

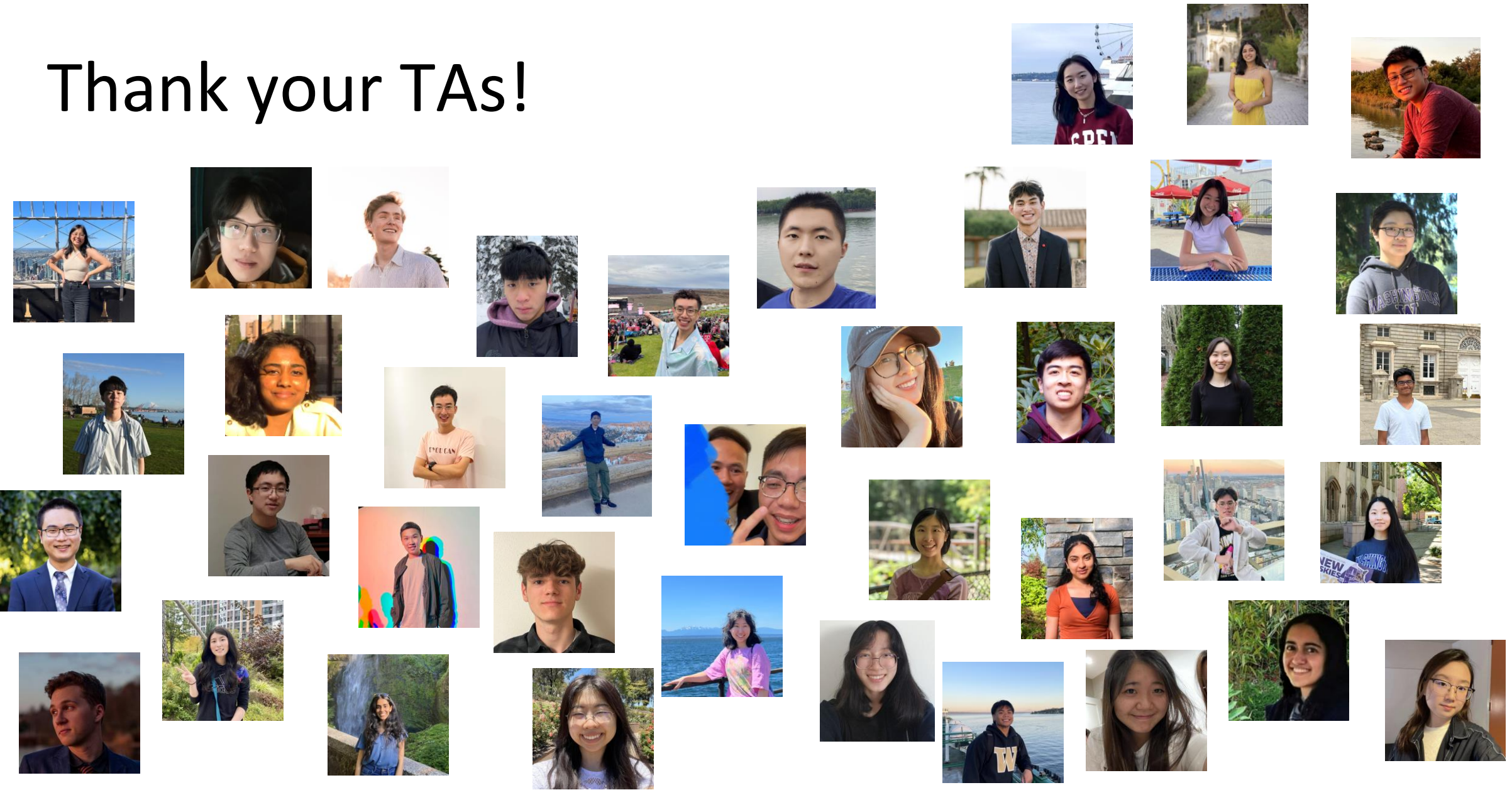
# You Made It!



# Announcements

- P3 due tonight
- R7 due Sunday
  - [Can use to extend P3](#)
- Final exam next Wednesday, 12:30-2:20
  - Read [exam policies](#)
    - One note page, no more than 8.5" x 11"
    - [Reference sheet](#) provided
    - Assigned seats
- [Grade guarantee calculator](#) posted later today

# Thank your TAs!



# Learning Objectives

*or, “What did I learn in this class?”*

- 1. Computational Thinking** Create an algorithm to solve a given problem and express that algorithm in a structured way (e.g. pseudocode)
- 2. Comprehension** Trace and predict the behavior of programs and systems
- 3. Code Writing** Write functionally correct Java programs that meet a provided specification using control structures, primitive data types, and basic data abstractions
- 4. Communication** Clearly and effectively describe the behavior of a given code snippet
- 5. Debugging** Identify errors in a method’s behavior & implement fixes for identified errors
- 6. Decomposition** Solve problems by breaking them into subproblems and recombining the solutions using techniques such as methods
- 7. Ethics/Impact** Describe ethical and sociotechnical issues related to software and technology and explain how their choices as programmers can impact those issues

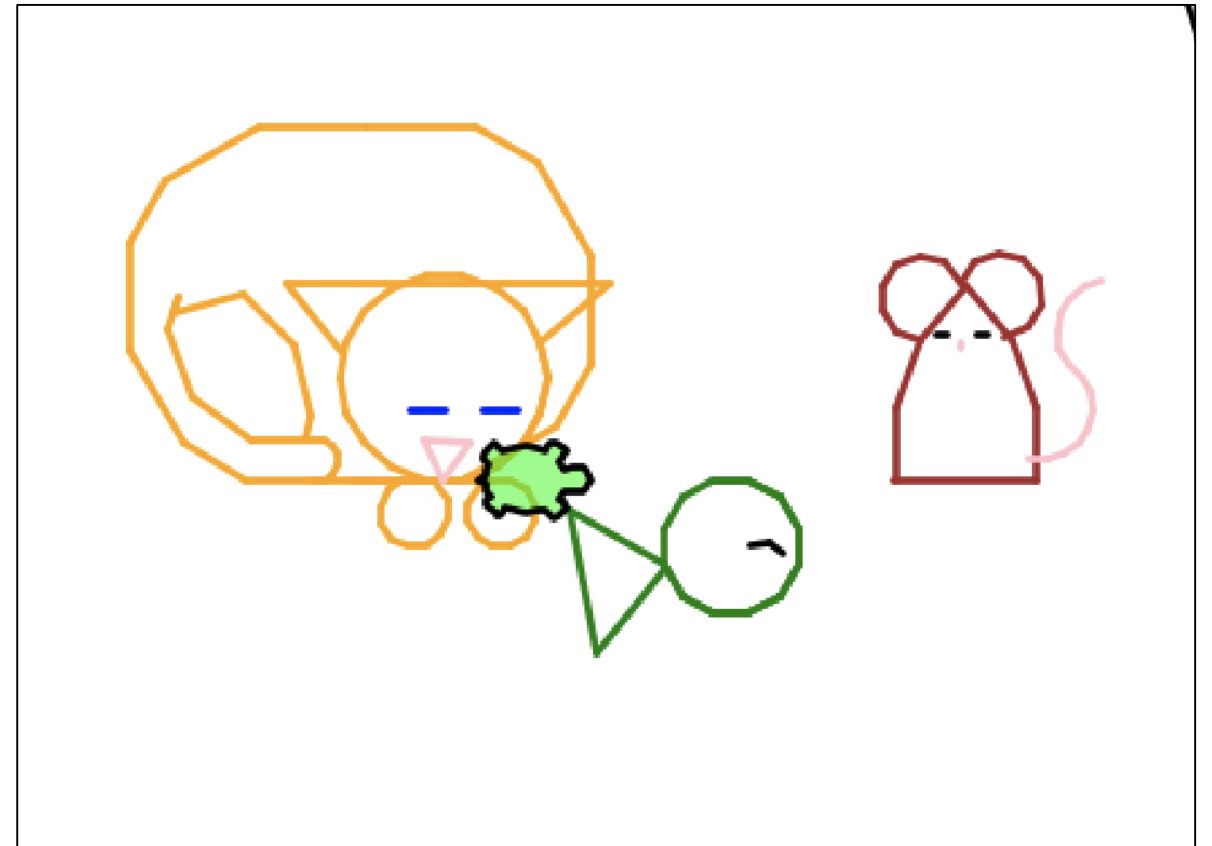


# Digression: My New Hobby

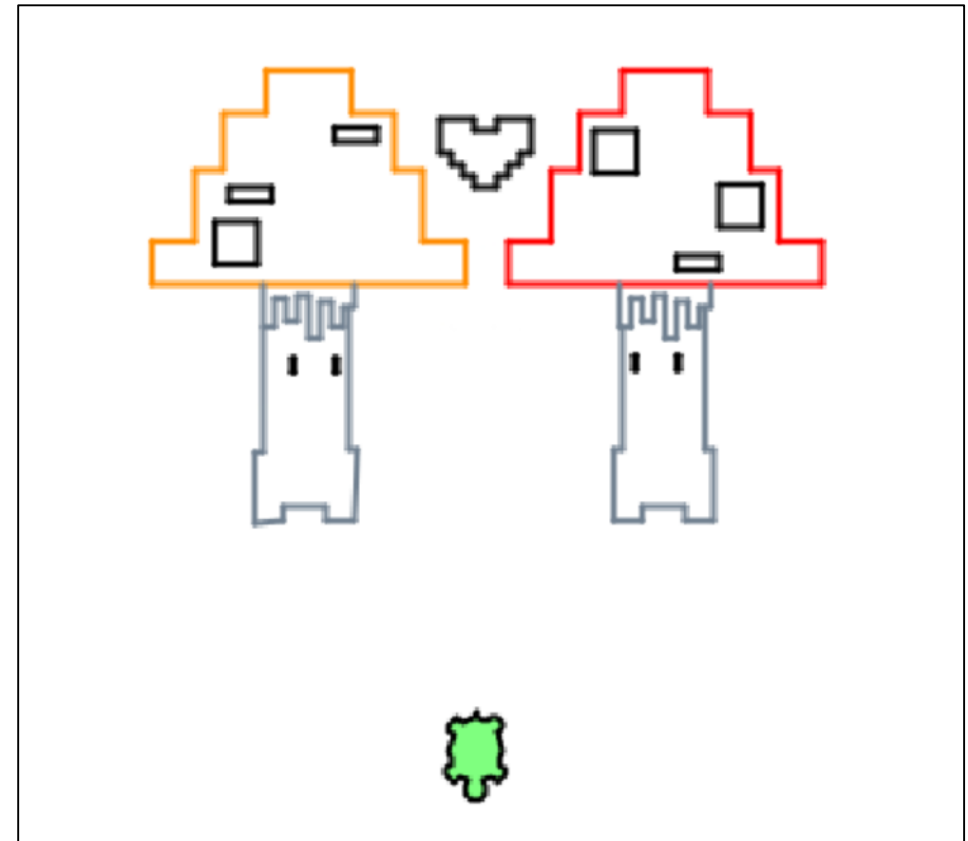
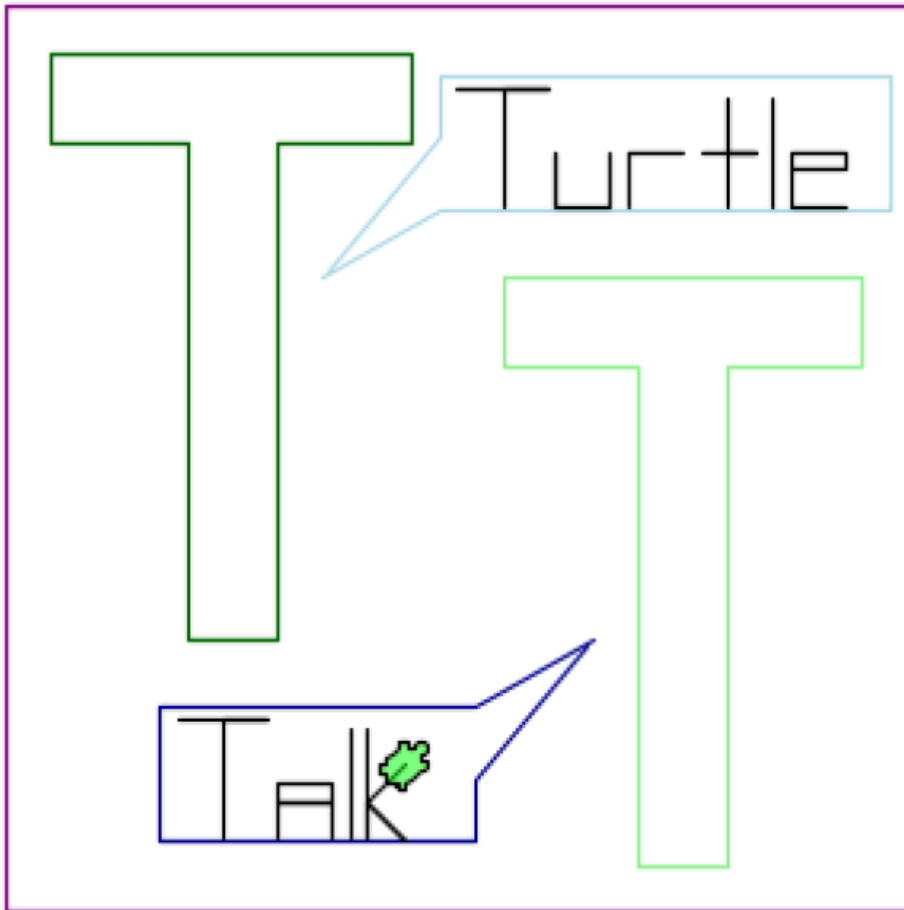
*Amigurumi*: Japanese art of creating crocheted or knitted stuffed toys



# You made some pretty cool crafts yourself!

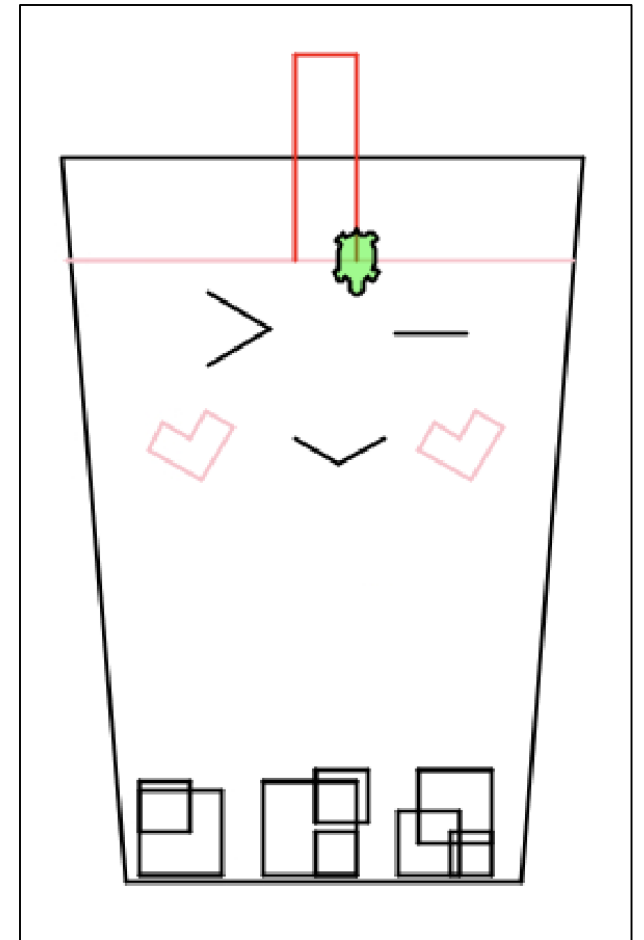
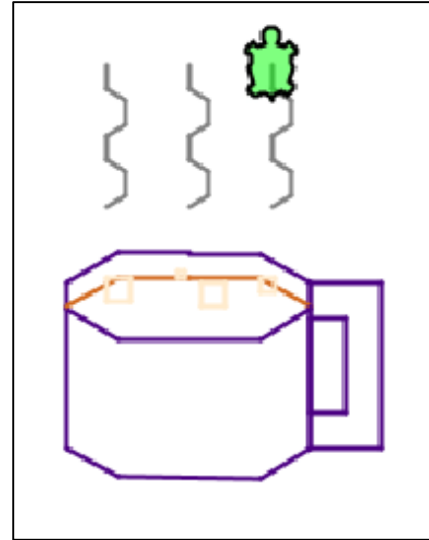
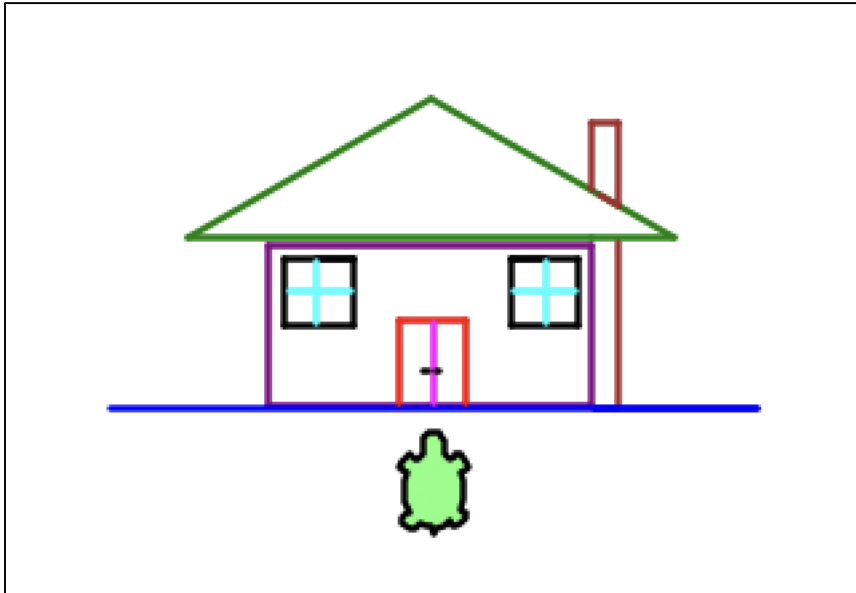


# You made some pretty cool crafts yourself!





# You made some pretty cool crafts yourself!



# 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 Courses

or “What can I do next?”

Course	Overview
<a href="#">CSE 122</a>	Introduction to Computer Programming II
<a href="#">CSE 123</a>	Introduction to Computer Programming III

## Majors

Course	Overview
<a href="#">CSE 311</a>	Mathematical foundations
<a href="#">CSE 351</a>	Low-level computer organization/abstraction
<a href="#">CSE 331</a>	Software design/implementation
<a href="#">CSE 341</a>	Programming languages (!)
<a href="#">CSE 340</a>	Interaction programming

## Non-majors

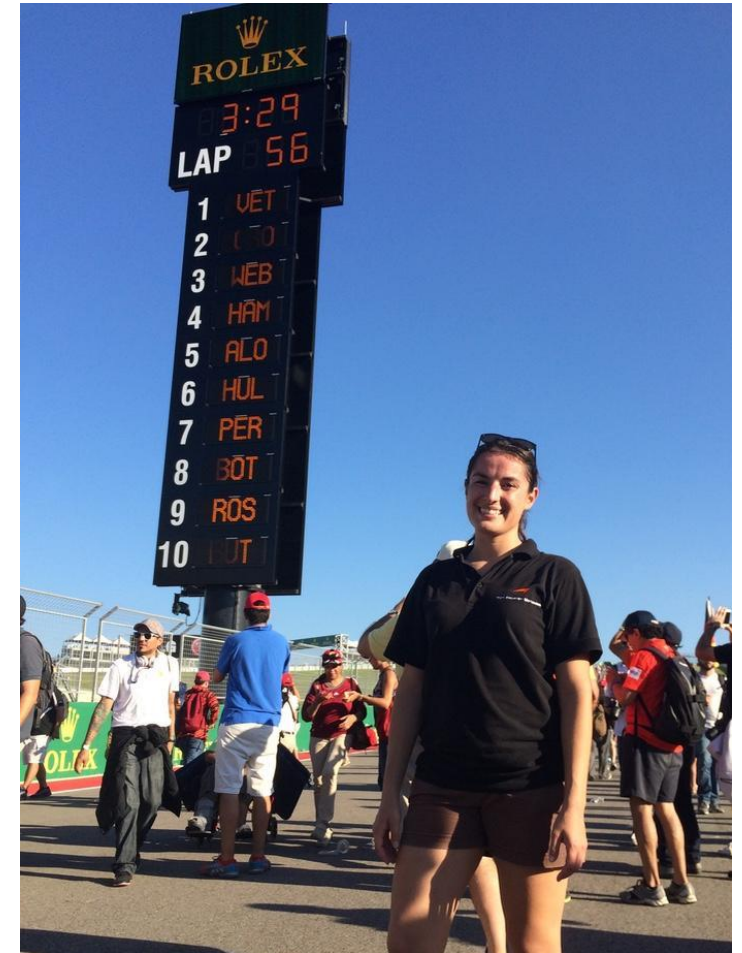
Course	Overview
<a href="#">CSE 154</a>	Intro. to web programming (several languages)
<a href="#">CSE 163</a>	Intermediate programming, data analysis (Python)
<a href="#">CSE 180</a>	Introduction to data science (Python)
<a href="#">CSE 373</a>	Data structures and algorithms (non-majors)
<a href="#">CSE 374</a>	Low-level programming and tools (C/C++)
<a href="#">CSE 416</a>	Intro. to Machine Learning

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)

# Thank you!!!



Ask Us (Almost) Anything!