

CSE 341: Programming Languages

Dan Grossman

Winter 2008

Lecture 17— Implementing languages, especially higher-order functions

Where are we

- Today:
 - Finish static vs. dynamic typing (arguments 2–5)
 - Learn how closures are actually implemented (key to hw 5)
- Friday: Modularity in Scheme
- Monday: Ruby basics
- Later: More concepts and contrasts
 - At least as important as programming details
 - (for life and, say, the final)

Implementing Languages

Mostly 341 is about language meaning, not “how can an implementation do that”, but it’s important to “dispel the magic”.

At super high-level, there are two ways to implement a language A :

- Write an *interpreter* in language B that evaluates a program in A
- Write a *compiler* in language B that translates a program in A to a program in language C (and have an implementation of C)

In theory, this is just an implementation decision.

HW5: An interpreter for MUPL in Scheme.

Most interesting thing about MUPL: higher-order functions.