CSE 121 Lesson 19:

Final Exam Review & Victory Lap!

Matt Wang

Spring 2024

TAs: Andy Anju Archit Arkita Autumn Christian

Hannah H Hannah S Heather Hibbah Janvi Jessie

Jonus Julia Luke Maria Mia Ritesh

Shayna Simon Trey Vidhi Vivian Gumball?

sli.do #cse121-19

Today's playlist:
CSE 121 lecture beats 24sp

Announcements, Reminders

- R7 due June 6th – all assignments eligible!
- today is the last day for IPL + instructor office hours
- Final Exam: **Wednesday, June 5th from 2:30-4:20 in KNE 120**
- TA-led Review Session: **Monday, June 3rd from 4:30-6:50 in JHN 102**
- other exam reminders:
  - please look at seating charts and let me know ASAP if you're not there!
  - review Exam Resource Bank!

Evaluations and Awards

- Course Evals are due **Sunday, June 2nd at 11:59 PM**
  - currently just 16% (36/219) – we can do better than that :’)
- CERSE survey – please see Dan Grossman’s email!
  - This is a <u>different</u> kind of feedback from course evals
- Bob Bandes TA Award nominations open!

Applications of CS

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

- *Detect and prevent toxicity online* & *recognize disinformation*
- *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 muder hornets?)*
- *Computational knitting* & *carpentry*
- *Create an interactive atlas of millions of refugee experiences*
- *Fix Olympic badminton* & *identify cheating in chess*
- *and so much more!*

… including your projects! (1/2)

- Computational Biology & Medicine (P2, P3)
  - fun fact: Matt did some DNA sequencing (P3+++) in grad school at UCLA!
  - at UW: Chris Thachuk, Linda Shapiro, Sara Mostafavi, Sui-In Lee; BIME & Med!
- Turtle (C0, C1)
  - fun fact: maps well to stitching & embroidery or laser cutting!
  - at UW: “Cultural-Centric Computational Embroidery”, CSE + iSchool, SIGCSE ’24

… including your projects! (2/2)

- Games & Graphics (C1, C3):
  - fun fact: Foldit (from UW) is a crowd-sourced game for protein folding!
  - at UW: many labs in CSE and iSchool’s GAMER group
- Social Computing (P1, C2):
  - at UW: Amy Zhang’s Social Futures Lab + so much of iSchool!
- and many side quests (in lecture, section, PCM): accessibility (e.g. UW CREATE), weather forecasting, chatbots, software tools, and lots of math

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

More 12X!

Course	Overview
CSE 122	Introduction to Computer Programming II
CSE 123	Introduction to Computer Programming III

Majors

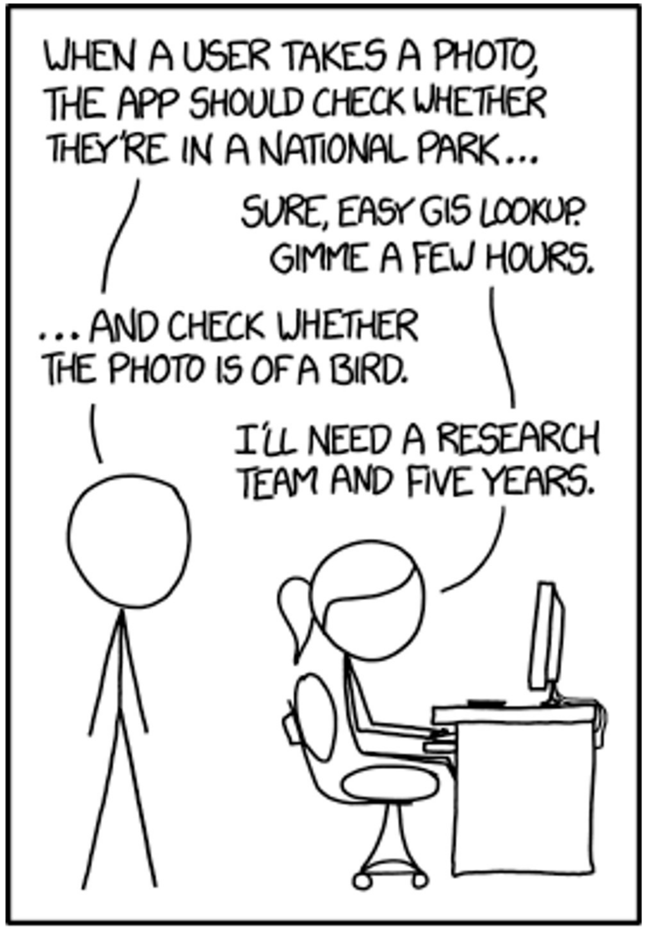
Course	Overview
CSE 311	Mathematical foundations
CSE 331	Software design/implementation
CSE 340	Interaction programming (mobile apps)
CSE 341	Programming languages (!!)
CSE 351	Low-level computer organization/abstraction

Other tech-related majors:
Informatics, ACMS, HCDE, Electrical & Computer Engineering, …

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

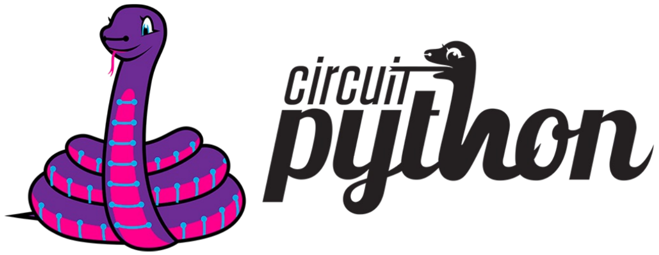
Frequently Asked Questions

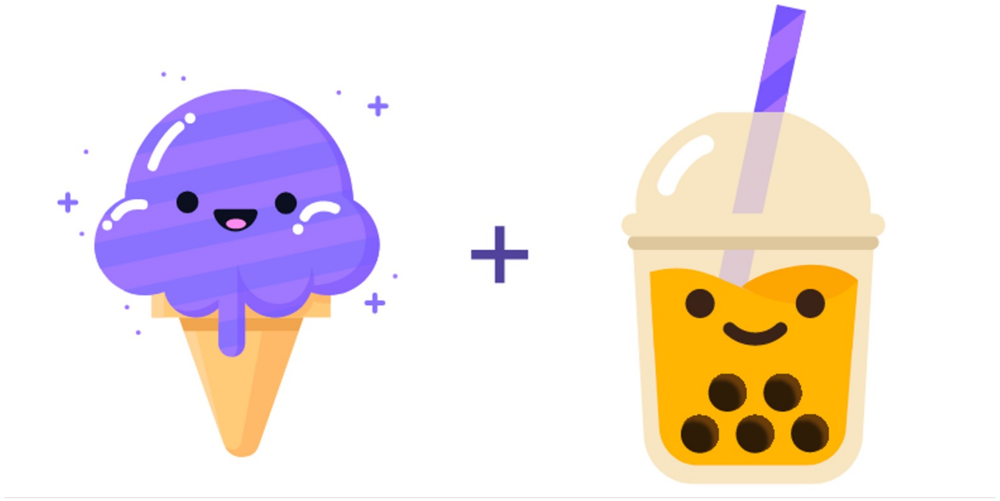
- *How can I get better at programming?*
  - Practice!
- *How can I learn to X?*
  - Search online, read books, look at examples :)
- *What should I work on next?*
  - Anything you can think of! (Here are some ideas)
  - Beware: it's hard to tell what's easy and what's hard.
- *Should I learn another language? Which one?*
  - That depends–what do you want to do?
- *What's the best programming language?*
  - 😠 (take CSE 341 or CSE 413)

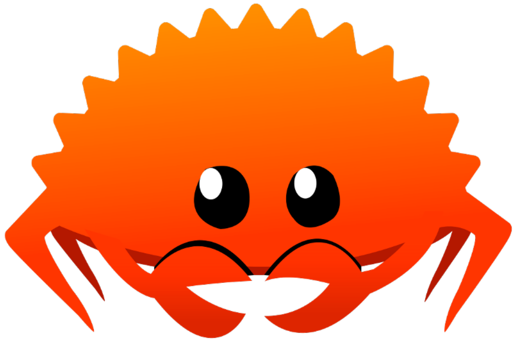

WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Aside: Cute Programming Language Logos







Summer Project: Tic Tac Toe (1/2)

Build your own Tic Tac Toe game (and “AI”)!

1. How would you represent a Tic Tac Toe game in Java?

   (hint: arrays will be very, very helpful!)

2. Write a method that tells you if a Tic Tac Toe game is won (or playable).
3. Write a method that gets input from the user and “makes” a move.
4. Wrap it up all up – into a nice two-player game!

Summer Project: Tic Tac Toe (2/2)

Wait, there’s more!

Make some “AI” that…

- just makes a random valid move (you should be able to beat this!)
- tries to make a “good” move (~ some if statements)
- <u>never loses</u>
  - Tic Tac Toe is a “solved game”: a perfect player will <u>never</u> lose.

Or, extend this idea to other grid-based games!

- similar-ish: connect four, checkers, battleship
- much harder: sudoku, chess, go, othello

Thank your lovely TAs!!!!!!

Thank you!

Ask Me (Almost) Anything!

sli.do #cse121-19