

Preliminary version of ?

Course Information

Course structure for Spring 2008

The course focuses on programming language concepts such as binding, evaluation, and types, exhibited on various representative programming languages, with an emphasis on the functional and logic programming paradigms. Students will gain working knowledge of several representative programming languages.

Initial familiarity with the procedural and functional paradigm is assumed, as described below.

Here is a selection of topics that may be treated in the course:

- Logic Programming:
 - Declarative and logic programming in general.
 - Prolog search, unification, resolution.
 - Impure features of Prolog.
- Typed Functional Programming:
 - Purely functional programs in Haskell.
 - The type system of Haskell.
 - Type reconstruction.
 - Higher-order functions for list processing.
 - Data types and their operators.
 - Programming in the point-free style.
 - Lazy evaluation.
 - Interaction with the environment, input-output, and state.
- Programming Language Design and Implementation:
 - Language syntax:
 - * Introduction to scanning and parsing techniques.
 - Language semantics:
 - * Introduction to semantics specification techniques.
 - Language implementation:
 - * Overview of interpretation and compilation.
 - * Exercises in small-language interpretation.

Programming

The course involves *extensive programming assignments*.

For Prolog, our reference implementation is SWI-Prolog version 5.6.14. For Haskell, our reference implementation is GHC version 6.6.1.

Assignments

The course grade will be determined by several (up to 5) short in-class quizzes (10% of the cumulative grade), a mid-term exam (15%), a final exam (covering the entire course) (25%), and 5 written homework assignments (50%).

Prerequisites in detail

Experience with developing substantial applications is required; UNM CS course 351 *Design of Large Programs* provides appropriate background (in Java). Exposure to functional programming is required; UNM CS course 257 provides appropriate background (in Scheme).

Undergraduate students entering 451 are expected to have completed all core CS classes at the 200 and 300 level, in accordance with the rules of the UNM CS undergraduate program. *Graduate* students entering 451 are expected to have taken courses equivalent to UNM CS courses 257 (programming in Scheme), 241 (programming in C), 361 and 362 (algorithms and data structures), and 251 and 351 (object-oriented programming).

Lectures

Mondays & Wednesdays, 2:00 - 3:15, in ME 300.

Instructor

Darko Stefanovic, office ECE 236C, phone 2776561, email darko — office hours Mondays & Wednesdays, 1:00-2:00.

Teaching assistant

Bryan Cheng, email bryan — office hours TBA

Mailing list

A mailing list will be used for class discussion amongst the students and for questions to the instructor and the teaching assistant. It may also be used for administrative announcements. See <http://www.cs.unm.edu/cgi-bin/mailman/listinfo/cs451>.

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.

Reading material

Required reading

William F. Clocksin and Christopher S. Mellish: *Programming in Prolog. Using the ISO Standard*, 5th edition, Springer Verlag, 2003, ISBN 3540006788.

Graham Hutton: *Programming in Haskell*, Cambridge University Press, 2007, ISBN 0521692695.

Homework and programming assignment hand-in policy

Homework assignments are due at midnight of the date assigned. Late homework will be accepted, but it will be penalized $2n^2\%$, where n is the number of days late. All work will be handed in electronically; hand-in mechanisms will be specified with each assignment.

Programming questions on homework assignments

When an assignment asks you to write a program, that means that you must design a program, type it into a computer, compile, and run, and then submit a listing of the program and its output, making sure that your set of test inputs is carefully designed. The textual layout of the program must be logically sound and aesthetically pleasing. The names used must be descriptive. Code comments and additional documentation must accompany non-trivial programs, and must provide informal argumentation that the program satisfies the specification. Occasionally you may be asked to provide a formal correctness proof of a program—include that proof as part of program documentation.

Cooperation and cheating

Feel free to *discuss* homework assignments with classmates and the instructor. However, *do not look at or copy another's solution*. If a problem appears too difficult, or you lack the background to solve it, you are expected to talk to the instructor promptly. Once you have the background necessary to solve a problem, you must provide your own solution. Exchanging homework solutions is cheating and will be reported to the University administration; students involved may not be permitted to continue in the class. You are responsible for exercising due diligence in protecting your homework files from unauthorized access. In case two or more students present essentially similar homework, all involved students will be reported to the University administration. Occasionally we will reuse problems that have been used in previous courses. Searching on the web for verbatim solutions is not the best application of your time, and you will receive no credit for it.

Each assignment handed in must be accompanied by the following statement: *“I pledge my honor that in the preparation of this assignment I have complied with the University of New Mexico Board of Regents’ Policy Manual.”*

Of particular relevance is Section 4.8, which reads as follows.

4.8 Subject: ACADEMIC DISHONESTY**Adopted: September 12, 1996****Applicability**

This policy applies to all students at the University with regard to academic activities and professional activities related to academic work.

Definition

“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.

Policy

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 who otherwise fails to meet the expected 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.

Implementation

The President may establish administrative policies and procedures for implementing this policy, which shall be published in the Pathfinder and the Faculty Handbook, together with this policy.