

Functional and Logic Programming

Prerequisites: A grade of C⁻ or better in CPSC 141, and CPSC 281; or permission of instructor.

Accommodations If there is any student in this course who, because of a disability, may have a need for special academic accommodations, please come and discuss this with me, or contact the Disability Services Centre located in the Teaching & Learning Building in Room 10-1048.

Web-page: <http://web.unbc.ca/~casper/Semesters/2011F/370.php>

Instructor: David Casperson; **Office:** T&L 10-2040; **Phone:** 960-6672; **Departmental Assistant:** Shannon Norum (T&L 10-2074 960-6490); **e-mail:** David.Casperson@unbc.ca .

Lecture times: MWF 08:30–09:20.
Room: 5-159. There are *no* assigned lab or tutorial times.

Office Hours: Scheduled for TW 13:00–14:00.

Text Books: None are required. [8] is recommended for SML.

Grading Scheme:

Homework:	25%	
Midterm 1:	20%	Wed, Oct 03
Midterm 2:	20%	Wed, Nov 07
Final Exam:	35%	3h in 2012-12-05 to 2012-12-15

I reserve the right to change the weight of any portion of this marking scheme. If changes are made, your grade will be calculated using the original weighting and the new weighting, and you will be given the higher of the two.

Programming Assignments: There will be approximately weekly programming assignments during the semester. Programming languages include SCHEME, PROLOG and STANDARD ML.

Course Content: An introduction to functional programming. Static versus dynamic typing. Strict versus non-strict evaluation. Some common functional programming languages. “Pure” versus “impure” functional programming.

Functions and partial functions. Cartesian products. Disjoint unions. “Currying”.

STANDARD ML. Builtin types and literals. Tuples. Lists. Declarations. Function declarations and function values. Product types and function types.

Recursion, tail recursion, and accumulator arguments. Higher order functions for lists.

SCHEME

Space and time complexity for functional programs and data structures.

An introduction to logic programming. PROLOG.

Facts. Rules. Goals. Variables. Conjunctions. Horn Clauses. The Unification algorithm. Accumulator arguments. Difference lists. Cuts. Negation. Arithmetic. Debugging.

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*, fourth ed., The MIT Press, 1996.
- [3] ———, *The seasoned Schemer*, The MIT Press, 1996.
- [4] Robert Harper, *Introduction to Standard ML*, Carnegie Mellon University, 1993, This is being replaced by [5]. Available from <http://www.smlnj.org/doc/literature.html#tutorials>
- [5] ———, *Programming in Standard ML*, working draft ed., Carnegie Mellon University, August 2002, available from <http://www-2.cs.cmu.edu/~rwh/smlbook/>
- [6] Larry C. Paulson, *ML for the working programmer*, second ed., Cambridge University Press, 1996, This is more in-depth than Ullman's book.
- [7] 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>.
- [8] Jeffrey D. Ullman, *Elements of ML programming*, ML97 (second) ed., Prentice Hall, 1998.