2023-Fall

- Created by Robert Cartwright, last modified on Aug 22, 2023

COMP 311 / COMP 544: Functional Programming (Fall 2023)

Syllabus	Online Book	Racket HW Guide	Racket HW Grading	Java HW Guide	HW Support Documents

Instructor	Robert "Corky" Cartwright		
Lectures	DCH 1064	**Lecture Times**	9:25 am–10:40 am TuTh
Instructor Email	cork@rice.edu	**Online Discussion**	Piazza – Rice Comp 311

Brief Description

This class provides an introduction to functional programming. Functional programming is a style of programming in which computations are solely expressed in terms of applications 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, parallel (multi-core) programming, and distributed computing. Coursework consists of a series of programming assignments in the Racket and Java programming languages followed by a discussion of frameworks for reasoning about functional and imperative programs supported by written homework assignments.

Grading, Honor Code Policy, Processes, and Procedures

Grading will be based on your performance on weekly programming assignments and two exams: a midterm and a final. 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

Office Hours

Instructor			
Corky Cartwright	TuTh	3pm-4pm	DCH 3104
		By appointment	TBA
Teaching Assistants			
Barret Glass	TBA	Jones Commons	TBA
Hunena Badat	TBA	Zoom	TBA

Textbooks and articles	There is no required textbook. We will follow the pedagogic approach of "How to Design Programs, First Edition" and extend it to other languages. The Second Edition of this book is the default at the website www.htdp.org but this web page contains a link to the first edition (at URL: https://htdp.org/2003-09-26/) at the bottom of the page. The two editions are very similar but this course tracks the first edition. We will draw material from a variety of sources, including: • Felleisen, Findler, Flatt, Krishnamurthi. "How to Design Programs, First Edition" MIT Press 2001. • Robert Cartwright, "The Elements of Object-Oriented Design", Unpublished notes. • John Hughes, "Why Functional Programming Matters", Online version from Chalmers dating from 1984; republished in 1989 and 1990. • 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. • Cartwright, "Why is Functional Programming Important?" (Slides skimming the advanced material covered in Comp 411)
Recommended 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
Development Environment	• DrRacket is recommended for all Racket homework assignments in this course. The interface is "textually transparent" as we will show in class. • DrJava is the supported IDE for Java in this course; it supports essentially the same textually transparent interface for Java that DrRacket does for Racket. You are also welcome to use a "professional" IDE such as IntelliJ or Eclipse, which have important features (particularly with regard to program refactoring) that DrJava lacks. • For students who want to explore Scala, a putative successor to Java that directly supports functional programming, there are multiple online IDEs and interpreters available.

Lecture Schedule (In Progress)

1	Tu	Aug 22	Motivation and the Elements (Constants) of Racket	Skim HTDP First Edition, Part 1 (Ch 1-8), Part 2 (Ch 9-10)	
	Th	Aug 24	Conditionals, Function Definitions, and Computation by Reduction		
2	Tu	Aug 29	Conditionals, Function Definitions, and Computation by Reduction	Homework 1 Review Ch 8 HTDP Part 2 (Ch 9-10)	Sep 04
3	Th	Aug 31	The Program Design Recipe for Racket, which focuses on using recursion to process lists and natural numbers	Preface, 9.4 HTDP Part 2 (Ch 11-13)	
4	Tu	Sep 05	Data Definitions, Data-driven Structural Recursion,	Homework 2 HTDP Part 3	Sep 11
5	Th	Sep 07	Mutually Recursive Definitions and Help Functions	HTDP Ch 15-17	
6	Tu	Sep 12	Local Definitions and Lexical Scope	Homework 3 HTDP Parts 5-6	Sep 18
7	Th	Sep 14	Lambda the Ultimate and Reduction Semantics	LawsOfEvaluation	
8	Tu	Sep 19	Functional Abstraction and Polymorphism		
9	Th	Sep 21	Functions as Values	Homework 4	Sep 28
10	Tu	Sep 26	Generative (Non-structural) Recursion	Homework 5 (long)*	Oct 11
11	Th	Sep 28	Lazy Evaluation and Non-strict Constructors		
12	Tu	Oct 03	Techniques for Implementing Lazy Evaluation		
13	Th	Oct 05	A Glimpse at Imperative Racket and Memoization	Sample Exam	
	Tu	Oct 10	Fall Recess	Sample Exam Key	

14	Th	Oct 12	On to Java! Midterm (Through Lecture 13 and HW 5) 7-10 pm	Homework 6 OO Design Notes	Oct 23
14	Tu	Oct 17	Adapting the HTDP Design Recipe to Java		
15	Th	Oct 19	Higher-order Functional Programming in Java	Homework 7	Oct 26
16	Tu	Oct 24	Four Key Idioms for Encoding FP in Java		
17	Th	Oct 26	The Singleton and Visitor Patterns	Homework 8	Nov 1
18	Tu	Oct 31	Java Generics and Their Role in FP in Java		
19	Th	Nov 02	Reasoning About Functional Programs	Homework 9*	Nov 8
20	Tu	Nov 07	First-order Programming Logic (an analog of ACL2 [UT Austin])		
21	Th	Nov 09	Theorem Proving Strategies	Homework 10	Nov 15
22	Tu	Nov 14	Hoare Logic		
23	Th	Nov 16	imperative Loop Invariants vs. Contracts for Help Functions	Homework 11	Nov 27
24	Tu	Nov 21	Reasoning About Procedure Calls		
25	Tu	Nov 28	Hoare Logic Applied to OO Code		
26	Th	Nov 30	The Future of FP and Programming Logic		

*Assignments marked with * are double assignments that count twice as much as regular assignments.