2019-Fall

COMP 311 / COMP 544: Functional Programming (Fall 2019)

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, Wednesday	8:30-10:30am	DCH 3104
	Thursday	1:30-3:00pm	DCH 3104
Teaching Assistants			
Ryuichi	Monday	11:30am-noon	DCH 3113
	Wednesday	After 5pm by appointment only	

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	• IntelliJ IDEA is recommended for all homework assignments in this course. See the setup instructions for working with Scala projects. • We are not recommending DrScala this semester due to incompatibility with Scala version 2.13. • For interactive coding we'll use a BeakerX Notebook rather than a traditional Scala REPL or an IntelliJ Scala Worksheet.

Lecture Schedule (Subject to Change Without Notice)

Week	Day	Date	Lecture Topic and Resources	Work Assigned	Work Due
1	Tu	Aug 27	Overview, Motivation		
	Th	Aug 29	Computation by Reduction, Types, Core Scala	Homework 0	
2	Tu	Sep 03	Ints, Doubles, Error Conditions, Programming with Intention		
	Th	Sep 05	The Design Recipe BeakerX Notebook: Source, PDF		
3	Tu	Sep 10	Conditionals, Functions on Ranges & Point Values, Compound Data BeakerX Notebook A: Source, PDF BeakerX Notebook B: Source, PDF		
	Th	Sep 12	Methods, Objects, Grading		
4	Tu	Sep 17	Abstract Datatypes	Homework 1	Homework 0
	Th	Sep 19	Recursively Defined Types		
5	Tu	Sep 24	Functions as Values		
	Th	Sep 26	1st-Class Functions, Imports		
6	Tu	Oct 01	Named Arguments, Varargs, String Interpolation, Packages	Homework 2	Homework 1
	Th	Oct 03	Generic Types, Type Hierarchy, Variance Supplement: Producers and Consumers		
7	Tu	Oct 08	Type Hierarchy, Variance, Generic Map Function		
	Th	Oct 10	Fold, Zip, Flatten, For Expressions		
8	Tu	Oct 15	Midterm Recess (no classes)		
	Th	Oct 17	Scala Immutable Collections, Call by Name	Homework 3	Homework 2
9	Tu	Oct 22	Monads, For-expression desugaring		
	Th	Oct 24	Operators, Accumulators BeakerX Notebook: Source, PDF		
10	Tu	Oct 29	Video: *Growing a Language*, by Guy L. Steele, Jr. Exam 1 at 7pm in DCH 1064		
	Th	Oct 31	Scala Parser Combinators		
11	Tu	Nov 05	Lazy and Infinite Sequences		
	Th	Nov 07	Semantics of Exceptions	Homework 4	Homework 3

12	Tu	Nov 12	Traits and Mixins BeakerX Notebook A: MTG with Mixins – Source, PDF BeakerX Notebook B: Stackable Mixins – Source, PDF		
	Th	Nov 14	Additional Scala Features		
13	Tu	Nov 19	State Monad BeakerX Notebook: Source, PDF		
	Th	Nov 21	Video: *What to Leave Implicit*, by Martin Odersky	Homework 5	Homework 4
14	Tu	Nov 26	No Class		
	Th	Nov 28	Thanksgiving Holiday (no classes)		
15	Tu	Dec 03	Course Wrap-Up		
	Th	Dec 05	Exam 2 (in class)		Homework 5
16	Tu	Dec 10	Study Day (no classes)		
	Sat	Dec 14	Final exam/project date scheduled by university (last day to to submit assignments for this course)		

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