CS 510 Types and Semantics – Fall 2005

Instructor: Andrew Tolmach
120-23 FAB (503) 725-5492
email: `apt@cs.pdx.edu`
Office Hours: TuTh 4-5pm & by appt.
Course web page: `http://www.cs.pdx.edu/~apt/cs510types`

Description

This course provides an introduction to the mathematics of program meaning (semantics) using the framework of type systems and typed languages. Topics include operational semantics; inductive proof techniques; the lambda-calculus; type safety; basic and advanced types systems including references, exceptions, and subtyping; types for object-oriented languages, and polymorphic types.

Prerequisites

The theoretical material in the course is self-contained, so there are no specific prerequisites, but a reasonable level of mathematical maturity is desirable. For example, you should be comfortable with proofs by induction.

Programming exercises will be in the OCAML language, so previous exposure to a functional language such as Standard ML (to the level of CS558) or Haskell (as in CS557) is quite desirable.

Readings

We will use the textbook "Types and Programming Languages," by Benjamin C. Pierce, MIT Press, 2002.

Requirements

There will be weekly homework assignments, a midterm, and a final exam (probably closed book). The homework assignments will include both theory problems and short programming exercises. Since answers are provided to many assignment questions in the back of the book, the assignments will be scored simply based on whether you turn them in or not; hence, they won't count for much of the grade. But doing them will be essential preparation for the exams.

The course grade will be distributed as follows:

Homework	20%
Midterm	35%
Final	45%

Although it will not be formally assessed, class participation is strongly encouraged, and may affect borderline grades.

You are strongly encouraged, though not required, to typeset your homework solutions using `latex`.

Computing Facilities

Some of the homework exercises will require use of the OCAML language. This is installed as a package (`ocaml-3.07`) on the CS Solaris machines. It is also very easy to install on your own personal machine (and doesn't require many resources). See the course web page for pointers.

Individual Work

It is permitted (even encouraged) for you to work together on homework assignments. However, all homework submissions must be written up (or typed in) individually; an important part of the course is learning how to write down

theoretical arguments, even after they are clear in your own mind.

Exams must be completed individually without any collaboration. Cheating on an exam will result in an automatic zero grade and the initiation of disciplinary action at the University level.

Disabilities

If you are a student with a disability in need of academic accommodations, you should register with Disability Services for Students and notify the instructor immediately to arrange for support services.

Tentative Schedule

This schedule is highly subject to change. You should always attempt to do the reading *before* the relevant class meeting.

dates	*Pierce chapters*	*topics*
Sep 27 & 29	1,(2),3,4	Introduction; Syntax and Operational Semantics
Oct 4 & 6	5,6,7	Untyped lambda-calculus
Oct 11 & 13	8,9,10	Types; Simply-typed lambda-calculus
Oct 18 & 20	11	Extensions
Oct 25 & 27	13,14	References and Exceptions
Nov 1		**Midterm** (in-class)
Nov 3	15	Subtyping
Nov 8 & 10	16,17,19	More subtyping; Object-oriented languages
Nov 15 & 17	20,22	Recursive Types; Type Reconstruction
Nov 22 & 24	23,24,25	Universal and Existential Types
Nov 29 & Dec 1	29,30	Higher-order Systems
Dec 6		**Final Exam (10:15-12:05)**