CS106B: Programming Abstractions

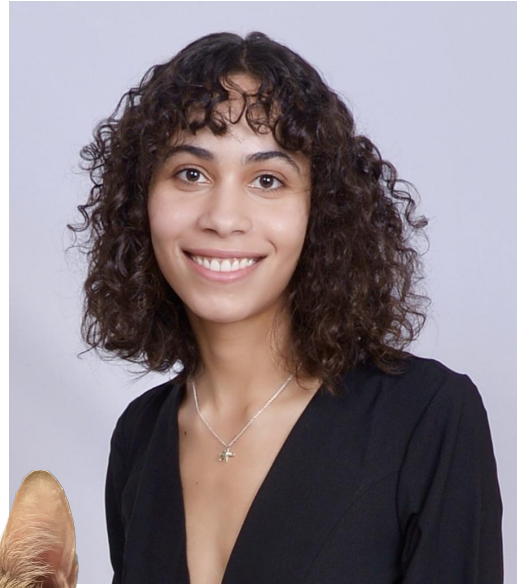
Elyse Cornwall and Amrita Kaur

June 26, 2023

Contributions made from previous CS106B Instructors

Elyse Cornwall

- Stanford BS CS ’22, MS CS candidate ’24
  - Studied theory and human-computer interaction
- CS106A Head TA during school year
- CS section leader for 2 years as an undergrad
- From Milwaukee, Wisconsin
- Here are some photos of my cat, Sidibou!



Amrita Kaur

- Call me Amrita (pronounced um-rih-tha)
- From Columbia, Maryland
- Stanford BS Biology ‘20, MS CS ‘22
- Taught CS110A (systems), CS109 (probability)
- Absolutely love tennis and F1
- Guilty pleasure is reality TV (Survivor!!)

Just a Few of Your 20+ Section Leaders!









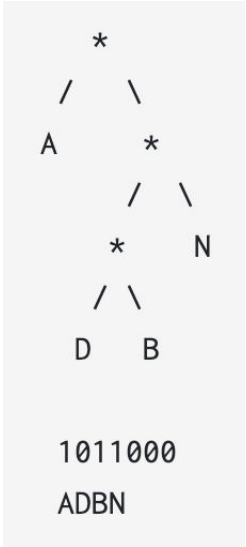


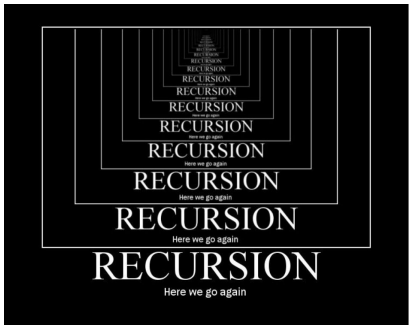

Let’s get to know each other!

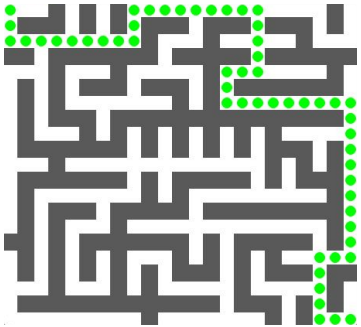
- Name (and pronouns if you’re comfortable)
- What school you go to/where you’re from
- Year + (prospective) major
- Pick one (or more!)
  - What is one hobby you want to pick up?
  - What is your strangest habit?
  - What are you looking forward to this summer?
  - Anything else you would like to share with each other :)

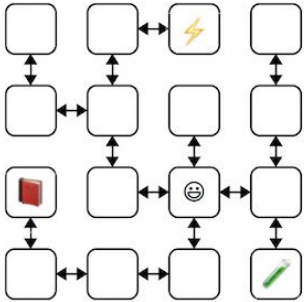
What is CS106B About?

- 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 an “abstraction”?

“The essence of abstraction is **preserving information that is relevant** in a given context, and **forgetting information that is irrelevant** in that context.”

– John V. Guttag

What is an “abstraction”?

“The essence of abstraction is **preserving information that is relevant** in a given context, and **forgetting information that is irrelevant** in that context.”

– John V. Guttag

Relevant: toasting the bread
Irrelevant: infrared radiation

What is an “abstraction”?

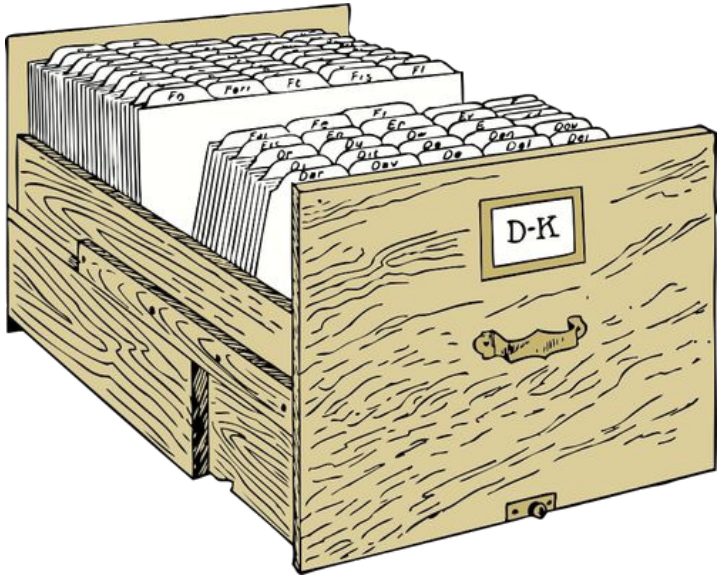
“The essence of abstraction is **preserving information that is relevant** in a given context, and **forgetting information that is irrelevant** in that context.”

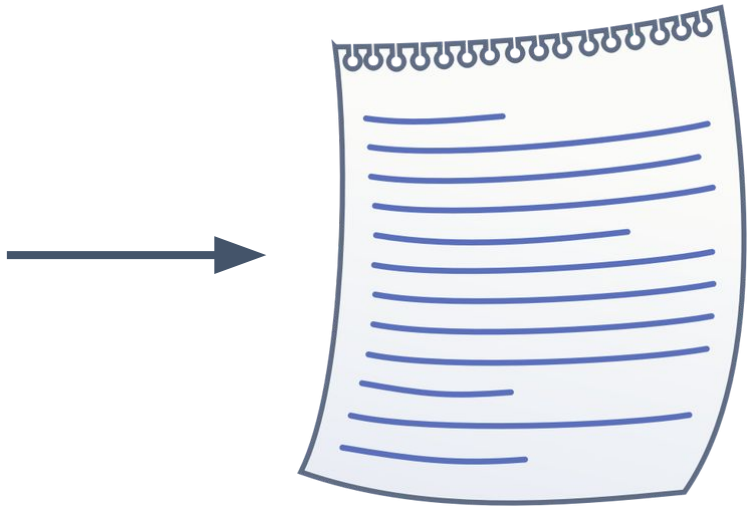
– John V. Guttag

```
bool isPerfect(long n)
```

Relevant: whether n is a “perfect number”
Irrelevant: how the function works

Example: Data Compression



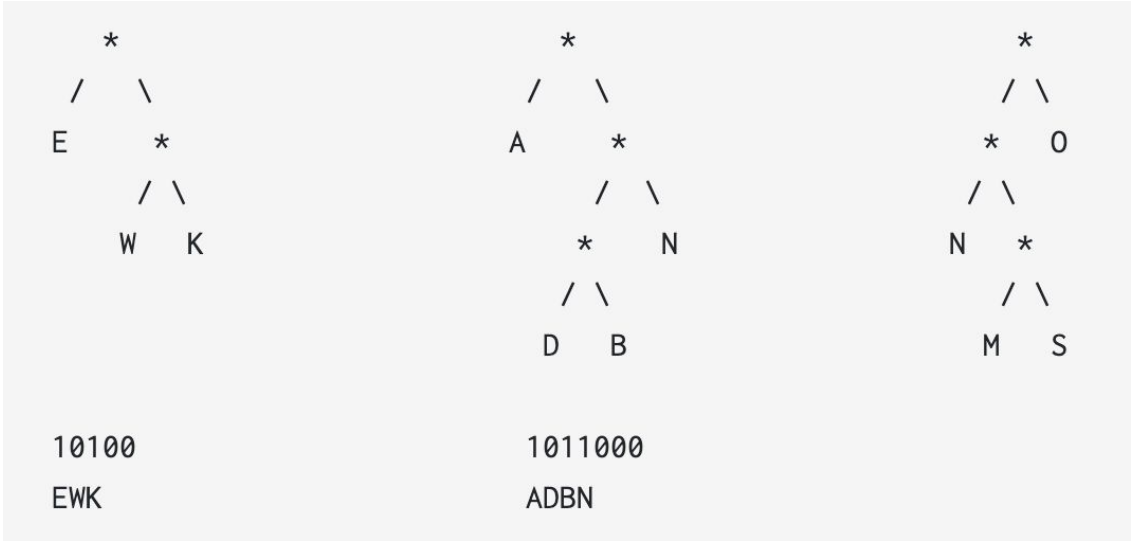


Represent all of this data…

… with only this much space!

Example: Data Compression

Using the abstraction: Huffman trees



Example: Data Compression

Building the abstraction: your C++ code

```
EncodedData compress(string messageText)
```

Roadmap

Roadmap

1. What is possible with tech and code? What isn't possible?

Roadmap

1. What is possible with tech and code? What isn't possible?
2. How can I use programming to solve problems that I otherwise would not be able to?

Roadmap

1. What is possible with tech and code? What isn't possible?
2. How can I use programming to solve problems that I otherwise would not be able to?
3. What makes for a "good" algorithm or data structure? Why?

Roadmap

Using Abstractions

Building Abstractions

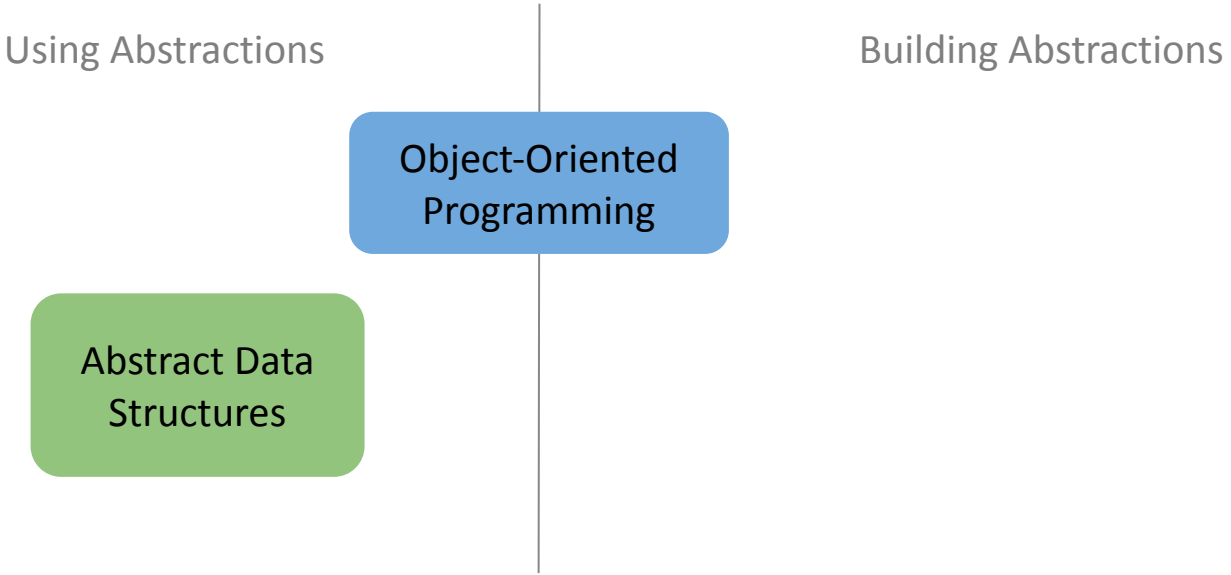
Roadmap

Using Abstractions

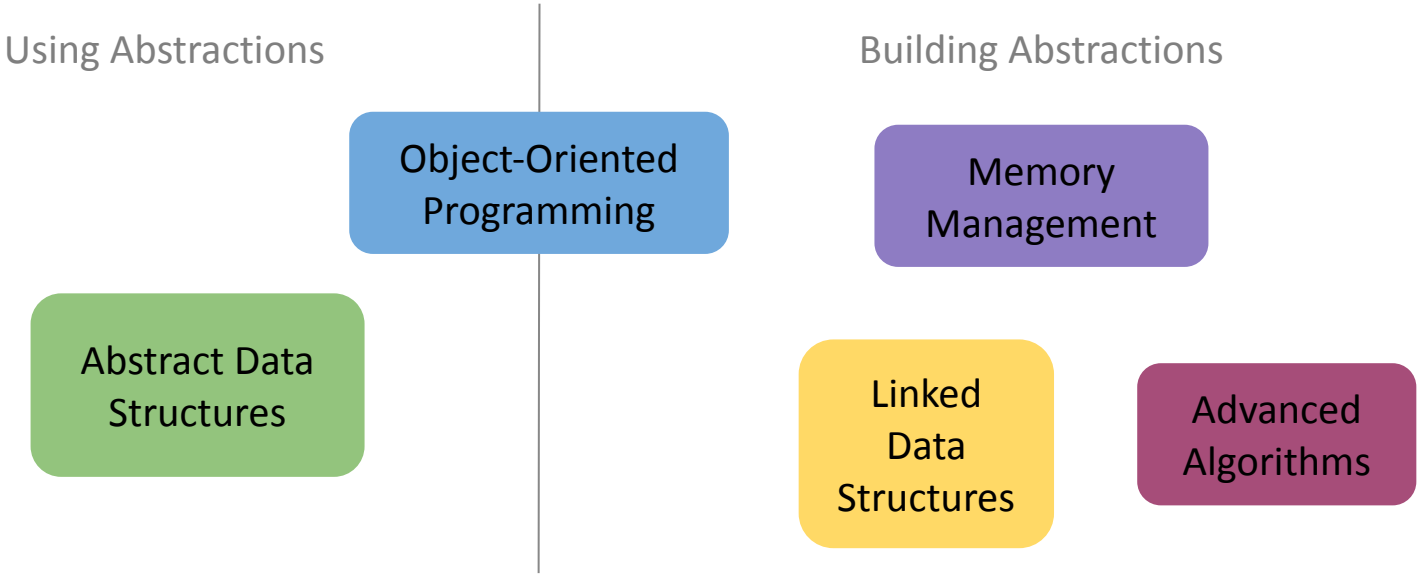
Building Abstractions

Abstract Data Structures

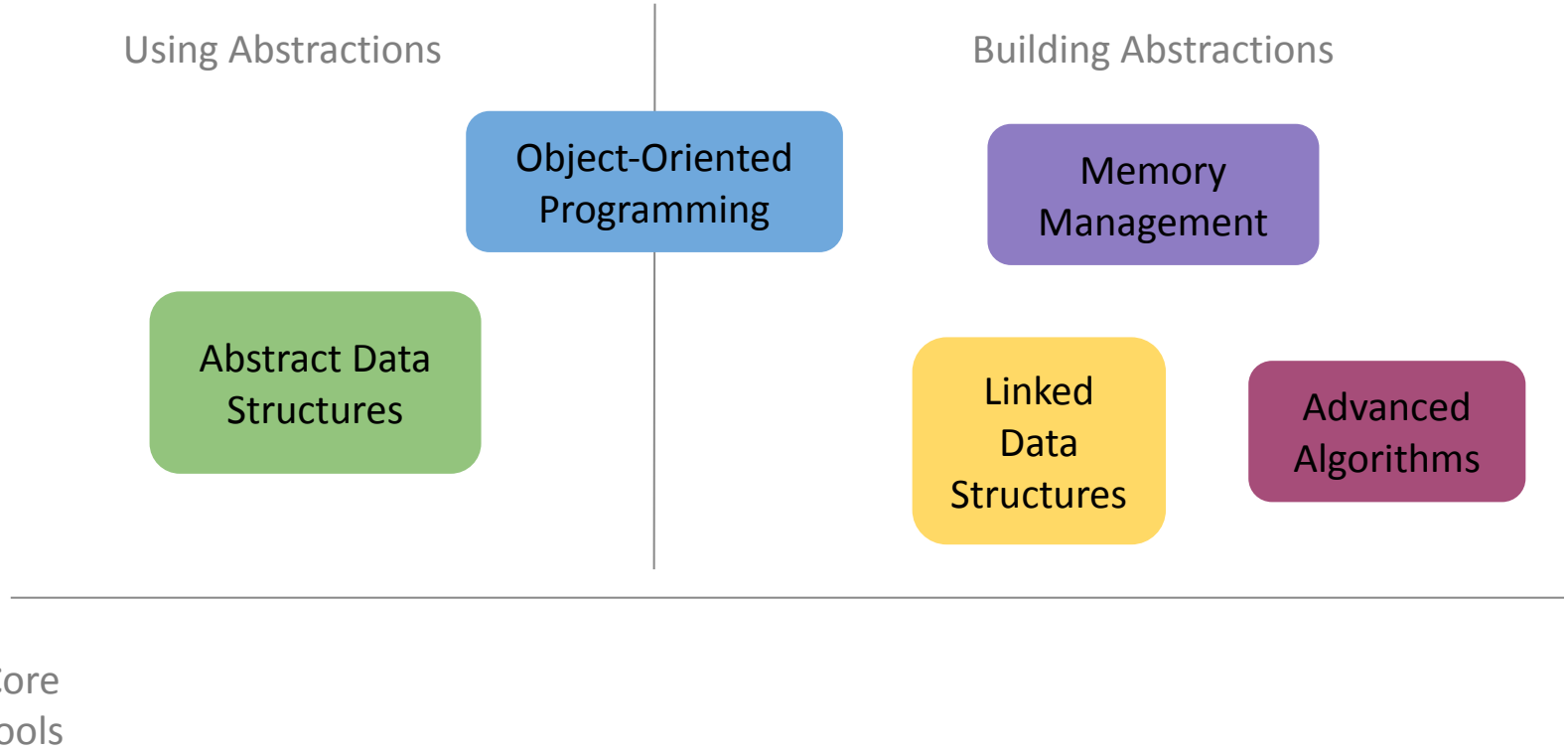
Roadmap


Using Abstractions
Building Abstractions
Object-Oriented Programming
Abstract Data Structures

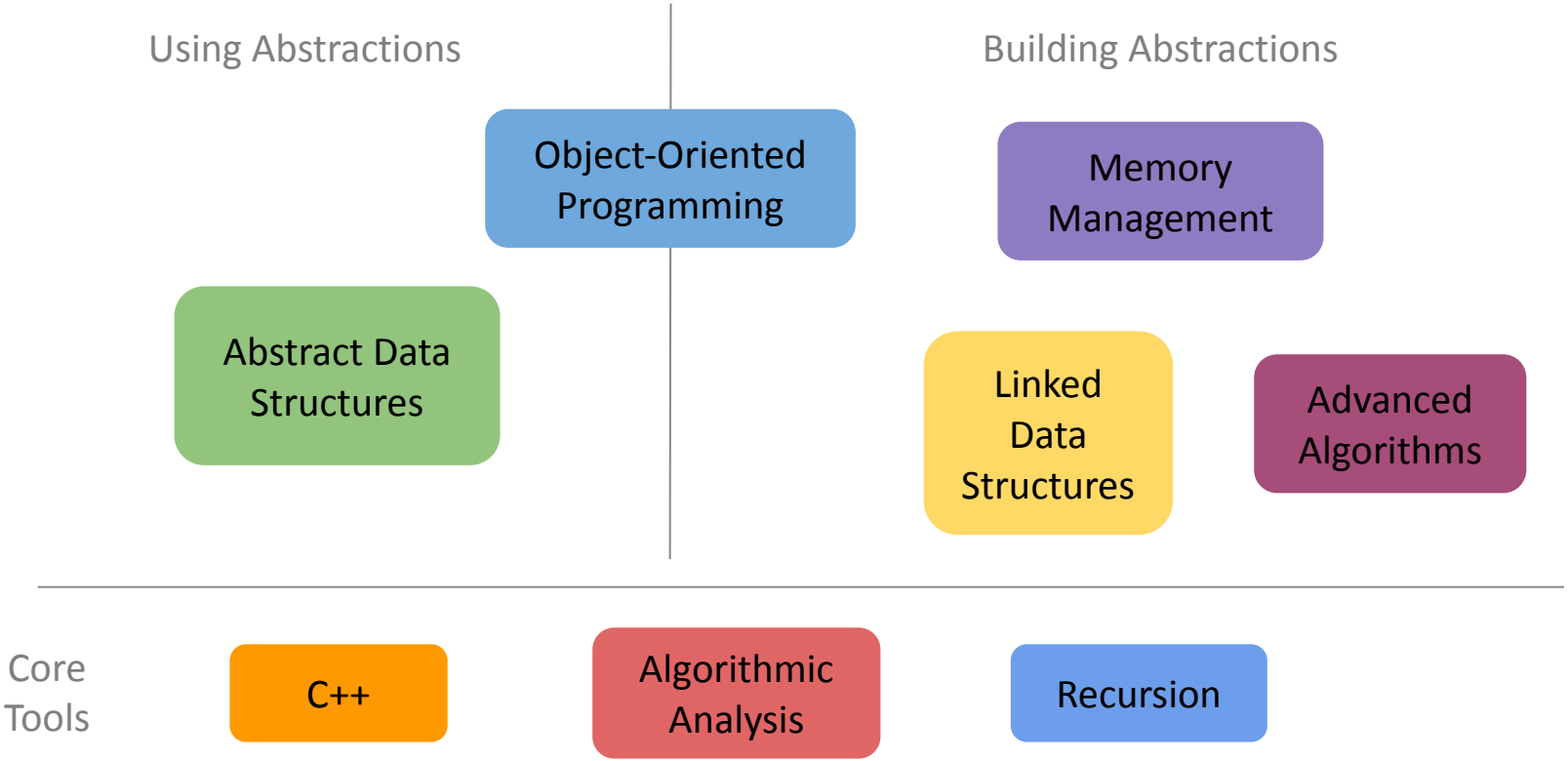
Roadmap


Using Abstractions
Building Abstractions
Object-Oriented Programming
Memory Management
Abstract Data Structures
Linked Data Structures
Advanced Algorithms

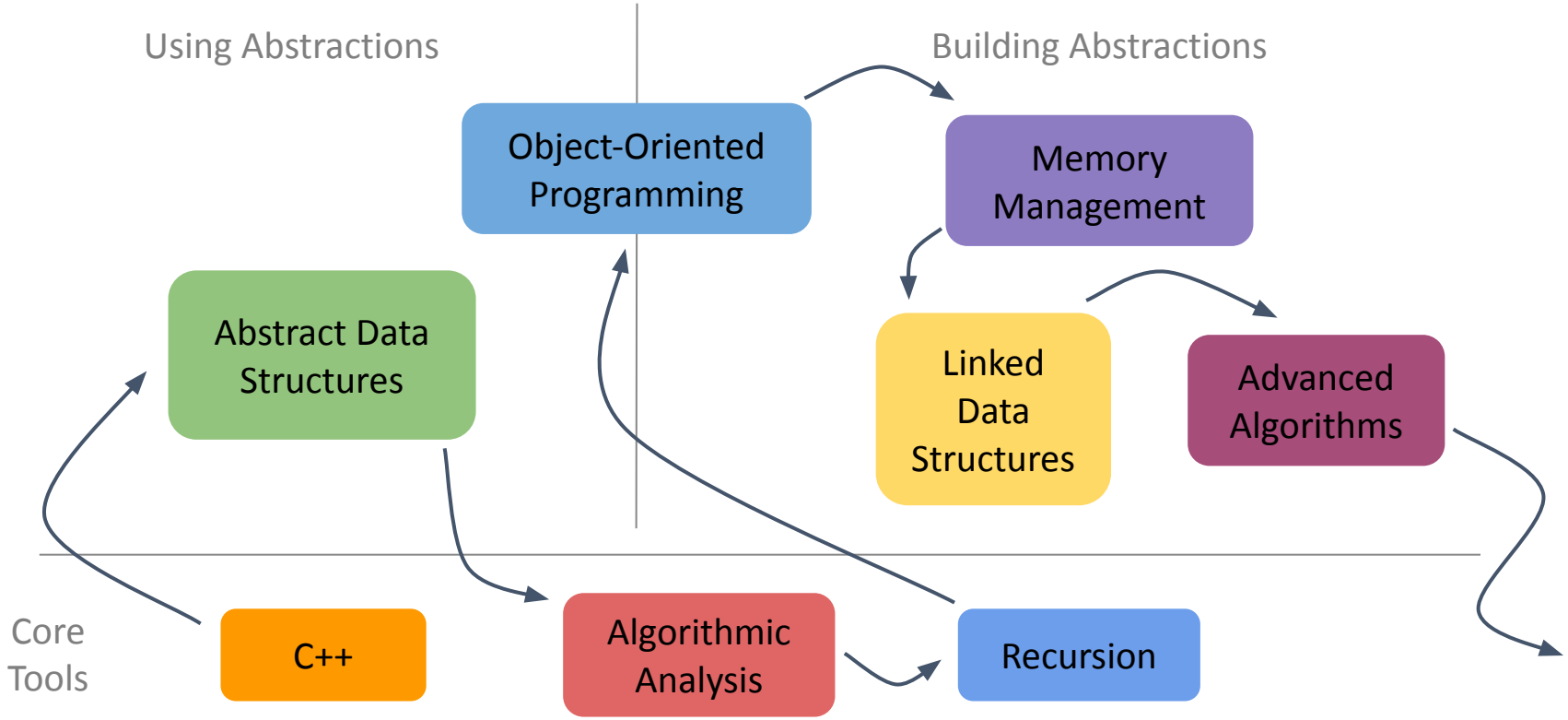
Roadmap


Using Abstractions
Building Abstractions
Object-Oriented Programming
Memory Management
Abstract Data Structures
Linked Data Structures
Advanced Algorithms
Core Tools

Roadmap


Using Abstractions
Building Abstractions
Object-Oriented Programming
Memory Management
Abstract Data Structures
Linked Data Structures
Advanced Algorithms
Core Tools
C++
Algorithmic Analysis
Recursion

Roadmap


Using Abstractions
Building Abstractions
Object-Oriented Programming
Memory Management
Abstract Data Structures
Linked Data Structures
Advanced Algorithms
Core Tools
C++
Algorithmic Analysis
Recursion

Class Norms

- Celebrate everyone's learning
  - Be kind
  - Don't shame others
  - You're not competing
  - Learning includes struggling
- Be actively engaged

Class Norms

You

- Celebrate everyone's learning
  - Be kind
  - Don't shame others
  - You're not competing
  - Learning includes struggling
- Be actively engaged

Us

Class Norms

You

- Celebrate everyone's learning
  - Be kind
  - Don't shame others
  - You're not competing
  - Learning includes struggling
- Be actively engaged

Us

- Prioritize your well-being
- Provide you with as many resources as possible
- Value your feedback

Course Mechanics

Course Website

https://cs106b.stanford.edu

And this URL will live on for the next several years, even after this quarter is over:

https://web.stanford.edu/class/archive/cs/cs106b/cs106b.1238

Prerequisites

- Should have either:
  - Taken CS106A
  - Taken AP CS (4 or 5 on exam)
  - Significant previous programming experience
- Take CS106A instead if you don't have much programming experience (this is where most students start!)
  - http://cs106a.stanford.edu
- Come talk to us if you're unsure!

Lecture and Attendance

- MTuWTh from 1:30-2:45pm in NVIDIA Auditorium
  - Most lectures will only be 60 mins, but could run over
- Lectures are recorded, but we don't want you to rely on that
- We want to encourage you to stay up-to-date with the lectures and come to class!
  - Opportunity to have 5% of your course grade (knocked off from final exam) come from lecture attendance
  - To get credit, you must either attend lecture or watch the recording before the next class, and fill out an attendance ticket on Gradescope
- Attendance will start next class (tomorrow)!

Sections and Participation

- Weekly 50-minute section led by one of our awesome SLs
  - Discuss and solve interesting problems
- Section signups open now on class webpage (not on Axess)
  - **Signups close Tuesday 5pm!**
  - Not first-come first-serve
- Attendance and participation are mandatory
  - ✓+ : Showed up to section on time, followed section norms, and participated in an engaged manner
  - ✓ : Showed up to section a little late and/or minimal participation
  - 0 : Did not show up to section
- Section starts this week!

Assignments

- 6 assignments
- Usually takes 10-20 hours/assignment
- Can be written problems, hands-on exercises with the tools, targeted coding tasks, and/or a larger complete program
- Due Fridays (Assns 0-2) or Wednesdays (Assns 3-6), 11:59pm PT
- Late Policy:
  - Due Date: small on-time bonus
  - Grace Period: automatic 24-hour extension for everyone, but no bonus
  - After Grace Period: deduct 15% per late day*, up to 5 days

*shit happens, so come talk to Amrita or Elyse about extensions in special circumstances

Assignment Grading

- Graded on functionality and style

+	Exceeds our expectations, “perfect”
✓+	Satisfies all the requirements, good functionality and style
✓	Meets the requirements, but perhaps with small problems
✓–	Has somewhat serious problems
–	Serious problems, does not show effort or understanding
0	Not submitted

- Interactive grading sessions with your SL
  - Get one-on-one feedback that’s more than just your grade
  - Mandatory!

Exams

- Midterm - Monday, July 17th from 7-9pm
  - Email us ASAP if you cannot make this
- Final - Friday, August 18th from 3:30-6:30pm
  - You MUST take the exam at this time
- In-person, closed-book, closed-note

Grading Scheme

55%	Assignments
10%	Midterm Exam
20-25%	Final Exam*
10%	Section participation
0-5%	Lecture Attendance*

* varies based on how much you complete lecture attendance ticket

Getting Help - LaIR and Office Hours

- LaIR - helper hours run by our section leaders
  - Sunday - Thursday, 5-9pm PT in Durand 353
  - Conceptual and debugging help
  - Starts this Tuesday (tomorrow!)
- Office Hours with Elyse and Amrita in Durand 303
  - Elyse: Mon 3-5pm (group), Thu 3-5pm (by appointment)
  - Amrita: Wed 3-5pm (group), Fri 10am-12pm (by appointment)
  - Group: conceptual help, high-level assignment help, and CS/life talk
  - Appointment: 15 minutes to talk about whatever you want!

Getting Help - Edstem

- Your go-to for online questions!
  - Questions about lecture - use the lecture megathread
  - Questions about assignments (conceptual or debugging)
    - We may tell you to come to LaIR
  - Questions about administrivia
- You should NEVER post assignment code in a public post

Getting Help - Friday Sessions

- Fridays from 1:30-2:45pm in NVIDIA Auditorium
- Conceptual review of the week's lectures and extra practice problems
- Completely optional, open to all students
- Led by two amazing SLs: Yasmine and Poojan!

Getting Help - YEAH Hours

- “Your Early Assignment Help” hours
  - Fridays from 3-4pm for assignments 1-3
  - Wednesday from 3-4pm for assignments 4-6
- Get started on assignments right after they’re released, with support from an SL
- Virtual (zoom link), will be recorded
- Completely optional, open to all students
- Led by an amazing SL: Bryant!

Stanford Honor Code

1. You cannot look at solutions that are not your own
    a. This includes AI-generated solutions
2. You cannot share your solutions with anyone but course staff
3. You should cite any non-staff collaboration in your submission
    a. This collaboration must still follow rules 1 and 2

We will run plagiarism detection software at the end of the quarter

Your First C++ Program: Hello World!

Starter Project - you'll need to complete Assignment 0 before you can run this code yourself :)

Announcements

- Assignment 0 (Installing Qt Creator) is due Friday at midnight
- Sign up for section - rank preferences by Tuesday 5pm!
  - Sections start this week, earliest ones are Wednesday
- Send OAE letters to Amrita and Elyse via email ASAP

Thank you!