2018-Fall

COMP 311 / COMP 544: Functional Programming (Fall 2018)

Instructors	Dr. Nick Vrvilo (2) Dr. Corky Cartwright	**TA**	Ryuichi Sai ryuichi@rice.edu
Lectures	Duncan Hall 1075	**Lecture Times**	4pm–5:15pm TR
Instructor Email	{nick.vrvilo,cork}@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. There will also be a final exam, as described in the syllabus.

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

Instructors			
Nick	Tuesday, Thursday	Available after class	—
Corky	Tuesday, Thursday	8:30am–10:30am	DCH 3104
	Wednesday	1:00pm–3:00pm	DCH 3104
Teaching Assistants			
Ryuichi	Monday	11am-noon	DCH 3109
	Tuesday, Thursday	noon-1pm	

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. • "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. • *What to Leave Implicit,* by Martin Odersky • *Impromptu: A Lightweight, dependently-typed async framework for Scala*, by Jon Pretty
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	Tu	Aug 21	Overview, Motivation		
	Th	Aug 23	Computation by Reduction, Types	Homework 0	
2	Tu	Aug 28	Core Scala, Doubles, Error Conditions		
	Th	Aug 30	Programming with Intention, The Design Recipe		
3	Tu	Sep 04	Conditionals, Functions on Ranges & Point Values, Compound Data		
	Th	Sep 06	Methods, Objects, Grading		Homework 0
4	Tu	Sep 11	Abstract Datatypes & DrScala	Homework 1	
	Th	Sep 13	Abstract Datatypes (cont.), Recursively Defined Types		
5	Tu	Sep 18	Recursively Defined Types (cont.), Functions as Values		
	Th	Sep 20	1st-Class Functions, Imports		
6	Tu	Sep 25	Variable Arity, Named Arguments, Exceptions, Format Strings	Homework 2	Homework 1
	Th	Sep 27	Exceptions, String Formatting, Generic Types		
7	Tu	Oct 02	Type Hierarchy, Variance, Generic Map Function		
	Th	Oct 04	Fold, Zip, Flatten, For Expressions		
8	Tu	Oct 09	Midterm Recess (no classes)		
	Th	Oct 11	Operators	Homework 3	Homework 2
9	Tu	Oct 16	Accumulators		
	Th	Oct 18	Call-by-Name/Value, Scala Immutable Collections		
10	Tu	Oct 23	*Growing a Language*, by Guy L. Steele, Jr. Midterm Exam: 7–10pm in DH1075		
	Th	Oct 25	Call by Name, Traits, Mixins		
11	Tu	Oct 30	Scala Parser Combinators		
	Th	Nov 01	Streams	Homework 4	Homework 3
12	Tu	Nov 06	Monads, For-expression desugaring		

	Th	Nov 08	...		
13	Tu	Nov 13	Additional Scala Features		
	Th	Nov 15	Semantics of Exceptions	Homework 5	Homework 4
14	Tu	Nov 20	Videos: *What to Leave Implicit* and *Impromptu*		
	Th	Nov 22	Thanksgiving Holiday (no classes)		
15	Tu	Nov 27	State Monad		
	Th	Nov 29	Course Wrap-up		Homework 5
16	Tu	Dec 04	Study Day (no classes)		
	F	Dec 07	Final Exam: 2–5pm in Duncan Hall 1075		

* Lectures slides not yet updated from last year are marked with an asterisk.

** The place and time of the final exam is set by the registrar. The current scheduling details are available on the registrar's page for this course.