CMPT 479 - D100 Special Topics in Computing Systems

Instructor(s): Nick Sumner

Calendar Objective/Description:
Current topics in computing systems depending on faculty and student interest.

Instructor's Objectives:
How can we create software that is maintainable, reliable, and secure? How can we treat software systems as subjects for analysis? How can we automate challenging tasks like finding vulnerabilities or even programming itself? This course examines both classic and cutting edge answers to these software engineering questions. This course will explore modern aspects of software engineering including design, reliability, performance, and security. Beyond manual design and programming issues, students will gain experience with techniques for automating aspects of software engineering and treating programs themselves as data that can be analyzed, transformed, or automatically generated. This course is a trial run for future offerings of CMPT 745: Software Engineering. The material will be hands-on, with several small projects in a variety of programming languages throughout the semester. Students will also be expected to complete a term project in a direction of their choice based on material from the course. The term project will involve building a tool that automates some useful analysis/task within software engineering. Students are expected to learn core techniques used in program analysis and to ultimately apply them. CMPT 379 is recommended but not required. Introductory projects will involve programming in C++. Term projects can be done using a language of student preference.

Prerequisites:
CMPT 300.

Topics:
- Static and dynamic program analysis
- Software security (offense and defense)
- Automated program synthesis
- Automated test generation
- Concurrency and parallelism
- Automated debugging & defect investigation
- Performance analysis
- Classic design and architecture

Grading:
Assignments: 50% Exams: 25% Term Project: 25% Grading criteria are subject to change.

Reference Books:
Working Effectively with Legacy Code, 978-0131177055
Principles of Program Analysis, 9783540654100
Engineering a Compiler, 9780120884780
Writing Solid Code, 9781570740558
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).