SWE 795 Fall 2017

"Program Analysis for Software Testing"

Fall 2017: Mondays, 4:30pm-7:10pm, Room 2026 Art and Design

Instructor: Prof. Jonathan Bell
Email: bellj@gmu.edu
Twitter: @_jon_bell_
Office: 4422 Engineering Building; (703) 993-6089
Office Hours: Anytime electronically, Mondays 3:30-4:30pm, or by appointment

Pre-Requisites:

Students are required to have previously taken a compilers course (E.g. CS 540) OR a testing class (e.g. SWE 637) OR special permission from the instructor to take this class. This class also assumes working knowledge of programming in Java and C/C++ (although most assignments will use Java). Students must also have working understanding of UNIX shell environments. **For CS MS students, this course will count towards the Programming Languages/Software Engineering Requirement.**

Objectives:

Learn different methods for analyzing software, with several applications in software engineering, particularly testing. We will study different analysis techniques, learn how they work by studying specific algorithms and tools, and discuss applications of the techniques. Our goal will be to explore the current research issues in this cutting edge area, to learn how to build software analysis tools, and to understand how these techniques can be applied to software development activities.

We will primarily focus on applications for testing software, including automatic test data generation. We will also consider using analysis techniques for other software related activities such as maintenance, reuse, metrics, and optimization. Some of the specific analysis techniques

to be studied are parsing, software representation methods (control flow graphs, data flow graphs, program dependency graphs), symbolic evaluation, constraints, program slicing, software coupling, and testability.

While we will consider analyses in various languages (e.g. applying to native x86 binaries, Java and JavaScript), there will be a particular emphasis towards Java analysis. Students will complete a research project involving dynamic analysis of Java applications.

Intended Audience:

This graduate-level course is intended for students in a MS degree program or in a PhD program. This course is intended for students who would like to become more productive software engineers in industry by applying analysis tools.

This is also intended for research-focused students who would like to prepare for research in programming languages, compilers or software engineering, or would like to prepare for research in other areas of computer science (E.g., security or systems) that involve applying software analysis and testing.

Learning Outcomes:

- Understand the abstract properties of different techniques for analyzing and testing software.
- Compute the outcome of these techniques on concrete code examples.
- Evaluate the suitability of different techniques for a given software and/or set of constraints.
- Run analysis and testing tools on actual software and interpret their results to improve the quality of their code.

Required Materials:

There is no required textbook for this class. Weekly readings will involve academic papers, which will be made available online. To complete the homeworks and project successfully, students will need access to a computer capable of running virtualization technology (e.g. VirtualBox).

Grading:

40% Homework
50% Final Project
5% Participation
5% Labs

PRELIMINARY SCHEDULE (subject to change, links will be added shortly):

Date	Topic	Reading (read before class)	Additional references
8/28	1. Introduction to program analysis; course overview; Lab: Java bytecode instrumentation HW1 out, due 9/5	**Bring your laptop so you can complete the lab in class.**	
9/4	(No class) labor day		
9/5	HW1 due. (No class) last day to drop class with no financial penalty		
9/11	2. Dynamic dataflow analysis and taint tracking Lab: Phosphor HW2 out, due 9/18	Phosphor, DyTAN	
9/18	3. Concurrency Lab: Concurrency HW2 due; HW3 out, due 9/25	Hybrid data race detection, TBA	
9/25	4. Model Checking Lab: JPF HW3 due, HW4 out, due 10/2	JPF, TBA	
10/2	5. Test Dependencies HW4 due, Project out, due 12/11	VMVM, DTDetector	
10/10	6. **(Note: TUESDAY)** Random Testing	RANDOOP, TBA	
10/16	7. Fuzzing	TBA	
10/23	OOPSLA (no class)		
10/30	8. Fuzzing part 2, evolutionary test generation	EvoSuite, TBA	
11/6	9. Dynamic Symbolic Execution	KLEE, SAGE	
11/13	10. Fault Tolerance	Rx, CROCHET	
11/20	11. Test Quality	TBA	
11/27	12. Test Coverage and Selection	Deflaker, Ekstazi	
12/4	13. Debugging	Delta Debugging, TBA	

12/11	(No class) Final project due
12/18	Project presentations (during exam time, normal room)

EDIT WITH VISUAL COMPOSER

Contact