Preliminary version of 13 August 2024

Course Information

Lectures

Lecture day/time: Tuesdays and Thursdays, 3:30–4:45
Lecture location: CENT-1026

Instructor

Darko Stefanovic
Email: `darko@cs.unm.edu`
Office hours: Tuesdays and Thursdays, 2:30-3:30
Office hours location: Farris Engineering Center 2020

Teaching assistant

None

Course topics and format

The course explores the theory and practice of programming in the purely functional programming language Haskell.

The course is intended for graduate students and advanced undergraduates. Experience with functional programming (at the level of UNM CS357) is required. Facility with discrete mathematics is recommended. Students outside CS should consult the instructor beforehand.

The course consists of lectures, exercises, and programming projects. Readings include the textbook and research articles.

Course objectives

At the completion of this course students will be able to:

1. *Construct* computer programs in a pure functional programming language to *solve* various application problems.
2. *Apply* compiler algorithms to symbolic input data processing in various application domains.
3. *Design* and *program* an interpreter for a programming language.

Textbook

Graham Hutton, *Programming in Haskell*, 2nd Ed., Cambridge University Press, 2016, ISBN-13: 978-1316626221.

Other useful books

Rebecca Skinner, *Effective Haskell*, 2023.

Christopher Allen and Julie Moronuki, *Haskell Programming from First Principles*, 2016.

Richard Bird, *Thinking Functionally with Haskell*, 2015.

Grading

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

- Class participation 30%
- Programming projects 70%

Communication

Canvas will be used for administrative announcements. Lecture notes and assignments will be uploaded to the Canvas page for the class.

Assignment hand-in policy

Assignments are to be submitted on-line, in Canvas. Detailed instructions will be provided with each assignment.

List of topics (subject to revision)

- Advanced review of pure functional programming
  - untyped lambda calculus
  - combinators
  - Haskell type system
  - equational reasoning / calculational proofs
  - list functionals and their algebraic laws
  - origami programming
  - efficiency

  - lazy evaluation
  - purely functional data structures (queus, leftist heaps, zipper, finger trees, etc.)
  - algorithmic reasoning
  - continuation-passing style
  - defunctionalization

- Monads and other standard type classes in Haskell
  - category theory
  - `(>>=)` and return
  - list monad
  - Maybe monad
  - `(>=>)` and Kleisli morphisms
  - State monad
    - threading global state
    - get and put
    - randomness
  - Parser monad
    - recursive descent parsing
    - parsing combinators
  - other various useful monads
  - type class Functor
  - type class Applicative
  - type class Monoid
  - guard and type class MonadPlus
  - fold and type class Foldable
  - traverse and type class Traversable
  - Monad Transformers
    - Writer monad and WriterT monad transformer
    - StateT monad transformer
    - lift and typeclass MonadTrans
    - typeclass MonadState
    - liftIO and typeclass MonadIO

- Further explorations
  - algebraic effect systems
  - abstract machines
  - programs and proods
  - Liquid Haskell
  - generalized algebraic data types
  - proof assistants

Credit-hour statement

This is a three credit-hour course. Class meets for two 75-minute sessions of direct instruction for fifteen weeks during the Fall 2024 semester. Students are expected to complete a minimum of six hours of out-of-class work (or homework, study, assignment completion, and class preparation) each week.

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 these 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 entire 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.

UNM statement of compliance with ADA

Every instructor should include an official statement in their course syllabus. The suggested syllabus statement should include the following text:

"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[s] 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."