Section 4

CSE 341 – Konstantin Weitz

Mutual Recursion

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

Mutual Recursion

Does this work?

```
fun f x =
  ...
  g y
  ...
fun g y =
  ...
  f x
  ...
```

Mutual Recursion

We can employ higher order functions
for a work around.

```
fun earlier (f,x) = … f y …
…
fun later x = … earlier(later,y)
```

Mutual Recursion

But ML gives us special syntax for this

```
fun f x =
  ...
  g y
  ...
and g y =
  ...
  f x
  ...
```

Modules

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

Modules - Invariants

- Ordering of operations
  - e.g. insert, then query
- Data kept in good state
  - e.g. fractions in lowest terms
- Policies followed
  - e.g. don't allow shipping request without purchase order
- Sensitive information hidden
  - e.g. force authentication for api use

Currying and Higher Order Functions

Questions?