

# COMP311-Fall-2015

**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 2015)

Instructor	Dr. Eric Allen	TAs	<ul style="list-style-type: none"><li>• Arghya "Ronnie" Chatterjee</li><li>• Yue Wang</li></ul>
Lectures	<a href="#">GRB W212</a>	Lecture Times	2:30PM - 3:45PM TR
Course Email	<a href="mailto:comp311@rice.edu">comp311@rice.edu</a>	Online Discussion	<a href="#">Piazza -- Rice Comp 311</a>

### 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

<a href="#">Course Syllabus</a>				
<a href="#">Homework Submission Guide</a>				
Office Hours	Eric	Tuesday	4PM - 5PM	DH 2062
	Ronnie	Friday	1:30PM - 2:30PM	DH 3002
	Yue	Thursday	4PM - 5PM	DH 3113

Textbooks	<p>There is no required textbook. We will follow the pedagogic approach of <a href="#">"How to Design Programs"</a> but in a typed context. We will also draw material from a variety of sources, including:</p> <ul style="list-style-type: none"> <li>• <a href="#">Felleisen, Findler, Flatt, Krishnamurthi. "How to Design Programs." MIT Press 2001.</a></li> <li>• <a href="#">Harold Abelson, Gerald Jay Sussman, Julie Sussman, "The Structure and Interpretation of Computer Programs." MIT Press 1985.</a></li> <li>• <a href="#">Odersky, Spoon, Venners. "Programming in Scala." Artima Press 2012.</a></li> <li>• <a href="#">Chiusano and Bjarnason. "Functional Programming in Scala." Manning Publications Co. August 2014.</a></li> <li>• <a href="#">Coursera: Functional Programming Principles in Scala by Martin Odersky.</a></li> <li>• <a href="#">edX: FP101x: Introduction to Functional Programming by Erik Meijer.</a></li> <li>• <a href="#">Okasaki. "Purely Functional Data Structures." Cambridge University Press. New York, NY. 1999.</a></li> <li>• <a href="#">The Apache Spark website.</a></li> </ul>
Online Videos	<ul style="list-style-type: none"> <li>• <a href="#">Guy L. Steele, Jr., Growing a Language</a></li> <li>• <a href="#">Martin Odersky, Working Hard to Keep Things Simple</a></li> <li>• <a href="#">Phil Wadler, Propositions as Types</a></li> </ul>
Development Environment	<ul style="list-style-type: none"> <li>• The <a href="#">DrScala Pedagogic IDE</a> is recommended for all homework assignments in this course.</li> <li>• You may also use <a href="#">IntelliJ IDEA</a>. See the <a href="#">setup instructions for working with Scala projects in this course</a>.</li> </ul>

## Lecture Schedule (Subject to Change Without Notice)

Week	Day	Date	Topic	Work Assigned	Work Due
1	Tues	Aug 25	Overview, Motivation, Core Scala		
	Thur	Aug 27	<a href="#">The Nature of Doubles, The Design Recipe</a>		
2	Tues	Sep 01	Type Checking, Conditional Functions, Compound Datatypes		
	Thurs	Sep 03	<a href="#">Grading, DrScala, Tests, Binary Methods, Operators</a>	Hwk 1	
3	Tues	Sep 08	<a href="#">Abstract Datatypes, Exceptions</a>		
	Thur	Sep 10	<a href="#">Exceptions, Overloading</a>		Hwk 1
4	Tues	Sep 15	<a href="#">Recursively Defined Datatypes</a>		
	Thur	Sep 17	<a href="#">First-Class Functions</a>	Hwk 2	
5	Tues	Sep 22	Functions as Values, Parametric Types, Covariance		
	Thurs	Sep 24	<a href="#">Checking Variance, For-Expressions</a>		
6	Tues	Sep 29	<a href="#">Translating For-Expressions, The Environment Model</a>		
	Thur	Oct 01	<a href="#">Lexical vs Dynamic Scoping, Call-by-Name, Traits 1</a>	Hwk 3	Hwk 2
7	Tues	Oct 06	<a href="#">Traits 2, Generative Recursion</a>		
	Thurs	Oct 08	<a href="#">More Generative Recursion, Accumulators</a>		
8	Tues	Oct 13	MIDTERM RECESS		
	Thur	Oct 15	<a href="#">More Accumulators, Tail Recursion</a>	Hwk 4	Hwk 3
9	Tues	Oct 20	<a href="#">Leftist Heaps, Functional Red-Black Trees</a>		
	Thur	Oct 22	<a href="#">Red Black-Trees Continued, Stream Processing</a>		
10	Tues	Oct 27	<a href="#">Guest Lecture: Shams Imam: Coroutines Scala-Project</a>		
	Thur	Oct 29	<a href="#">Variable Assignment and Environments</a>		
11	Tues	Nov 03	<a href="#">Mutable Objects, Equality, Memoization</a>		
	Thur	Nov 05	<a href="#">The State Monad, Mechanical Proof Checking</a>	Hwk 5	Hwk 4
12	Tues	Nov 10	<a href="#">Programs as Proofs and the Curry Howard Isomorphism</a>		
	Thur	Nov 12	<a href="#">Additional Scala Features, Extractors, Parser Combinators</a>		
13	Tues	Nov 17	<a href="#">More Parser Combinators, Actors and Concurrency</a>		
	Thur	Nov 19	<a href="#">Tactical Theorem Proving</a>	Hwk 6	Hwk 5
14	Tues	Nov 24	<a href="#">Guest Lecture: Robert "Corky" Cartwright: The Y Combinator</a>		
	Thur	Nov 26	THANKSGIVING		
15	Tues	Dec 01	<a href="#">Functional Distributed Computing</a>		

	Thur	Dec 03	<a href="#">Course Wrap Up</a>		Hwk 6
			<a href="#">Homework 6 Survey Results</a>		