CMPT 383 - D100 Programming Langs.

Instructor(s): Gregory Baker

Calendar Objective/Description:
Programming Langs.

Instructor's Objectives:
The objective of this course is to give the student a better understanding of non-imperative programming, and other important distinctions between languages. Various concepts and principles underlying the design and use of modern programming languages are considered. We will take a detailed look at a pure functional programming language, and a language that promotes concurrency.

Online offering notes: you will need a computer with a webcam and reliable Internet access. The computer should be powerful enough to run a virtual machine: at least 8 GB memory, 20 GB disk, and a reasonably decent processor. There will be 3-4 quizzes during the semester which must be completed during the lecture time. Otherwise, lectures will be posted as a “watch party” where we can watch together (and ask questions in a forum), but they can also be viewed later.

Prerequisites:
see go.sfu.ca

Topics:
- Expressing algorithms functionally
- Functional programming in Haskell
- Type systems in programming languages
- Compilers, interpreters, and runtime environments
- Challenges and techniques in concurrent programming
- Concurrent programming in Go

Grading:
Weekly exercises 15%; assignments 35%; quizzes 24%; final project 26%. May be adjusted: if so, changes will be announced in the first week of the semester.

Students must attain an overall passing grade on the weighted average of exams in the course in order to obtain a clear pass (C- or better).

Recommended Books:
Programming in Haskell, Graham Hutton, Cambridge University Press, 9781316626221

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).