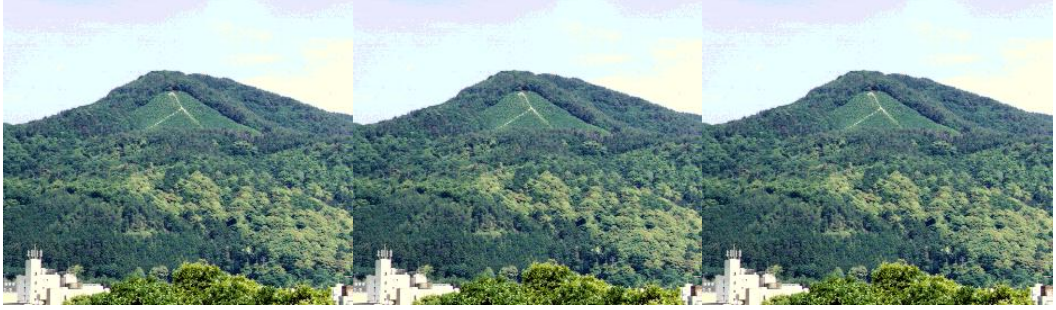
Parallel Functional Programming

Stephen A. Edwards

Columbia University

Fall 2022



Instructor

Prof. Stephen A. Edwards

sedwards@cs.columbia.edu

http://www.cs.columbia.edu/~sedwards/

462 Computer Science Building

Email me for appointments, or just come by


```
primes = filterPrime [2..]
  where filterPrime (p:xs) =
          p : filterPrime [x | x <- xs, x `mod` p /= 0]
```

Purely Functional · Declarative · Lazy · Statically Type-Inferred · Parallel

Sequential Haskell in the first half · Parallel in the second half

Prerequisites

Data structures (COMS W3134, W3137, or equivalent)

- You must be fluent in at least one programming language

- You must dream about lists and trees

- You do not need prior experience in a *functional* programming language; that's what this course is for

Assignments and Grading

70 %	Homework assignments
30 %	Final Project (alone or in pairs)

This is a coding[†] class

The homework must be your own code

The project may be done alone or in pair

[†]More precisely, mostly debugging,
with a little bit of bugging

Collaboration

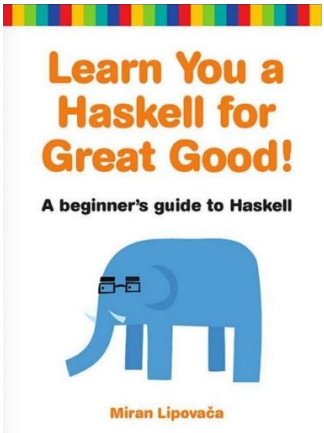
You may seek outside help, including from other students, on homework, but

- You must write all of your own code. No copying or copying-with-modification of any code. No looking at other student's code as reference as you write your own.
- You must cite all people and resources you consulted. For example, you might add a comment like

```
{– I collaborated with Haskell Curry, Jim Backus, Alonzo Church,
  and Grace Hopper on this assignment, and consulted
http://hackage.haskell.org/package/base–4.12.0.0/docs/Data–List.html
https://stackoverflow.com/questions/211216
http://www.cis.upenn.edu/~cis194/fall16/policies.html
–}
```

See also `http://www.cs.columbia.edu/education/honesty/`

Recommended Texts

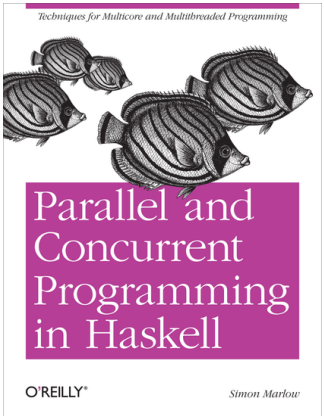


Miran Lipovača.
Learn You a Haskell for Great Good!
No Starch Press, 2001.

`http://learnyouahaskell.com/`

Excellent introductory text. We will be following it for roughly the first half of the class.

Recommended Texts

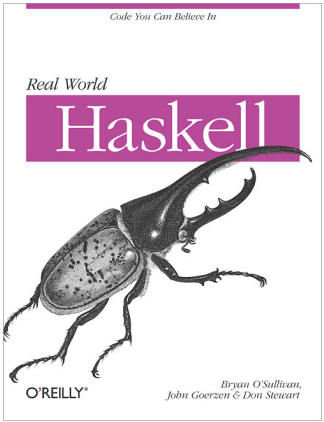


Simon Marlow.
Parallel and Concurrent Programing in Haskell.
O'Reilly, 2013.

`https://simonmar.github.io/pages/pcph.html`

Like its title says. Assumes a reasonable understanding of Haskell. We will be following it for the second half of the class.

Recommended Texts

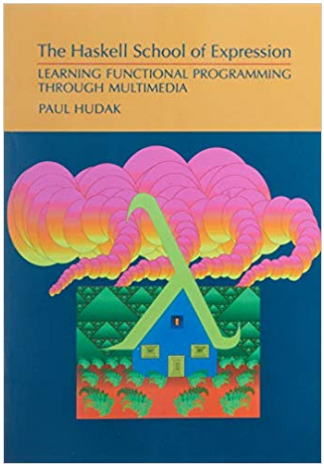


Bryan O'Sullivan, Don Stewart, and John Goerzen.
Real World Haskell.
O'Reilly, 2009.

`http://book.realworldhaskell.org/`

Also an introductory text on Haskell that starts at the beginning, it quickly focuses on practical, real-world aspects of writing Haskell programs, such as elaborate I/O, and interfacing with external libraries.

Recommended Texts

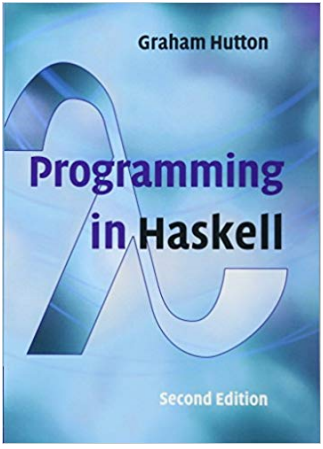


Paul Hudak.
The Haskell School of Expression.
Cambridge University Press, 2000.

`http://www.cs.yale.edu/homes/hudak/SOE/`

An idiosyncratic approach to learning Haskell based on multimedia (graphics, animation, and sound) ultimately leading to domain-specific langauges.

Recommended Texts

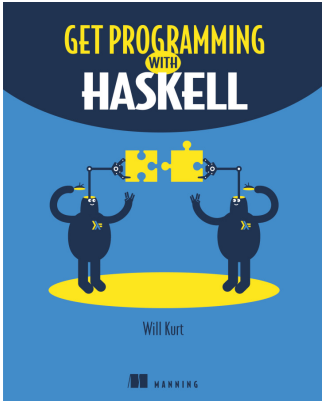


Graham Hutton.
Programming in Haskell.
Second Edition, Cambridge University Press, 2016.

`http://www.cs.nott.ac.uk/~pszgmh/pih.html`

Another introductory Haskell text, this one written by a professor from the University of Nottingham

Recommended Texts



Will Kurt.
Get Programming with Haskell.
Manning, 2018.

`https://www.manning.com/books/get-programming-with-haskell`

Another introductory Haskell text, written more like a textbook