CSI 3120 Programming Language Concepts  
(3 units)  
Course Components: Lecture, Laboratory, Tutorial  

Fall 2018  
Course Outline  


Course Objectives  
(see Chapter 1 of the Mitchell textbook):  

- To understand the design space of programming languages, which includes concepts and constructs from past programming languages as well as those that may be used more widely in the future. Also, to understand some of the major conflicts and trade-offs between language features, including implementation costs.  
- To develop a better understanding of the languages we currently use by comparing them with other languages.  
- To understand the programming techniques associated with various language features. More generally, to study conceptual frameworks for problem solving, software construction, and development via the study of programming languages.  
- To revisit functional, imperative, and object-oriented features via the use of the OCaml programming language.  

Professor:  
Dr. Amy Felty  
SITE 5-068  
afelty@uottawa.ca  

Lectures:  
- Simard Hall (SMD), Room 425  
- Wednesday 10:00–11:30  
- Friday, 8:30–10:00  

Laboratory/Tutorial  
- LAB Group 1: Monday, 17:30–18:50, SITE 0130  
- LAB Group 2: Tuesday, 13:00–14:20, SITE 2060  
- TUT: Tuesday, 11:30–12:50, Fauteux Hall (FTX), Room 147A  

Office Hours:  
Tuesday 9:00–11:00
Required Textbooks:


Evaluation:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>30%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>45%</td>
</tr>
</tbody>
</table>

Assignments 30% (approximately 6 or 7, must be done individually)
Midterm Exam 25% (will take place during tutorial time)
Final Exam 45%

Course Web Page:

http://www.eecs.uottawa.ca/~afelty/csi3120/

Course Outline:

- An introduction to programming language concepts
- An introduction to OCaml
- Types and functional programming in OCaml
- Imperative programming in OCaml
- Object-oriented programming in OCaml
- Syntax and semantics
- Algol and ML (the prehistory of OCaml)
- Scope, procedures, and storage management
- Control in sequential languages
- Data abstraction and modularity
- Basics of concurrency
- Basics of scripting languages