Section 4

CSE 341 – Spencer Pearson
Mutual Recursion

What if we need function f to call g, and function g to call f?
Mutual Recursion

Does this work?

```
1 fun f x =
2   ...
3   ...
4   ...
5   fun g y =
6     ...
7     ...
8     ...
```
Mutual Recursion

We can employ higher order functions for a work around.

fun _earlier f x = ... f y ...
fun later x = ... _earlier later y
val earlier = _earlier later
Mutual Recursion

But ML gives us special syntax for this

```
1 fun f x = y =
2     ...
3     ...
4     ...

5 fun g
6 y =
7     ...
8     ...
```
Modules

Remember that signatures are good both for organization and management and for maintaining invariants
Modules - Invariants

• Ordering of operations
  o e.g. insert, then query

• Data kept in good state
  o e.g. fractions in lowest terms

• Policies followed
  o e.g. don't allow shipping request without purchase order
Currying and Higher Order Functions

Questions?