# CPEN 400P: Program Analysis for Reliability and Security

## Staff and Contact:

**Instructor: Karthik Pattabiraman**

**Teaching Assistants:** Abraham Chan, Mohsen Salehi

**Website:** https://piazza.com/ubc.ca/winterterm22022/cpen400p/resources

## Course Logistics (Lectures, Office hours and Lab):

| What ? | When ? | Where ? |
|---|---|---|
| Lectures | Tuesdays and Thursdays (11 AM to 12:30 PM) | See schedule |
| Karthik's Office hours | Tuesdays ( 4 PM to 5 PM) | KAIS 4048 |
| Labs | Online | Online |

## Motivation

Software today pervades every aspect of our society, and is used in many critical scenarios from financial systems to medicine and automated driving. The consequences of a failure or malfunctioning of software are dire, and can lead to loss of money and human lives. Further, software often has to satisfy various requirements beyond functional correctness, such as reliability, security, quality and performance, in order to be deployed in practical scenarios. Therefore, it is challenging to ensure the quality and reliability of large-scale software systems.

This course will investigate the principles of systematic techniques to analyze large-scale software systems, and ensure that they satisfy their requirements. While software testing can check for functional correctness, it cannot ensure adherence to attributes beyond functional correctness. Further, testing by itself is necessarily incomplete, and hence cannot even guarantee functional correctness. Therefore, this course will investigate techniques beyond traditional software testing. Because of the large scale of software systems today, it is challenging to apply techniques that require manual intervention. The course will thus focus on scalable and automated techniques.

## Course Introduction

This is a fourth year elective course on program analysis. We will cover a variety of techniques from static analysis to dynamic analysis to fuzzing. The primary language we will use for the class is C/C++, though this is NOT a course on C/C++ or any other language. We will cover the following topics in class (rough outline):

- Static Analysis Techniques to analyze the program (source) code prior to its deployment.
- Dynamic Analysis Techniques to find violations that arise during the program execution.
- Fault Injection/Fuzz testing Techniques to find corner cases that lead to program failures.
- Advanced topics such as model checking, safety analysis, program monitoring and repair.

## Pre-requisites and background

The prerequisites for this class are **CPEN 221 and CPEN 321. You are responsible for all materials covered in those classes**.

However, we do not require any familiarity with program analysis or compilers to take this class, though of course, such familiarity will be helpful. **We will not cover C++ in this class – you will have to pick it up on your own from textbooks or online tutorials.** The assignments will require substantial amounts of C++ programming.

## Course Logistics and Policies

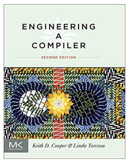
1. **The lectures will be conducted in person, with the exception of a few online classes (these will be announced later).** You are expected to attend all lectures - we'll not record lectures or stream them online.
2. Labs will be held mostly online - there'll be no need to show up for any of the labs either physically or virtually. Assignment submissions will be done online (via Github), and do not require physical presence.
3. The in-person class session will involve live problem-solving - not all the solutions will be posted online. Attendance will not be taken at these events, and they will NOT be recorded. Exam questions will be similar to these problems, and so you are strongly encouraged to attend them.
4. We will use Piazza for online discussions and for answering questions about the assignments. You must sign up for a Piazza account by the end of this week. Questions about the class must be posted to Piazza[1] and NOT emailed to the TAs or the professor (you can send a private note to "Instructors" if needed).
5. During Karthik's office hours, he will only answer questions about the class lectures and/or problem solving sessions. **No questions about the assignments will be answered during this time.**
6. Any questions on the assignments **must** be posted on Piazza as **public notes**, and must be addressed to all the "Instructors". These will be answered by TAs. No questions will be answered about the assignments **48 hours** before the deadline. **No solution code should be posted on Piazza or else you'll get a 0 for the assignment. Note that individual help on assignments will be provided only via the labs, not Piazza.**
7. The labs will serve as the TAs' office hours, and are the only way to get help with the assignments on a one-on-one basis. We will not set up any sessions outside of these lab times to help with the assignments.
8. Assignments will be done in groups of two. You may not share code or discuss solutions with any other group of students. We reserve the right to call upon you to explain the details of your group's solution. Failure to do so will result in you getting a 0 for the assignment, and be considered academic misconduct.
9. All deadlines are hard unless you have a documented emergency. In both cases, you may be called upon to produce documentation related to the nature of the emergency. The weightage of any missed component will automatically be moved to the final exam for you - there's no need to inform us or seek our permission.
10. There is to be no collaboration for assignments (except with your group partner), exams etc. - any violation of this policy will be treated as academic misconduct, and dealt with accordingly.
11. Finally, it is your responsibility to keep up with all course announcements, lectures and assignments. You are also expected to monitor Piazza on a regular basis. **No email will be sent for any announcements**.

## Course Grading

We will have regular programming assignments in this class (total of 5 assignments over 12 weeks). Each assignment will count for 10% of your total grade and will cumulatively constitute 50% of the course grade. The assignments will be due roughly every other week and will involve a substantial amount of C++ programming.

The course will have two exams –a midterm exam, and a final exam. These will account for 15% and 25% of your grade respectively. We will also have a programming proficiency test in class which will test basic knowledge of C++ programming – this will count for 5% of your grade. The proficiency test will be held approximately one week into the course. Finally, you will get participation points of up to 5% for online discussions on Piazza.

## Textbook



There is no textbook for this course, though the book "Engineering a compiler" by Cooper and Torczon  (see left image) is highly recommended, at least for the first part of the course. Lecture notes will be posted on the course web page approximately every week.  If you want a quick primer on C++, I recommend the free, online book "Thinking in C++, 2nd edition" by Bruce Eckel. It is also recommended that you familiarize yourself with the Unix command line and simple shell scripts.

[1] Like most online services, Piazza is hosted outside Canada. Please send me an email if you are uncomfortable using Piazza.