Preliminary version of 14 July

Course Information

Registering

For the Fall 2008 semester, this course is listed as CS591 (section 30056) and CS491 (section 30051).

Lectures

Tuesdays and Thursdays, 2:00–3:15, in Mechanical Engineering 208

Instructor

Darko Stefanovic, office hours Mondays 2:20–3:00 and Tuesdays 3:20–4:00 in ECE 236C

Teaching assistant

None

Course topics and format

The course offers an introduction to modern programming techniques and programming language features, as well as an introduction to the theory used to describe and define programming languages. The course is intended for first-year graduate students, but advanced undergraduates are welcome as well. No specific prerequisites are needed. The course consists of lectures, written assignments, programming assignments, two mid-term examinations, and a final examination. Programming assignments will be carried out in teams of two.

Assignments

Two midterm exams, final exam (covering the entire course), several short written homework assignments to consolidate lecture material, several programming assignments.

Textbooks

Required reading

Benjamin C. Pierce, *Types and Programming Languages*, MIT Press, 2002, ISBN-10: 0262162091.

Optional reading

Graham Hutton, *Programming in Haskell*, Cambridge University Press, 2007, ISBN-10: 0521871727.

Richard Bird, *Introduction to Functional Programming*, Prentice Hall, 2nd edition, 1998, ISBN-10: 0134843460

Robert Harper, *Practical Foundations for Programming Languages* (working draft on-line)

Michael L. Scott, *Programming Language Pragmatics*, Morgan Kaufmann, 2nd edition, 2005, ISBN-10: 0126339511

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:

Homeworks 50%

Exams 50% (15% each midterm exam, 20% final)

Homework and programming assignment hand-in policy

Late assignments will be penalized $3n^3$%, where n is the number of days late.

Lecture Plan

1. organizational; Haskell introduction
2. prelude types and classes
3. functions and list comprehensions; unit testing; literate programming
4. recursive and higher-order functions
5. declaring types and classes
6. interactive programs
7. lists in depth: map, filter
8. lists in depth: foldr, scanr
9. trees with folds
10. binary heap trees, rose trees
11. efficiency: lazy evaluation, accumulating parameters, tupling
12. efficiency: fusion and deforestation

13. modules and abstract data types
14. infinite data structures; approximation ordering; cyclic structures; streams
15. monads in depth
16. combining monads
17. syntax
18. operational semantics
19. lambda calculus syntax and reduction
20. programming in the lambda calculus
21. combinators and combinator reduction
22. types
23. simply typed lambda calculus
24. simple extensions (ascription; let-bindings; records; variants; recursion)
25. type reconstruction
26. unification
27. universal polymorphism

Mailing list

A mailing list will be used for class discussion. It may also be used for administrative announcements.

UNM statement of compliance with ADA

Qualified students with disabilities needing appropriate academic adjustments should contact the instructor as soon as possible to ensure their needs are met in a timely manner. Handouts are available in alternative accessible formats upon request.