

# CMSC388F: Functional Pearls

## Description

This course will explore elegant examples of functional programming. The first half will provide an introduction to Haskell and four important abstractions: monoids, functors, applicative functors, and monads. We will focus on simple and plentiful examples. The second half will cover a wide variety of functional programming techniques and applications.

## Details

- **Course:** CMSC388F
- **Prerequisites:** C- or better in CMSC330
- **Credits:** 1
- **Seats:** 30
- **Lecture Time:** Fridays, 10 - 10:50AM
- **Location:** CSI 3120
- **Semester:** Spring 2019
- **Textbook:** Learn You a Haskell for Great Good! by Miran Lipovača
- **Facilitators:** Cameron Moy, Ben Mariano
- **Advisor:** David Van Horn

## Topics

- Haskell
- Abstractions
  - Monoids
  - Functors
  - Applicative Functors
  - Monads
- Techniques
  - Equational Reasoning
  - Laziness
  - Zippers
  - Propositions as Types
  - Randomized Testing
  - Parsing

## Grades

Grades will be maintained on the department grades server.

You are responsible for all material discussed in lecture and posted on the class repository, including announcements, deadlines, policies, etc.

Your final course grade will be determined according to the following percentages:

Percent	Title	Description
70%	Assignments	Four individual programming assignments.
30%	Final Project	Final group project writing and implementing an elegant functional solution.

Any request for reconsideration of any grading on coursework must be submitted within one week of when it is returned. No requests will be considered afterwards.

Week	Topic	Assignment
01 (2/01)	Hello, Haskell	A1 Out
02 (2/08)	Sudoku	
03 (2/15)	Typeclasses	
04 (2/22)	Monoids	A1 Due, A2 Out
05 (3/01)	Functors	
06 (3/08)	Applicative Functors	A2 Due, A3 Out, Project Out
07 (3/15)	Monads	
08 (3/22)	Spring Break	
10 (4/05)	Laziness	A3 Due, A4 Out
11 (4/12)	Zippers	
12 (4/19)	Propositions as Types	A4 Due
13 (4/26)	Randomized Testing	
14 (5/03)	Parsing	Project Due
15 (5/10)	Presentations	

## Assignments

Assignments must be submitted electronically following the instructions given in each project assignment. They may not be submitted by any other means. It is your responsibility to test your program and verify that it works properly before submitting. All projects are due at 11:59:59 PM on the indicated date.

Assignments may be submitted up to 24 hours late for a 10% penalty. If you submit both on-time and late, your project will receive the maximum of the penalty-adjusted scores.

You will be responsible for developing your own tests and for using appropriate testing techniques. We expect your projects to use proper style and

documentation.

## Outside-of-class communication with course staff

We will interact with students outside of class in primarily two ways: e-mail and office hours by appointment.

Name	Office	E-mail
Cameron Moy	3452 AVW	<a href="mailto:camoy@cs.umd.edu">camoy@cs.umd.edu</a>
Ben Mariano	3452 AVW	<a href="mailto:benmar@cs.umd.edu">benmar@cs.umd.edu</a>
David Van Horn	3439 AVW	<a href="mailto:dvanhorn@cs.umd.edu">dvanhorn@cs.umd.edu</a>

## Excused Absence and Academic Accommodations

See the section titled “Attendance, Absences, or Missed Assignments” available at Course Related Policies.

## Disability Support Accommodations

See the section titled “Accessibility” available at Course Related Policies.

## Academic Integrity

Note that academic dishonesty includes not only cheating, fabrication, and plagiarism, but also includes helping other students commit acts of academic dishonesty by allowing them to obtain copies of your work. In short, all submitted work must be your own. Cases of academic dishonesty will be pursued to the fullest extent possible as stipulated by the Office of Student Conduct.

It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit <http://www.shc.umd.edu>.

## Course Evaluations

If you have a suggestion for improving this class, don’t hesitate to tell the instructor or TAs during the semester. At the end of the semester, please don’t forget to provide your feedback using the campus-wide CourseEvalUM system. Your comments will help make this class better.