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 Access Resource Centre located in Teaching & Learning 10-1048.


Instructor: David Casperson; Office: T&L 10-2040; Phone: 960-6672; Departmental e-mail: David.Casperson@unbc.ca

Lecture times: MWF 13:30–14:20. Room: 5-159. There are no assigned lab or tutorial times.

Office Hours: Scheduled for:
Mon 11:00-12:00
Mon 14:30-16:00.

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

Grading Scheme:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>25%</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>20%</td>
</tr>
<tr>
<td>Midterm 2</td>
<td>20%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>35%</td>
</tr>
</tbody>
</table>

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 Haskell.

Approximate Course Content:

GENERAL
- An introduction to functional programming and languages.
- Mathematical concepts: Relations, Functions, partial functions, Cartesian products, Disjoint unions, “Currying”.
- Programming: recursion, tail recursion, tail recursion strategies.

HASKELL:
- Basic syntax.
- Types, polymorphic types, type classes
- Creating data structures.
- Exploiting laziness. Downsides to laziness.
- Monads, do-notation, monadic programming.

SCHEME:
- Basic syntax.
- Space and time complexity for functional programs and data structures.
- the pure \( \lambda \)-calculus.
- Combinators.

PROLOG
- An introduction to logic programming.
- The Unification algorithm.
- Programming strategies: Accumulator arguments, difference lists.
- Cuts. Negation.
- Arithmetic.
- Debugging.
References


