Computing Science Course Outlines
CMPT 383 - D100 Comparative Programming Languages

Instructor(s): Toby Donaldson

Calendar Objective/Description:
Various concepts and principles underlying the design and use of modern programming languages are considered in the context of procedural, object-oriented, functional and logic programming languages. Topics include data and control structuring constructs, facilities for modularity and data abstraction, polymorphism, syntax, and formal semantics.

Instructor’s Objectives:
This course explores various concepts and principles underlying the design and use of modern programming languages. A major focus of the course will be on functional programming, an elegant and useful set of techniques that are slowly finding their way into many mainstream languages such as C++, JavaScript, Python, Swift, and more. Declarative programming in Prolog will be discussed, and, time permitting, we will see how the Go languages supports concurrency.

Prerequisites:
CMPT 225, and (MACM 101 or (ENSC 251 and ENSC 252)).

Topics:
- Lisp/Scheme: basic functional programming
- Haskell: typed functional programming
- Prolog: declarative programming
- Go: Concurrency (time permitting)

Grading:
The marking scheme will be provided in the first week of class.

Required Books:

Recommended Books:
Concepts of Programming Languages (11th Edition), Robert Sebesta, Addison-Wesley, 2015, 9780133943023, It is okay to use the 10th edition, if you prefer.


Reference Books:
Programming Language Pragmatics, Michael L. Scott, Morgan Kaufmann, 2015, 9780124104099, General discussion of programming language topics.

Academic Honesty Statement:
Academic honesty plays a key role in our efforts to maintain a high standard of academic excellence and integrity. Students are advised that ALL acts of intellectual dishonesty will be handled in accordance with the SFU Academic Honesty and Student Conduct Policies (http://www.sfu.ca/policies/gazette/student.html).