CS 583 – Advanced Functional Programming
Syllabus for Spring 2019

CS 583, Functional Programming, is a four-credit course for graduate students. This course introduces to advanced topics of functional programming languages. Topics to be covered include: type and constructor classes, monads, functional/persistent data structures, and graph reduction.

Prerequisites: You should have taken CS 581 and have a good understanding of the following concepts.

- Haskell: Data types, higher-order functions, representing abstract syntax, writing interpreters
- Lambda Calculus: Syntax, free variables, beta-reduction, normal forms, reduction strategies

If you haven’t taken CS 581, please talk to me in person to see whether you are well enough prepared for CS 583.

Instructor: Martin Erwig
Email: erwig@oregonstate.edu
URL: oregonstate.edu/~erwig
Office: KEC 3045
Office Hours: Mondays 10am - 11am
Wednesdays 3pm - 4pm

Learning Objectives: On completion of the course, students must demonstrate the following abilities.

1. Students will be able to write sophisticated functional programs employing higher-order function, currying, type classes, and monads.
2. Students can employ the Haskell type class system. In particular, they will be able to design classes and define instances for specific types.
3. Students will be able to define and use monad data types. They will be able to write programs in monadic style, and they can convert programs into monadic style. Students will be able to define parsers using monadic parser combinators.
4. Students can define and use functional data structures. They can analyze their complexity and compare them with imperative data structures. Students will be able to explain the essential differences between functional and imperative data structures, and they can also describe their respective advantages/disadvantages.
5. Students understand the graph reduction technique for implementing functional languages.
Tentative Lecture Syllabus (subject to change):

Week 1. Introduction, Haskell, higher-order functions, infinite data structures
Week 2. Type classes, constructor classes
Week 3. Monads, state (transformer) monad
Week 4. Parser combinators
Week 5. More on monads (IO & functional imperative programming, maybe monad transformers)
Week 6. Review for midterm, midterm exam
Week 7. Functional data structures, amortization, functional queues
Week 8. Advanced functional data structure techniques
Week 9. Graph reduction, miscellaneous topics
Week 10. Final project presentations

Plus: Student presentations of papers relevant to the projects along the way.

Important date
Thursday, May 9  10:00am - 11:00am  Midterm Exam

Grading
35%  Midterm exam
35%  Functional programming project (graded for correctness, functionality, and style).
15%  Programming workshop (brief peer-reviewed programming assignments)
15%  Class participation & presentations

Final Project
Students must demonstrate proficiency in functional programming and explain in their final submission what functional programming concepts (such as, higher-order functions or monads) they have employed and how.

Important Note
Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should know, or who need special arrangements in the event of evacuation, should make an appointment with the instructor as early as possible, however, no later than the first week of the term. In order to arrange alternative testing the student should make the request at least one week in advance of the test. Students seeking accommodations should be registered with the Office of Services for Students with Disabilities.