

LEC 15

CSE 123

Victory Lap & Next Steps

Questions during Class?
Raise hand or send here

sli.do #cse123



BEFORE WE START

Talk to your neighbors:

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

Respond on sli.do!

Instructor: Trien Vuong

TAs:	Cora	Nhan
	Jonah	Nichole
	Maitreyi	Shiven

Music: [CSE 123 26su Lecture Tunes](#)

Announcements

- C3 due tonight **Friday** (8/21) at 11:59pm
- R6/R-Trien due Friday (8/21) at 11:59pm
 - Eligible assignments for R6: C1, C2, P2
 - R-Trien open to all assignments w/ feedback released
- IPL closes tomorrow (8/20)
- **Final Exam tomorrow, Thursday (8/20) in your assigned section. And on Friday (8/21) during lecture time (12:00 – 1:00 PM) in AND 205.**
 - You must bring your ID in order to receive your exam.
 - Typical rules for quizzes, note sheet (8.5" x 11" double-sided, typed or handwritten)
- Please fill out course evaluation by Sunday night!
- Please fill out TA evaluations 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
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 programmer



Be a better person

[Recap] Topics Covered

- Implementing Abstract Data Types (ADTs)
 - ArrayList (int[] elementData, int size)
 - LinkedList (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
- Binary Trees
 - Binary Search Trees (BST) & Runtime
- Advanced Object-Oriented Programming (OOP)
 - Inheritance, Polymorphism, Abstract classes
- Exhaustive Search / Recursive Backtracking
 - Dead ends / Choose, explore Un-choose

Assessable
content

**You've
learned
A LOT!!!**
(hopefully)

Future Courses

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!

Also: bringing computational thinking to other fields!

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!

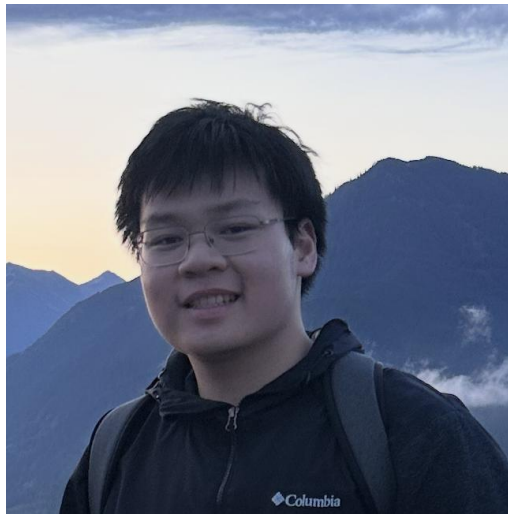
Future Projects

- 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](#)]
 - Really, anything you want!!!*

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?