

Advanced Functional Programming

Prerequisites: Permission of instructor.

Web-page: <http://casper.unbc.ca/Semesters/2019F/499.php> (not yet created)

Instructor: David Casperson; **Office:** T&L 10-20800; **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:	30%
Report:	20%
Participation:	50%

Text Books: none required.

References: [2] is a good online book. Many others and definitive language references can be found through the `haskell.org` web-site.

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

Course Content: The course principally explores advanced HASKELL ([2, 1]) programming: · exploring strictness, · advanced pattern-matching syntax, · purely functional data structures, · standard type classes (`Monoid`, `Foldable`, `Traversable`, `Functor`, `Applicative`, monads, and monadic transformers), · polymorphic recursion, · fixed-point operators at both the value and type level,

The goal of this course is to explore the connection between effective programming and various HASKELL-based abstraction techniques.

Topic selection and order are driven partly by the instructor's whim, partly by student consensus. Students are expected to follow up on meeting discussions and find related materials and ideas.

References

- [1] P. R. Hudak and J. H. Fasel, *A gentle introduction to haskell*, ACM SIGPLAN Notices **27** (1992), no. 5, 1–53.
- [2] Miran Lipovaca, *Learn you a haskell for great good!: A beginner's guide*, 1st ed., No Starch Press, San Francisco, CA, USA, 2011.
- [3] 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.