

Advanced Programming Languages

Prerequisites: Permission of instructor.

Web-page: <http://web.unbc.ca/~casper/Semesters/2013S/720.php> (not yet created)

Instructor: David Casperson; **Office:** T&L 10-2040; **Phone:** 960-6672;
Administrative Assistant: Marva Byfield; **AA's Phone:** 960-6490;
e-mail: David.Casperson@unbc.ca .

Lecture times: as arranged. This is a reading course.

Grading Scheme: There will be no formally assigned examinations.

Office Hours: To be scheduled.

| | |
|----------------------|-----|
| Homework: | 90% |
| Language Comparison: | 10% |

Text Books: none required.

References: [5, 3] are good online books. [7, 10, 6] are definitive language references.

Students are expected to report on material read, and solve mutually agreed upon homework problems as invented by the instructor.

Course Content: The calendar says:

Topics for this course may include advanced study of general programming language design concepts, formal reasoning about programs and languages, pragmatic evaluation of language properties, and case studies of specific languages. The course may be used to communicate programming language theory and practice specific to students' project or thesis research needs.

The course contains an introduction to logic and functional programming through examination of particular languages including Standard ML ([11, 8, 1, 3, 9]), Haskell ([5, 4]), Scheme ([2, 10]), and Prolog ([?]). The goal of this course is to use illustrations from particular languages to introduce more fundamental ideas such as the distinction between strict and lazy programming languages and the difference between statically-typed and dynamically-typed languages. Particular attention is paid to modelling semantics (in particular monads and game-theoretic semantics), concurrency, and other topics relevant to individual students' research.

References

- [1] Matthias Felleisen and Dan Friedman, *The little MLer*, MIT Press, 1998, in the UNBC library.
- [2] Daniel P. Friedman and Matthias Felleisen, *The little schemer (4th ed.)*, MIT Press, Cambridge, MA, USA, 1996.
- [3] Robert Harper, *Programming in Standard ML*, working draft ed., Carnegie Mellon University, August 2002, available from <http://www-2.cs.cmu.edu/~rwh/smlbook/>
- [4] P. R. Hudak and J. H. Fasel, *A gentle introduction to haskell*, ACM SIGPLAN Notices **27** (1992), no. 5, 1–53.
- [5] Miran Lipovaca, *Learn you a haskell for great good!: A beginner's guide*, 1st ed., No Starch Press, San Francisco, CA, USA, 2011.
- [6] Simon Marlow, *Haskell 2010 language report*.
- [7] Robin Milner, Mads Tofte, Robert Harper, and David Macqueen, *The Definition of Standard ML - Revised*, rev sub ed., The MIT Press, May 1997.
- [8] Larry C. Paulson, *ML for the working programmer*, second ed., Cambridge University Press, 1996, This is more in-depth than Ullman's book.
- [9] Riccardo Pucella, *Notes on programming in in Standard ML of New Jersey*, Cornell University, January 2001, also available from <http://www.smlnj.org/doc/literature.html#tutorials>.
- [10] Michael Sperber, R. Kent Dybvig, Matthew Flatt, Anton van Straaten, Robby Findler, and Jacob Matthews, *Revised [6] report on the algorithmic language scheme*, 1st ed., Cambridge University Press, New York, NY, USA, 2010.
- [11] Jeffrey D. Ullman, *Elements of ML programming*, ML97 (second) ed., Prentice Hall, 1998.
- [12] P. L. Wadler, *Comprehending monads*, Proceedings of the 1990 ACM Conference on LISP and Functional Programming, Nice (New York, NY), ACM, 1990, pp. 61–78.