Functional and Logic Programming

Prerequisites: A grade of C$^-$ or better in CPSC 142, 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/2010F/370.php`

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

Lecture times: MWF 12:30–13:20. **Room**: 5-159. There are *no* assigned lab or tutorial times.

Office Hours: Tentatively scheduled for W 14:30-15:20, R 14:00-15:00.

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

References:

Grading Scheme:

Homework:	25%	
Midterm 1:	20%	Wed, Oct 13
Midterm 2:	20%	Wed, Nov 10
Final Exam:	35%	3h in 2010-12-06 *to* 2010-12-17

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 Assigments: 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 programning 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.