

LEC 18

CSE 121

Victory Lap!

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

*Talk to your neighbours:**What are your spring break plans?*Music: 🌸 [CSE 121 25sp Lecture Tunes](#) 🌸**Instructor:** Miya Natsuhara

TAs:	Chloë	Hibbah	Sushma
	Ailsa	Julia	Kelsey
	Johnathan	Sahej	Shayna
	Christian	Ruslana	Hannah
	Merav	Hanna	Zach
	Judy	Maitreyi	
	Janvi	Ayesha	

Announcements, Reminders (general)

This week:

- Programming Assignment 3 deadline **extended to Thursday, June 5!**
- Last day of IPL is Friday, June 6 (last day of instruction)
- Friday's class will be TA-led final exam review
- Miya's Friday office hours cancelled

Next week:

- Gumball & friends visit on Monday, June 9 1:00-2:30pm around Drumheller Fountain
- TA-led review session: **Tuesday, June 10th from 3:30-6:20pm, JHN 075**
- R7 (+ extra resub) due **Thu, June 12th** – all assignments eligible
- Final Exam: **Thursday, June 12th from 2:30 - 4:20pm in KNE 120**
 - look at seating charts and let me know ASAP if you're not there!
 - review [Exam page of website](#) (with policies & resources)

Pause: Exam Questions?

Will answer some live,
and put the rest in the class megathread for today!



sli.do
#cse121

Evaluations and Awards

Please give us feedback!

- [Course Eval](#)s are due **Sunday, June 8th at 11:59 PM**
- [TA Evals](#) are *also* due **Sunday, June 8th at 11:59 PM**

[Bob Bandes TA award](#) nominations open!

- thought your TA was the goat? write them a nomination!
- fun fact: some of our faculty won the award when *they* were TAs!
- fun fact #2: a 121 TA won the award last year :)

(Optional) Using `String[] args`!

`String[] args` is just a parameter to our `main` method...but we never call `main`, so how do we pass anything to `args`?

When we run our program from the *terminal*, we can pass "command-line arguments" to the `main` method, and they become the contents of `args`

```
javac MyProgram.java
```

```
java MyProgram these 7 words will go in args
```


You did it!



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 murder 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 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, 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, 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
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

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

Generalizing *beyond* CS

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
- 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

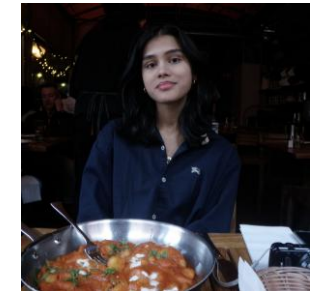
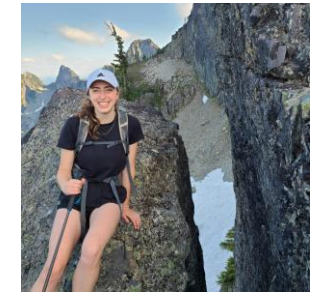
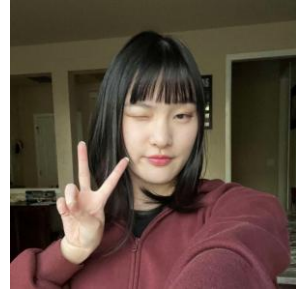
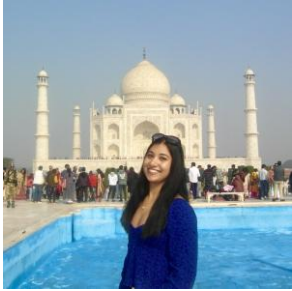
Should I learn another language? Which one?

- That depends – what do you want to do?



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

Thank your lovely TAs!



Thank you!

Ask Me (Almost)
Anything!



[sli.do #cse121](https://app.sli.do/event/eFdfh6UeGMUusiokyvNLYJo)

<https://app.sli.do/event/eFdfh6UeGMUusiokyvNLYJo>

