## CS 457/557 Functional Languages - Spring 2018

Instructor:
Andrew Tolmach
120-23 FAB
725-5492
email: tolmach@pdx.edu
Office Hours: Tu1-2pm & 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.

## Textbook

The required text is by Graham Hutton, *Programming in Haskell*, 2nd. ed., Cambridge University Press, 2016.

There are lots of excellent alternative books and other learning resources, many of which are available free on the web. See www.haskell.org/documentation for a list.

Lecture notes, which are also executable Haskell files, will be made available via the course home page.

## Requirements

Grading will be based on the following components:

| | |
|---|---|
| Weekly graded homework assignments (6 in all) | 55% |
| Midterm exam | 20% |
| Final project | 25% |
| **No** final exam | |

The weekly assignments are the focus of the course; they are intended to help you learn the Haskell language and the functional approach to programming.

The midterm is intended to confirm your mastery of the core elements of Haskell.

The final project will give you a chance to write a small application in Haskell. Typical projects will build on existing libraries for graphics, music, file manipulation, etc. A set of suggested projects will be provided, but you are encouraged to come up with your own ideas instead. Projects may be done individually or in teams of two. A preliminary version of the project will be due in week 8, and the final version will be due at the last class meeting in week 10.

## Computing Facilities

To do the homework, you will need to use the Haskell interpreter called `ghci` with a small subset of the standard libraries. To the the project, you are likely to want the native code compiler `ghc` and a more extensive set of libraries. The simplest way to get these tools is to install the Haskell Platform on your own machine from `http://haskell.org/platform`. (The current version of the Platform is 8.2.2. For the homework and most projects, any recent version should work fine.) Installation is simple, but does consume a lot of disk space. The Platform is also installed on the CS department `linuxlab` machines; version 8.2.2 is available using the `addpkg` command.

You should bring a laptop capable of running `ghci` (either locally or remotely) and your favorite editor to each class meeting.

## Mailing List

Important information will be distributed throughout the term via a mailing list called `cs457list@cs.pdx.edu`. You can subscribe to this list via the course home page.

## Academic Integrity

Except for the project, 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.

## Tentative Schedule

This schedule is subject to change!

| | | Reading | What's due | Topics |
|---|---|---|---|---|
| Apr | 2 | 1–3 | | Introduction; Overview |
| | 4 | | HW0 | Getting started; Simple examples |
| | 9 | 4–7 | | Programming with Lists |
| | 11 | | HW1 | More about lists |
| | 16 | 8–9 | | Trees |
| | 18 | | HW2 | Algebraic data types |
| | 23 | | | Polymorphism; Type checking |
| | 25 | | HW3 | Type classes |
| | 30 | 10–11 | | IO Actions |
| May | 2 | | HW4 | Testing program properties |
| | 7 | | | **Midterm Exam (2-3:50pm)** |
| | 9 | | P1 | Functions as data; data structure design |
| | 14 | | | Guest lecture: Functional Logic Programming |
| | 16 | 16 | HW5 | Guest lecture: Proving program properties |
| | 21 | 12 | | Monads |
| | 23 | 15 | P2 | Laziness |
| | 28 | | | MEMORIAL DAY HOLIDAY – NO CLASS |
| | 30 | | HW6 | Advanced topics: Performance and implementation issues |
| Jun | 4 | | | Advanced topics: Parallelism |
| | 6 | | P3 | Advanced topics: TBD |
| | 13 | | | **NO Final Exam** |

Class meets from 2-3:50pm. Each class will normally consist of lecture material followed by a series of programming exercises for you to attempt on your laptops, either individually or in small groups. Therefore, **it is important to bring a laptop equipped with a working Haskell system to each class meeting.**

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

HW0 through HW7 mark the days when homework assignments are due. P1 marks the day when project topics are due; P2 marks the the day when the intermediate project submission is due; P3 marks the day when the final project submission is due.