Course: COMP 311/544
Term: Fall 2020
Lecture Format: Online using Zoom
Lectures: Tu/Th 9:40–11:00am

Course Title

Functional Programming

Instructor Contact Information

Name: Robert "Corky" Cartwright
Office: Online and Duncan Hall 3104
Email: cork@rice.edu

Teaching Assistants

Name: Agnishom Chattopadhyay
Office: Online
Email: agnishom@rice.edu

Name: Chunxiao Liao
Office: Online
Email: cl117@rice.edu

Name: Andrew Obler
Office: Online
Email: ajo2@rice.edu

Course Objectives and Learning Outcomes

This class provides an introduction to the functional model of computation which serves as the core of nearly all modern programming languages and in selected languages like Haskell, their entirety. In the functional model of computation, a program simply consists of a finite set of function definitions and a body that is an expression constructed from defined and primitive functions. (Conventional constants can be viewed as functions of no arguments.) Most functional programming languages are *higher-order*: functions can be passed as parameters to other functions. Many contemporary mainstream languages are not higher-order but they typically support some form of data package (*e.g.*, objects in Java) that enable higher-order functions to be formulated as ordinary (first-order) functions that take data packages representing functions as arguments. The functional model of computation has a long informal history in mathematics that was ultimately formalized by Alonzo Church and his students in a language called the Lambda Calculus in the 1930's, predating the creation of digital computers.

Functional programming has become increasingly popular in recent years because:

- it fosters a *systematic* approach to program design;

- it enables many computations to be expressed more simply at a modest cost in execution overhead;
- It facilitates the decomposition of computations into independent subcomputatoins that can be efficiently mapped to multiple cores or, in very large scale problems, clusters of cores; and
- it simplifies the design of many modern applications with web interfaces because at any point in evaluating a thread, the rest of any computation can be formulated as a function called a *continuation*.

Functional programs are easy to reason about because theorems describing the behavior of functions on common types of data can readily be proved formally using a combination of equational reasoning (substituting equals for equals) and structural induction.

Required Texts and Materials

There are no required commercial textbooks or software tools for the class. All of the texts that we will use are freely available online. All of the software tools that we will use are open source applications that are supported on Windows 7 and 10, Mac OS X, and Linux.

Grade Policies

Grading will be based on your performance on weekly programming/written assignments (50%) and two exams (a midterm [25%] and a final [25%]). Late assignment submissions will not be accepted but each student will be given an allotment of 7 slip days that can be used in whole day increments to postpone the due dates for assignments. Slip days are best saved for the more challenging assignments given late in the course. Some extra credit points may be available in selected programming assignments and exams.

Attendance Policies

All course lectures and discussions will take place online. Lectures will be delivered using Zoom at the specified meeting time for the course (TuTh at 9:40am-11:00am). Unless geographic time differences make it difficult, students are expected to listen to lectures in real-time as they are being given. In this context students can ask questions during lecture providing interactive explanations and feedback. If a student misses a lecture, he or she is expected to view the lecture asynchronously on Zoom as soon as possible..

Rice Honor Code

In this course, all students will be held to the standards of the Rice Honor Code, a code that you agreed 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 at http://honor.rice.edu/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. Each graded assignment will stipulate what resources students are allowed to use in completing the assignment.

Homework and Exam Submissions: All submitted homework and exam submissions are expected to be the result of your own personal effort. You are free to discuss course material and approaches to problems with your other classmates, the teaching assistants and the professor, but you should never misrepresent someone else's work as your own. If you use any material from external sources, you must provide proper attribution. You should not share your solutions or make them publicly available.

Disability Support Services

If you have a documented disability or other condition that may affect academic performance you should: 1) make sure this documentation is on file with Disability Support Services (Allen Center, Room 111 / adarice@rice.edu / x5841) to determine the accommodations you need; and 2) talk with me to discuss your accommodation needs.

Syllabus Change Policy & Course Web Site

This syllabus is only a guide for the course and is subject to change with advance notice. The latest syllabus information for the course will always be available at the course website.

This term we will be using Piazza for class discussion. The system is designed to foster quick responses to questions by the instructor, teaching assistants, and classmates. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza.

Find our class page at https://wiki.rice.edu/confluence/display/FPSCALA/2020-Fall