Hello! I’m Cody!!!

- One of the two main section leaders (with Eric).
  - We’ll probably alternate weeks.
  - Remember: Cody is the fancy one.
- I’m one of the 5th year masters students.
- Previously TA’d many times including 341 twice.
- This is one of my favorite classes!
The REPL

- Read-Eval-Print-Loop
- Meant for iterative development and real-time testing.
- Usually load a file using `use`, test functions, edit source, restart REPL, and repeat.
- Very powerful tool for convenience.

Note: It’s dangerous to call `use` more than once in the same REPL.

Shadowing of Variable Bindings

SML Example

```sml
1 val x = "Hello World!";
2 val res = x*2; (* Is this 4 or a type error? *)
```

- There is no assignment in SML.
- However, a variable name can be bound multiple times.
- When looking up a variable, the latest binding in the current scope is used.
- Any previous bindings are said to be shadowed.
- A similar shadowing effect occurs in other languages like Java.
- This is the reason calling `use` more than once on the same file can cause problems.

Dealing with error messages from SML

We’ve Got Errors...

```sml
1 val x = 34;
2 y = x + 1; (* Missing val *)
3 val z = if y then 34 else x < 4;
4 val q = if y > 0 then 0;
5 val a = -5;
6 val w = 0;
7 funn = 34; (* fun is a keyword *)
8 val v = x / w; (* Can’t (/) ints *)
```

- If it’s ever possible to use a shadowed variable? Yes! And no.
- It can be possible to uncover a shadowed variable when the latest binding goes out of scope.

FAQ

Is it ever possible to use a shadowed variable? Yes! And no.

Using a Shadowed Variable

```sml
1 fun absOriginal x = abs x; (* Save abs function *)
2 fun (x, y) = (absOriginal x, absOriginal y);
```

FAQ: Is it ever possible to use a shadowed variable? Yes! And no.
- It can be possible to uncover a shadowed variable when the latest binding goes out of scope.

Using a Shadowed Variable

```sml
1 val x = "Hello World!";
2 fun add1 (x : int ) = x+1; (* Shadow x in function body *)
3 val y = add1 2;
4 val z = x+++!; (* "Hello World!!!" *)
```

FAQ: Is it ever possible to use a shadowed variable? Yes! And no.
- It can be possible to uncover a shadowed variable when the latest binding goes out of scope.

Using a Shadowed Variable

```sml
1 val x = "Hello World!";
2 fun add1 (x : int ) = x+1; (* Shadow x in function body *)
3 val y = add1 2;
4 val z = x+++!; (* "Hello World!!!" *)
```
Boolean Operators

The Operators

- `andalso` (same as Java’s `&&`)
- `orelse` (same as Java’s `||`)
- `not` (just a function)

Why can `not` be a function while the others cannot?

- Because `andalso` and `orelse` may not evaluate both its left and right sides. They short-circuit evaluation.
- Be careful to always use `andalso` instead of `and`.
- `and` is completely different. We will get back to it later.

Exercises

- Write the `xor` function.
- Given three `int`s, return their min and max in a pair.
- Write a function that computes the `n`th Fibonacci number.
- Implement a function that, given a real `x`, results in a real equal to the equation `x^2 - x/2 + 5.0`. Calculate `f(-2)` with it.

Solutions

```ml
1 fun xor1 (b1 : bool, b2 : bool) = if b1 then not b2 else b2;
2 fun xor2 (b1 : bool, b2 : bool) = (b1 orelse b2) andalso not (b1 andalso b2);

3 fun minmax (a : int, b : int, c : int) = (Int.min (a, Int.min (b, c)), Int.max (a, Int.max (b, c)));

3 fun fib (n : int) = if n < 2 then n else fib (n-1) + fib (n-2);
3 fun f (x : real) = x * x - x/2.0 + 5.0;
```