

```
fun append (xs,ys) =  
  if xs=[]  
  then ys  
  else (hd xs)::append(tl xs,ys)  
  
fun map (f,xs) =  
  case xs of  
    [] => []  
  | x::xs' => (f x)::(map(f,xs'))  
  
val a = map (increment, [4,8,12,16])  
val b = map (hd, [[8,6],[7,5],[3,0,9]])
```

Programming Languages

Dan Grossman

University of Washington

Welcome to Part B!

Re-Welcome!

Part A covered a lot – now let's build on it!

Challenging opportunity to learn *the fundamental concepts* of programming languages

With hard work, patience, and an open mind, this course makes you a much better programmer

- *Poor* course summary: “Uses ML, [Racket](#), and Ruby”

Now:

- Another functional language – easier to pick up
- Powerful idioms “delaying evaluation”
- Implementing your own programming language with closures
- Static vs. dynamic typing

(Lack of) introductory material

Similar format to Part A

So “dive right in”

(As needed, consult course information and Part A materials)