UNIVERSITY *of* WASHINGTON

LEC 19

CSE 122

Victory Lap & Next Steps

Questions during Class?

Raise hand or send here

sli.do   #cse122

BEFORE WE START

Talk to your neighbors:
What was your favorite part of this quarter?
Doesn't have to be about CSE 122 ☺

(Put AMA questions in sli.do!)

Instructors   **Tristan Huber & Hunter Schafer**

TAs
Ambika	**Evelyn**	**Poojitha**
Andrew	**Jacob**	**Rishi**
Audrey	**Jaylyn**	**Rucha**
Autumn	**Jin**	**Shivani**
Ayush	**Joe**	**Shreya**
Ben	**Kevin**	**Steven**
Colton	**Leon**	**Suhani**
Di	**Megana**	**Yijia**
Eesha	**Melissa**	**Ziao**
Elizabeth	**Mia**	

Announcements

- Lots of information posted on Wednesday!
    - C3 deadline tonight (Fri, June 2)
    - Resub 7 now open (due 11:59pm on Tues 6/6)
    - Bob Bandes TA Award
    - Course Evaluations
    - IPL closes after today (6pm)
    - Review Session Monday (June 5)
    - Cheddar visit on Monday (June 5)

You Made It!

Lecture Outline

- **Looking Back** ◀

- Looking Forward

- Thank You!

CSE 121 (or CS1) vs. CSE 122

CSE 121 / AP CS / CS1 or **Other Programming Experience**

- Print statements
- Data types (int, String, boolean)
- Methods / Functions
  - Parameters
  - Returns
- Control structures
  - Loops
  - Conditionals
- File I/O
- Arrays
- **Computational Thinking** (language agnostic)

CSE 122 – Computer Programming II

- Decomposing large problems into smaller, manageable, subproblems
- Using data structures
  - List
  - Stacks / Queues
  - Sets
  - Maps
  - 2D Arrays
- Object Oriented Programming
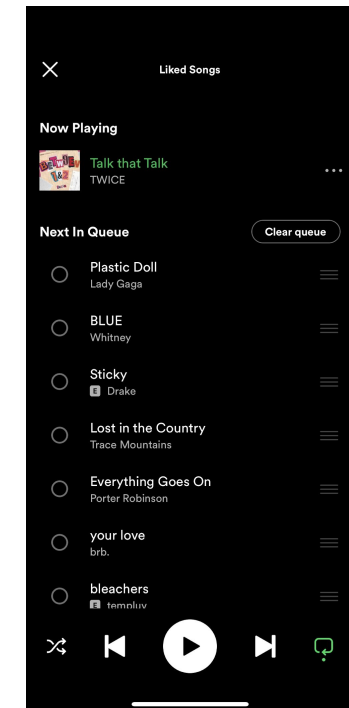 - Interfaces
  - Separation of Concerns

Why 122?

1. Build a strong foundation of data structures that will let you tackle the biggest problems in computing

122 Data Structures
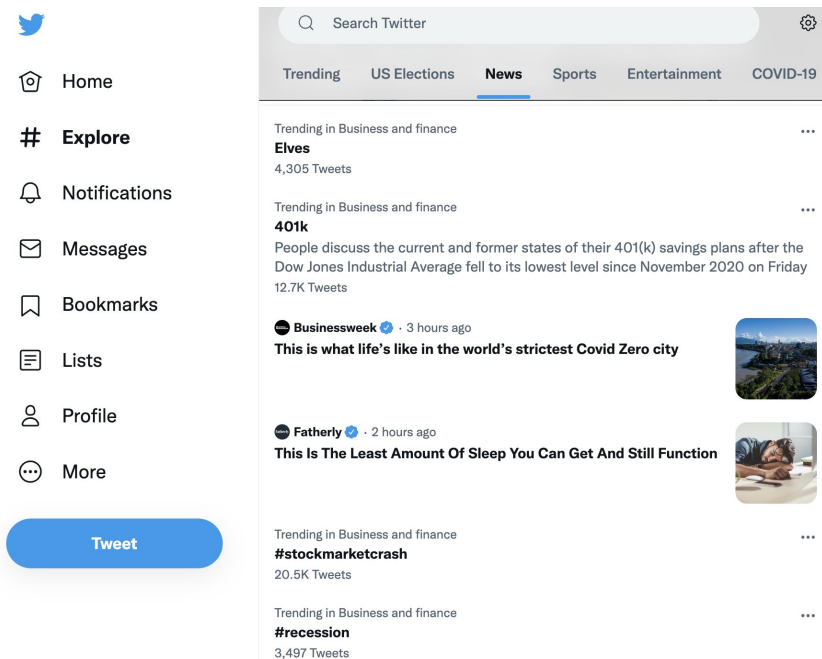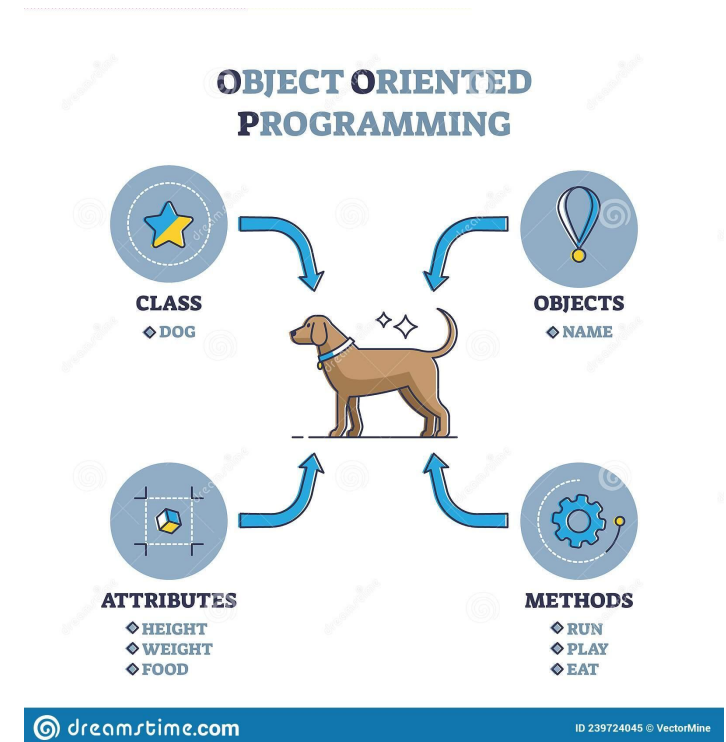
Source: Twitter 9/23
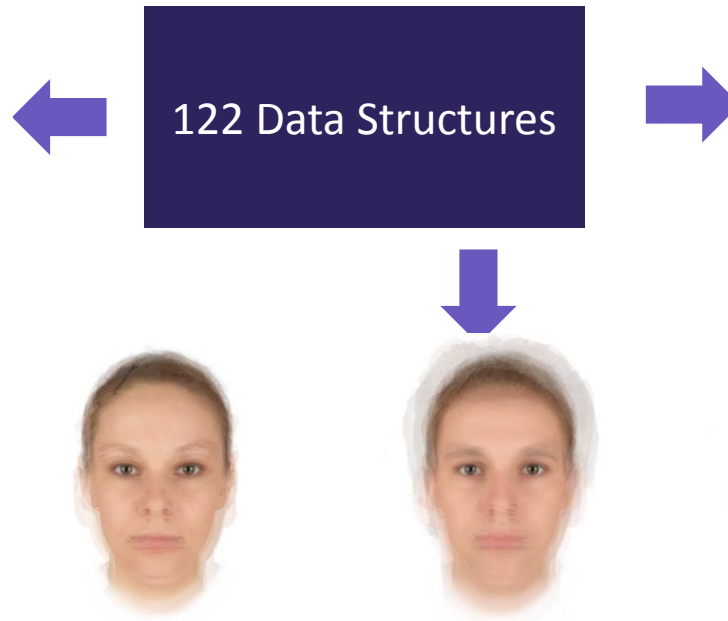
Source: Ethical CS

Source: Hunter's Spotify

Why 122?

2. Learn an important structural pattern for representing **objects** in code to make our code more **reusable** and **maintainable** and **easier to understand.**

- Java is designed around this idea of objects. We haven't been leveraging that yet!

- Used in almost every real-world software project.

OBJECT ORIENTED PROGRAMMING

CLASS
◇ DOG

OBJECTS
◇ NAME

ATTRIBUTES
◇ HEIGHT
◇ WEIGHT
◇ FOOD

METHODS
◇ RUN
◇ PLAY
◇ EAT

dreamstime.com   ID 239724045 © VectorMine

Review So Far

CS Concepts
- Abstraction
- Debugging
- Client/Implementer
- Object Oriented Programming
- Encapsulation
- Testing
- Third Party Libraries

Java Language
- Intro to Java (e.g., File Processing)
- Iterators and For-each Loops
- Exceptions
- Interfaces
- References
- JUnit*

Data Structures
- Lists
- Stacks
- Queues
- 2D Arrays
- Sets
- Maps

Java Collections
- Arrays / 2D Arrays
- ArrayList
- LinkedList
- Stack
- TreeSet / TreeMap
- HashSet / HashMap
- Interfaces for Java Collections

Lecture Outline

- Looking Back

- **Looking Forward** ◀

- Thank You!

What Can Come Next?

- Some ideas
  - Work on a project
  - Learn a new language
  - Learn a new library
  - Take more courses
  - Explore CS beyond programming

- The general idea though is… whatever you want!
  - You've learned an extremely powerful set of skills, use it on what you are most interested in pursuing!

What Project?

- Add a Graphical User Interface (GUI) to an assignment

- Automate some boring tasks in your life
    - Maybe even automate writing code with good style?

- Organize and process data from your life (favorite quotes, your calendar, etc.)

- What are you currently doing that a computer could do?

- List of some project ideas (UW CSE alum)

What Language?

- Expanding your Java knowledge with a project is valuable. Or use a project to learn a new language!

- Pick a project, see what similar projects use! No wrong language to learn, certain tasks favor certain languages
  - iOS: Swift
  - Android: Java, Kotlin
  - Client-side web: Javascript (many frameworks to choose from)
  - Beautiful visuals: Processing
  - Data Processing + Machine Learning: Python
  - Data Management: SQL
  - Embedded systems: C / C++ / Rust

- Learn a new programming paradigm
  - Functional languages: Racket, Haskell, Scala, (now, Java 8!)

What Library?

Here are just a FEW examples. There is so much more!

- Human language (aka "natural language processing")
    - http://nlp.stanford.edu/software/

- Building games
    - http://lwjgl.org/
    - http://jbox2d.org/ (with physics!)

- Processing biological data
    - http://biojava.org/wiki/Main_Page

- Accessing Facebook data
    - http://restfb.com/

- Make a website backed by Java
    - https://www.jetbrains.com/help/idea/your-first-spring-application.html

What Classes?

- CSE 123 is the most common next class. Continue the story, learn how data structures are implemented

- Other courses
  - CSE 154: Web Programming (HTML/CSS/Javascript)
  - CSE 163: Intermediate Data Programming (Python)

- *Large* set of CSE courses for *both* Allen School majors and students from all over UW campus. Many exciting courses, many (but not all) require CSE 123.
  - Allen School Majors
  - All UW Students

- Courses in Tech Related Majors: INFO, AMATH, HCDE, DXARTS, …

What is CSE?

The changing nature of the field: From smaller/faster/cheaper to tackling societal challenges
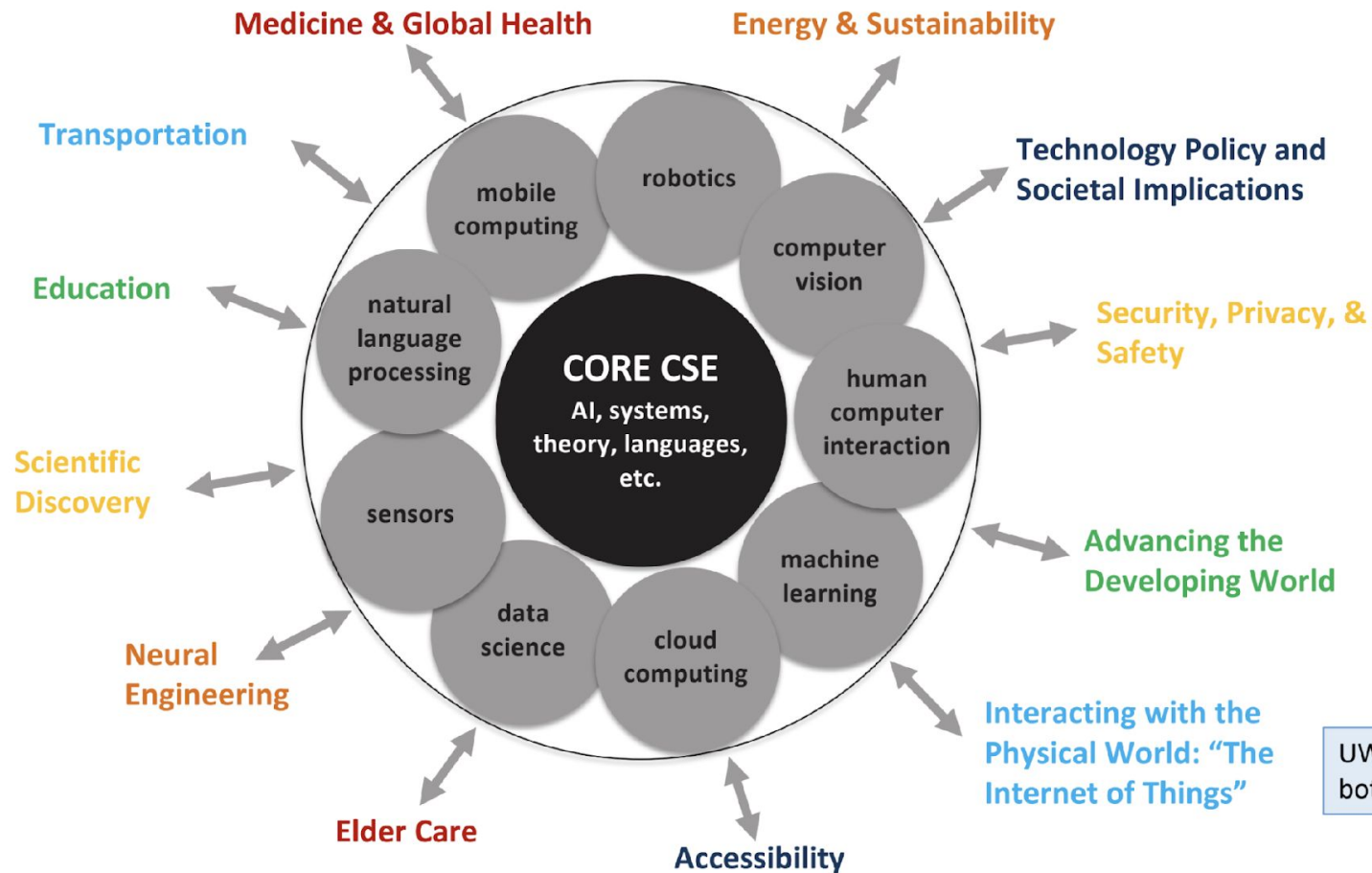
UW has led this modern view of the field – both in concept and in implementation

Research Beyond Programming

## Learn a new CS Topic

- Investigate how to best distribute relief funds

- Digitize basketball players

- Help deaf/hard-of-hearing people identify sounds

- Detect and prevent toxicity online

- Recognize disinformation online

- Make movies

- Improve digital collaboration

- Design algorithms that are more fair and better respect privacy

- Fix Olympic badminton

- And so much more!

Attend Weekly Meetings

- Change – technologies for low-income regions

- Dub – human-computer interaction and design

- ComputingEd@UW – computer science education

Read a Book! (links on pictures)
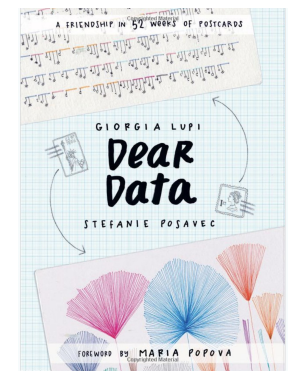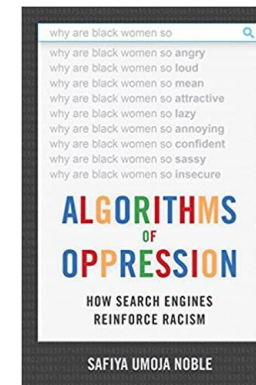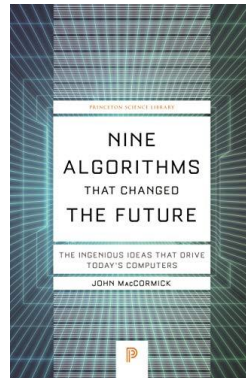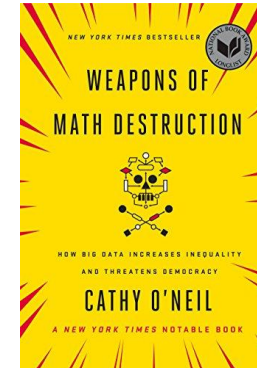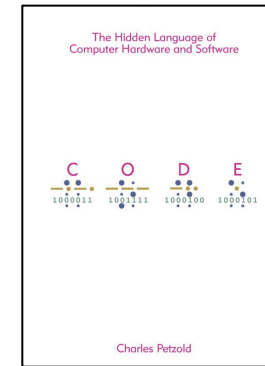
Lecture Outline

- Looking Back

- Looking Forward

  - Demo: Web Programming & Java

- **Thank You!** ◀

Thank You! (Students)

- This is still a very new course! We are always looking for feedback on how to improve the class for you and for future students! Thank you for your patience and understanding as we develop everything. ☺️
  - We *really* value your feedback!
  - Let us know what's working and what isn't working for you
  - Something that went well in another course? Tell us about it!
- **Please fill out the Course Evaluation by Sunday 6/4 at 11:59 pm** to provide feedback about the course!

Thank You! (TAs)

This course wouldn't work without your amazing TAs! Thanks to them for running the course!
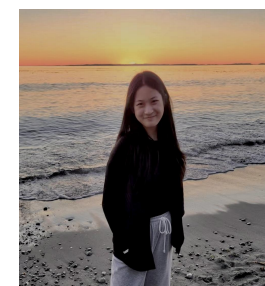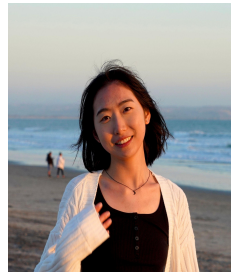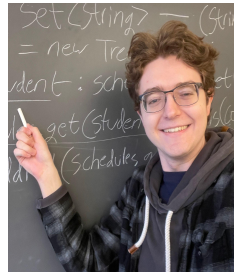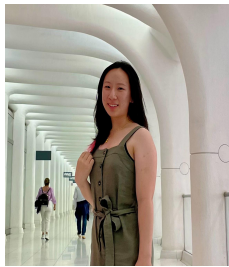
Ask Us Anything
AMA

sli.do   #cse122