CS 558: Software Foundations
Fall 2020

Matthew R. Lakin

August 11, 2020

Course information

Lectures

Day/time: Tuesdays and Thursdays 11:00–12:15
Location: Classes will take place on Zoom.

Instructor

Matthew Lakin
Email: `mlakin@cs.unm.edu`
Office hours: Tuesdays 3–5pm
Location: Office hours will take place on Zoom.

Course delivery

This class will be offered via the Remote Scheduled modality for Fall 2020. This means that classes will be streamed on Zoom at the scheduled class meeting time. Classes will also be recorded and uploaded to UNM Learn to enable offline viewing. Additional class materials will also be uploaded to UNM Learn where appropriate. Office hours will also take place on Zoom. **Zoom access information for classes and office hours will be posted to the class UNM Learn page.**

COVID-19 statement

Due to the COVID-19 pandemic, UNM and the School of Engineering will require students, staff, and faculty to follow all health guidelines of the New Mexico Department of Health. Specifically, students, faculty and staff will adhere to social distancing guidelines, will wear masks in all buildings on campus (students are expected to provide their own masks), will maintain a distance of at least six feet from others, and will wash their hands frequently when on campus.

Course topics and format

This course studies the theory used to describe and define programming languages and to guide their implementation. Our approach is type-based, in the spirit of our textbook, Pierce's Types and Programming Languages (TAPL). As a prelude, the course offers a brief overview of functional programming techniques and of programming language features found in the purely functional programming language Haskell.

The course is intended for first-year graduate students, but advanced undergraduates are welcome as well. No specific courses are prerequisites, but programming experience and mathematical maturity are necessary. Experience with functional programming (at the level of UNM CS357) and discrete mathematics is strongly recommended.

The course will provide students with the background they need for CS550.

The course consists of lectures, homework assignments (primarily programming based), quizzes, two mid-term examinations, and a final examination.

Assignments

There will be two midterm exams, and a final exam covering the entire course. These may be administered on paper or via UNM Learn.

Programming-based homework assignments may be given: in the early part of the course these tasks will be drawn from the general domains of mathematics, science, and engineering, to practice programming skills; in the later part of the course the tasks will correspond to implementation of programming language theory. Short written homework assignments may be given to consolidate lecture material; they may take the form of short algebraic proofs of program fragment equivalence, or consideration of small language extensions. Homework assignments will be submitted via UNM Learn.

There will also be regular quizzes administered online via UNM Learn.

Textbooks

- Benjamin C. Pierce, Types and Programming Languages, MIT Press, 2002, ISBN-10: 0262162091.
- Miran Lipovača, Learn You a Haskell for Great Good, No Starch Press, 2011, ISBN-13: 978-1-59327-283-8. Accompanying webpage: http://learnyouahaskell.com.

Grading

You are expected to attend class regularly, read any assigned reading before class, and participate in class discussion. The grade will be determined as follows:

- Homeworks: 40% total
- Exams: 50% (15% for each midterm, 20% for the final)
- Quizzes: 10%

Note that no requests for grade changes will be considered after the final day of classes. There will also be no extra credit assignments or "do-overs" for homeworks, exams, or quizzes.

Communication

The Loboweb email list functionality will be used for administrative announcements. Lecture notes and homework assignments will be uploaded to the UNM Learn page for the class.

Topics

The topics covered in class will be a subset of the following:

- Topics in functional programming
  - functional programming and Haskell
  - prelude types and classes
  - functions and list comprehensions; unit testing; literate programming; interactive programs
  - recursive and higher-order functions
  - declaring types and classes
  - lists in depth: map, filter, and their algebraic laws

  - lists in depth: foldr, scanr, and their algebraic laws
  - trees with folds, binary heap trees, rose trees
  - efficiency: accumulating parameters, tupling, fusion, and deforestation
  - modules and abstract data types
  - lazy evaluation and infinite data structures; approximation ordering; cyclic structures; streams
  - monads
  - metaprogramming in a functional programming language
- Topics in programming language semantics
  - syntax
  - operational semantics
  - simple imperative languages
  - lambda calculus syntax and reduction
  - programming in the lambda calculus
  - combinators and combinator reduction
  - types
  - simply typed lambda calculus
  - simple extensions (ascription; let-bindings; records; variants; recursion)
  - references
  - exceptions
  - subtyping
  - recursive types
  - type reconstruction
  - unification
  - universal polymorphism
  - program transformations

Academic integrity statement

Each student is expected to maintain the highest standards of honesty and integrity in academic and professional matters. The University reserves the right to take disciplinary action, up to and including dismissal, against any student who is found guilty of academic dishonesty

or otherwise fails to meet the standards. Any student judged to have engaged in academic dishonesty in course work may receive a reduced or failing grade for the work in question and/or for the course.

Academic dishonesty includes, but is not limited to, dishonesty in quizzes, tests, or assignments; claiming credit for work not done or done by others; hindering the academic work of other students; misrepresenting academic or professional qualifications within or without the University; and nondisclosure or misrepresentation in filling out applications or other University records.

All students will be required to sign and submit a warning regarding issues of academic integrity and possible sanctions prior to any submissions being graded.

Accommodation statement

In accordance with University Policy 2310 and the Americans with Disabilities Act (ADA), academic accommodations may be made for any student who notifies the instructor of the need for an accommodation. It is imperative that you take the initiative to bring such needs to the instructor's attention, as I am not legally permitted to inquire. Students who may require assistance in emergency evacuations should contact the instructor as to the most appropriate procedures to follow. Contact Accessibility Resource Center at 277-3506 for additional information.

If you need an accommodation based on how course requirement interact with the impact of a disability, you should contact me to arrange an appointment as soon as possible. At the appointment we can discuss the course format and requirements, anticipate the need for adjustments and explore potential accommodations. I rely on the Disability Services Office for assistance in developing strategies and verifying accommodation needs. If you have not previously contacted them I encourage you to do so.

Title IX statement

In an effort to meet obligations under Title IX, UNM faculty, Teaching Assistants, and Graduate Assistants are considered "responsible employees" by the Department of Education (see pg 15 - http://www2.ed.gov/about/offices/list/ocr/docs/qa-201404-title-ix.pdf). This designation requires that any report of gender discrimination which includes sexual harassment, sexual misconduct and sexual violence made to a faculty member, TA, or GA must be reported to the Title IX Coordinator at the Office of Equal Opportunity (http://oeo.unm.edu). For more information on the campus policy regarding sexual misconduct, see: https://policy.unm.edu/university-policies/2000/2740.html