Programming Abstractions

CS106B

Cynthia Lee

Today's Topics

1. **Quick final exam discussion**
   - Details/logistics, topics, sources for practice problems
2. **Quarter wrap-up**
   - Putting it all together: what have we accomplished together this quarter?
3. **What next?**
   - Options for continuing your passion for CS after this quarter is done
   - Preview of CS107: security exploits in C/C++
     - FYI: if you are interested in security, join a new student group:
     - *Applied Cybersecurity is new a Stanford group focused on teaching students practical skills in exploiting and protecting systems. Past workshops include password cracking and lockpicking. For information regarding upcoming white hat hacking events and information on how to join our mailing list, visit our website: applied-cybersecurity.stanford.edu/*

Final Exam

Logistics

- Open notes: 2 pages (4 sides)
- Open textbook
- 3 hours
- In this room (Gates B01)

Final Exam Topics

- ADTs
- Recursion
- Backtracking
- Objects and classes
- Big-O analysis
- Pointers and dynamic memory
- Trees: heaps, binary search trees, tries, types of traversals
- Hashing
- Graphs: Dijkstra's, A*, Kruskals, BFS, DFS
- Inheritance, Polymorphism
- Sorting algorithms

Final Exam Study Strategy

- Don’t memorize things—either write it in notes, or learn the concept
  - If you’ve got flash cards, you’re approaching this with the wrong mindset
  - No big multiple choice/true-false section where “memorized” facts would be tested
- Don’t read the book (except in a targeted way)
  - Computer science is about creating things, so do some practice problems
  - Re-do Socrative questions from lecture, do old section problems, do CS106B practice exams as warm-up for our practice exams
  - Look at lecture slides or book **as needed** for review of things you identify as weak points when solving problems
- Don’t stress
  - Most of the really mind-bending topics (recursion, pointers) were on the midterm, and you’ve had more time to let those settle in

Big O Quick Reference (see also http://bigocheatsheet.com/)

What	Cost
• Hash table average case (good design)	$O(1)$
• Balanced trees • Heap, BST with balancing such as Red-Black • Binary search on sorted array	$O(\log n)$
• Linked list find • Inserting into beginning/middle of array • Hash table worst case • Unbalanced tree (e.g. BST) worst case	$O(n)$
• Good sorting • Mergesort, Heapsort, Quicksort (expected)	$O(n \log n)$
• Bad sorting • Insertion, Bubble, Selection, Quicksort (worst case)	$O(n^2)$

Quarter Wrap-Up

What did we set out to do in the beginning?

Where are we now?

Stanford University

Goals for this Course

- **Learn how to model and solve complex problems with computers.**
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - Bring added rigor to your understanding of algorithmic performance, so you can quantitatively compare approaches for solving problems.

From here on out, there are no obvious answers to any problem worth your hourly rate. ☺

- Programming is all about exploring new ways to **model** and **solve** problems.

- There are CHOICES and TRADEOFFS in how we model these and how we implement them! (array or linked list? BST or hash table?)

- Skilled computer scientists recognize that any problem worth tackling has *many* possible models and *many* possible solutions, none of which is clearly better than the others in all dimensions—**tradeoffs!**

That’s a lot of material to cover in 10 weeks

You are part of a very challenging course,
in the best CS department *in the world*,
and you are so, so close to completing this course!

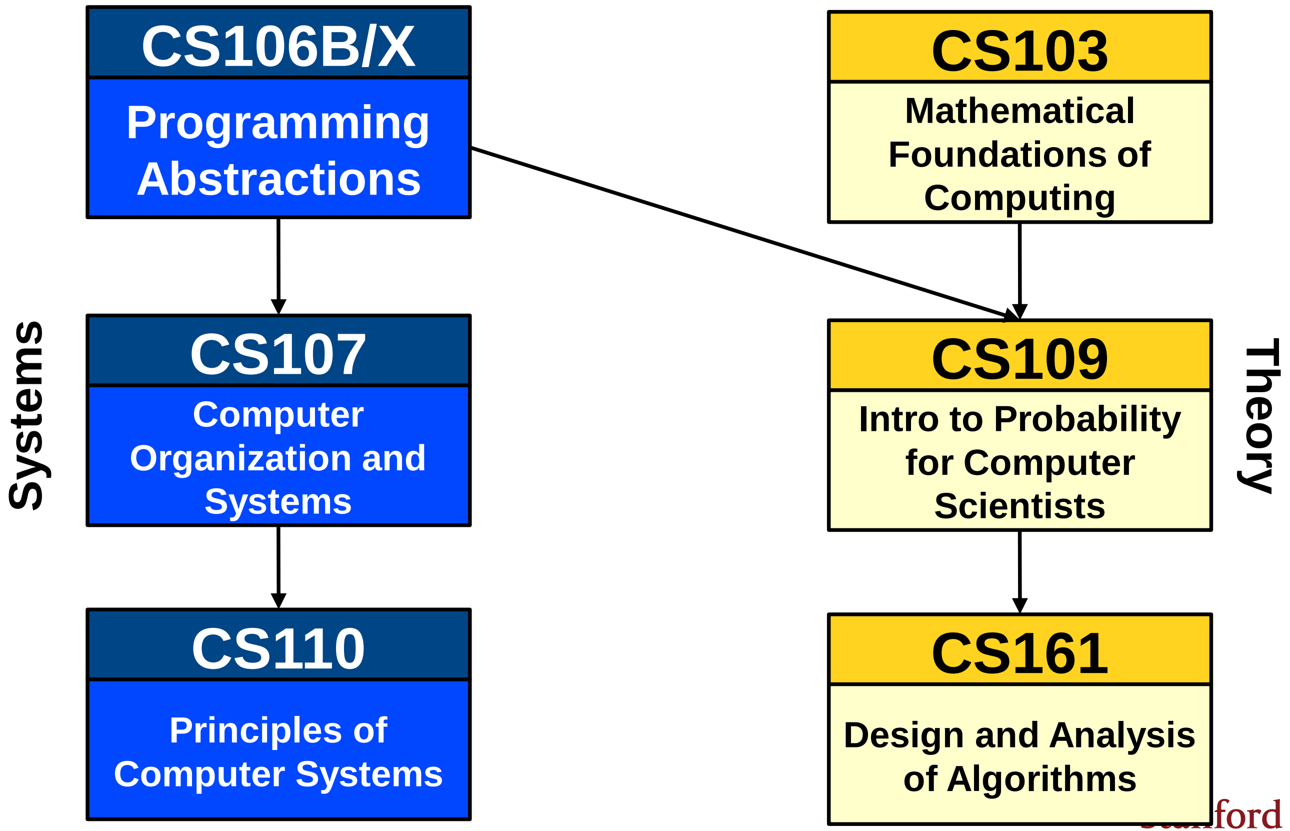
Congratulations!!
You’ve almost made it through CS106B!

- *So…what next?*

What comes next?

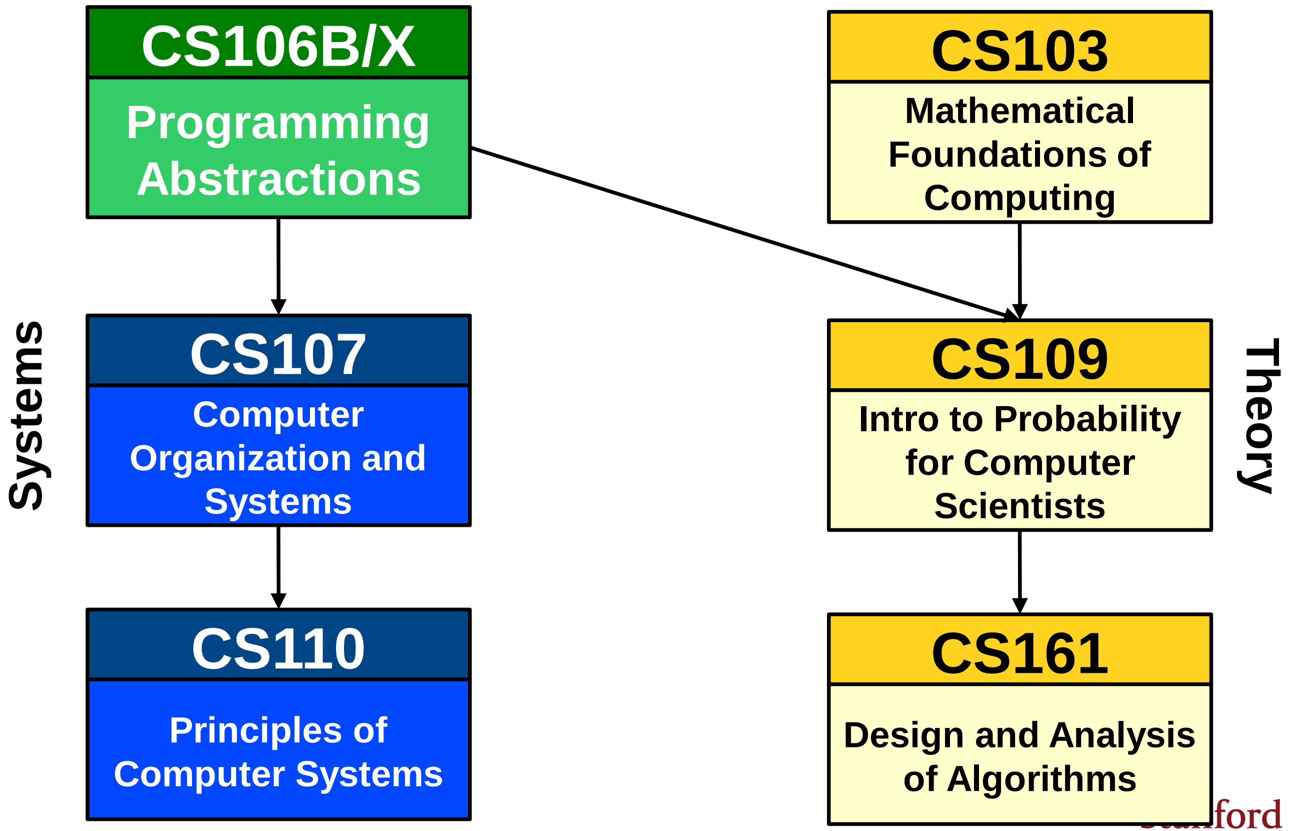
You’re conquering this mountain, let’s find some more ☺

The CS Core

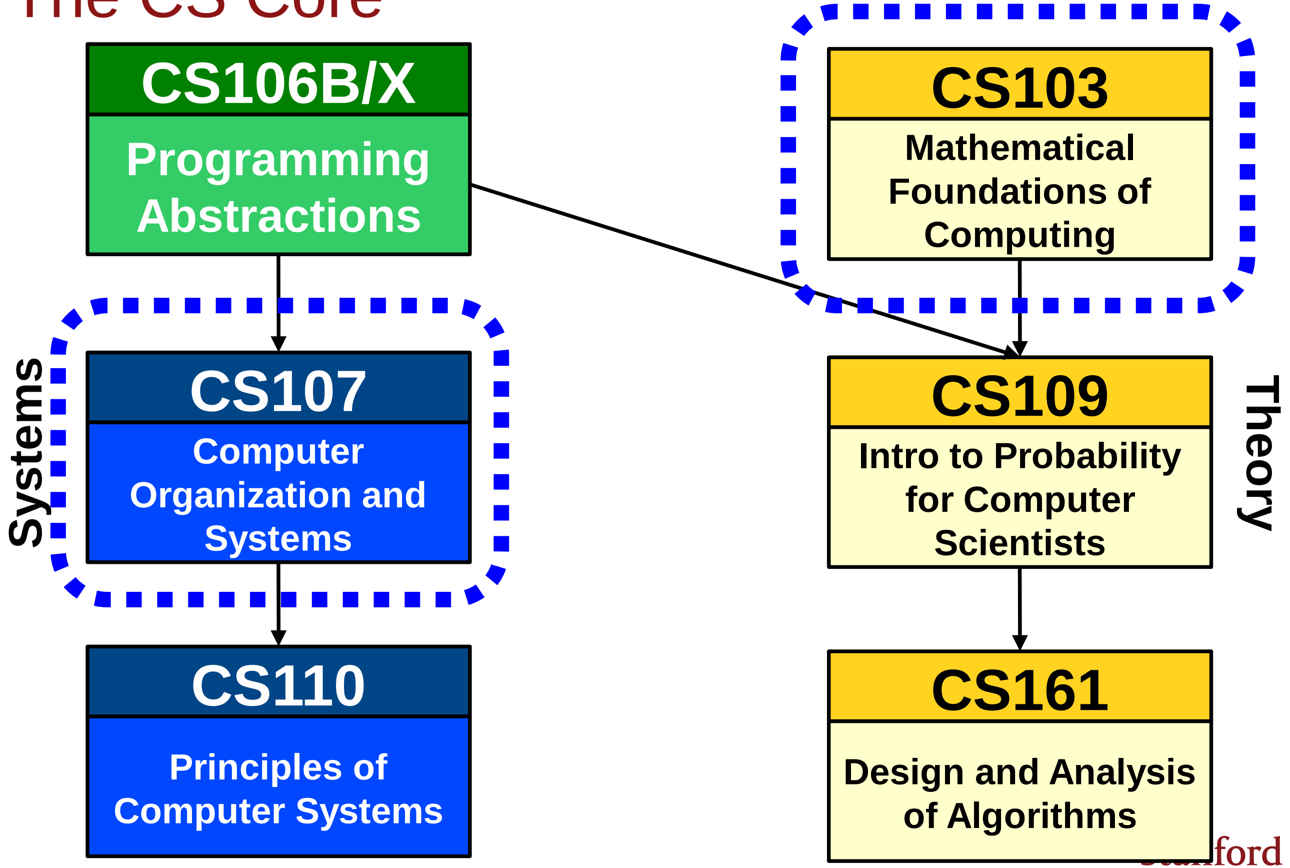

CS106B/X
Programming Abstractions
CS107
Computer Organization and Systems
CS110
Principles of Computer Systems
CS103
Mathematical Foundations of Computing
CS109
Intro to Probability for Computer Scientists
CS161
Design and Analysis of Algorithms
Systems
Theory

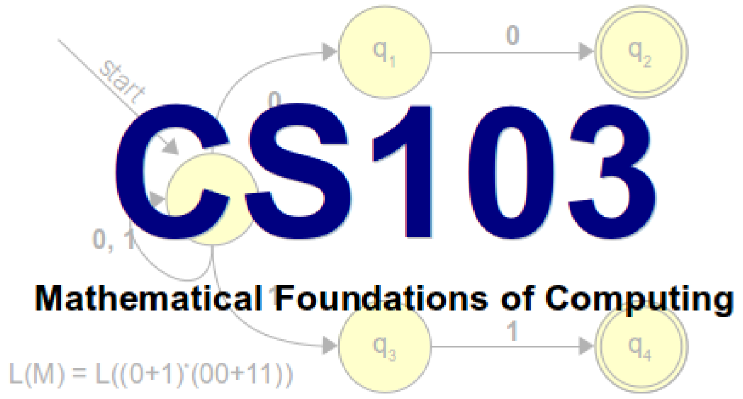
Stanford University

The CS Core


CS106B/X
Programming Abstractions
CS107
Computer Organization and Systems
CS110
Principles of Computer Systems
CS103
Mathematical Foundations of Computing
CS109
Intro to Probability for Computer Scientists
CS161
Design and Analysis of Algorithms
Systems
Theory

The CS Core


CS106B/X
Programming Abstractions
CS103
Mathematical Foundations of Computing
Systems
CS107
Computer Organization and Systems
CS109
Intro to Probability for Computer Scientists
Theory
CS110
Principles of Computer Systems
CS161
Design and Analysis of Algorithms


CS103
Mathematical Foundations of Computing
start
q_1
q_2
q_3
q_4
0
0
1
0, 1
$L(M) = L((0+1)^*(00+11))$

Can computers solve all problems?

Spoiler alert: no!

Why are some problems harder than others?

We can do find in an unsorted array in O(N), and we can sort an unsorted array in O(NlogN). Is sorting just inherently a harder problem, or are there better O(N) sorting algorithm yet to be discovered?

How can we be certain about this?

Stanford University

CS107
Computer Organization and Systems

How do we encode text, numbers, programs, etc. using just 0s and 1s?

Where does memory come from? How is it managed?

How do compilers, debuggers, etc. work?

CS107 in the news: Heartbleed

- In April 2014, security experts warned that users of thousands of major websites needed to change their passwords due to potential exposure caused by the "Heartbleed" vulnerability

- Heartbleed exploited a **buffer overrun** bug in OpenSSL
  - SSL is the layer that secures web interactions, i.e., it's what make the "s" in "https://" mean something

CNET › Security › How to protect yourself from the 'Heartbleed' bug

How to protect yourself from the 'Heartbleed' bug

A new security bug means that people all across the Web are vulnerable to having their passwords and other sensitive data stolen. Here's what consumers can do to protect themselves.

by Richard Nieva @richardjnieva / April 8, 2014 2:37 PM PDT

139 / 4.4K / 1.6K / 896 / more +

Codenomicon/CNET

A major new security vulnerability dubbed Hea
implications for the entire Web. The bu
stored, including private data s

See also: The H

CS107 in the news: Heartbleed

- The protocol allows you to send “heartbeat” messages, which basically say:
  - › *`Are you still there? If you are, repeat this word back to me: "hello" [0x0005 bytes].`*
  - › Each char is one byte, so 5 letters
- Unfortunately, the software also let you send messages like this:
  - › *`Are you still there? If you are, repeat this word back to me: "hello" [0xFFFF bytes].`*
  - › That’s 65535 bytes—much more than the length of `"hello"`!
  - › So the software would continue for-looping past the end of the `"hello"` array, sending information back
  - › Which causes an error, right? **RIGHT??** Turns out, no.

CNET › Security › How to protect yourself from the 'Heartbleed' bug

How to protect yourself from the 'Heartbleed' bug

A new security bug means that people all across the Web are vulnerable to having their passwords and other sensitive data stolen. Here's what consumers can do to protect themselves.

by Richard Nieva @richardjnieva / April 8, 2014 2:37 PM PDT

139 / 4.4K / 1.6K / 896 / more +

Codenomicon/CNET

A major new security vulnerability dubbed Hea
implications for the entire Web. The bug
stored, including private data s

See also: The H

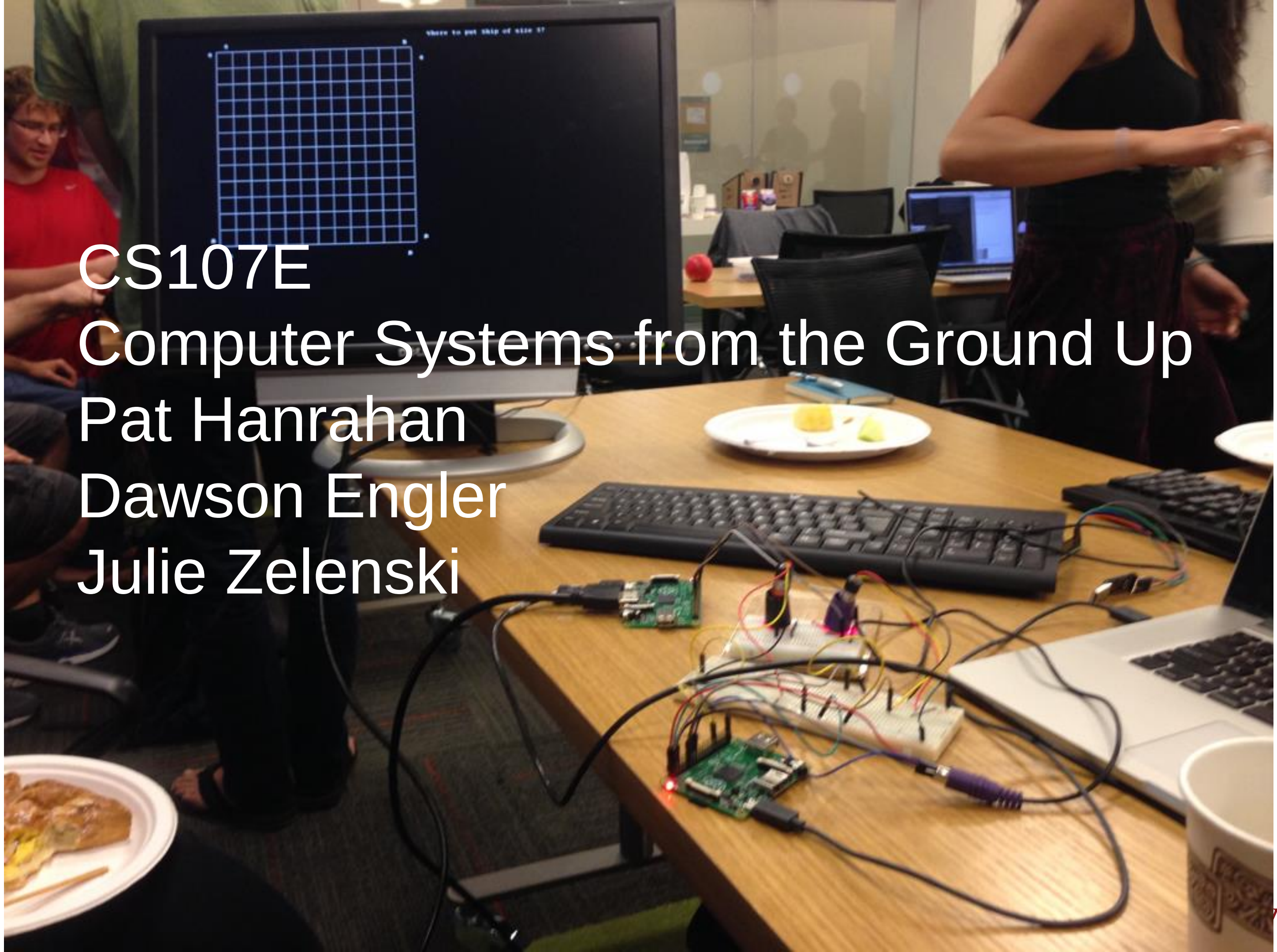
It's

What CS107 Isn't

- CS107 is ***not*** a litmus test for whether you can be a computer scientist.
  - You can be a *great* computer scientist without enjoying low-level systems programming.
- CS107 is ***not*** indicative of what programming is “really like.”
  - CS107 does a lot of low-level programming. You don't have to do low-level programming to be a good computer scientist.

CS107E
Computer Organization and Systems—*Embedded*

- **Counts for prerequisites etc. the same as regular CS107**, but covers the topics with a new twist: embedded work on Rasberry Pi



CS107E

Computer Systems from the Ground Up

Pat Hanrahan

Dawson Engler

Julie Zelenski

ersity

FAQ (vs CS107)

- Same goals: understand how computers represent data and execute programs; tools
- Different approach: bare metal on the Raspberry Pi; build a working personal computer from scratch
- Logistics
  - Same format: weekly assignments and labs
  - Assignments build on each other
  - No exams, but a final project
  - ARM vs X86
  - More hardware (thinker Maker Faire, breadboard)
  - Enrollment limited to 40; Application through cs107e on Axess; Due 12/12; still lots of openings!!
  - Will be offered in winter and spring

Other CS Courses

Stanford University

CS9
Interviewing for Software Jobs

- 1 unit, 1 meeting per week, little to no outside work
- Prereq: 106B/X
- Practice **real** job interview questions
  - Additional topics such as resume polish, negotiating once you have multiple offers, differences between roles (Project Management vs Developer vs Test Engineer)
- ***Special guests from industry!***

Taught by Cynthia Lee, Keith Schwarz

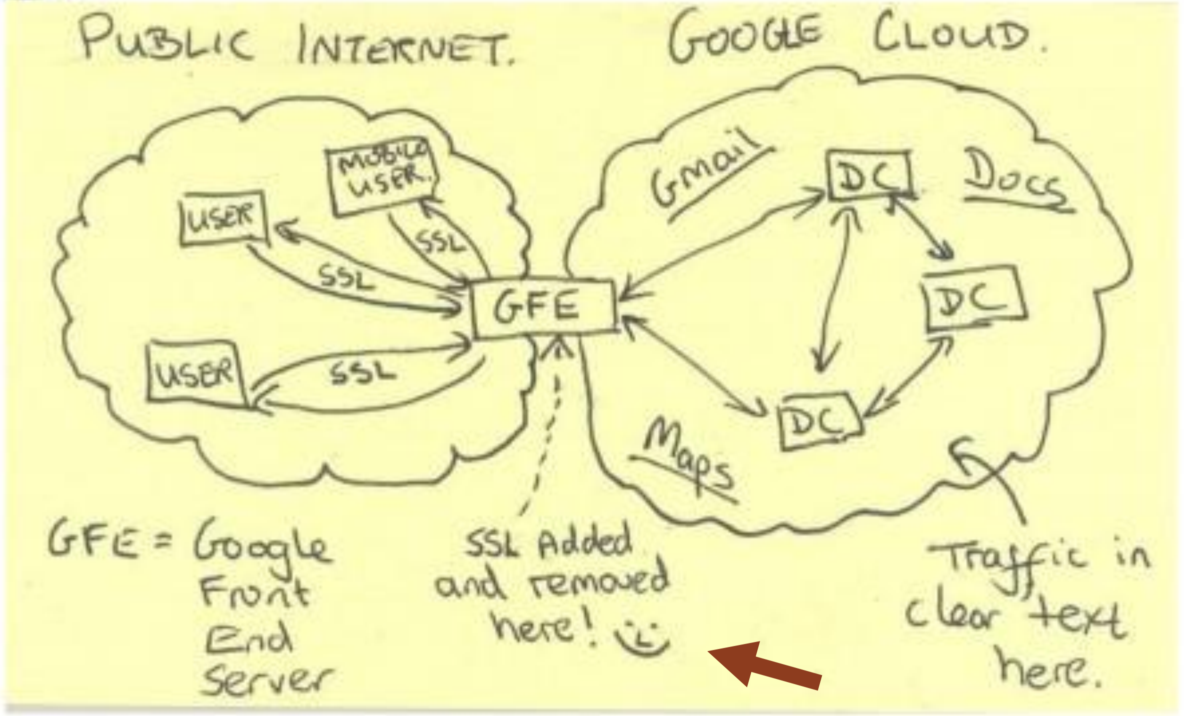
Offered each autumn quarter

CS181
Computers, Ethics, and Public Policy

- Some sample news headlines recently:
  - Edward Snowden reveals that NSA knows more about you than your parents do
  - GamerGate: about harassing women, or about ethics in game journalism?
  - How should AirBnB be taxed?
  - The password to launch the US nuclear arsenal was 00000000

We have the power to control and create technology, but how should we use it?

Current Efforts - Google


PUBLIC INTERNET.
GOOGLE CLOUD.
MOBILE USER.
USER
USER
SSL
SSL
SSL
GFE
Gmail
DC
Docs
DC
DC
Maps
GFE = Google Front End Server
SSL Added and removed here! :-)
Traffic in clear text here.

CS108
Object-Oriented Systems Design

- ***How do you build large software systems in a team?***
- Introduction to things you need to know for work in the “real world”:
  - Unit-testing frameworks
  - Object-oriented design
  - Multithreaded applications
  - Databases and web applications
  - Source control

Stanford University

CS193
Language-specific courses

- Misc. offerings throughout the year, focused on specific technologies:
  - CS193A: Android Programming
  - CS193C: Client-Side Web Technologies
  - CS193I: iOS Programming
  - CS193L: Lua Programming
  - CS193P: iPhone and iPad programming
- Great for learning particular technologies.

Options besides CS Major

CS Minor: only 5 more classes!

- 103, 107, 109, two your choice—fun!

CS Coterminal MS degree

- Earn an MS in CS while you are here earning your BS
- Possible for CS majors **and** other majors
  - ex: Math major, CS co-term