

UC Berkeley EECS Lecturer Michael Ball

# Computational Structures in Data Science



# Lecture 1: Welcome to DATA C88C / CS88!



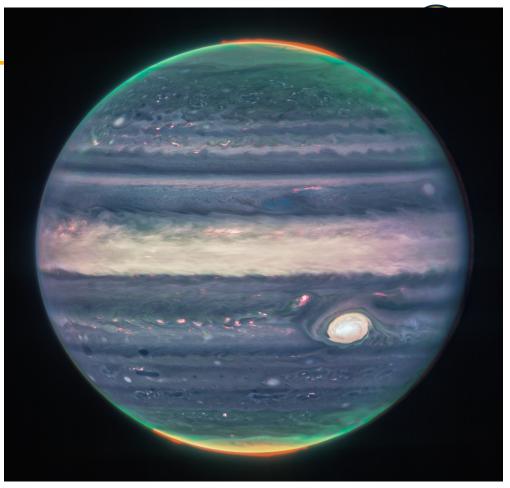
#### In The News

# Webb's Jupiter Images Showcase Auroras, Hazes

Webb NIRCam composite image of Jupiter from three filters – F360M (red), F212N (yellow-green), and F150W2 (cyan) – and alignment due to the planet's rotation. Credit: NASA, ESA, CSA, Jupiter ERS Team; image processing by Judy Schmidt. With giant storms, powerful winds, auroras, and extreme temperature and pressure conditions, Jupiter has a lot going on.

Author Alise Fisher
Posted on August 22, 2022
Categories James Webb Space Telescope
NASA:

https://blogs.nasa.gov/webb/2022/08/22/webbs-jupiter-images-showcase-auroras-hazes/



### Goals today



- Introduce you to
  - the field
  - the course
  - the team
- Answer your questions
- •Big Ideas:
  - Abstraction
  - Data Type





#### CS88 Team – Instructor: Michael



- Michael Ball
  - <u>-\_ball@Berkeley.edu</u> You're best off by using Ed! ☺
  - 784 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



### About Me...A little more unscripted



- I graduated from UC Berkeley in 2015!
- I enjoy Cal Football, but I'm not the biggest sports fan. ©
- I really enjoy photography, videography, and digital media.
- Berkeley is a special place. Be studious, but go explore and find yourself.
- I probably cannot see you raise your hand.

### CS88 Team



### **Head Teaching Assistants**



#### Anjali Gurajapu [she/her]

Office Hours: Wednesday 4pm - 5pm

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!:)



#### Jessica Lin [she/her]

Office Hours: TBD

#### linjessica@berkeley.edu

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

### **Teaching Assistants**



#### Amit Sant [he/him]

Office Hours: Tuesday 11am - 12pm

#### amitsant2000@berkeley.edu

Hello, my name is Amit, and I am a fourth 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: Wednesday 4pm - 5pm

#### hetal.shah@berkeley.edu

Hello! I am a senior 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!



#### Karim Kaylani [he/him]

Office Hours: Tuesday 10am - 11am

#### 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 a tutor this semester and meet you all, never hesitate to reach out to me about anything anytime! :)



#### Lukas Chang [he/him]

Office Hours: Monday 10am - 11am

#### lukasc@berkeley.edu

Hi everyone I'm Lukas, a 4th 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!



#### Sebastian Zhao [he/him]

Office Hours: Wednesday 2pm - 3pm

#### sebbyzhao@berkeley.edu

Hiya! I'm a CS and DS double major from Erie Pennsylvania (underneath the lake). I'm a huge fan of pastries and baking them, digital art (impressionism is my favorite movement, I like Monet a lot), and bedroom pop. Feel free to talk to me about anything, I am a huge believer in conversations about random stuff!



#### Tommy Joseph [he/him]

Office Hours: TBD

#### tommy11jo@berkeley.edu

Hi I'm a fourth-year CS major from Southern California and this is my fourth semester TA-ing for CS 88. I enjoy bread, long walks in nature, and pickleball! Feel free to reach out about anything! .)

#### **Tutors**





Christine Zhang [she/her]

Office Hours: Friday 10am - 12pm

cyuzhang@berkeley.edu

Hi! I'm Christine and I'm a 2nd year EECS major. I love playing badminton, cooking, working out, and eating Chipotle. Other things I like are noodles, socks, water, plushies, and pretty much any kind of music. Reach out to me anytime for anything! I'd love to meet you:)



#### Ethan Yoo [he/him]

Office Hours: Thursday 10am - 12pm

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, hiking). I'm at your assistance and am looking forward to the school year!



#### Michelle Chen [she/her]

Office Hours: Monday 11am - 1pm

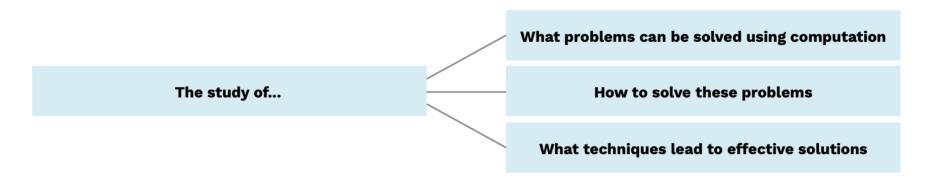
michelle.chenn@berkeley.edu

hi! im michelle, a second year majoring in cs and econ from singapore!! a little about me: i love art, photography, traveling and food so hmu if u have (or want) recs. excited to meet u all!!

& Joanna Yoo ...who just joined!

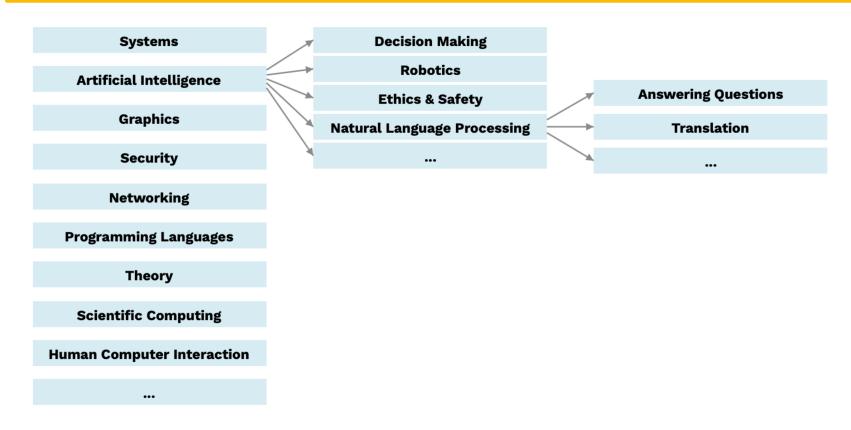
### **Computer Science**





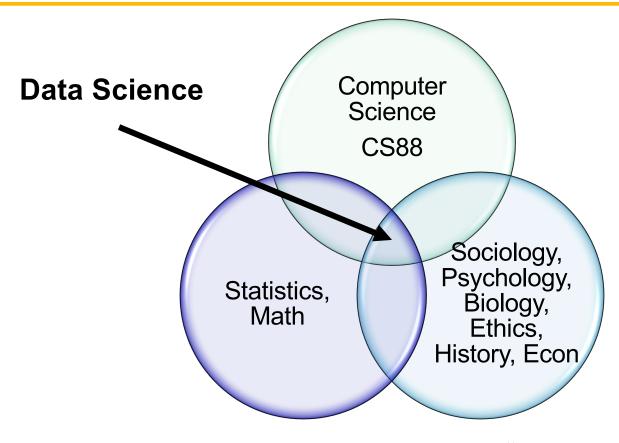
### Computer Science, Some Ideas...Definitely Not Exhaustive!





### Computer Science & Data Science (One View)





### So...COVID...



- It's like the 6<sup>th</sup> "not yet normal" semester.
- Please stay home if you feel sick.
- Attendance is **not** required!
- Lectures are recorded, labs will not be
- We will do our best to figure out online/remote lab times.
- We will have online OH (TBD)

### CS88 – Computational Structures in Data Science



- Deeper understanding of the computing concepts introduced in DATA 8
  - –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 (kind of!) data-centric approach

#### **Course Culture**



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

#### 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. (Projects)
- Collaboration has limits
  - Please don't read/copy/use 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 others direct code.
- We have a very particular set of skills and we will use them.

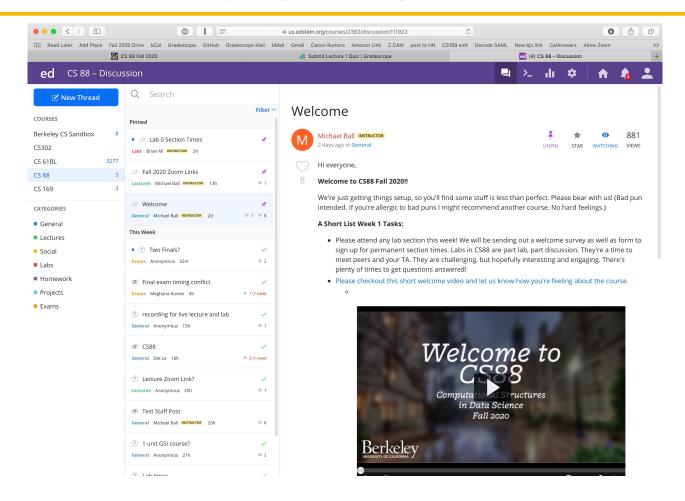
#### Where will we work?



- Your laptop
  - Using an editor and a terminal
- c88c.org (or 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!



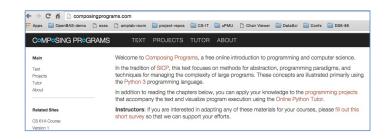
### Ed For Class Discussion: https://go.c88c.org/qa1



#### **Course Structure**



- •2 lectures, 1 lab each week
- •Lecture introduces concepts (quickly!), answers why questions.
- Labs (12) 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
- You get one "drop" of each lab and homework
- Readings: <a href="http://composingprograms.com">http://composingprograms.com</a>
  - Same as cs61a



#### **Class Format**



- Tues / Thurs Lectures:
- Each lecture has a series of short self-check questions
- Labs will try to rely on Tues' lecture
- 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.





- 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!





- Projects: 100 points between 2 projects
  - -Start early! "Checkpoint" assignments
- Slip Days: 9 total
  - -Use up to 3 on any assignment
  - -We apply the 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.

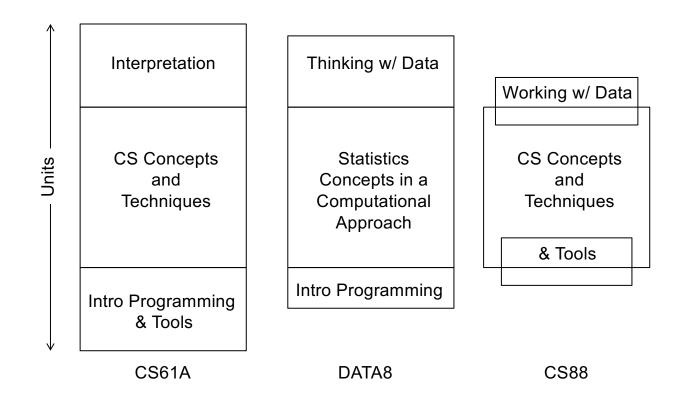
#### Class Format: Exams



- 1 midterm and 1 final exam, in person or remote
- Midterm 2 hours, Oct 19, 7-9PM
- 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 if you're not on campus.
- Unlimited handwritten cheat sheets

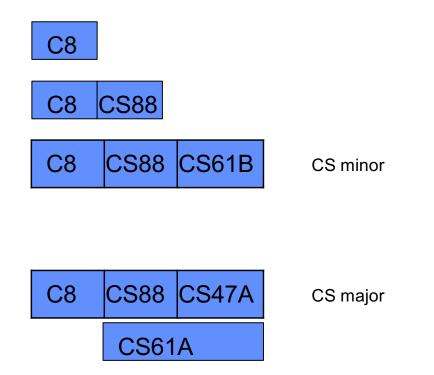
### How does CS88 relate to CS61A?





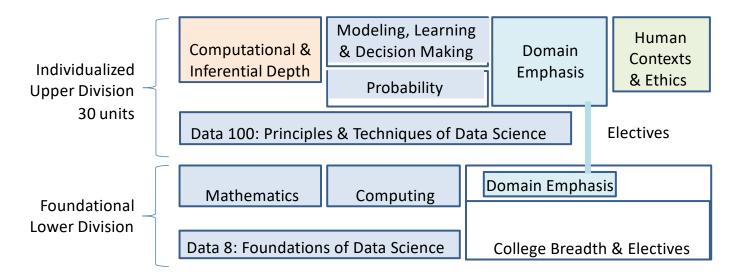
# Opportunities for students





# The Data Science Major

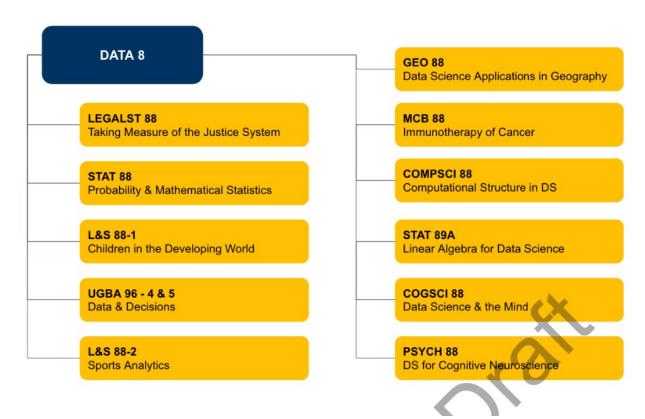




https://data.berkeley.edu/academics/data-scienceundergraduate-studies/data-science-major/declaringmajor

### **Explore The World of Data 8 Connectors!**





### 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
- Data 8 is corequisite!
  - You ideally should have taken, or be enrolled in Data 8
  - If not, you can get by, but you might be missing some context.

### **Your Tasks**



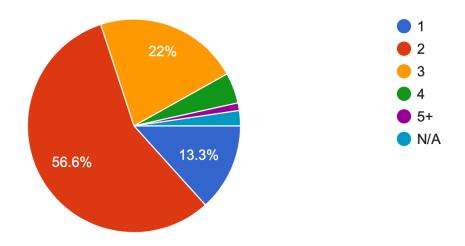
- Lecture 1 Quiz On Gradescope:
  - https://go.c88c.org/1
- Watch Ed for info about section signup
  - Out tomorrow at Noon!
- Fill out the Welcome Survey!
  - https://go.c88c.org/welcome

Welcome, and Good luck!

### Survey Results So Far



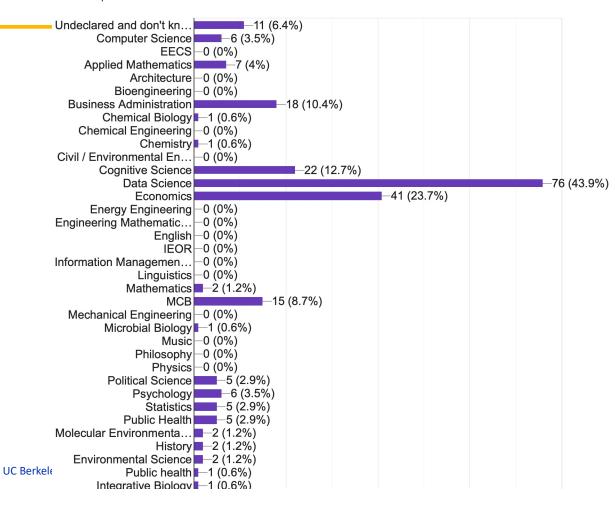
How many years have you attended UC Berkeley, including this one? 173 responses



### Survey Results So Far

#### What is/are your major(s)?

173 responses

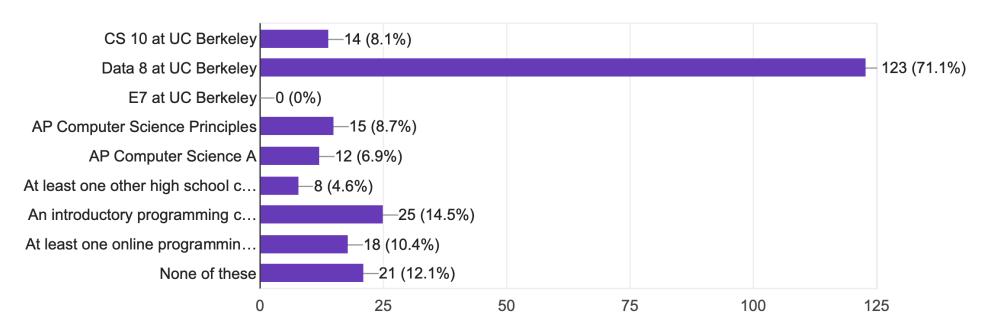


## **Survey Results**



### Which of the courses below have you taken?

173 responses



### **Learning to Program**



- Yes! It can be tricky!
- Most of you have taken Data 8, but that's it!
- The range of experiences are wide.
- It is OK if this is your first time programming!!





# 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."  $\label{eq:constraint}$
- Generalization
  - "The process of formulating general concepts by abstracting common properties of instances"  $\,$
- Technical terms: Compression, Quantization, Clustering, Unsupervized Learning



# **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!





# 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

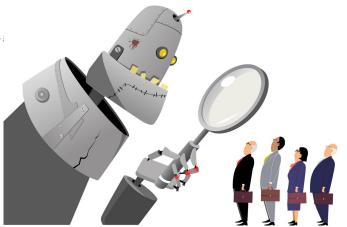
This is where / how / when / by whom it is actually built, which is done according to the interface, specification, or contract.

## **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: Al and hiring decisions.

• Abstraction makes things special and that creates dependencies. Dependencies

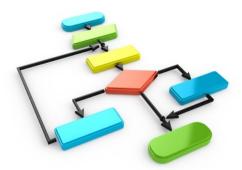


## Algorithm



- •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

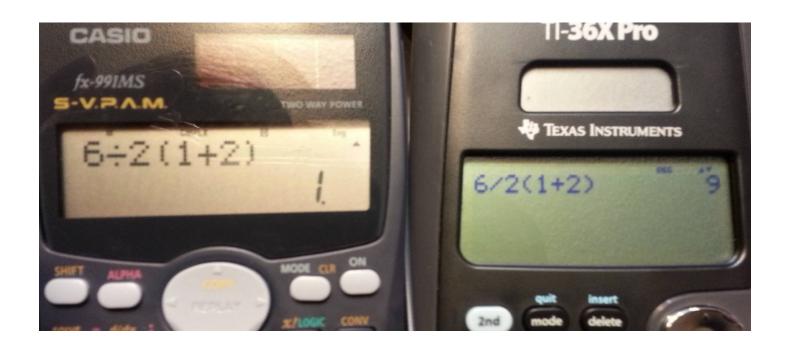


- •An algorithm is a description that can be expressed within a finite amount of space and time.
- Executing the algorithm may take infinite space and/or time, e.g. ``calculate all prime numbers".
- •In CS and math, we prefer to use well-defined formal languages for defining an algorithm.

$$6 \div 2(1+2) = ?$$
1 or 9

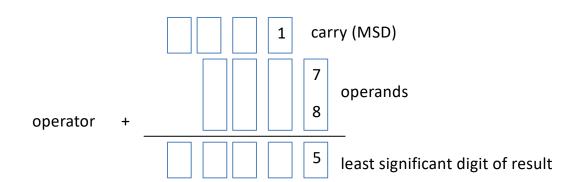
# Algorithm: Well-Definition





# 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**

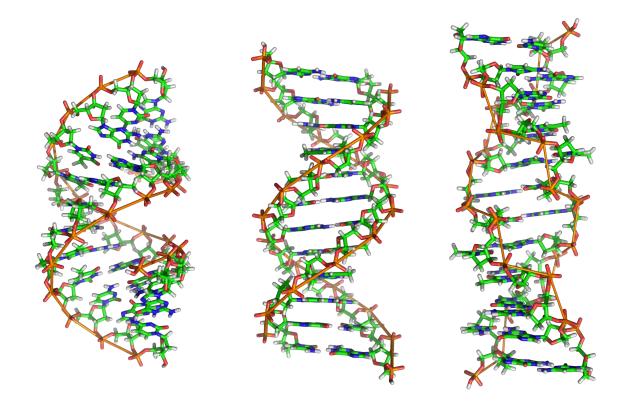
**Observations** 

### Information

Reduction of uncertainty in a model (measured in bits)

# Data or Code?





UC Berkeley | Computer Science 88 | Michael Ball | http://cs88.org

#### Data or Code?



#### Data or Code?



# Here is some information! Integer

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,
```

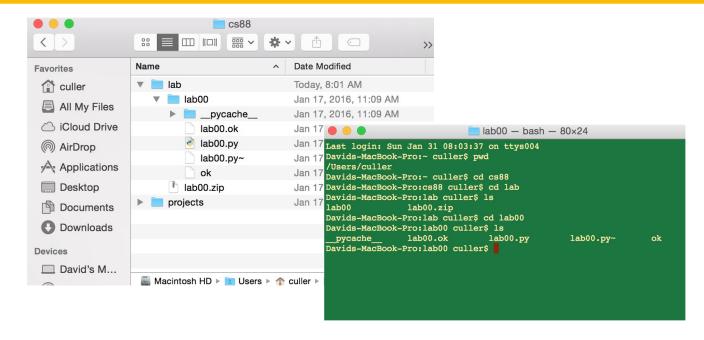
# Machine-executable instructions (byte code)

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