CPEN 400P: Program Analysis for Reliability and Security

Lecture 0: Course Orientation

(Term 2, 2021)

What this course is about ?

Program Analysis (i.e., static and dynamic analysis of software systems)

- Explores the use of automated analysis techniques
- Focus on finding bugs and vulnerabilities in software systems
- Focus is on techniques that go beyond testing (unit test, integration test etc.)
- Typically use techniques from compilers, runtime systems, formal methods

Goal is to provide assurance in software to minimize bugs and vulnerabilities

- Not to guarantee the absence of software defects (unlike formal methods)
- Not to improve software performance or resources (unlike compiler design)
- Not on techniques that involve human/social processes (e.g., code review)

Focus will be on automated analysis of **real-world, large-scale software systems**

Pre-Requisites

CPEN 221: Software Design

CPEN 321: Software Engg.

You should also be comfortable

1. Hacking large code bases in C++ as most of the assignments will require this
2. With algorithms and data-structures as most of the lectures will use these
3. Some level of mathematical maturity (but we'll not do formal proofs etc.)

NOTE: You do NOT need to have taken an undergraduate course on compiler design, though you may have to read up on some concepts as needed from them

Topics covered (non-exhaustive list)

Compilers and Intermediate Representations

Static analysis of programs

Dynamic analysis of programs

Symbolic execution

Fuzz testing

Resilience Evaluation and Security Analysis

Model-checking

Relationship to other courses

Undergraduate compiler design courses

- Typically focus on front-end concerns such as scanning, parsing etc.
- Focused on performance improvement and code generation
- Some overlap in terms of analysis (e.g., dataflow analysis, Alias analysis)

Software Testing course (e.g., CPEN 422)

- Typically focus on test generation (e..g, unit tests, integration testing)
- Systematically assess goodness of test cases via coverage etc.
- Some overlap in terms of code analysis and corner-case testing (e.g., fuzz)

Teaching Staff

Karthik Pattabiraman, Professor, ECE dept.

- Handle lectures, exams and course logistics
- First time I’m teaching this course (created it)

TAs

- Abraham Chan, PhD student
- Udit Agarwal, MASc student
- Handle all labs and assignements

Course Logistics

Lectures delivered in class (no recording of lectures, but slides will be available)

- We'll do class activities during regular lecture times in class (2-3 per class)
- Exam questions will be similar to the class activities for the most part
- Karthik will have office hours, but will NOT answer any questions about assignment

Attendance at labs optional, but they'll serve as TA office hours (no other OHs)

- Assignment submission and grading will be done via online submission

All course communication will be exclusively via Piazza (see last slide)

- We'll simply delete all emails sent to us, as also Canvas messages etc.
- It's your responsibility to keep up with course announcements on Piazza

Evaluation components

Programming Proficiency Test (5%)

Class Participation via Piazza (5%)

Exams (40%) - 15% Midterm Exam and 25% Final Exam

Assignments (50%) - Five assignments each counting for 10% of the marks

1. You do NOT need to pass the exams in order to pass the course
2. Weight of any component that you don’t submit will be moved to final exam
3. No solutions will be provided for the assignments

Programming Proficiency Test (PPT)

Tests basic knowledge of C++ programming

Doesn’t need much preparation if you’re familiar with C++ programming

Open book, open notes, open web - but no collaboration allowed with anyone

WIll be held in class on Jan 20th - we’ll use HackerRank online platform

Test cases will be provided; 5 programming questions; no partial credit

We’ll return grades to you the same day. If you can’t pass the PPT, you should seriously consider dropping the course as you won’t be able to do the assignments

Assignments

To be done in groups of 2 - it's up to you to choose your partner

Each assignment will require considerable time and effort

Assignments will require modifying large software packages (e.g., LLVM, KLEE etc.). You need to be comfortable with reading and modifying large code bases.

We'll be using Github to do the assignments and submit them (we'll take care of creating the repos and adding you to them. Detailed instructions will be posted.)

Questions about assignments may only be asked **publicly** on Piazza or during TA office hours (for one on one help). **No solution code** should be posted on Piazza.

Exams

Will be written in paper and pen (not online). Will be closed-book, closed-notes

Consist of problem solving as well as multiple choice questions

Problem solving questions will be similar to class activity (not the same of course)

Require you to understand and apply the material (memorization won’t help !)

No need to pass the exams individually to pass the course….

Class Participation

Piazza participation counts for class participation points (5%)

- Asking and answering questions on Piazza
- Both quantity and quality of posts important
- Reading Piazza posts doesn’t count for participation
- No rubric will be provided for this component
- Private posts should be only for personal situations (we’ll delete them if not)

NOTE: Though coming to class and participating in class doesn’t count for points, you’re encouraged to do so as the exams will test you on material covered in class

Course Policies

All deadlines are hard (no extensions will be given)

Any missed submissions will automatically be carried over to final exam weightage

No scaling of marks; no extra credit work

Collaboration in assignments with anyone except your partner is academic misconduct. Collaboration in exams with anyone is academic misconduct.

Both you and your assignment partner get the same grade for assignments. You are expected to pull your weight - we reserve the right to ask you alone questions

No email should be sent to the teaching staff whatsoever; only use Piazza.

Should you take this course ?

Do NOT take this course if you're looking for an easy elective in your 4th year

Do NOT take this course if you're not comfortable with writing significant code

Do NOT take this course if you don't want to invest considerable time in solving assignments, especially in learning the ins and outs of these software systems

Do NOT take this course if you're not comfortable with material in CPEN 221, 321

Do take this course if you want to learn a rich class of techniques, if you like algorithms and implementing them in real-world systems, and software building

Do take this course if you want to develop useful skills for (specialized) industry

TODOs in the next couple of weeks…

Join Piazza forum for the course - see link below. No signup code.

piazza.com/ubc.ca/winterterm22021/cpen400p

Find a partner to do the assignment with and let us know via private Piazza post

Start preparing for the PPT if you're not comfortable/need to brush up C++

Get familiar with Git and also install Clang, LLVM etc (tutorial provided online)