

Functional and Logic Programming



Web-pages: <https://web.unbc.ca/~casper/Semesters/2022-05F/370.php>
<https://moodle.unbc.ca/>

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

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

Accommodations: Students in this course who, because of a disability, may have a need for special academic accommodations should discuss this with the instructor, or contact the Access Resource Centre (<http://www.unbc.ca/access-resource-centre>) located in the Library Building in Room 5-157.

Grading Scheme and Dates:

Homework:	25%	
First class:		2022-09-07
Midterm 1:	20%	Fri, Oct 14
Midterm 2:	20%	Wed, Nov 16
Last class:		2022-12-06
Final Exam:	35%	in 8-19 Dec

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

Lecture times: MWF 11:30-12:20. **Room:** 5-122. There are *no* assigned lab or tutorial times.

Office Hours: Scheduled for:

M W 10:00-11:00

Please feel to interrupt me whenever you can find me, or schedule alternate times.

Text Books: None are required. [5] is a great introduction to HASKELL conveniently found on the web. [6, 1] are both in the library.

References

- [1] W. F. Clocksin and C. S. Mellish, *Programming in prolog*, Springer Verlag, 1981.
- [2] Matthias Felleisen and Dan Friedman, *The little MLer*, MIT Press, 1998, in the UNBC library.
- [3] Daniel P. Friedman and Matthias Felleisen, *The little Schemer*, fourth ed., The MIT Press, 1996.
- [4] ———, *The seasoned Schemer*, The MIT Press, 1996.
- [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] Richard A. O'Keefe, *The craft of prolog*, Logic Programming, MIT Press, 1990.
- [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] 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.

Students who successfully complete CPSC 370 will:

- have an appreciation of the functional logic programming paradigms;
- be cognizant of the associated theory and language implementation techniques (for instance, the λ -calculus, closures, unification, and thunks),

and consequently will be stronger programmatic problem solvers in whatever programming environment they work.

Approximate Course Content:

GENERAL

- Characteristics of functional programming languages,
- Basic syntax.
- Language concepts: Static and dynamic typing. Strict and non-strict evaluation. “Pure” versus “impure” languages.
- Mathematical concepts: Relations, Functions, partial functions, Cartesian products, Disjoint unions, “Currying”, categories, monads, combinators
- Programming: recursion, tail recursion, tail recursion strategies.

HASKELL:

- Types, polymorphic types, type classes
- data structures.
- immutability and its consequences
- Laziness: its implementation consequences
- Monads, do-notation, monadic programming.

PROLOG

- What logic programming is.
- Facts. Rules. Goals. Variables. Conjunctions. Horn Clauses.
- The Unification algorithm.
- Programming strategies: Accumulator arguments, difference lists.
- Cuts. Negation.
- Arithmetic.