LEC 19

CSE 123

Victory Lap & Next Steps

BEFORE WE START

Talk to your neighbors:

What was your favorite thing you learned about this quarter? Why?

Instructors: Brett Wortzman
Miya Natsuhara

TAs:

Arohan	Neha	Rushil	Johnathan	Nicholas
Sean	Hayden	Srihari	Benoit	Isayiah
Audrey	Chris	Andras	Jessica	Kavya
Cynthia	Shreya	Kieran	Rohan	Eeshani
Amy	Packard	Cora	Dixon	Nichole
Trien	Lawrence	Liza	Helena	

Music: CSE 123 25wi Lecture Tunes

Questions during Class?

Raise hand or send here

sli.do #cse123

Announcements

- C3 due tonight (3/14) at 11:59pm
- R7/R-Gumball due Sunday (3/16) at 11:59pm
  - Two forms to submit! Both on the Ed board.
  - R7 open to C3 if you need two extra days instead of another resub
  - R-Gumball open to all assignments w/ feedback released
- IPL closes end of day today (3/14)
  - Not open this weekend or next week
  - Message board will remain available
- **Final Exam Tuesday (3/18) @ 12:30pm-2:30pm in KNE 110/120**
  - Seating chart now posted on the course website!
  - Typical rules for quizzes, note sheet (8.5” x 11” double-sided, typed or handwritten)
- Please fill out course evaluation by Sunday night!

You Made It!

[Recap] Why 123?

1. To solve more complex problems by leveraging more complex programming structures / patterns

2. To better rationalize specific design decisions
   - How to "best" structure classes to reduce redundancy
   - Which ADT implementations are "most" appropriate to use

$\}$ **Be a better programmer**

3. To understand and critically analyze intersections between Computer Science and society
   - Search engines, algorithmic art, machine learning, etc.
   - Developing informed opinions on current issues

$\}$ **Be a better person**

[Recap] Topics Covered

- Advanced Object-Oriented Programming (OOP)
  - Inheritance, Polymorphism, Abstract classes
- Implementing Abstract Data Types (ADTs)
  - ArrayIntList (int[] elementData, int size)
  - LinkedIntList (ListNode front)
  - Java's ArrayList & LinkedList (int size, ListNode back)
- Runtime (Complexity & Big O notation)
- Recursion
  - Recursive definitions (n! = n * (n – 1)!)
  - (Implicit) Base and Recursive cases
  - Public / private pairs
  - LinkedLists w/ recursion (x = change(x))
- Binary Trees
  - Binary Search Trees (BST) & Runtime
- Exhaustive Search / Recursive Backtracking
  - Dead ends / Choose, explore Un-choose

Assessable content

- Machine Learning & Hashing

You've learned A LOT!!!

(hopefully)

Future Courses

Also: bringing computational thinking to other fields!

CSE Majors

Course	Overview
CSE 311	Mathematical foundations
CSE 351	Low-level computer organization/abstraction
CSE 331	Software design/implementation
CSE 340	Interaction Programming
CSE 341	Programming languages
CSE 344	Data Management (databases)

https://www.cs.washington.edu/academics/ugrad/current-students

- Tons of options for everyone!
  - Self study always valid too!

Non-CSE Majors

Course	Overview
CSE 154	Intro to web programming
CSE 163	Intermediate programming, data analysis
CSE 180	Introduction to data science
CSE 373	Data structures and algorithms
CSE 374	Low-level programming and tools
CSE 412	Data Visualization
CSE 416	Intro. to Machine Learning

https://www.cs.washington.edu/academics/ugrad/nonmajor-options/nonmajor-courses

Applications of CS

or "What can I do with what I learned?"

- Detect and prevent toxicity online
- Digitize basketball players
- Help DHH people identify sounds
- Figure out how to best distribute relief funds
- Recognize disinformation online
- Make movies
- Improve digital collaboration
- Fix Olympic badminton
- And so much more!

Future Projects

- At this point, you know 90% of the fundamentals you need to accomplish practically any project
    - Hurdle will typically be learning the syntax of a new language, using GitHub, importing external libraries, etc.

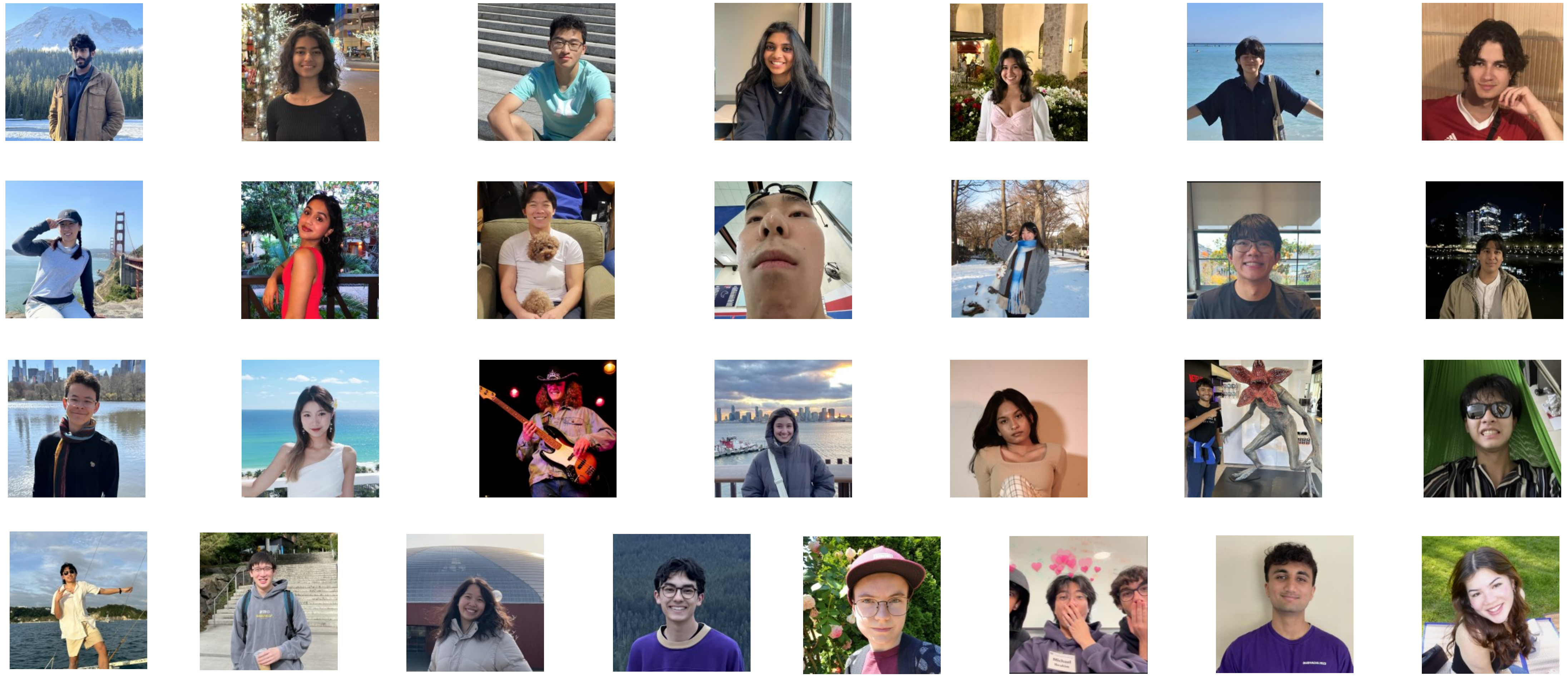
- Some ideas:
    - Make a Minecraft mod! (Java) [link]
    - Make a Discord bot! (Python) [link]
    - Make personal website! (HTML, CSS, Javascript) [link]
    - Convert a project from this course into a more user-friendly application
        - C1, make a Graphical User Interface (GUI) [link]
        - P3, refine the Email class until you get an accuracy you're happy with
    - Really, anything you want!!!*

*Warning: it's often hard to tell what computers can do easily and what they struggle with…

Frequently Asked Questions

- How can I get better at programming?
  - Practice!
- How can I learn to X?
  - Search online, read books, look at examples
  - Start with something that already works (try github), then make changes!
- What should I work on next?
  - Anything you can think of! (See previous slide for some ideas)
- Should I learn another language? Which one?
  - Depends on what you want to do!
    - Python: Data Science & Machine Learning
    - JavaScript: Web Dev
    - C / C++: Systems Programming
- What's the best programming language?
  - 😠 (take CSE 341/CSE 413)

Thank your TAs!



Thank You!

- Thank you for participating, asking questions, engaging with course materials & resources!
  - And thank you for the feedback if you filled out the course evaluation :)
- Thank your amazing TAs!
- Any final questions before we wrap?