Preliminary version of 1 June 2025

Course Information

General

This course counts for 3 credits. Students outside CS should consult the instructor beforehand. Prerequisite: CS 251L.

Course objectives

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

1. Design and program algorithms in a purely functional programming language.
2. Apply purely functional algorithms to symbolic data processing in various application domains.

Lectures

times to be announced

Instructor

Darko Stefanovic, office hours to be announced

Teaching assistant

to be announced

Electronic device policy

Laptops, tablets, phones, and all screens must be turned off before entering the classroom and must remain off during class time.

Topics and format

In Fall 2025, the course will be an informal introduction to functional programming techniques, using the language Haskell. The course will consist of lectures, exercises, four midterm exams, and a final exam.

Textbook

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

Other assigned reading materials will be provided free of charge.

Grading

You are expected to attend class regularly, *read the assigned reading before class*, and participate in class discussion. The grade will be determined as follows: 20% each midterm exam, 20% final exam.

Grading option change requests will not be considered after the last class period.

Topics (tentative)

1. Week 1: introduction to functional programming and Haskell: prelude types and classes (textbook chapters 1–3)
2. Week 2: functions and list comprehensions; polymorphism (chapters 4–5)
3. Week 3: recursive and higher-order functions (chapters 6–7)
4. Week 4: declaring types and classes (chapter 8)
5. Week 5: lists in depth: map, filter, foldr, and their algebraic laws
6. Week 6: trees with folds, binary heap trees, rose trees
7. Week 7: interactive programming (chapter 10)
8. Week 8: algebraically structured programming: functors, applicatives, and monads (chapter 12)
9. Week 9: monadic parsing (chapter 13)
10. Week 10: foldables (chapter 14)

11. Week 11: lazy evaluation and infinite data structures (chapter 15)
12. Week 12: reasoning about pure programs (chapter 16)
13. Week 13: survey of other functional programming languages

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."