COMP311

***NOTE:** This page is for an old offering of the course. To find the latest course offering, please visit https://comp311.rice.edu/.*

COMP 311: Functional Programming (Fall 2016)

Instructor	Dr. Eric Allen Dr. Corky Cartwright	**Graduate TAs**	• Arghya "Ronnie" Chatterjee • Lechen Yu
Co-Instructor	Dr. Sanak Tarlar	**Undergraduate TAs**	• Chris Brown • Cannon Lewis • Jake Nyquist
Lectures	DCH 1075	**Lecture Times**	8:00AM - 9:15AM TR
Course Email	comp311_staff@rice.edu	**Online Discussion**	Piazza -- Rice Comp 311

Description

This class provides an introduction to concepts, principles, and approaches of functional programming. Functional programming is a style of programming in which the key means of computation is the application of functions to arguments (which themselves can be functions). This style of programming has a long history in computer science, beginning with the formulation of the Lambda Calculus as a foundation for mathematics. It has become increasingly popular in recent years because it offers important advantages in designing, maintaining, and reasoning about programs in modern contexts such as web services, multicore programming, and distributed computing. Course work consists of a series of programming assignments in the Scala programming language and various extensions.

Grading, Honor Code Policy, Processes, and Procedures

Grading will be based on your performance on weekly programming assignments. All work in this class is expected to be your own, and you are expected not to post your solutions or share your work with other students, even after you have taken the course. Please read the Comp 311 Honor Code Policy for more details on how you are expected to work on your assignments.

All students will be held to the standards of the Rice Honor Code, a code that you pledged to honor when you matriculated at this institution. If you are unfamiliar with the details of this code and how it is administered, you should consult the Honor System Handbook. This handbook outlines the University's expectations for the integrity of your academic work, the procedures for resolving alleged violations of those expectations, and the rights and responsibilities of students and faculty members throughout the process.

Accommodations for Students with Special Needs

Students with disabilities are encouraged to contact me during the first two weeks of class regarding special needs. Students with disabilities should also contact Disabled Student Services in the Ley Student Center and the Rice Disability Support Services .

General Information

Course Syllabus
Homework Submission Guide

Office Hours	<table><tr><td>Eric</td><td>By Appointment</td><td>-</td><td>-</td></tr><tr><td>Corky</td><td>Wednesday Tuesday, Thursday</td><td>2pm-4pm 9:15am-10:30am</td><td>DCH 3110 DCH 3104</td></tr><tr><td>Sagnak</td><td>Thursday</td><td>9:15am - 11:15am</td><td>DCH 2062</td></tr><tr><td>Lechen</td><td>Wednesday</td><td>1pm - 2pm</td><td>DCH 2069</td></tr><tr><td>Chris</td><td>Tuesday</td><td>1pm - 3pm</td><td>Duncan Commons</td></tr><tr><td>Cannon</td><td>Monday</td><td>3pm - 5pm</td><td>Jones Commons</td></tr><tr><td>Jake</td><td>Wednesday</td><td>1:55pm - 3:55pm</td><td>Will Rice Commons</td></tr></table>
Textbooks	There is no required textbook. We will follow the pedagogic approach of "How to Design Programs" but in a typed context. We will also draw material from a variety of sources, including: • Felleisen, Findler, Flatt, Krishnamurthi. "How to Design Programs." MIT Press 2001. • Harold Abelson, Gerald Jay Sussman, Julie Sussman, "The Structure and Interpretation of Computer Programs." MIT Press 1985. • Odersky, Spoon, Venners. "Programming in Scala." Artima Press 2012. • Chiusano and Bjarnason. "Functional Programming in Scala." Manning Publications Co. August 2014. • Coursera: Functional Programming Principles in Scala by Martin Odersky. • edX: FP101x: Introduction to Functional Programming by Erik Meijer. • Okasaki. "Purely Functional Data Structures." Cambridge University Press. New York, NY. 1999. • The Apache Spark website. • Why is functional programming important ? by Corky Cartwright
Online Videos	• "Working Hard to Keep it Simple" by Martin Odersky • "Growing a Language," by Guy L. Steele, Jr.
Development Environment	• The DrScala Pedagogic IDE is recommended for all homework assignments in this course. • You may also use IntelliJ IDEA. See the setup instructions for working with Scala projects in this course.

Lecture Schedule (Subject to Change Without Notice)

Conditional Functions on Ranges, Point Values, and Compound Datatypes

Semantics of Type Checking, Binary Methods, Abstract Datatypes

For Expressions, Monads, The Environment Model of Reduction

Call-by-Name, Environment Model of Type Checking, Generative Recursion

Week	Day	Date	Topic	Work Assigned	Work Due
1	Tues	Aug 23	Overview, Motivation		
	Thur	Aug 25	What are Types, Core Scala	Hwk 0	
2	Tues	Aug 30	Doubles, Programming with Intention, The Design Recipe		
	Thurs	Sep 01	Functions on Ranges, Point Values, Compound Datatypes		
3	Tues	Sep 06	Methods, Grading, DrScala		
	Thur	Sep 08	Abstract Datatypes	Hwk 1	
4	Tues	Sep 13	Subtyping of Arrow Types, Exceptions		
	Thur	Sep 15	Abstract Datatypes 2, Recursively Defined Types		
5	Tues	Sep 20	Recursively Defined Types 2, Functions as Values		
	Thurs	Sep 22	Higher-Order Functions	Hwk 2	Hwk 1
6	Tues	Sep 27	Functions as Values, Parametric Types		
	Thur	Sep 29	Currying, Fold, Flatmap, and For Expressions		
7	Tues	Oct 04	For Expressions, Monads, The Environment Model		
	Thurs	Oct 06	"Growing a Language," Guy L. Steele, Jr.	Hwk 3	Hwk 2
8	Tues	Oct 11	MIDTERM RECESS		
	Thur	Oct 13	Scala Collections Classes, Traits		
9	Tues	Oct 18	Call-by-Name, Type Environments, Generative Recursion		

	Thur	Oct 20	Strategies for Generative Recursion	Hwk 4	Hwk 3
10	Tues	Oct 25	Accumulators		
	Thur	Oct 27	Functional Data Structures		
11	Tues	Nov 01	Streams, State, Mutation		
	Thur	Nov 03	Mechanical Proof Checkin, The Curry-Howard Isomorphism	Hwk 5	Hwk 4
12	Tues	Nov 08	The State Monad		
	Thur	Nov 10	Additional Scala Features, Extractors, Parser Combinators		
13	Tues	Nov 15	More Parser Combinators, Actors and Concurrency		
	Thur	Nov 17	Tactical Theorem Proving	Hwk 6	Hwk 5
14	Tues	Nov 22	Project Fortress		
	Thur	Nov 24	THANKSGIVING		
15	Tues	Nov 28	Functional Distributed Computing		
	Thur	Dec 01	Course Wrap Up		Hwk 6