



# CSE341: Programming Languages

## Lecture 9

### Function-Closure Idioms

Dan Grossman

Fall 2011

## *More idioms*

- We know the rule for lexical scope and function closures
  - Now what is it good for

A partial but wide-ranging list:

- Pass functions with private data to iterators: Done
- Combine functions (e.g., composition)
- Currying (multi-arg functions and partial application)
- Callbacks (e.g., in reactive programming)
- Implementing an ADT with a record of functions

# Combine functions

Canonical example is function composition:

```
fun compose (g,h) = fn x => g (h x)
```

- Creates a closure that “remembers” what `g` and `h` are bound to
- Type `('b -> 'c) * ('a -> 'b) -> ('a -> 'c)`  
but the REPL prints something *equivalent*
- ML standard library provides this as infix operator `o`
- Example (third version best):

```
fun sqrt_of_abs i = Math.sqrt (Real.fromInt (abs i))  
fun sqrt_of_abs i = (Math.sqrt o Real.fromInt o abs) i  
val sqrt_of_abs = Math.sqrt o Real.fromInt o abs
```