



The Beauty and Joy of Computing



UC Berkeley
Teaching Professor
Dan Garcia

Welcome & Abstraction

BJC... We know you'll love it!

Watch the student testimonials about the course, what it means to them, and how it has changed their lives. Inspiring!

youtu.be/oG1OItm4JyU?si=bzdwD9tL5OFbxRfi



Learning to Think Like a Computer

April 4, 2017



In “The Beauty and Joy of Computing,” the course he helped conceive for nonmajors at the University of California, Berkeley, Daniel Garcia explains an all-important concept in computer science — abstraction — in terms of milkshakes.

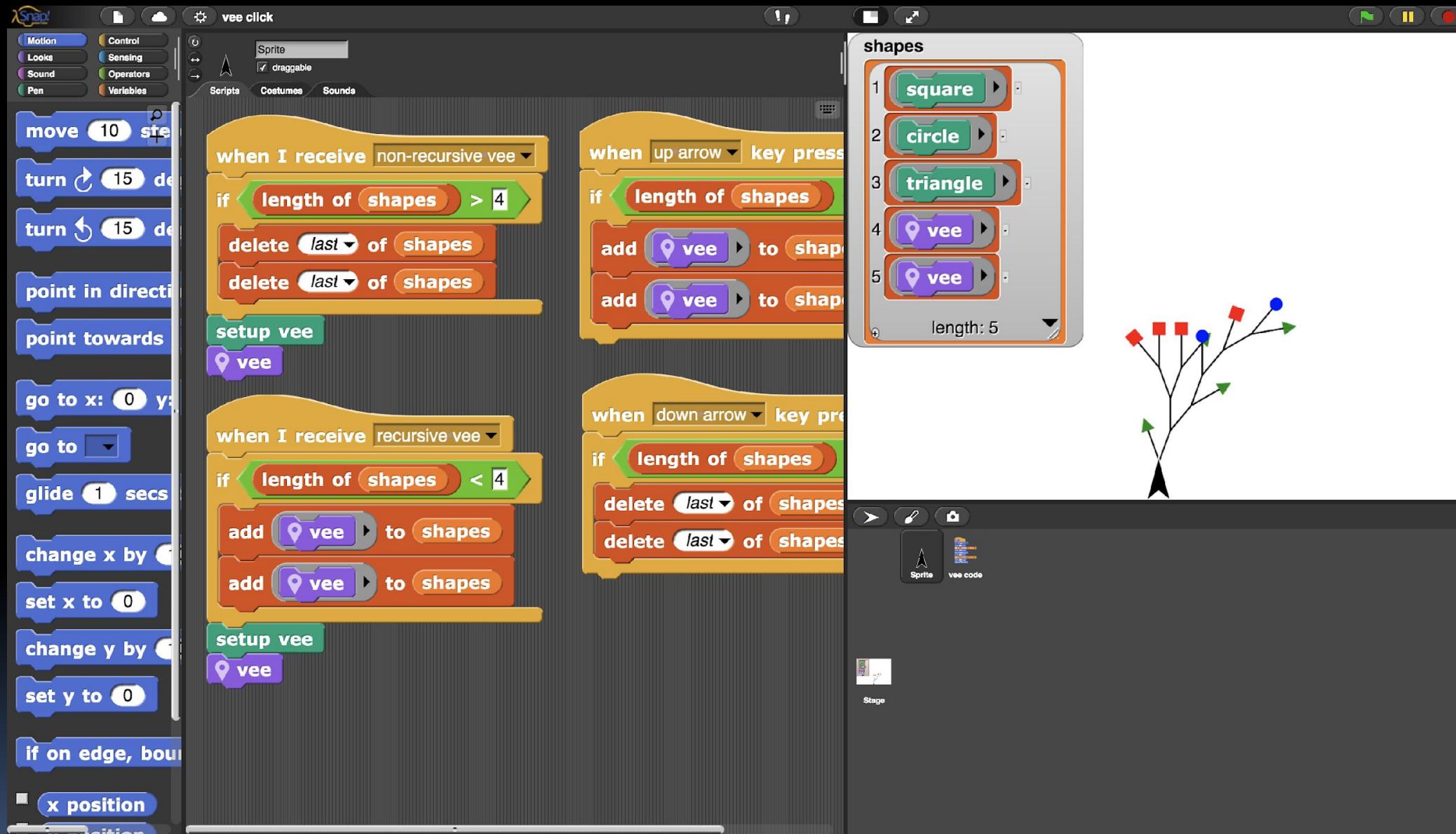
BJC in one slide

- Big Ideas of Programming
 - Abstraction
 - Algorithms
 - Recursion
 - Functions-as-data
- Beauty and Joy
 - Pair Programming
 - Five Projects
 - Final project your own choice!!
 - "Paper" of your own choice!!
- Big Ideas of Computing
 - HowStuffWorks
 - Computers (binary numbers)
 - Research Summaries
 - AI
 - HCI
 - Social Implications of Computing (2-3)
 - Saving the World with Computing
 - Set you up for follow-up 61A
 - We also teach Python!





Incredibly easy-to-learn coding in Snap!



Worldwide community of *Snap!* users

snap.berkeley.edu

The screenshot shows the homepage of snap.berkeley.edu. At the top, there's a navigation bar with links for "Run Snap!", "Explore", "Forum", a search bar, and a user account dropdown for "dan Garcia". Below the navigation is a large, vibrant illustration of various flowers in purple, green, blue, and orange. In the center of the illustration, the "Snap!" logo is displayed. To the left of the logo, a Scratch script is shown:

```
when mouse down?
  set [color v] to [pick random -10 to 10]
  go to [mouse-pointer v]
repeat (7)
  stamp
  turn (360 / 7) degrees
next costume
```

Below the illustration, the text "low floor" is written in a stylized purple font. At the bottom of the page, a welcome message reads "Welcome, dan Garcia!". A descriptive paragraph follows: "Snap! is a broadly inviting programming language for kids and adults that's also a platform for serious study of computer science." At the very bottom, there are five buttons: "Run Snap! Now", "My Projects", "My Public Page", "Example Projects", and "Reference Manual".



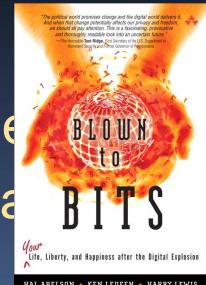
Format & Textbooks (OH starts week)

2) Format (8 hrs/wk * 15 wks) + ~4/hr/wk of proj/hw/reading

| CS10 schedule | Monday | Tuesday | Wednesday | Thursday | Friday | |
|---------------|-------------|-------------|---------------|-------------|-------------|-------|
| 9:00 | OH@411Soda | OH@411Soda | OH@411Soda | OH@411Soda | OH@411Soda | 9:00 |
| 10:00 | | | | | | 10:00 |
| 11:00 | | | | | | 11:00 |
| 12:00 | | | | | | 12:00 |
| 1:00 | Lec@160AAPB | | Lec@160AAPB | | Lec@160AAPB | 1:00 |
| 2:00 | | | | | | 2:00 |
| 3:00 | | Lab@430Soda | DanOH@777Soda | Lab@430Soda | | 3:00 |
| 4:00 | OH@411Soda | | OH@411Soda | OH@411Soda | | 4:00 |
| 5:00 | | | | | OH@411Soda | 5:00 |
| 6:00 | | OH@411Soda | | Dis@60Evans | | 6:00 |
| 7:00 | | | | OH@411Soda | | 7:00 |

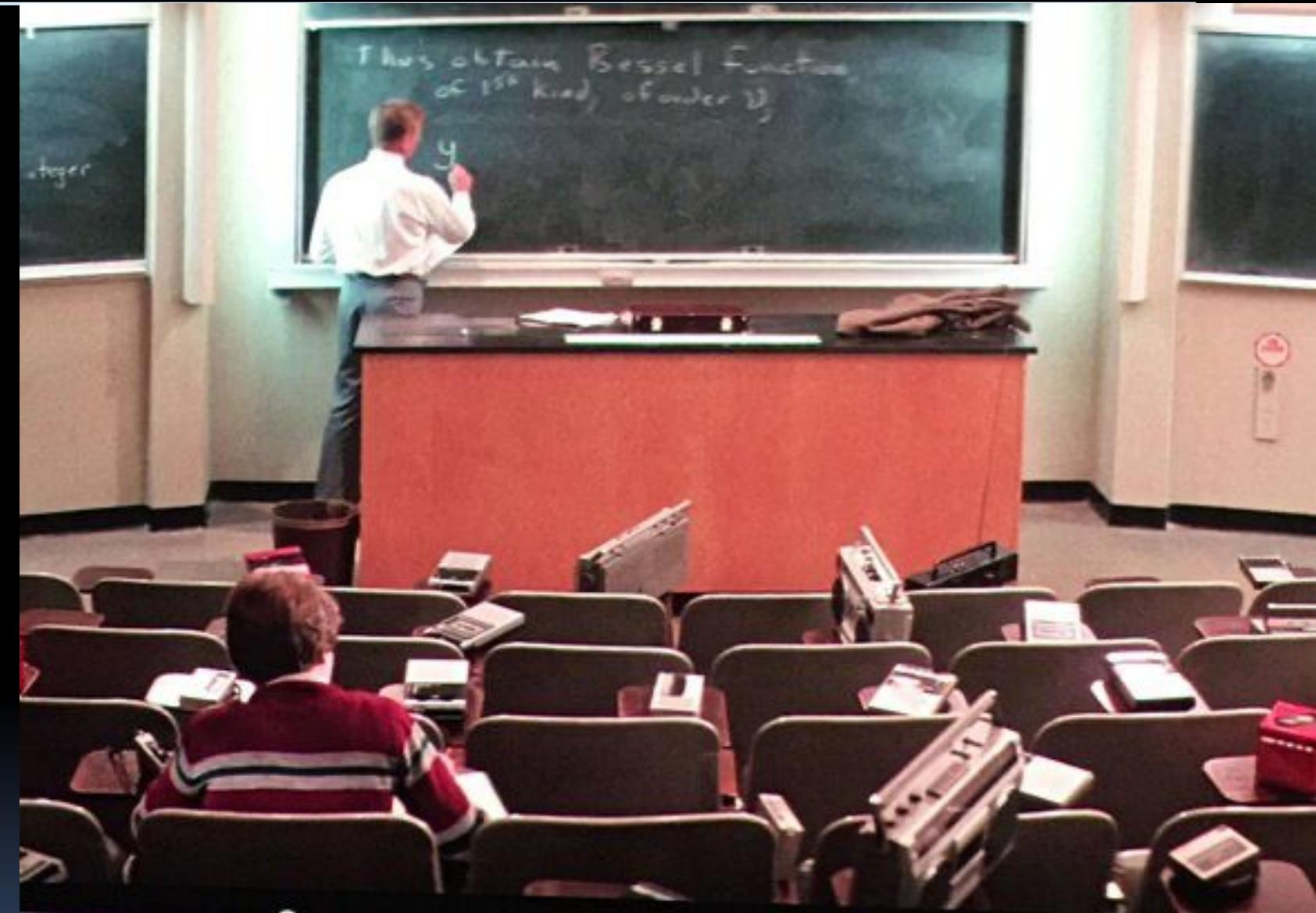
- Selected Reading

- Taken from great book (“Blown to Bits”) + articles + video
- Current events EVERY LECTURE (e.g., Algorithmic Bias)



Expectations ...

- Lecture attendance **required**
 - We'll use iclicker for this
- Discussion attendance **required**
 - ...or submit worksheet to Gradescope
- **If you need more time for labs or projects** fill extension form
 - If you receive an extension, it is NOT considered late; "late" means you missed deadline and did not ask for an extension
- Mastery-based learning!



Scene from “Real Genius” ... students “mailing in” lecture attendance



Let's check enrollments...

CURRENT ENROLLMENT

Total Open Seats: 16

Enrolled: 54

Capacity: 70

Waitlisted: 0

Waitlist Max: 200

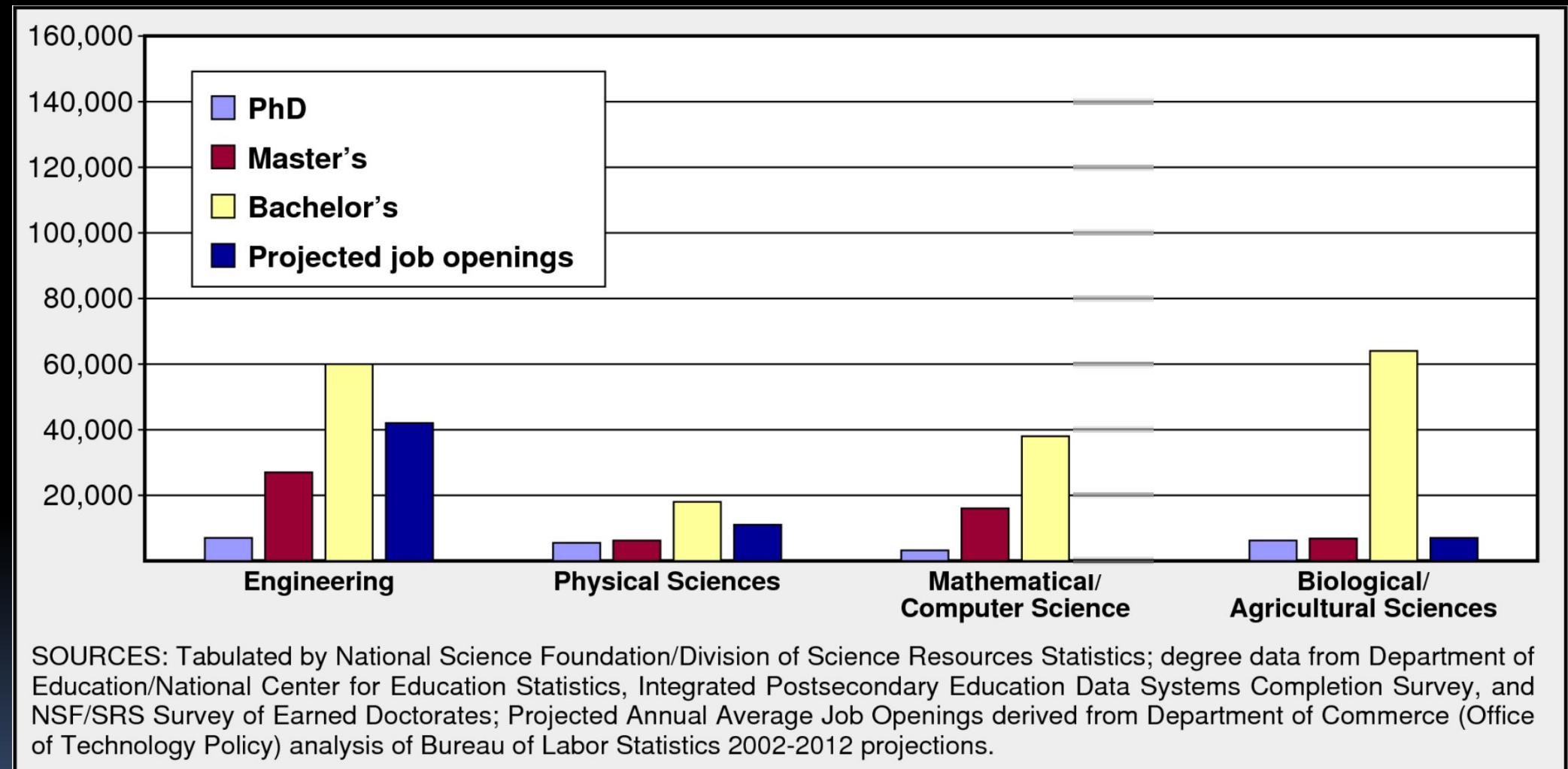


Other Introductory Courses

- Foundations of Data Science, Data Science 8 (CS C8 in the catalog)
 - Foundations of data science from three perspectives: inferential thinking, computational thinking, and real-world relevance. Given data arising from some real-world phenomenon, how does one analyze that data so as to understand that phenomenon? It teaches critical concepts and skills in computer programming and statistical inference, in conjunction with hands-on analysis of real-world datasets, including economic data, document collections, geographical data and social networks. It delves into social and legal issues surrounding data analysis, including issues of privacy and data ownership. (You learn how to use Python as a tool, not how to program)
- Structure and Interpretation of Computer Programs, CS 61A
 - Introduction to programming and computer science. This course exposes students to techniques of abstraction at several levels: (a) within a programming language, using higher-order functions, manifest types, data-directed programming, and message-passing; (b) between programming languages, using functional and rule-based languages as examples. It also relates these techniques to the practical problems of implementation of languages and algorithms on a von Neumann machine. There are several significant programming projects. (You learn how to program

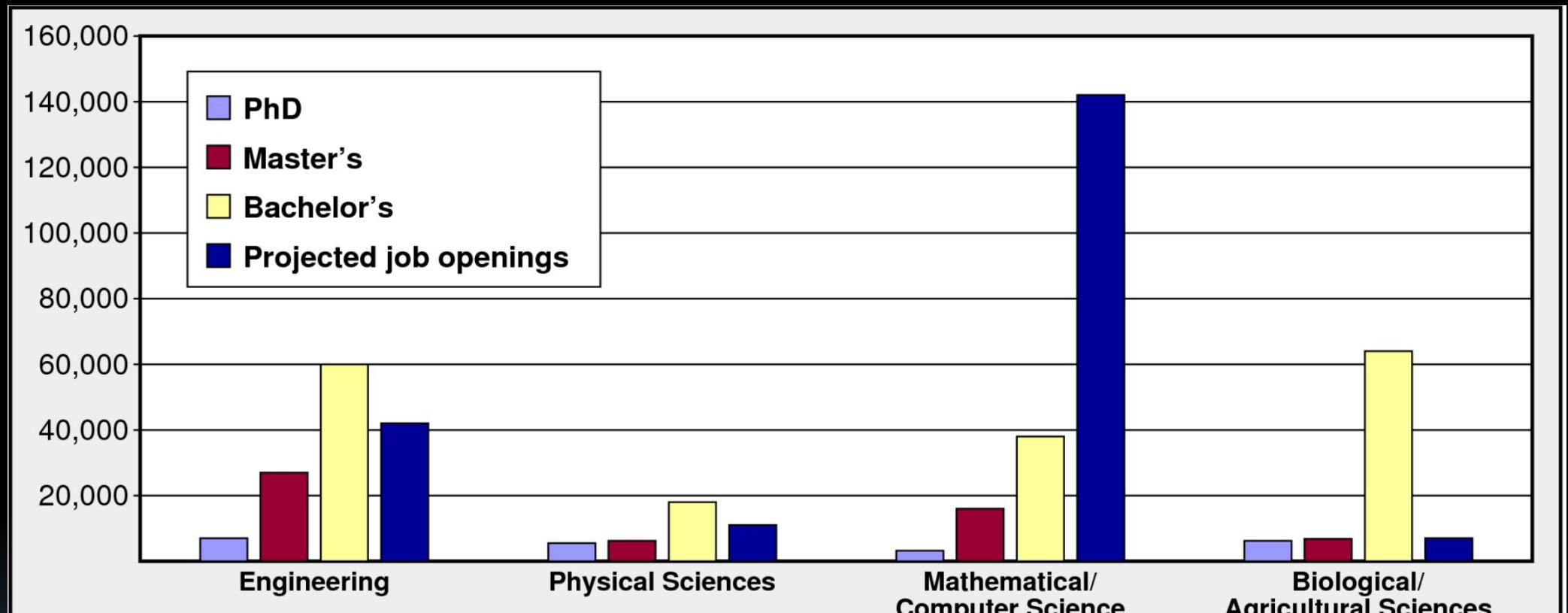
How does a 6-figure/year job sound?

Annual degrees and jobs in broad engineering fields



How does a 6-figure/year job sound?

Annual degrees and jobs in broad engineering fields



SOURCES: Tabulated by National Science Foundation/Division of Science Resources Statistics; degree data from Department of Education/National Center for Education Statistics, Integrated Postsecondary Education Data Systems Completion Survey, and NSF/SRS Survey of Earned Doctorates; Projected Annual Average Job Openings derived from Department of Commerce (Office of Technology Policy) analysis of Bureau of Labor Statistics 2002-2012 projections.



Ed for {ask,answer}ing questions

ed CS10 Fa22 – Ed Discussion

New Thread

COURSES +

- CS10 Fa22
- CS10 Su22 235
- CS61C FA22 4

CATEGORIES

- General
- Lecture
- Discussion
- Lab
- Projects
- Exams

Search Filter ▾

[Important] Section Signups #1

Vedansh Malhotra STAFF 3 days ago in General 13 5

Note: The following post contains important information about lab and discussion sections this semester. Please read the post in its entirety!

5

Welcome CS-10 students, staff is so excited to get this semester started! Woohoo 😊

While there will be an official *Welcome and Logistics* announcement in the near future, we wanted to release section sign ups early to give you enough time to make adjustments to your schedule, and modify section sizes if needed. Below is a short summary of lab and discussion sections. To reiterate: **this is not an official logistics post, and more detailed information regarding lab, discussion, and other class components will be announced soon.**

LAB:

In CS10, you must **enroll in one lab section**. Each lab section will meet twice a week (either Monday/Wednesday or Tuesday/Thursday) for two hours each. Lab section signups will be first come, first served. All lab sections are held in Sutardja Dai 200, which is not accessible from within Sutardja Dai hall. Instead, Sutardja Dai 200 access is right next to [Yali's Cafe](#), in the direction of the Campanile.

Lab sections are designed to reinforce concepts taught in lecture and give you more programming practice before attempting other class assignments. It is a place to experiment, learn, ask questions, and have fun! There will be lab assignments that will be posted on the website before lab sections.

Waitlisted Students: Please **do not** fill out the form linked below. You may still attend a section in the meanwhile, but the signups should only be used after you are officially enrolled in the class.

Lab Section Signup Link: <https://www.wejoinin.com/sheets/uamez>
Please fill this out by Tuesday, 08/23 at 2359hrs (11:59PM)

DISCUSSION:

Students should also **enroll in one discussion section**. These are meant to solidify your skills on the material covered in lecture and prepare you for the course's exams, by working on a worksheet of questions together as a group.

Discussions are 50 minutes long after Berkeley time, and are held once a week on either Mondays or Fridays. All discussion sections except for Friday's are located in Soda Hall, while Friday's discussion



"A's for All (as time and interest allow)"

- The “A” bar from earlier semesters doesn’t change
- Any points you lose on ANY grading element can be earned back **with no penalty** if you keep on working
- You can work to get the grade you want!
 - Rather than getting one shot on projects, you can keep working with the (auto)grader until you get it perfect
 - You’ll have multiple-chances on exams
 - If you don’t have the grade you want by the end of the semester, we’ll give you an incomplete
 - You can also say “eh, I’m done” and get whatever grade you earned at the end of the semester

Amazing Staff cs10.org/sp25/staff

UCS2

S



Stacey
(Head
TA)



Victoria
(Head
TA)

UCS1

S



Aanvi



Dream



Harry



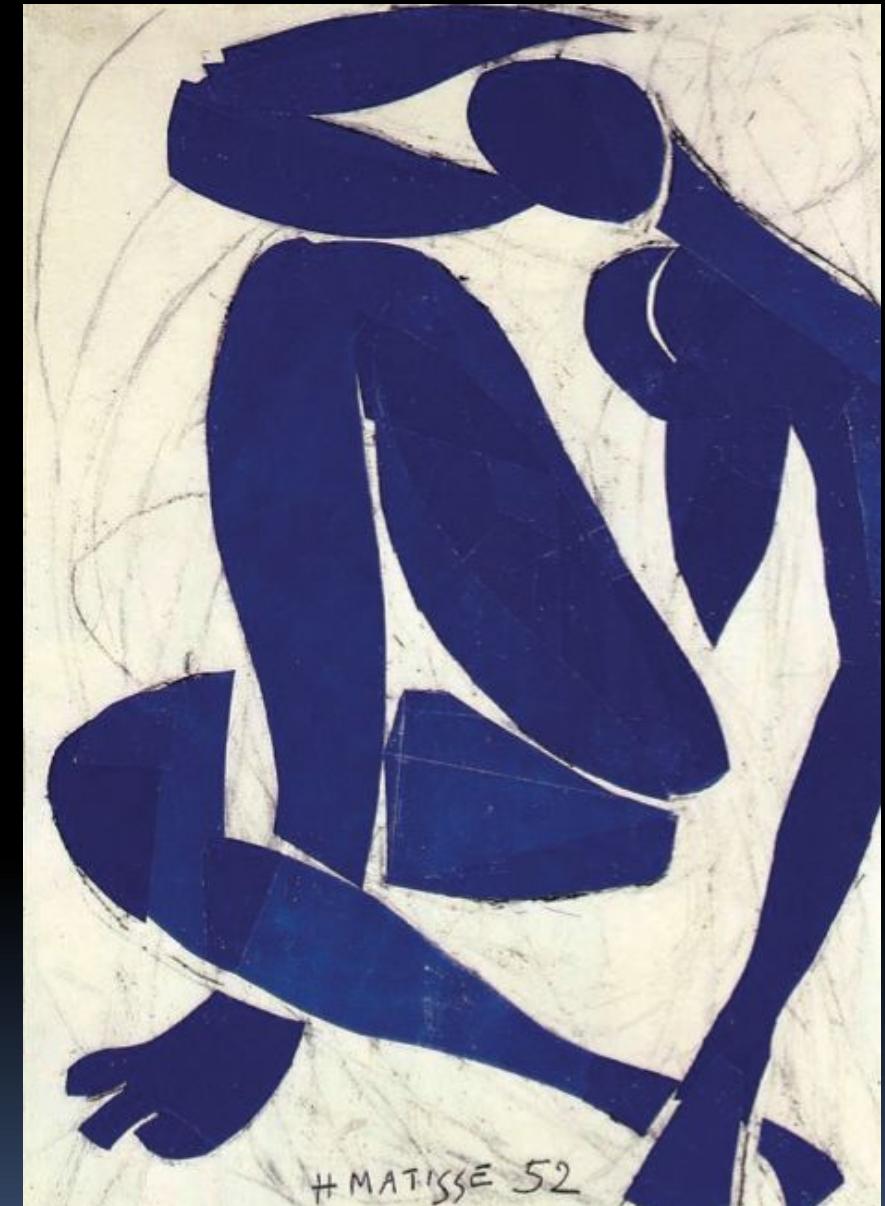
Marius



Raka

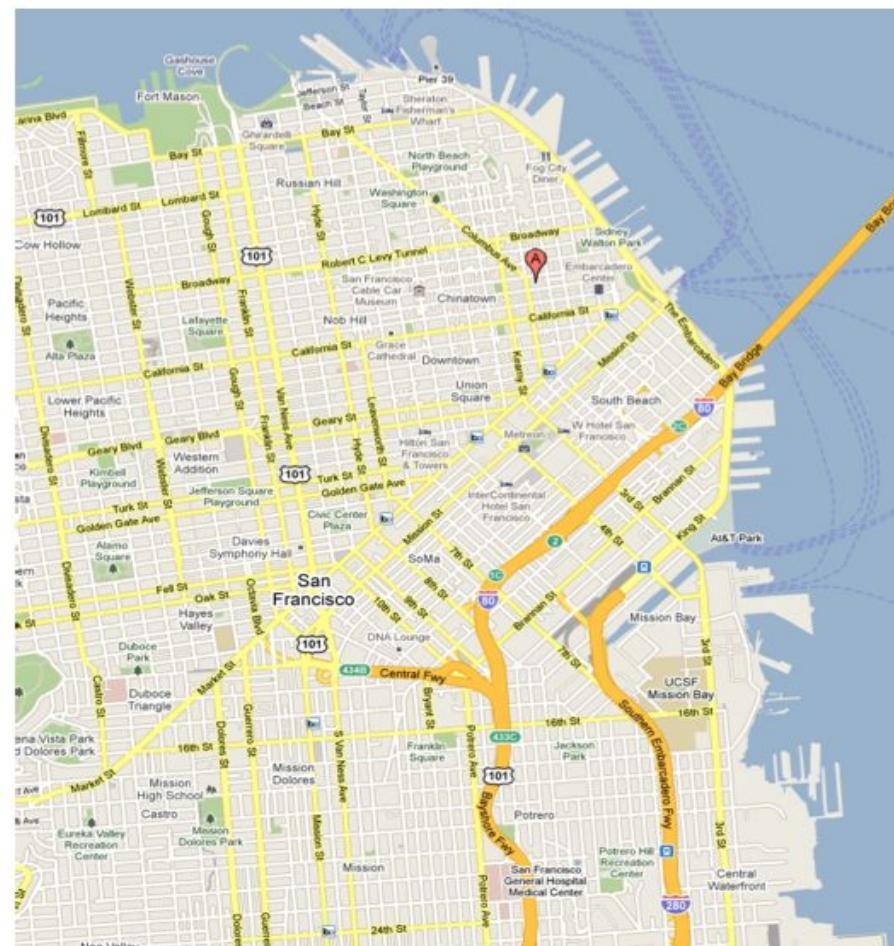
Abstraction

- Detail removal
 - “The act or process 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”

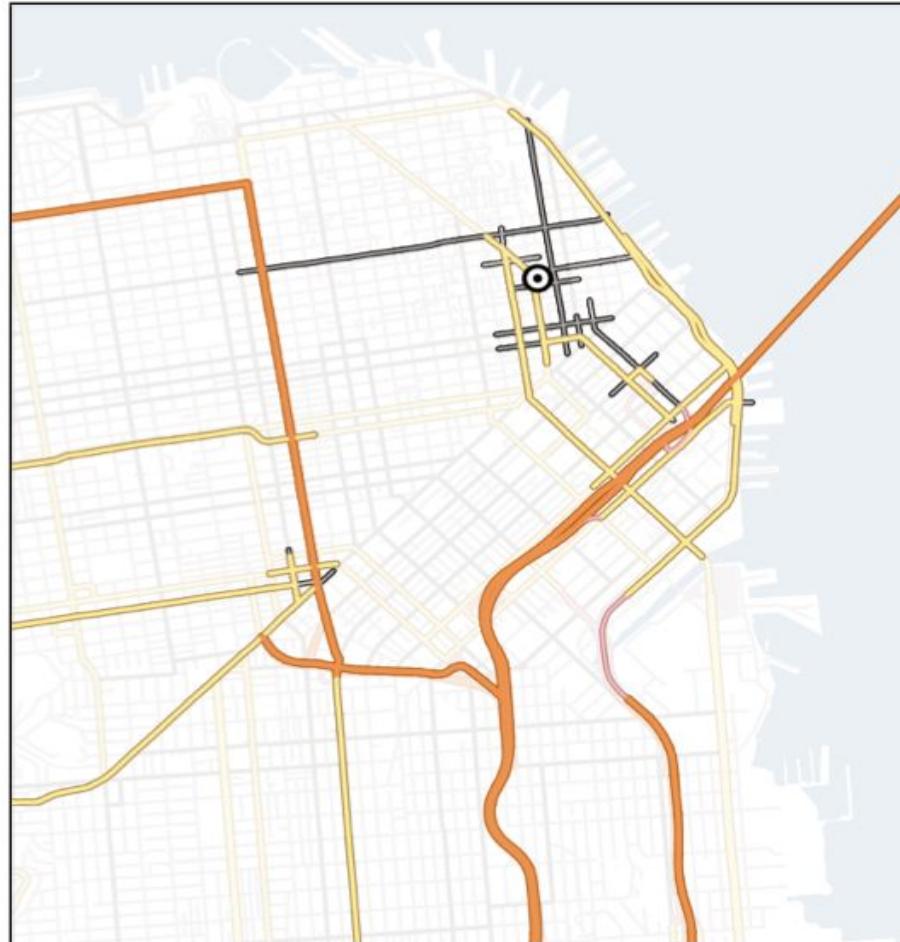


Henri Matisse “Naked Blue IV”

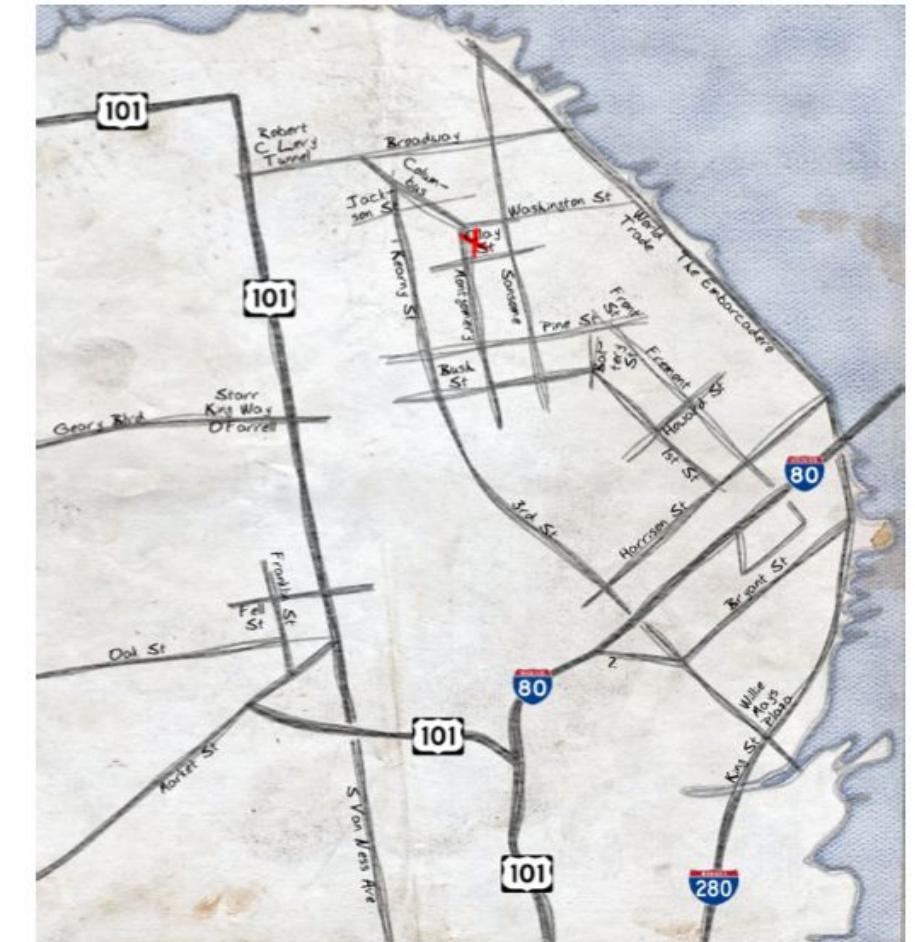
Detail Removal



General Purpose Online Map



Selected Roads

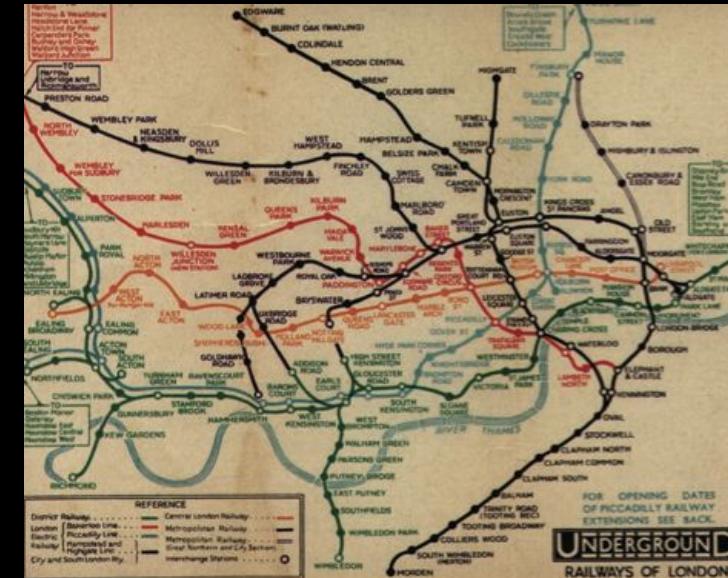


Our Result

Automatic Generation of Detail Maps
Prof. Maneesh Agrawala, among others

Detail Removal (in BJC)

- You'll want to write a project to **simulate a real-world situation**, or play a game, or ...
- 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.

Generalization Example

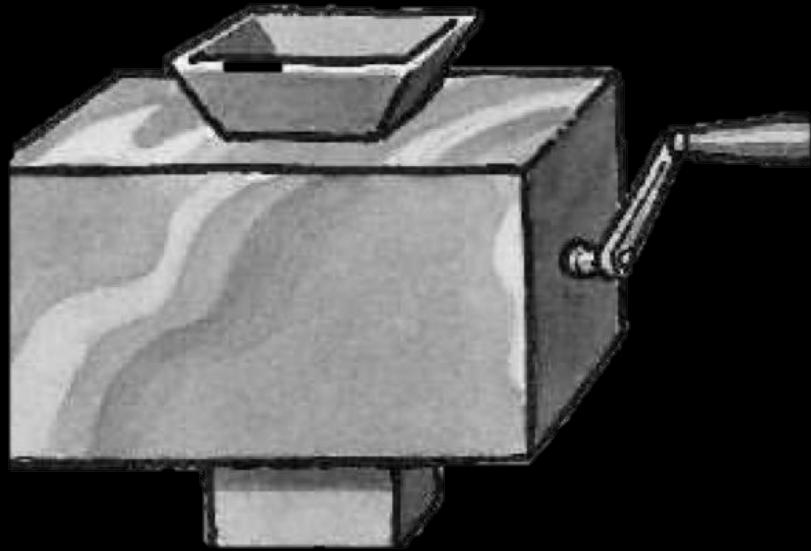
- You have a farm with many animal kinds.
- Different food for each
- You have directions that say
 - To feed dog, put dog food in dog dish
 - To feed chicken, put chicken food in chicken dish
 - To feed rabbit, put rabbit food in rabbit dish
 - Etc...
- How could you do better?
 - To feed <animal>, put <animal> food in <animal> dish



Generalization (in BJC)

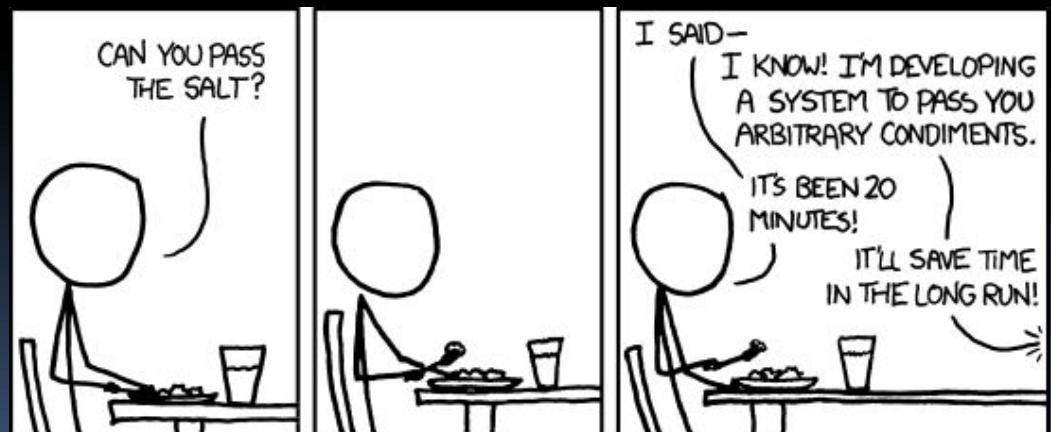
- You are going to learn to write functions, like in math class:

$$y = \sin(x)$$



“Function machine”
from Simply Scheme (Harvey)

- You should think about **what inputs make sense to use so you don't have to duplicate code**



The Power of Abstraction,

everywhere!

Examples:

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



Garcia

Summary

- Abstraction is one of the big ideas of computing and computational thinking
 - Detail Removal
 - Generalization
- Think about driving. How many of you know how a car works? How many can drive a car? Abstraction!



Someone who drove in 1930 could still drive a car today because they've kept the same Abstraction!
(right pedal faster, left pedal slow)

...they probably would have trouble starting a new one though,
or shifting a car from "park" into
"drive"