CS 421 Syllabus

Programming Languages

Summer 2024

Course Description

This course has four major themes.

In the first part of the course you will learn how to program in a functional programming language Haskell, some of the mathematics behind functional programming, and how to write an interpreter using Haskell.

In the second part of the course you will learn about parsing, or how to transform text into the data structures that our interpreters need to work.

In the third part of the course you will learn about the mathematics used to reason about the meanings of programs and programming languages. In the final part you will learn about some of the tradeoffs available in different languages and the effects those choices can have.

Course Prerequisite

- CS 233
- CS 374

In particular, we expect you to be familiar with recursion, time complexity, and state machines; and be able to make use of formal mathematical notations.

Course Goals and Objectives

What really goes into a language? Why is one language better than another (or all the all the same)? After this course you will know how to pick the right language for a given task and have the skills to implement that language if you need to. We want you to become more than a consumer of programming languages: we want you to be a producer.

By the end of the course, you will be able to:

- Learn to program in a functional programming language
- Understand the evaluation of programs in a functional programming language
- Be able to write a type checker / type inferencer given a formal type system
- Be able to write a lexer and a parser using recursive-descent parsing
- Program translation: be able to write a syntax-directed translator from abstract syntax to intermediate representations
- Recognize and use major methods of specifications of dynamic syntax
- Write an interpreter based on the formal optional semantics of a language
- Identify different design decisions available in programming languages and reason about their effects

Textbook and Readings

We do not have an official textbook for the course. However, many students have found the online book *Learn You a Haskell for Great Good* to be useful in learning the implementation language of the course. Also, some weeks will have written notes prepared by the instructor, and also some optional readings taken from research literature for students motivated to dig deeper.

Course Outline

There are three sections to this course. There is an MCS online section (DSO) that is taken by remote students and delivered via Coursera. There are two "on campus" sections. One is for graduate students (PG) and the other for undergraduates (PU). The sections are referred to as "on campus", but because everything is online those will be delivered via Coursera as well. The main difference that students will notice is how exams are proctored. Section DSO is worth four credit hours. Section PG has a four hour option (in which case it is just like section DSO) or a three hour option. Section PU only has a three hour option. The four hour sections have all the same assignments as the three hour sections, and also require a final project.

This course is 12 weeks long. You should expect to invest somewhere between 12–16 hours a week, but of course individual timings can vary greatly.

Week	**Start Date**	**Topics**
1	May 13	• Getting started with Haskell • Recursion • Higher order functions
2	May 19	• Algebraic Data Types • Implementing an evaluator for with basic arithmetic • Implementing closures and function calls
3	May 26	• Big Step Semantics • Lambda Calculus • Continuation Passing Style
4	June 2	• Type Classs • Monads • The State Monad
5	June 9	• Exam 1 (June 7 – 14) • Grammars • Regular Languages
6	June 16	• LL Parsing • LR Parsing • Combinator Parsing
7	June 23	• Small Step Semantics • Typing Semantics • Unification
8	June 30	• State • Objects • Hoare Logic
9	July 7	• Prolog • The Cut Operator
10	July 14	• Exam 2 (July 12 – 19)
11	July 21	• Variables
		• Parameters • Metaprogramming
12	July 28	• Final Exam (July 26 – Aug 2)

Assignment Deadlines

For all assignment deadlines, please refer to the Course Assignment Deadlines, Late Policy, and Academic Calendar page.

Elements of This Course

The course is comprised of the following elements:

About the course: this 4-credit hour course is 8 weeks long. The course is composed of two parts, with the part 1 starting from week 1 to week 3. There will be an exam in week 4, covering topics from weeks 1 to 3. Part 2 starts from week 4. There is a second exam in week 7, covering topics from week 4 to 6. The last day of the course will have a final exam (cumulative) to cover content in the entire course. For details about each component of the course, see the course component description below.

Lecture Video: in this course, content will be presented as lecture videos. Each week's content is broken into several short-duration videos. Each lecture video is no longer than 30 minutes. You may stream lecture videos for playback within the broswer by clicking on their titles, or you may download the lecture videos and watch them offline. You can also download each lecture's slides by clicking the Resources tab in each lecture video.

Any typos or corrections in the slides will be in an `dso-errata.pdf` document posted near this syllabus, since the coursera platform does not have a convenient way to post corrections next to the videos themselves.

Quizzes/Activities: Each week will feature one or two quizzes designed to reinforce the content of the week. The format will vary; for example, the quiz could be a few multiple-choice questions, or a small programming assignment. You will be allowed unlimited attempts for each graded quiz with your highest attempt score used toward your final grade. Your top 10 scores will be used to calculate the final grade. (i.e. the instructor will drop your lowest 2 quiz scores) There is no time limit on how long you take to complete each attempt at the quiz. Graded quizzes will be used when calculating your final score in the class.

Machine Problems: There are six programming assignments, called machine problems (MPs). You are allowed to get help on the machine problems, but you need to cite your help.

General Exams and Final Exam (proctored via ProctorU): There are three proctored exams in the class. Each is a 1-hour exam. These exams will be broken into mastery areas. The final exam will retest some of these mastery areas: if you show an improvement in a particular mastery area, we will upgrade your general exams score using your final score. The exams can be only taken during a specific window.

Final Project If you are taking this course for four credit hours, you need to complete a final project. Instructions and timelines will be posted on the course web site.

Important note: all assignments in this course are completed via PrairieLearn. Once an assignment is completed, the grading information will be transferred back to Coursera automatically. Links of each assignment to PrairieLearn will be provided in each week.

Please note, in order to access course materials and assignments, you will need to pay the Coursera fee ($158) for this course in addition to the tuition.

Grading Distribution and Scale

Your final grade will be calculated based on the activities listed in the table below. Your official final course grade will be listed in Enterprise. The course grade you see displayed in Coursera may not match your official final course grade.

Assignment	Number	Percentage (4 hour)	Percentage (3 hour)
Quizzes	12*	10	12.5
Machine Problems	6	30	37.5
General Exams	2	20	25
Final Exam	1	20	25
Final Project	1	20	0
Total		100	100

- For quizzes, the lowest two scores are dropped.
- Most graduate students will take this course for 4 hours, most undergraduates will take this for 3 hours.

Letter Grade	Percent
A+	97
A	93
A-	90
B+	87
B	83
B-	80
C	70
D	60
F	Below 60

Student Code and Policies

A student at the University of Illinois at the Urbana-Champaign campus is a member of a University community of which all members have at least the rights and responsibilities common to all citizens, free from institutional censorship; affiliation with the University as a student does not diminish the rights or responsibilities held by a student or any other community member as a citizen of larger communities of the state, the nation, and the world. See the University of Illinois Student Code for more information.

Academic Integrity

All students are expected to abide by the Campus Regulations on Academic Integrity found in the Student Code of Conduct. These standards will be enforced and infractions of these rules will not be tolerated in this course. Sharing, copying, or providing any part of a homework solution or code is an infraction of the University's rules on academic integrity. We will be actively looking for violations of this policy in homework and project submissions. Any violation will be punished as severely as possible with sanctions and penalties typically ranging from a failing grade on this assignment up to a failing grade in the course, including a letter of the offending infraction kept in the student's permanent university record.

Again, a good rule of thumb: Keep every typed word and piece of code your own. If you think you are operating in a gray area, you probably are. If you would like clarification on specifics, please contact the course staff. Disability

Accommodations

Students with learning, physical, or other disabilities requiring assistance should contact the instructor as soon as possible. If you're unsure if this applies to you or think it may, please contact the instructor and Disability Resources and Educational Services (DRES) as soon as possible. You can contact DRES at 1207 S. Oak Street, Champaign, via phone at (217) 333-1970, or via email at disability@illinois.edu.