Life After CS106B

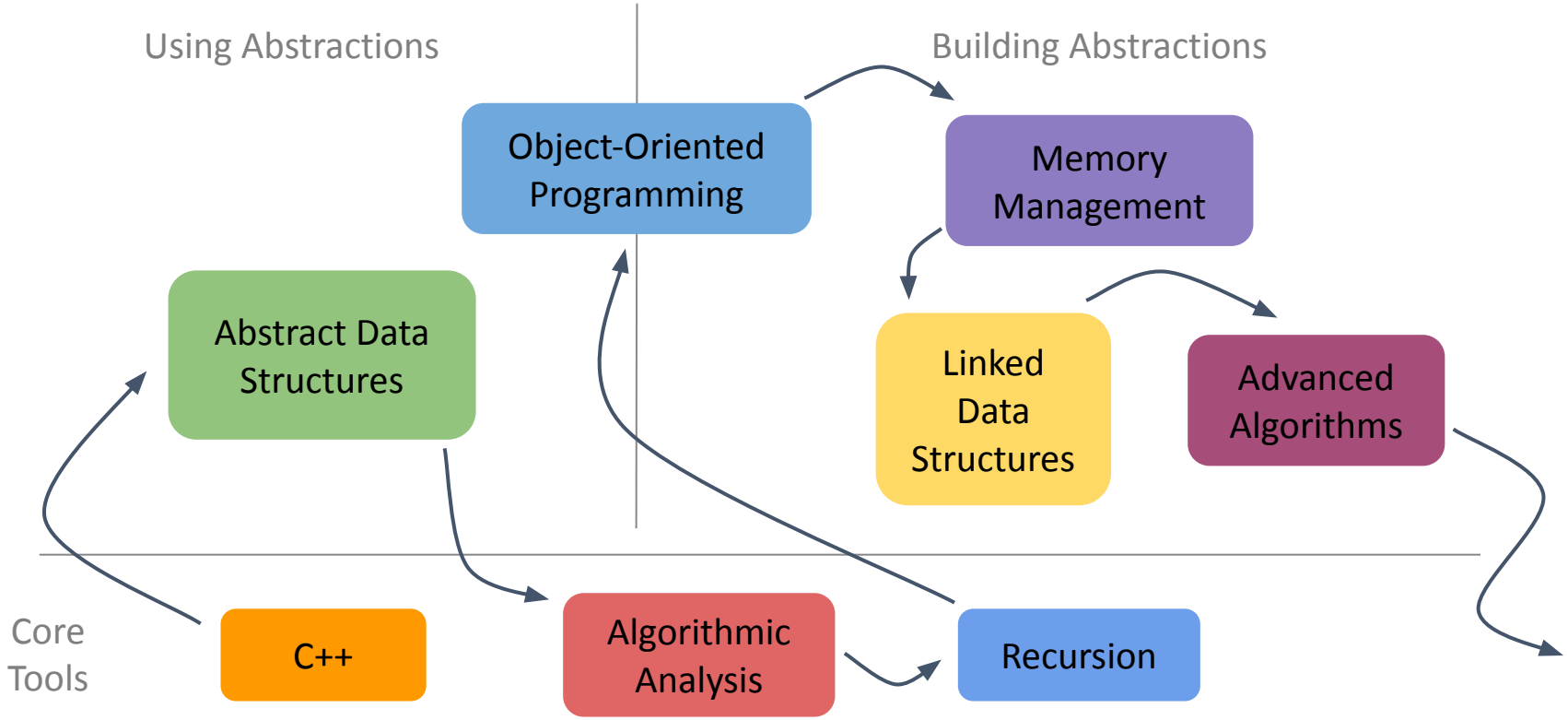
Elyse Cornwall

August 16, 2023

Announcements

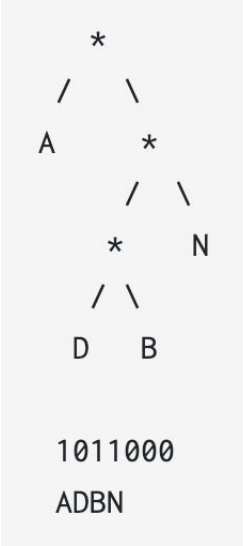
- Course evals are now open - find these on Canvas
  - This is a chance for you to provide feedback to Stanford on the instructors and CS106B in general (not your SL)
- No lecture tomorrow
- Assignment 6 **hard deadline** Thursday at 11:59pm
  - Group office hours today 3-5pm in Durand 303
  - Sign up for Elyse's 1:1 office hours tomorrow 3-5pm (a few slots left!)
  - LaIR Wednesday and Thursday night

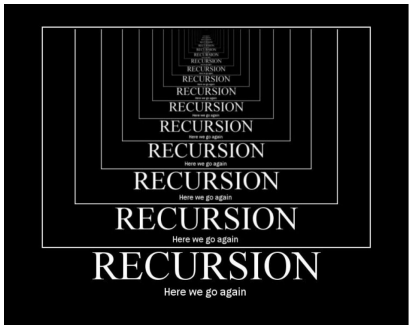
Where Have We Been?

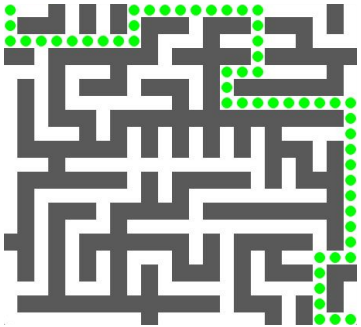


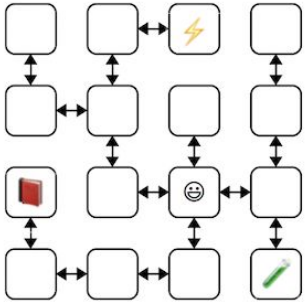
What is CS106B About? (Slide from Day 1)

- Solving interesting problems and processing large datasets
- Creating and managing complex data structures
- Analyzing the efficiency of your solutions
- Practicing good programming style and coding practices
- Gaining familiarity with the C++ programming language



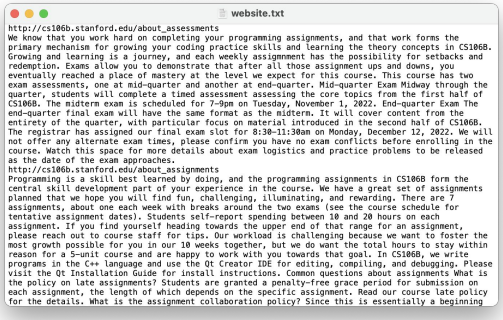




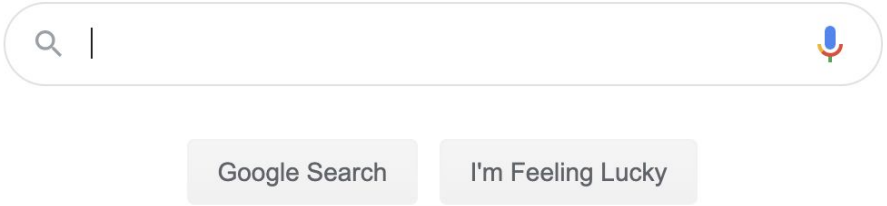


What is CS106B About?

- **Solving interesting problems and processing large datasets**


website.txt

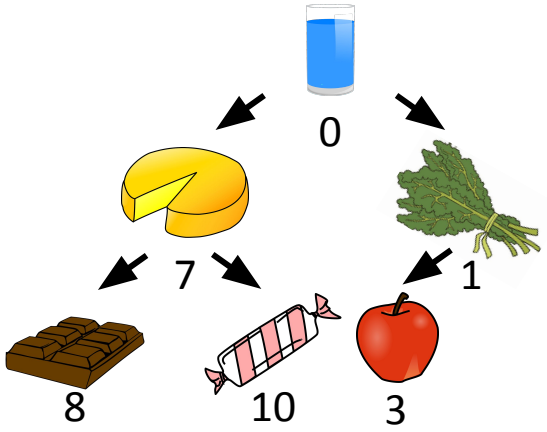
http://cs106b.stanford.edu/about_assessments
We know that you work hard on completing your programming assignments, and that work forms the primary mechanism for growing your coding practice skills and learning the theory concepts in CS106B. Growing and learning is a journey, and each weekly assignment has the possibility for setbacks and redemption. Exams allow you to demonstrate that after all those assignment ups and downs, you eventually reached a place of mastery at the level we expect for this course. This course has two exam assessments, one at mid-quarter and another at end-quarter. Mid-quarter Exam Midway through the quarter, students will complete a timed assessment assessing the core topics from the first half of CS106B. The midterm exam is scheduled for 7-9pm on Tuesday, November 1, 2022. End-quarter Exam The end-quarter final exam will have the same format as the midterm. It will cover content from the entirety of the quarter, with particular focus on material introduced in the second half of CS106B. The registrar has assigned our final exam slot for 8:30-11:30am on Monday, December 12, 2022. We will not offer any alternate exam times, please confirm you have no exam conflicts before enrolling in the course. Watch this space for more details about exam logistics and practice problems to be released as the date of the exam approaches.
http://cs106b.stanford.edu/about_assignments
Programming is a skill best learned by doing, and the programming assignments in CS106B form the central skill development part of your experience in the course. We have a great set of assignments planned that we hope you will find fun, challenging, illuminating, and rewarding. There are 7 assignments, about one each week with breaks around the two exams (see the course schedule for tentative assignment dates). Students self-report spending between 10 and 20 hours on each assignment. If you find yourself heading towards the upper end of that range for an assignment, please reach out to course staff for tips. Our workload is challenging because we want to foster the most growth possible for you in our 10 weeks together, but we do want the total hours to stay within reason for a 5-unit course and are happy to work with you towards that goal. In CS106B, we write programs in the C++ language and use the Qt Creator IDE for editing, compiling, and debugging. Please visit the Qt Installation Guide for install instructions. Common questions about assignments What is the policy on late assignments? Students are granted a penalty-free grace period for submission on each assignment, the length of which depends on the specific assignment. Read our course late policy for the details. What is the assignment collaboration policy? Since this is essentially a beginning



What is CS106B About?

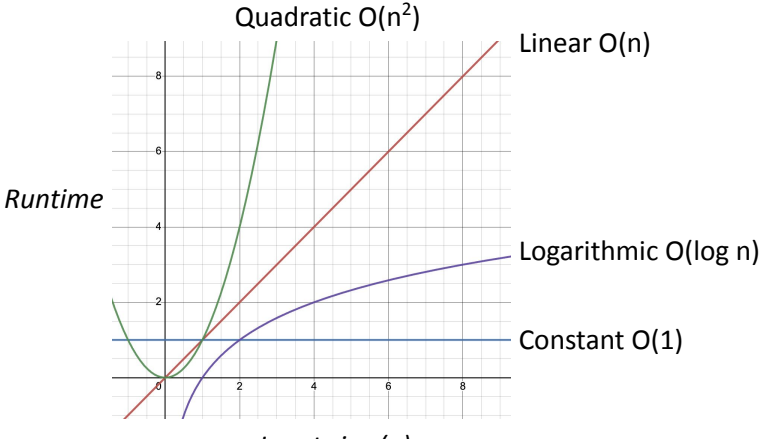
- **Creating and managing complex data structures**

```
class PQHeap {
public:
    PQHeap();
    ~PQHeap();
    void enqueue(DataPoint element);
    DataPoint dequeue();
    DataPoint peek() const;
    bool isEmpty() const;
    int size() const;
    void clear();
private:
    int getParentIndex(int child) const;
    int getLeftChildIndex(int parent) const;
    int getRightChildIndex(int parent) const;
};
```



What is CS106B About?

- **Analyzing the efficiency of your solutions**

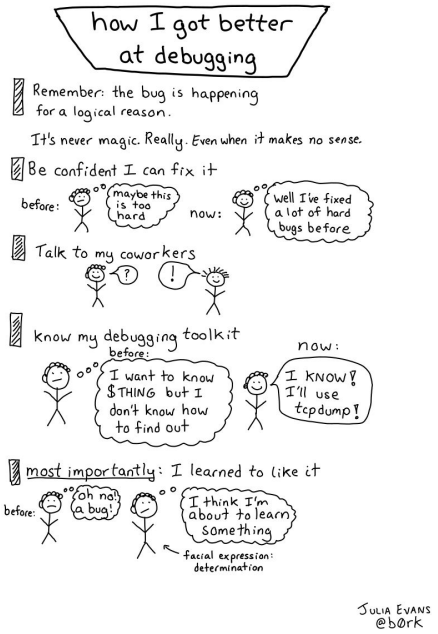


Q8. Give the results from your time trials and explain how they support your prediction for the Big-O runtimes of **enqueue** and **dequeue**.

```
Correct (STUDENT_TEST, perfect.cpp:122) Multiple time trials of findPerfects on increasing input sizes
    Line 127 TIME_OPERATION findPerfects(size) (size =    10000) completed in    0.145 secs
    Line 127 TIME_OPERATION findPerfects(size) (size =    20000) completed in    0.533 secs
    Line 127 TIME_OPERATION findPerfects(size) (size =    40000) completed in    2.089 secs
    Line 127 TIME_OPERATION findPerfects(size) (size =    80000) completed in    8.241 secs
```

What is CS106B About?

- **Practicing good programming style and coding practices**

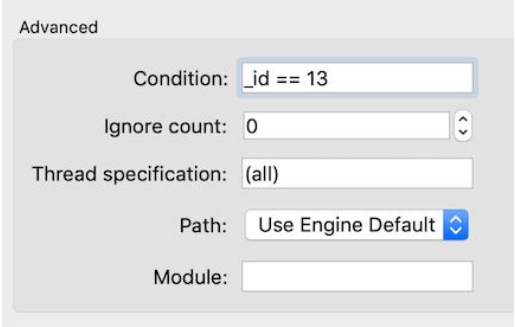


```
// ugh
if (count == 0) error("not found");
for (int i = 0; i < n; i++) draw(i);
if (condition)
    doFirst();
    doSecond(); // inside? Indent looks
```

```
// better
if (count == 0) {
    error("not found");
}
for (int i = 0; i < n; i++) {
    draw(i);
}

if (condition) {
    doFirst();
    doSecond();
}
```

Name	Value	Type
[statics]		
num	28672	int
q	<5 items>	Queue<int> &
front	1	int
-	2	int
-	3	int
-	4	int
back	5	int
s	<0 items>	Stack<int>



What is CS106B About?

- **Gaining familiarity with the C++ programming language**

Containers library

array (C++11)
vector – deque
list – forward_list (C++11)
set – multiset
map – multimap
unordered_map (C++11)
unordered_multimap (C++11)
unordered_set (C++11)
unordered_multiset (C++11)
stack – queue – priority_queue

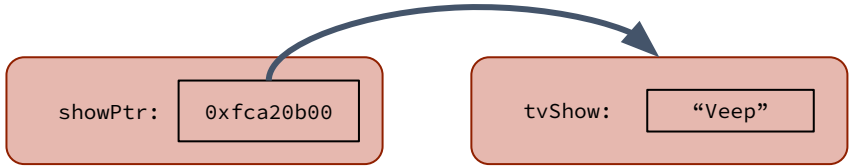
The Stanford libcs106 library,

Collection classes

Several of the classes represent collections of other objec

Each of these collections is documented in its own page,

Vector<ValueType>	A Vector is an index
Grid<ValueType>	A Grid is an indexe
GridLocation	A GridLocation stru
GridLocationRange	A GridLocationRan
Stack<ValueType>	A Stack is a linear s
Queue<ValueType>	A Queue is a linear
PriorityQueue<ValueType>	A PriorityQueue is a
Map<KeyType, ValueType>	A Map maintains ar
HashMap<KeyType, ValueType>	A HashMap is a hig
Set<ValueType>	A Set is an ordered
HashSet<ValueType>	A HashSet is a high
Lexicon	A Lexicon is a highl



What Comes Next?

Continuing Your CS Journey

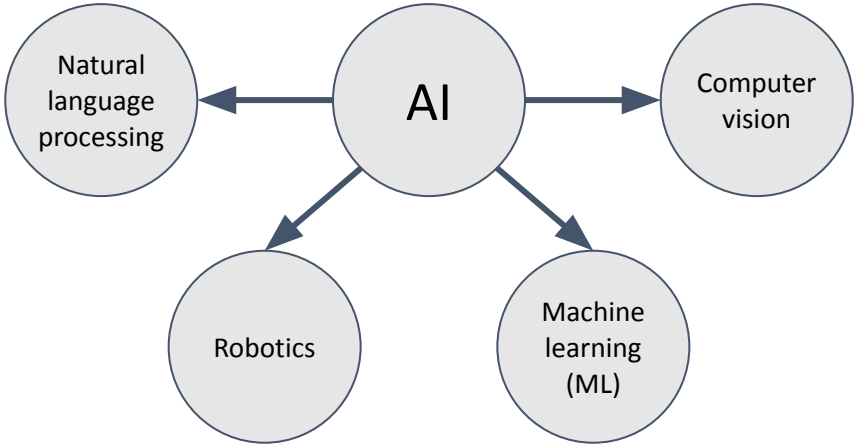
- If you already have a domain in mind, you might start to specialize!
- You could explore a breadth of core topics if you're considering majoring or minoring in CS

The Stanford CS Tracks

Artificial Intelligence (AI)

- Logic, probability, statistics
- Classes like CS221, CS124, and CS229

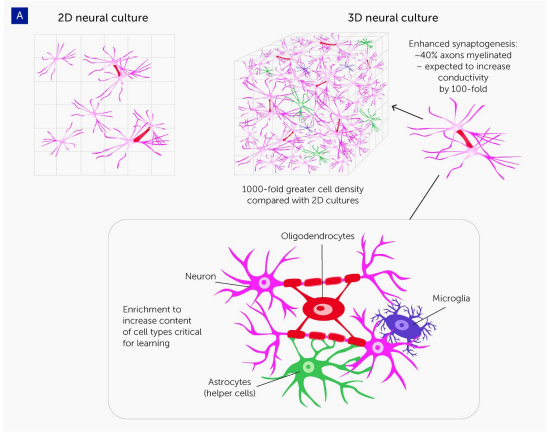
And much more!



The Stanford CS Tracks

Computational Biology (Biocomp)

- Biological and medical applications of CS
- Advanced algorithms, databases, networking, modeling, and simulation
- Classes like CS173A and CS279

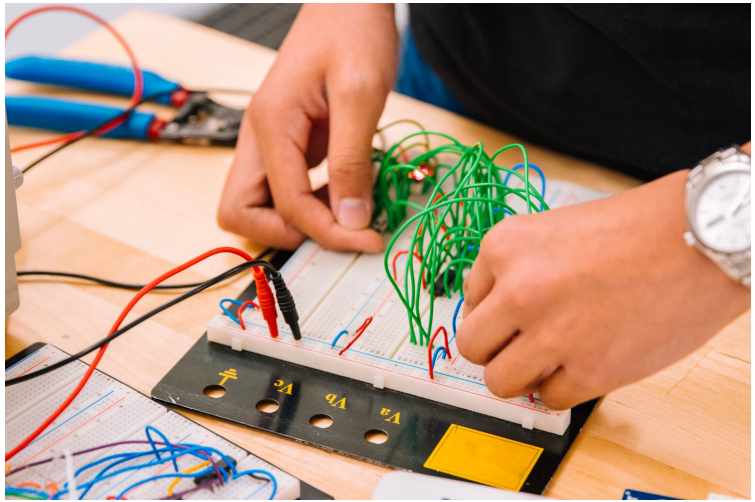


Abbreviations: **2D/3D**, 2/3-dimensional; **MAP2**, microtubule associated protein 2; **OLIG2**, oligodendrocyte transcription factor; **GFAP**, glial fibrillary acidic protein

The Stanford CS Tracks

Computer Engineering

- Combination of CS and electrical engineering
- Networking, digital systems, and robotics and mechatronics
- Classes like EE108 and EE180



The Stanford CS Tracks

Visual Computing (Graphics)

- Creating and manipulating images, videos, 3D geometry, and simulated virtual worlds
- Intersection of AI/ML, graphics, and systems
- Classes like CS148, CS248, and CS231N

The Stanford CS Tracks

Visual Computing (Graphics)

- Creating and manipulating images, videos, 3D geometry, and simulated virtual worlds
- Intersection of AI/ML, graphics, and systems
- Classes like CS148, CS248, and CS231N

Elyse's CS148 Project

The Stanford CS Tracks

Human-Computer Interaction (HCI)

- Studying how people interact with computational devices
- “Front-end” coding, user-centered design, psych, social science
- Classes like CS147, CS247, and CS278

The Stanford CS Tracks

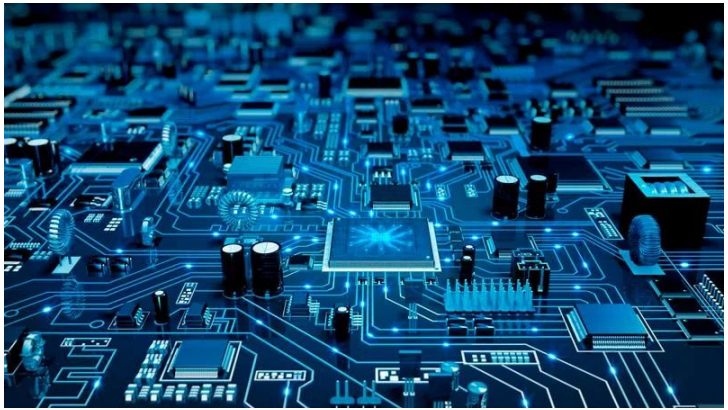
Information

- Creating, processing, and understanding digital information
- Data storage, query languages, data integrity, data extraction
- Classes like CS124 and CS145

The Stanford CS Tracks

Systems

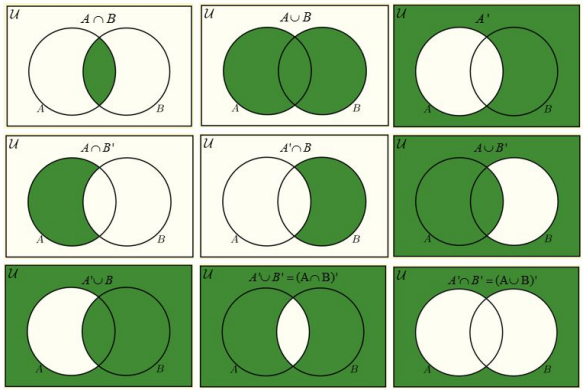
- Design and implementation of compilers, databases, networks, and operating systems (think memory, pointers, bits and bytes)
- Classes like CS107, CS111, CS112, and CS143

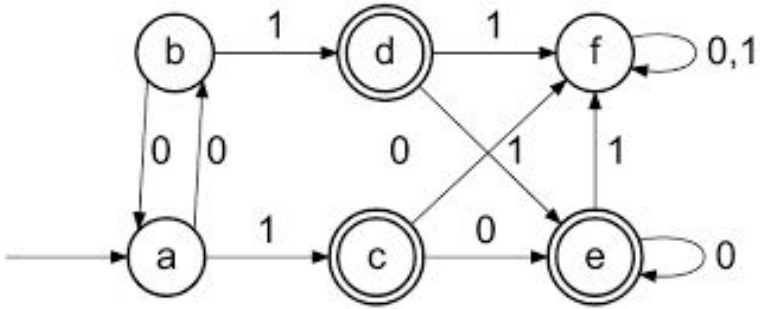


The Stanford CS Tracks

Theory

- Computational techniques, their capabilities and limitations
- Data structures, algorithms, program analysis, and logic
- Classes like CS103, CS154, and CS168





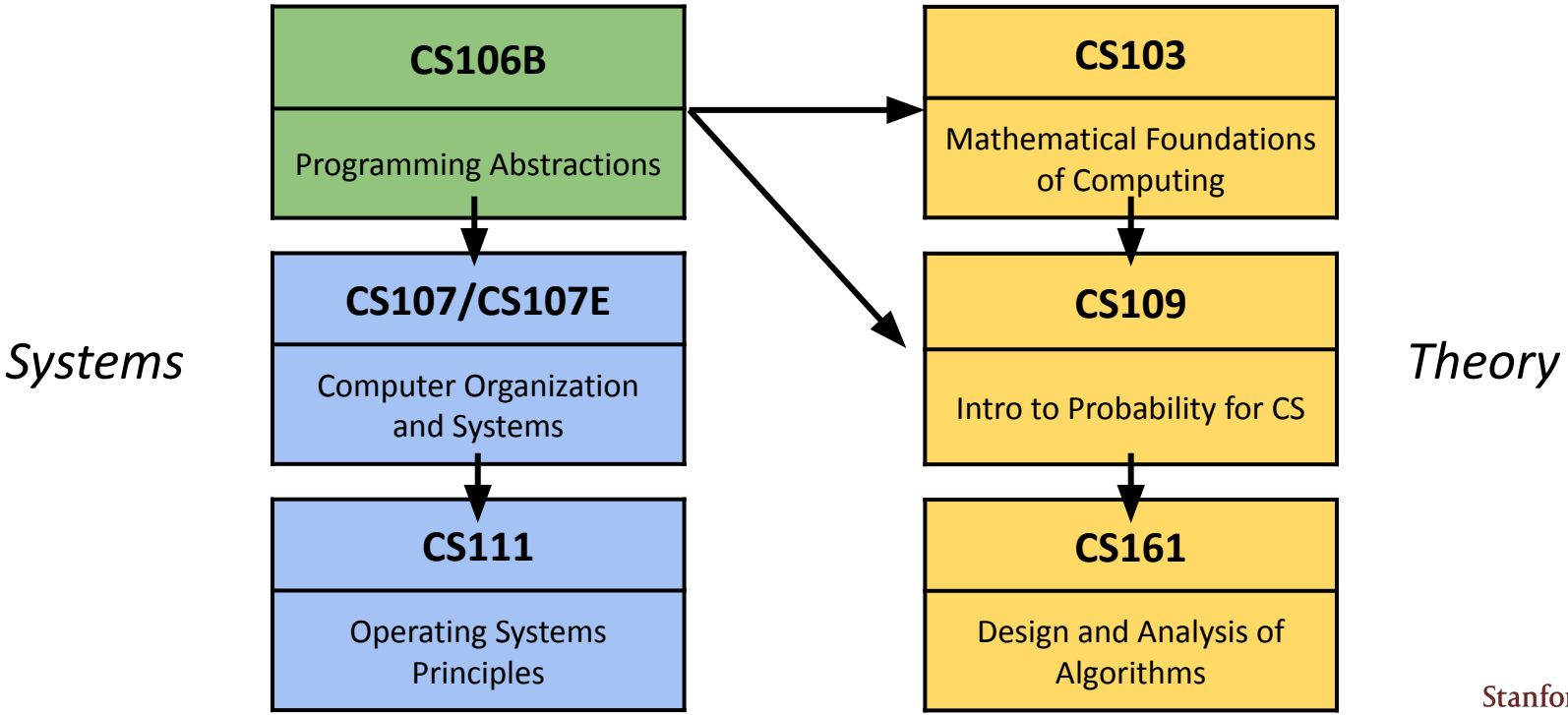
The Stanford CS Tracks

- Artificial Intelligence (AI)
- Computational Biology (Biocomp)
- Computer Engineering
- Visual Computing (Graphics)
- Human-Computer Interaction (HCI)
- Information
- Systems
- Theory

Beyond Stanford, these are the main areas of computer science study (maybe with the addition of security).

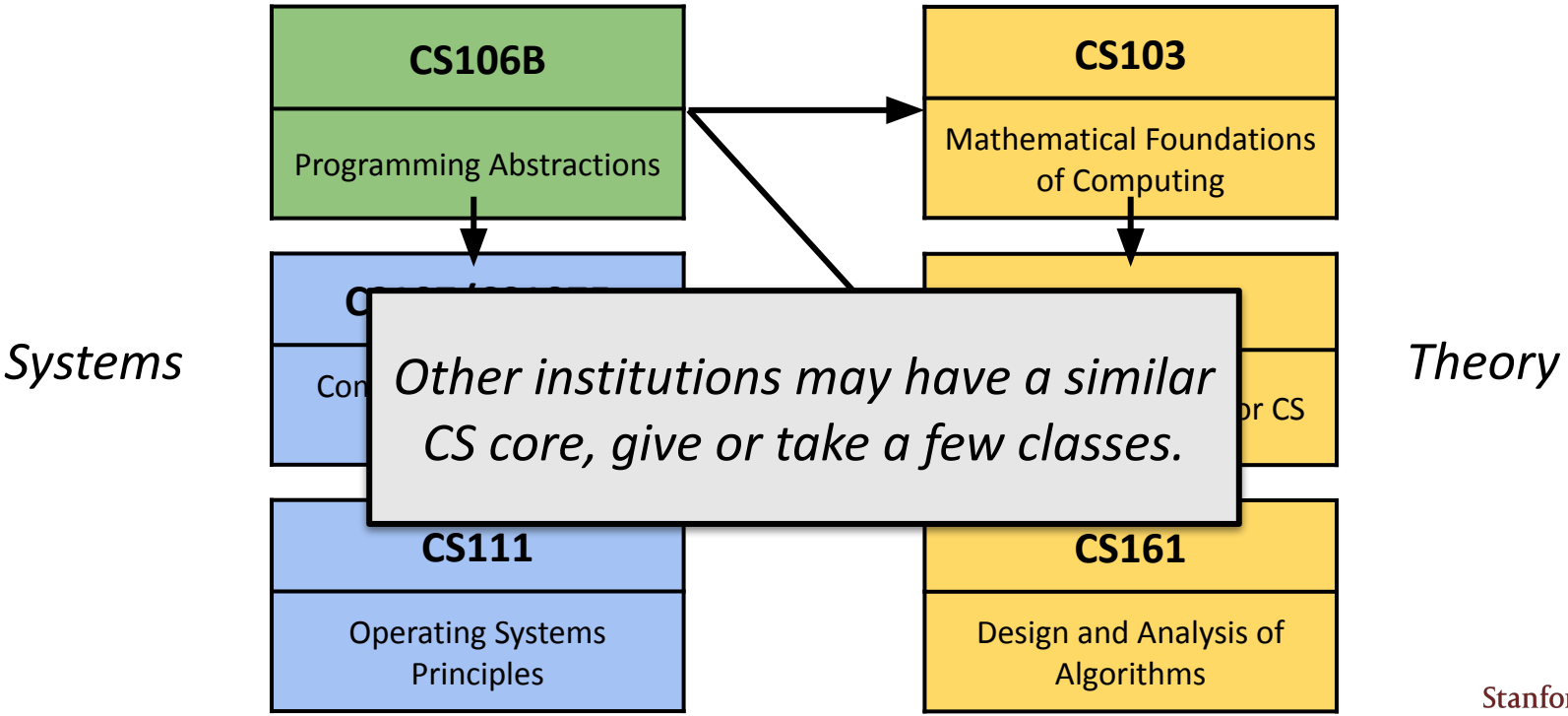
The Stanford CS Core

A breadth of CS classes that all CS majors/minors must take



The Stanford CS Core

A breadth of CS classes that all CS majors/minors must take



Practical Applications

What if I want to build a website or app?

- CS193A - Android Programming
- CS193C - Client-Side Web Technologies
- CS193P - iOS Programming
- CS193Q - Acceleration Intro to Python
- CS193X - Web Programming Fundamentals

More C++

Many core classes are taught in C/C++:

- CS107
- CS111
- CS103

If you want to focus more on the language itself:

- CS106L: Standard C++ Programming (check out the course reader)

Beyond Stanford

Online resources

- Codecademy
- Coursera, edX, Udemy, and other MOOCs
- Khan Academy
- MIT OpenCourseWare

Strategies for keeping up your code skills

- Work on a project that interests you
- Find other people to collaborate with
- Join open-source projects

Teaching!

Teaching is a great way to solidify your CS knowledge, and it's fun 😊

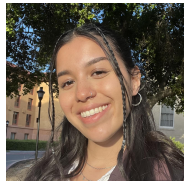
- If you're a Stanford undergrad, you can apply to section lead!
- You can apply to teach intro CS in the next iteration of Stanford's Code in Place (no Stanford affiliation required)

👥 Three Questions

- What's something you had never heard about before CS106B that you know now?
- What's something you were looking forward to learning in CS106B that you know now?
- What's something you learned in CS106B that you'd like to explore further after the class ends?










Thank you for a wonderful quarter!!
Good luck on the final!