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

Fall 2019  
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:  
- Wednesday 10:00-11:20, Morisset Hall (MRT), Room 218  
- Friday, 8:30-9:50, Colonel By Hall (CBY), Room C03  

Laboratory/Tutorial  
- LAB Group A01: Monday, 17:30–18:50, SITE 0130  
- LAB Group A02: Tuesday, 13:00–14:20, SITE 2060  
- LAB Group A03: Friday, 10:00–11:20, CBY B02  
- TUT Group A05: Thursday, 17:30–18:50, CBY B205  
- TUT Group A04: Friday, 16:00–17:20, CBY B205
Office Hours:     Tuesday 14:00–16:00

Required Textbooks:


Evaluation:

Assignments 30% (approximately 6 or 7, must be done individually)
Midterm Exam 25%
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
- Inductive data types
- Polymorphic higher-order programming
- Data abstraction and modularity
- Mutable data and imperative programming
- Object-oriented programming
- Syntax and semantics
- Scope, procedures, and storage management
- Control in sequential languages
- Concurrent and distributed programming
- Combinators, pipelines, and scripting