Course Objective: This course is an upper-division undergraduate CS course on the functional style of programming. The course is also appropriate for graduate students interested in learning about functional programming. Functional programming is often considered to be esoteric by mainstream software engineers, but there are deep insights and advanced practical programming techniques to be learned from a course on functional programming that you will not typically learn in any depth in other CS courses. This course brings together many of the theoretical concepts and programming skills that you have learned in your other courses up to this point in your studies. The emphasis will be on using surprisingly powerful higher-order programming constructs and learning advanced programming abstractions. You will learn about the lambda calculus, an alternative to Turing machines as a formulation of computation, you will learn about typeful programming, and you will learn how to implement read-eval-print-loop (repl) interpreters using an elegant technique called monadic parsing. The primary functional programming language to be studied is Haskell, a lazily evaluated, statically typed “pure” language with static type inferencing capabilities. Time permitting, we will also study some elements of Scheme, an eagerly evaluated, dynamically typed language directly modeled on the lambda calculus. At the completion of this course, you will think differently about programming and software engineering, even if you never program in a functional language again after this course.

Prerequisites: The following courses are required prerequisites: CS 107 and CS 161. If you lack one of these prerequisites, please see the instructor before registering for the course. Prior experience programming with a functional language (e.g., SML, O’Caml) or a scripting language (e.g., Python, Ruby) may be helpful but is not required.

Textbooks: There is one required textbook and one optional but recommended textbook:

Required: Programming in Haskell, by Graham Hutton.
Optional: Real World Haskell, by Bryan O’Sullivan, John Goerzen and Don Stewart.

The Real World Haskell book is also available online at http://www.realworldhaskell.org. All other material will be provided as handouts or you will be given a pointer to online material, such as http://www.haskell.org.

Lectures: Regular class attendance is to your advantage as a lot of information will be provided that is not on the lecture notes. Lecture notes are handed out at the beginning of each class. Electronic copies of the lecture notes are placed on the course web page by the end of each class day. It is your responsibility to keep up with the course lecture material by attending each class.

Office Hours: Office hours will be held as stated above unless otherwise announced in class or via email. You may also request an appointment in person, by telephone, or via e-mail.

Email: Feel free to send questions via email to the instructor or TA or class email alias. The instructor and/or the TA will respond as soon as possible.
Homework: Homework assignments will consist of assigned readings and problem sets (including short programs) that will be assigned regularly.

Programs: One of the primary goals of this course is to gain basic familiarity with different functional programming language concepts and their manifestation in one or more functional languages. To achieve this goal, there will be 4-5 programming assignments. The instructor assumes that you have access to a Stanford or personal Linux, MacOS™ X or Windows™ computer system for programming. Language systems used in this course are available on multiple platforms. Information for how to obtain language systems will be provided in class. We will use the Glasgow Haskell Compiler (GHC) system, version 6.10.1. I suggest you use the Eclipse IDE, version 3.2.2, for your Haskell programming assignments with the eclipseFP plug-in framework for functional programming. For Scheme, we will use PLT Scheme, which includes its own graphical IDE. See:

http://www.haskell.org/ghc
http://www.eclipse.org
http://eclipsefp.sourceforge.net
http://www.plt-scheme.org

You may discuss the general nature of an assignment with classmates, the TA and the instructor, but you are to write the actual programs on your own, without assistance from classmates, other people, or resources other than documentation of language features in books or online tutorials. Using Internet search engines to search for solutions to assignments is expressly forbidden, but you are encouraged to look at examples from the textbook and online tutorials or sample code as an aide to learning how to use a functional language.

Exams: There will be a mid-term examination and a final examination according to the following schedule. The exams will be based upon material from the lectures, assigned readings, homework assignments, and programming assignments. The instructor grades all exams. Questions regarding the grading of the midterm exam must be submitted in writing by the next class meeting after the exam is returned to the student.

<table>
<thead>
<tr>
<th>Mid-Term Exam</th>
<th>Wednesday, May 6th. Course material covered up to this date.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Exam</td>
<td>Monday, June 8th. Comprehensive but weighted towards 2nd half of course</td>
</tr>
</tbody>
</table>

Grading: Each exam counts 25%, programs collectively count 35%, homework assignments collectively count 15%. The final grade is assigned based on the General University Grading System. Final grades will be calculated based on a standard grading scale: A range: 90-100, B range: 80-89, C range: 70-79, D range: 60-69. The instructor reserves the right to set the lower-bound for each grade scale in a consistent manner, but you are guaranteed that the lower bound will not increase. For example: a 90 final course average guarantees a grade in the A range, but a 89.49 final course average may not qualify for a grade in the A range. However, if 89.49 qualifies for a grade in the A range, then 79.49 qualifies for the B range, 69.49 qualifies for the C range, and 59.49 qualifies for the D range.

Important Dates: Monday May 25th is Memorial Day so there will be no class that day. All other important Spring Quarter dates are on the web at http://registrar.stanford.edu/academic_calendar/

Code of Conduct: Students should be fully aware of Stanford University’s policy on academic honesty. Policies on handling academic dishonesty will be followed by the instructor and TA.

Extenuating Circumstances: If you have difficulty meeting the requirements of this course because of extenuating personal circumstances, you are to notify the course instructor at the earliest possible date so that your unique situation can be discussed in confidence. If you encounter an unexpected medical or family emergency, or a random act of Nature, that will cause you to unexpectedly miss a class assignment or exam, you must notify the instructor at the earliest possible date in order to qualify for any possible special consideration.