



# CSE 505: Programming Languages

Winter 2005

Instructor: Craig Chambers

# Goals

- ◆ Study main advanced concepts in modern PLs
- ◆ Gain experience using concepts in moderate-sized projects
  - A **few** key languages **deeply**
- ◆ Learn techniques for precisely & formally specifying PL features
  - Become a competent *reader* of PL theory
- ◆ Expose areas of current research
- ◆ See how PL ideas are relevant more broadly

# Course outline (part I)

## ◆ Functional languages

- programming using first-class functions
- polymorphic type checking and inference
  
- ML as main project language
- exposure to Haskell, Scheme
- lambda calculus, type theory, simple proofs
  
- project: evaluator, type inferencer in ML for Scheme-like language

# Course outline (part II)

- ◆ Object-oriented languages
  - various object/class/inheritance models
  - various method dispatching models
  - subtyping and its implications
  
- Diesel as main project language
- exposure to Smalltalk, Self, EML, AspectJ
  
- project: Self interpreter in Diesel

# Coursework

## ◆ Readings

- language reference manuals & tutorials
- some key papers from the literature
- “Types and Programming Languages” book

## ◆ Two implementation projects

## ◆ Periodic homework

## ◆ Final exam

# Why take 505?

- ◆ PL is central to CS, software development
  - many interactions with compilers, software eng.
  - active in research and practice
- ◆ PL ideas applicable in other areas
  - checking & enforcing security in systems
  - describing interfaces to components
  - domain-specific notations for music, DB queries, networking, animation, ...