Introductions

Yuma Tou

yumatou@uw.edu

5th-year Master’s Student

I took 341 2 years ago :( but it was super fun :)

Taylor Blau

ttaylorr@cs.washington.edu

Not a 5th-year Master’s Student (I’m a Junior)

Huge fan of 341 (it was my petition class)

I work part-time on Git at GitHub
Introductions (your turn)

1. What’s your name?
2. What year are you?
3. What did you do over Spring break?
Administrivia

- **Course webpage**: [https://cs.uw.edu/341](https://cs.uw.edu/341) (announcements, schedule, homeworks, *Emacs installation*)
- **Discussion list** (Google group): [cse341-19sp-discussion@cs.uw.edu](mailto:cse341-19sp-discussion@cs.uw.edu) (general questions)
- **Staff list**: [cse341-staff@cs.uw.edu](mailto:cse341-staff@cs.uw.edu) (private questions, etc.)
- **Private feedback**: [https://feedback.cs.uw.edu](https://feedback.cs.uw.edu)

Both one of our favorite classes (not a joke).
Demo

- Fix some syntactic errors
  - Both versions available on the course webpage afterwards
- We will take care of the first two
- You will work on the rest in groups
Demo (retrospective)

- Overwhelming? That’s OK.
- SML has helpful, but (at times) confusing errors.
- You will likely run into some of them on your homework.
- You’ve got this; we’re here to help.
More syntax!

- Boolean operations:
  - `e1 andalso e2` basically like Java’s `&&`
  - `e1 orelse e2` basically like Java’s `||`
  - `not e1` basically like Java’s `!`

- Comparison operations:
  - `>=` `>` `<=` `<` `=`
  - `<>` this is “not equals”
Testing your SML code

- All the homework assignments require you to also turn in testing code.

- No unit testing framework, but you can write simple tests like:
  ```sml
  val test1 = (cube 4 = 64);
  ```
  and check that everything is true when you run it!