CMPT 379 - D100 Principles of Compiler Design

Instructor(s): Anoop Sarkar

Calendar Objective/Description:
Principles of Compiler Design

Instructor's Objectives:
As Steve Yegge said, “If you don't know how compilers work, then you don't know how computers work.” (http://steve-yegge.blogspot.ca/2007/06/rich-programmer-food.html) This is a course for those who are interested in the design and implementation of programming languages. Compilers let us to use a high-level programming language by translating programs into low-level machine code. Understanding how compilers work is essential if you want to be a good programmer. In this course, you will build a working compiler using lex, yacc and LLVM (it's ok if you don't know what those terms mean).

Prerequisites:
see go.sfu.ca

Topics:
- Overview of a compiler
- Lexical Analysis: regular expressions
- Simple Parsing: context-free grammars, top-down and bottom-up parsing
- LL(1) parsing: efficient top-down parsing
- Shift-reduce parsers: introduction to bottom-up parsing
- SLR/LR parsing: fast and efficient bottom-up parsing
- Type checking: checking semantics of programs
- Semantics and code generation: from a high-level language to assembly language
- Optimization: an introduction to various types of code optimization

Grading:
The grade distribution will be handed out at the start of classes.

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).

Reference Books:

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).