

## CS 457/557 Functional Languages - Fall 2005

Instructor:

Andrew Tolmach

120-23 FAB

725-5492

email: [apt@cs.pdx.edu](mailto:apt@cs.pdx.edu)

Office Hours: TuTh4-5 & by appt.

Course home page: <http://www.cs.pdx.edu/~apt/cs457>

### Description

This course provides a gentle introduction to the ideas and techniques of functional programming, using the Haskell programming language. Functional languages are great for both rapid prototyping and serious software engineering. The key idea is to program declaratively using function definitions rather than operationally using assignments. Haskell's features include higher-order functions, strong typing, polymorphism, user-defined algebraic data types, automatic storage management, lazy evaluation, and powerful abstraction and modularization facilities. The course will concentrate on developing practical programming skills in Haskell, with a brief look at the theoretical underpinnings.

### Textbooks

The required text is by Paul Hudak, *The Haskell School of Expression*, Cambridge University Press, 2000.

An excellent auxiliary reference is Simon Thompson, *Haskell: The Craft of Functional Programming*, 2nd. ed., Addison-Wesley, 1999.

Two additional short readings are listed in the schedule below; more may be assigned from time to time.

Lecture notes will be available electronically in postscript format via the course home page.

Lots of on-line resources are available at <http://www.haskell.org>.

### Requirements

Weekly graded homework assignments (8 in all)	40%
Midterm exam	30%
Final exam	30%

The weekly assignments are intended to help you master the basic material of the course. Each homework assignment is to be handed in (on paper) at the beginning of class on the Tuesday on which it is due.

The midterm and final exams are intended to check that you have mastered the fundamentals of the language. They will be open-book, open-notes.

### Computing Facilities

To do the homework, you will need to use the Haskell interpreter called HUGS 98. In particular, you'll want the Nov2002 release; some of the graphics code in the Hudak book does *not* work properly with more recent releases. A copy of this Hugs release is available on the Solaris network in the CS department in the `hugs` package. Alternatively, you can download it for use on your own machine (either Windows or linux) from <http://haskell.org/hugs>. (Make sure to get the Nov2002 release.) It is easy to install and its resource demands are modest. Any supporting materials (program files, etc.) will be made available on the course web page.

## Mailing List

Important information will be distributed throughout the term via a mailing list called `cs4571list`. You can subscribe to this list via the course home page.

## Tentative Schedule

	Reading	What's due	Topics ( <i>Example Applications</i> )
Sep 27	1,2		Introduction; Using Hugs; Lists
29			Data Types ( <i>Shapes</i> ); Modules
Oct 4	3,4	HW1	IO Actions; Graphics ( <i>Drawing Shapes</i> )
6			
11	5,6	HW2	Polymorphism; Higher-order functions; Maps and folds ( <i>Perimeters of shapes</i> )
13			
18	7,8,HJ	HW3	Algebraic data types; Trees ( <i>Regions</i> )
20			
25	9,10	HW4	More higher-order functions: Currying, sections, composition; ( <i>Drawing Regions</i> )
27			
Nov 1	11		<b>Midterm Exam (2-3:50pm)</b>
3			Proving Program Properties
8	12,13	HW5	Qualified types and type classes ( <i>Animations</i> )
10			
15	14,H	HW6	Streams; Laziness; Infinite data structures
17			
22	18,19	HW7	Higher-order types; Monads ( <i>Robot Control Language</i> )
24			<b>Thanksgiving Holiday – no class</b>
29		HW8	
Dec 1			Pragmatics
5			<b>Final Exam (Monday 10:15am-12:05pm)</b>

Class will normally meet from 2-3:30pm.

Schedule is subject to change.

Numbers in the readings column are chapter numbers from the Hudak textbook.

Letters in the readings column are keyed as follows:

HJ = P. Hudak and M. Jones, *Haskell vs. Ada vs. C++ vs. Awk vs. ... , An Experiment in Software Prototyping Productivity*, July 1994, <http://haskell.org/papers/NSWC/jfp.ps>.

H = J. Hughes, *Why Functional Programming Matters*, 1984, <http://www.math.chalmers.se/~rjmh/Papers/whyfp.html>.

Additional readings may be assigned from time to time.

### Plagiarism

Unless specifically indicated otherwise, all homework assignments and exams must represent your own, individual work. It is permissible to discuss the assignment with other students, but you must program and write up the solution by yourself. *Do not, under any circumstances, copy another person's program and submit it as your own.* Writing code for use by another or using another's code in any form (even with their permission) will be considered cheating. In particular, cheating will result in an automatic F for the assignment in question, 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.