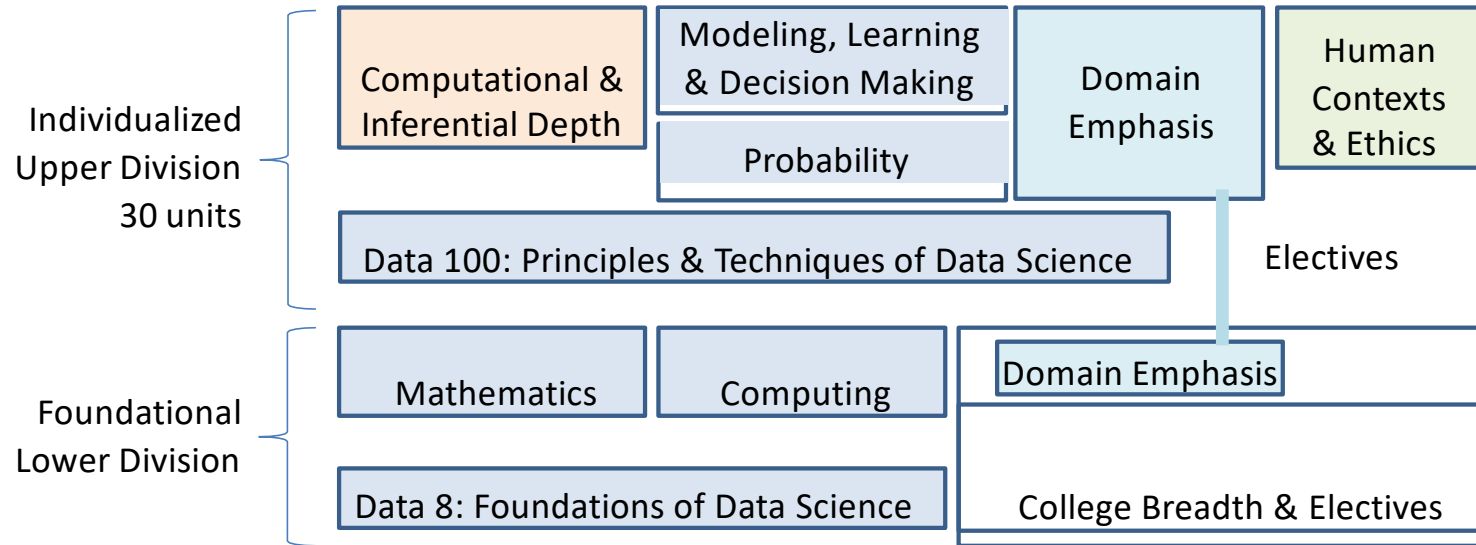
C8 CS88 CS61B

CS minor

C8 CS88 CS47A
CS61A

CS major

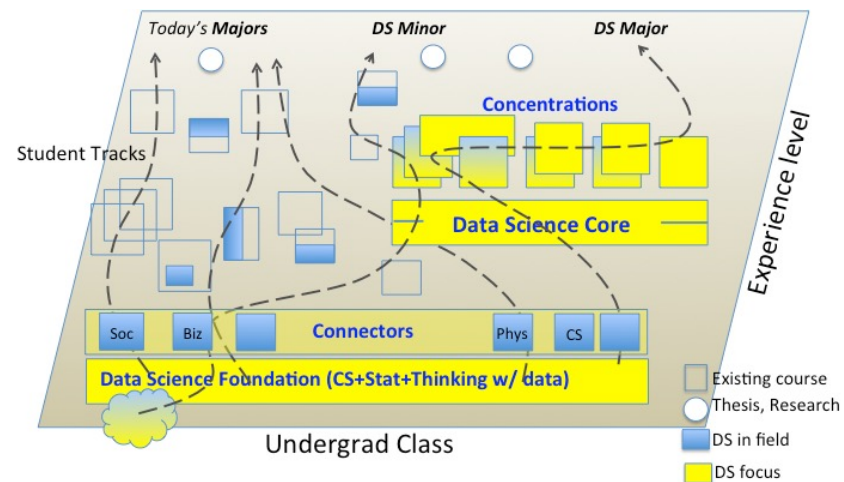
The Data Science Major





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



Poll: Check In



- Are you enrolled in Data 8?
 - A. I took it Spring 2021 or earlier
 - B. I took it Fall 2021 ("last semester")
 - C. I'm taking it right now
 - D. I am trying to enroll in Data 8
 - E. I am not taking Data 8

Poll: Check In



- Where are you right now?
- A. I made it to Berkeley!
- B. I'm somewhere in California
- C. I'm somewhere else in the US
- D. I'm somewhere internationally for the semester
- E. I've made it to Space where there is no COVID.



Ed For Class Discussion: Try it!

The screenshot shows a web browser window displaying an Ed discussion page for CS 88 Fall 2020. The page is titled "Welcome" and is posted by Michael Ball, an instructor, 2 days ago in the General category. The post content includes a greeting, a welcome message for the course, and a list of tasks for the week. A video player at the bottom of the post shows a video titled "Welcome to CS88" with the subtitle "Computational Structures in Data Science Fall 2020". The video player also displays the Berkeley University of California logo.

Welcome

Michael Ball **INSTRUCTOR**
2 days ago in General

UNPIN STAR WATCHING VIEWS 881

Hi everyone,

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.

Video Player: Welcome to CS88
Computational Structures in Data Science
Fall 2020
Berkeley UNIVERSITY OF CALIFORNIA

8/26/2020

26



Where will we work?

- Your laptop
 - Using an editor and a terminal
- cs88.org
- datahub.berkeley.edu
 - Not as often, but an option
- Ed Discssion: us.edstem.org
 - Can write and run (!!) python in you own posts!



Pro-student Grading Policies

- EPA

- Rewards good behavior

- Effort

- » E.g., Office hours, doing every single lab, hw, reading Ed posts

- Participation

- » E.g., Raising hand in lec or discussion, asking questions

- Altruism

- » E.g., helping other students in lab, answering questions on Ed



Your Tasks

- Lecture 1 Quiz On Gradescope
- Watch Ed for info about section signup.

Welcome, and Good luck!



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?

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

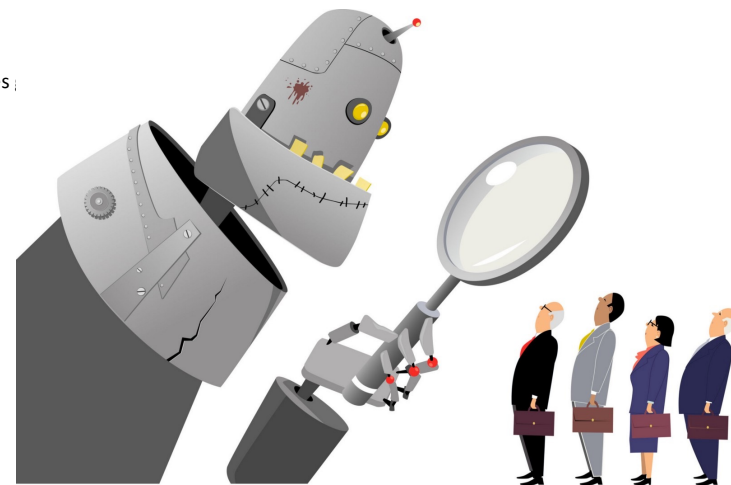
Below the abstraction line

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. Dependencies

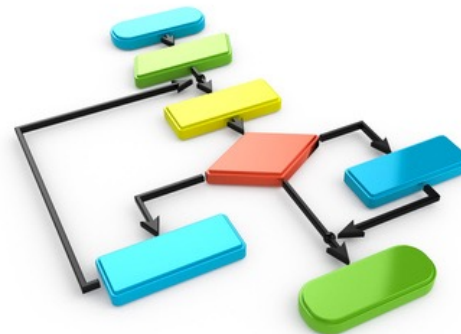




Algorithm

- 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.





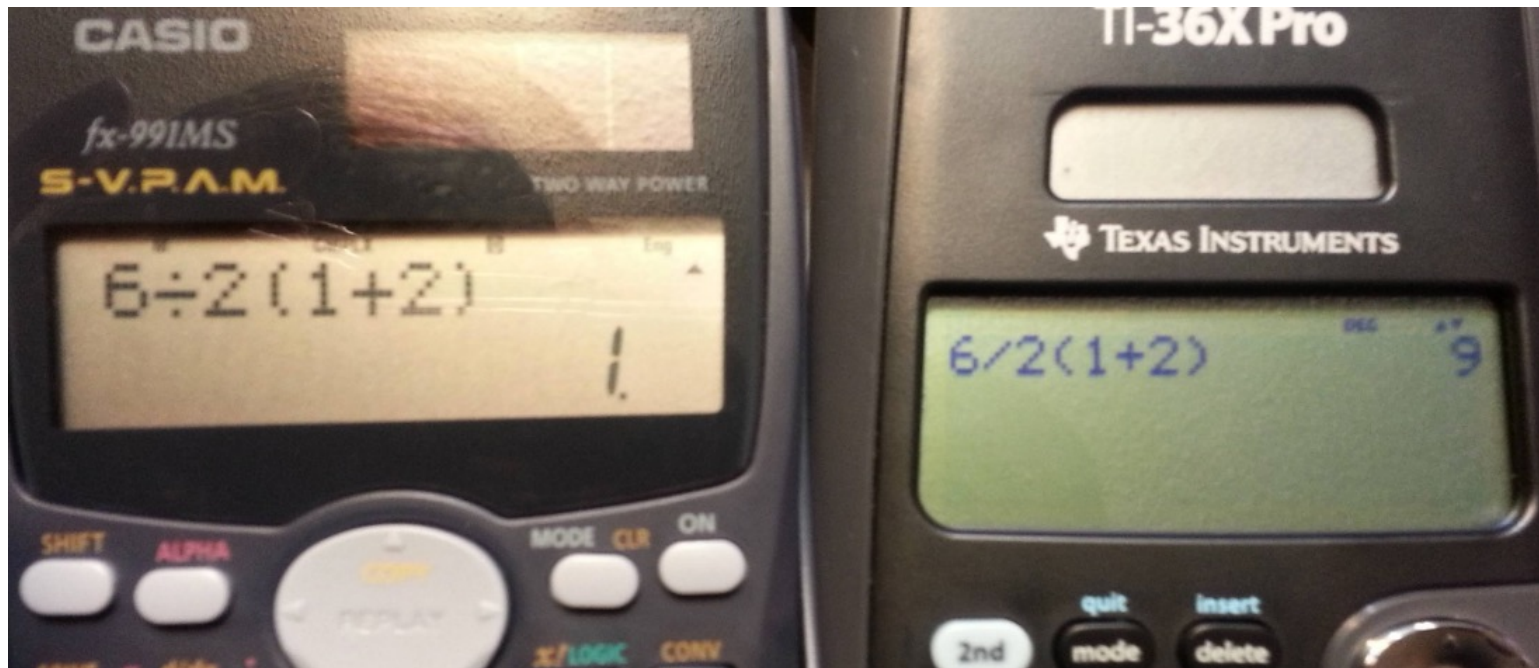
Algorithm: Properties

- 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.

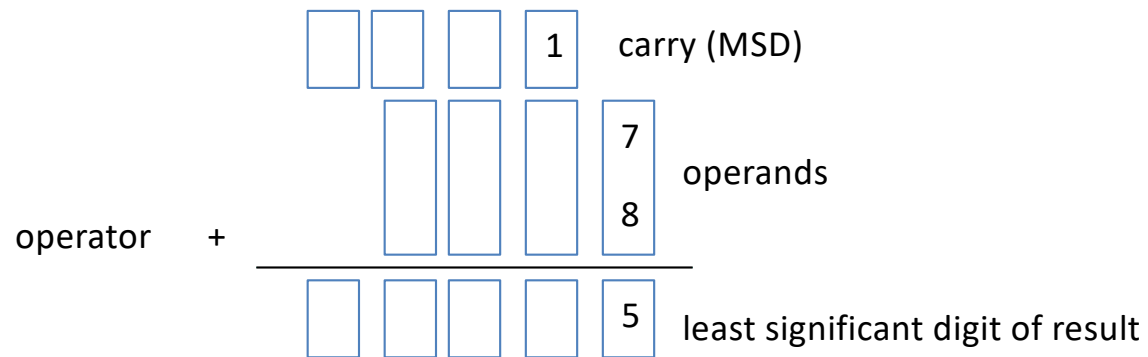
$$6 \div 2(1+2) = ?$$

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)

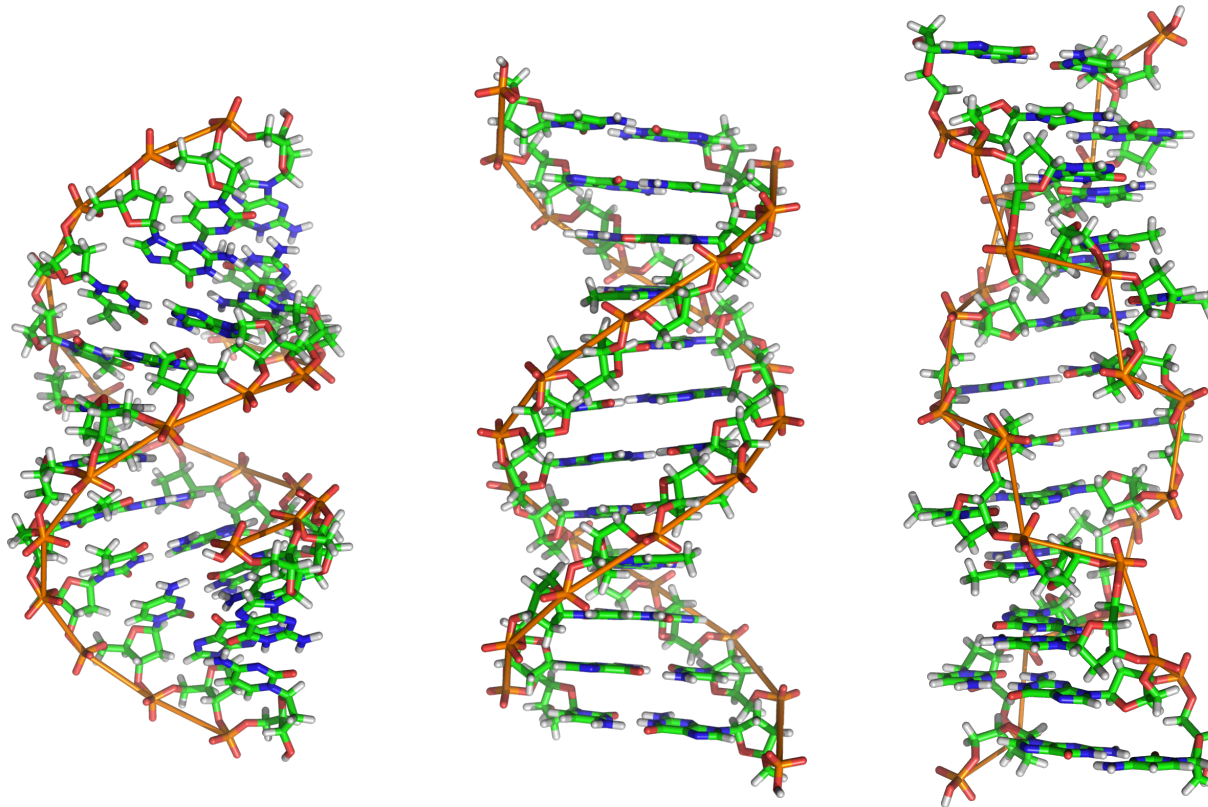
Data

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 10100220 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,
```

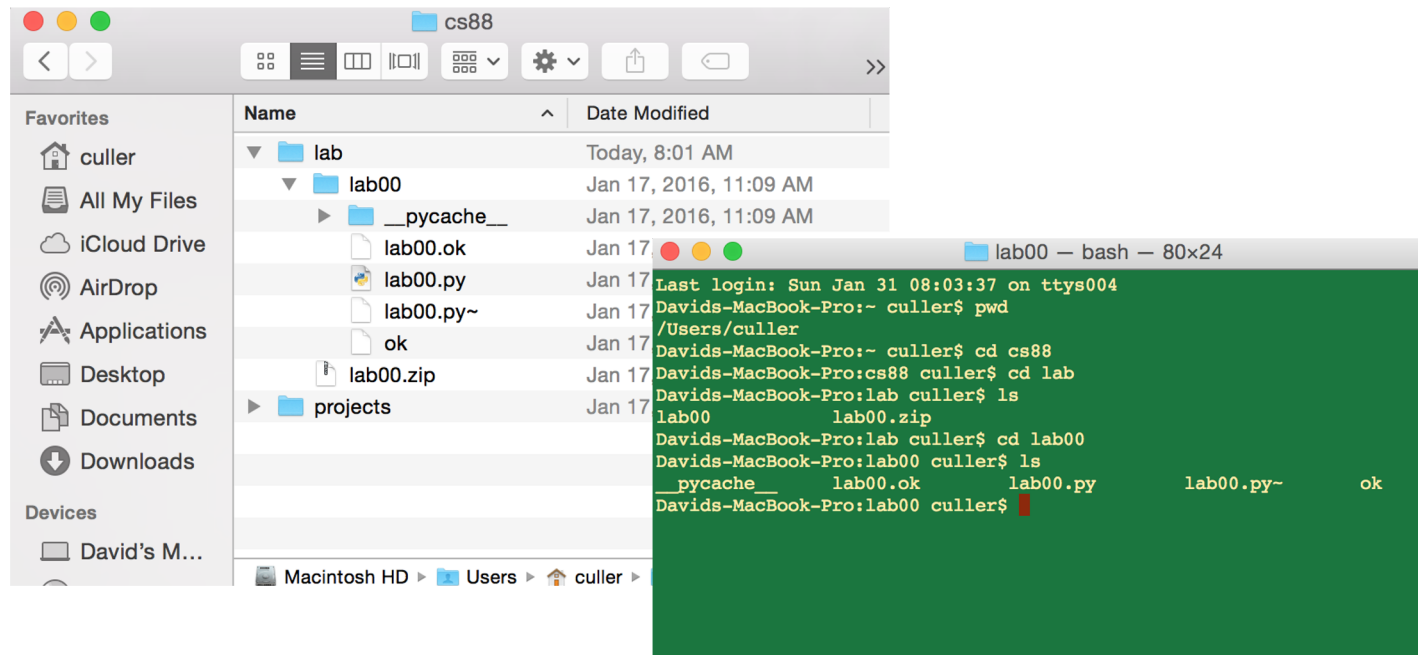
Machine-executable
instructions (byte code)



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