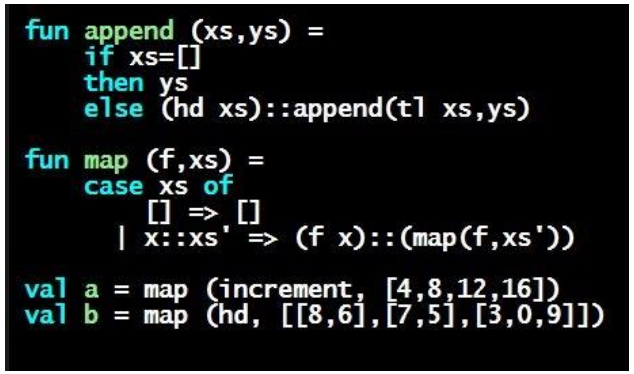

```
fun append (xs,ys) =
    if xs=[]
    then ys
    else (hd xs)::append(tl xs,ys)

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

## Part C Course Structure

# *Part C: Three weeks*

- Week 1:
  - "Typical" assignment structure (auto-grader, peer review)
  - But different style: small-ish extensions to provided code
  - And logistical hoops to support multiple Ruby versions
- Week 2:
  - Relatively few videos
  - More challenging assignment with ML and Ruby portions
  - Logistical hoops for multiple Ruby versions *and* ML + Ruby
- Week 3:
  - Again relatively few videos
  - Then "just" a "Final Exam" covering Part B and Part C