LEC 19

CSE 121

Victory Lap!

BEFORE WE START

Talk to your neighbors:

What are your winter break plans?

Music: 🎷 CSE 121 25au Lecture Tunes 🎶

Instructors: Brett Wortzman & James Weichert

TAs:

Trey	Ava	Caleb	Elden	Anya
Amogh	Reese	Anum	Suyash	Minh
Samrutha	Hayden	Abdul	Sthiti	TJ
Dalton	Aki	Janvi	Paul	Zach
Ailsa	Spencer	Navya	Shayna	Cayden
Ryan	Savannah	Sam	Jesse	Johnathan
Anant	Tamsyn	Jessica	Nhan	

Questions during Class?

Raise hand or send here

sli.do #cse121

Announcements, Reminders

- R7 (and R-Extra) due **Wednesday, December 10th**
  - all assignments are eligible for resubmission!
- Today is the last day for IPL and instructor office hours
- Final Exam: **Wednesday, December 10th from 12:30 - 2:20pm**
  - review the Exam page of website (with policies & resources)
- TA-led review session: **Monday, December 8th from 4:30-7:00pm in ARC 147**

Evaluations and Awards

Please give us feedback!

- **Course Evaluations** are due **Sunday, December 7th at 11:59 PM**
  - Lecture A Eval
  - Lecture B Eval
- **TA Evaluations** are *also* due **Sunday, December 7th at 11:59 PM**

Bob Bandes TA Award nominations open!

- thought your TA was goated? write them a nomination!
- fun fact: some of our faculty won the award when *they* were TAs!

You Made It!

Applications of Computer Science

or "What can I do with what I learned?" – outside of just "write code":

- Help deaf & hard-of-hearing people identify sounds
- Develop a programming language that celebrates the world's languages
- Build battery-free robots & put them on insects (and... track murder hornets?)
- Detect and prevent toxicity online & recognize disinformation
- Computational knitting & carpentry
- Create an interactive atlas of millions of refugee experiences
- Fix Olympic badminton & identify cheating in chess
- and so much more!

… including our assignments! (1/2)

- Computational Biology & Medicine (P2, P3)
  - in CSE: Chris Thachuk, Linda Shapiro, Sara Mostafavi, Su-In Lee, Luis Ceze
- Computational Art (C0, C1)
  - UW CSE has many unique intersections of CS + art!
  - "Cultural-Centric Computational Embroidery" (CSE + iSchool)
  - "Computational Illusion Knitting", "How to Knit Objects Weird"
  - "WasteBanned: Supporting zero waste fashion design"

… including our assignments! (2/2)

- Games & Graphics (C1, C3)
  - at UW: many labs in CSE and iSchool's GAMER group
  - fun fact: Foldit is a crowd-sourced game for protein folding
    - David Baker shared this year's Nobel Prize in Chemistry, in part for this!!
- Social Computing (P1)
  - at UW: Amy Zhang's Social Futures Lab + so much of iSchool
- Computer Security (C2)
  - in CSE: Franzi Roesner, David Kohlbrenner, Nirvan Tyagi
- and many side quests (in lecture, section, PCM): accessibility (e.g. UW CREATE), weather forecasting, chatbots, and lots of math

Closing the Loop (1/5)

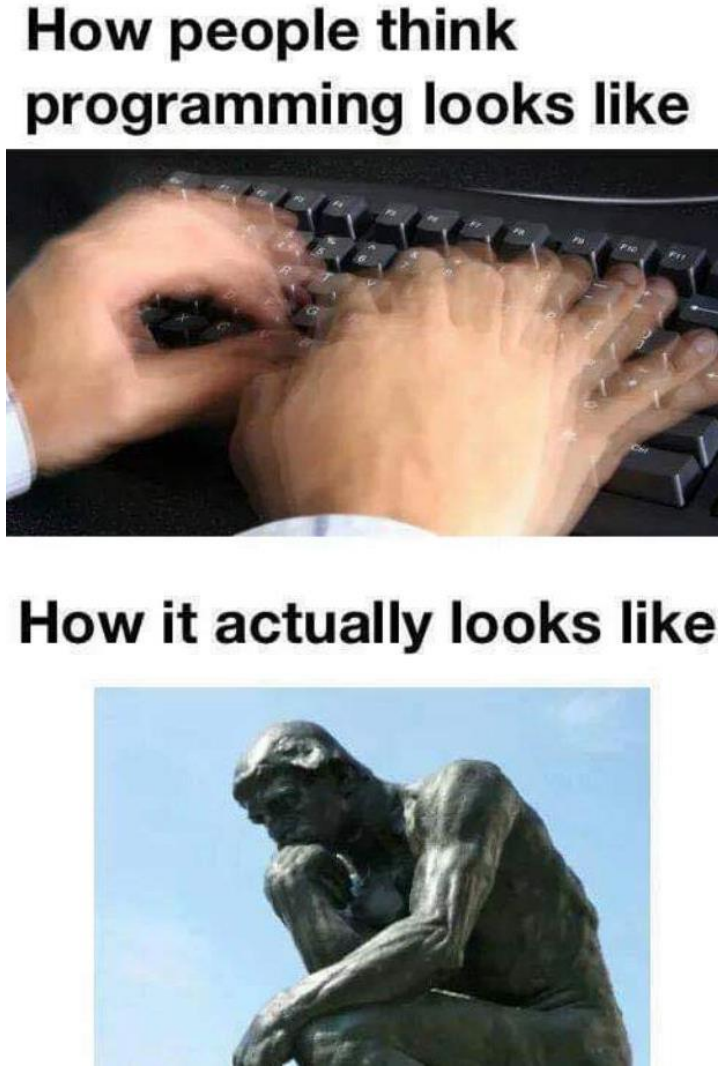
Other applications and intersections of computer science:

- **Economics and Finance**
  - predicting stock market trends
- **Environmental and Climate Science**
  - forecasting climate change
  - impact of energy use and e-waste from computers
- **Biology and Medicine**
  - detecting cancer

Closing the Loop (2/5)

Expectations vs. reality of CS

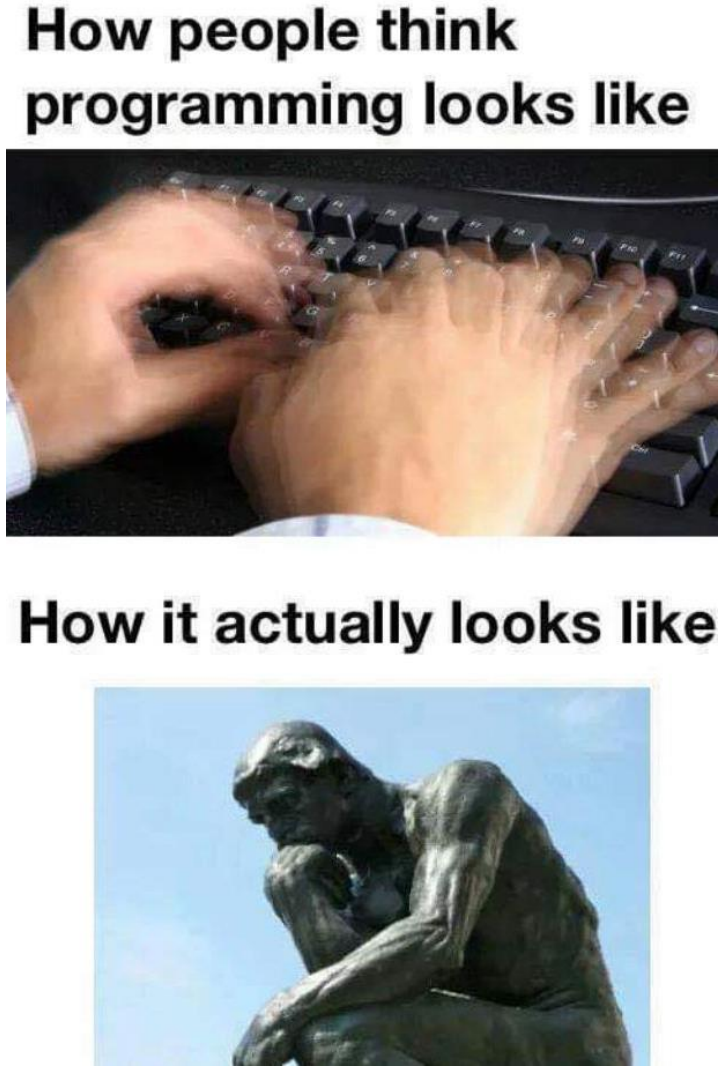
- **Not (just) writing and debugging code**
  - "Now I see [coding] more as step-by-step problem solving, breaking tasks into smaller parts, testing, and writing clear code"
  - "coding is a little like creating art, you get to implement your own creativity"
  - "[CS] is also about thinking clearly and breaking big problems into small steps"



Closing the Loop (3/5)

Expectations vs. reality of CS

- **Who codes (and uses code):**
  - "computer science felt kind of intimidating and 'only for tech people.' Now I see it more as step-by-step problem solving"
  - "I have now learned and realized that the concepts and skills that I have learned in this class has helped me better understand my own field that I am interested in"



Closing the Loop (4/5)

Reflecting on your journey (so far) learning CS!

- **Some good surprises…**
  - “I came into this course not expecting too much, but as the quarter ends, I'm realizing just how much I've learned, and I'm pretty proud of myself”
  - “it taught me a lot about persistence and opened my mind to the different possibilities”
  - “This course was a lot more fun than I thought it'd be”
  - This course gives me a new perspective on coding. That was amazing”
- **…and challenged expectations**
  - “Computer science is much harder than I thought it would be.”

Closing the Loop (5/5)

Some responses that made us chuckle:

- “I should check my code and output more carefully in each homework, reduce the resubmission times.”
- “Honestly, I really hated coding and CS in general since both of my parents are software engineers and they pushed it on me. This course took my opinion and switched it completely”
- “Before this course, I was the #1 coding hater…even though I still don’t see coding as my strongest skill, I’ve become more comfortable and less afraid of it.”

Future Courses

or "What can I do next?"

Non-majors

Course	Overview
CSE 154	Intro to web programming (several languages)
CSE 160	Intro programming, data analysis (Python)
CSE 163	Intermediate programming, data analysis (Python)
CSE 180	Introduction to data science (Python)
CSE 373	Data structures and algorithms (in Java)
CSE 374	Low-level programming and tools (C/C++)
CSE 412	Intro to Data Visualization
CSE 416	Intro. to Machine Learning
CSE 493E	Accessibility

More 12X!

Course	Overview
CSE 122	Data structures, object-oriented programming
CSE 123	More OOP, recursion

Majors

Course	Overview
CSE 311	Mathematical foundations
CSE 331	Software design/implementation
CSE 340	Interaction programming (mobile apps)
CSE 341	Programming languages
CSE 351	Hardware / Software Interface
CSE 480	Social impacts of computing

Related majors: Informatics, ACMS, HCDE, ECE, …

See: https://www.cs.washington.edu/academics/ugrad/current-students and https://www.cs.washington.edu/academics/ugrad/nonmajor-options/nonmajor-courses

Generalizing *beyond* Computer Science

Some of you said, ”I’m glad I took this class, but no more CS for me”

That’s totally valid!

Some lessons from this class that *could* apply more broadly:

- how to break big problems into smaller subproblems
- how to isolate what part of a system is broken
- attention to detail
- understanding basics of how software works
- how to learn (and reflect) effectively

Frequently Asked Questions

How can I get better at programming?

- Practice!

How can I learn to X?

- Classes, books, videos, or self-learn!
- CS (as a field) has lots of free resources :)

What should I do next?

- Anything you're interested in!
- but: hard to tell what's easy and what's hard

Should I learn another language? Which one?

- That depends – what do you want to do?

Thank your **fabulous** TAs!

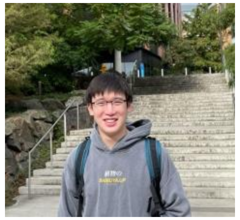










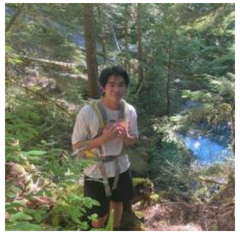













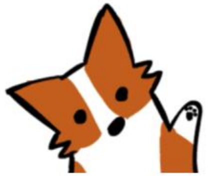












Thank you!

Ask Us (Almost) Anything!

sli.do #cse121