Fall 2019
EE382C-16 Software Testing
MW 9:00 a.m. – 10:30 a.m. ECJ 1.318

Instructor

Sarfraz Khurshid
EER 7.880
x1-8244
khurshid@ece.utexas.edu
Office hours: Tuesday, Wednesday 10:30am to 11:30am

Teaching Assistant

Yang Hu
huyang0905@gmail.com
Office hours: TBD

Catalog entry

Basic concepts and techniques for testing software and finding bugs. Topics include the testing process, unit, integration and system testing, manual and automatic techniques for generation of test inputs and validation of test outputs, and coverage criteria, and focus on functional testing.

Prerequisites

Electrical Engineering 422C (or 322C) with a grade of at least C-. Knowledge of Java will be beneficial but is not required; language constructs necessary for this course will be introduced in the class. Students must be able to write correct technical English.

Description

This course first introduces the basics of software testing theory and practice, and then presents some recently developed techniques for systematically finding bugs in programs and improving their reliability. A NIST report from 2002 estimates that software failures cost the US economy $59.5 billion dollars annually and over a third of this cost could be saved using a better infrastructure for testing. It is widely accepted that testing currently accounts for more than one half of the cost of software development. Learning the techniques and tools presented in this course is likely to significantly increase the students' productivity as software developers and testers and improve the quality of the code they develop.

Textbook (recommended)

Introduction to Software Testing by Paul Amman and Jeff Offutt. Cambridge University Press. ISBN: 9781107172012.

Deliverables and grading

There will be four to six problem sets, and three mid-term exams. In addition, graduate students will work in teams of 2 or 3 students each on a semester long project on designing and building a test automation tool (e.g., to monitor code coverage, to perform symbolic execution, or to perform mutation testing), and give a 15-20 minutes presentation during the last class week as well as submit a written final project report.

The problem sets will account for 20%, the mid-term exams for 60%, and the project for 20% of the grade.

The (tentative) dates for the three mid-term exams are: 9/25, 10/28, and 11/25.

Lab resources

Students will need to have access to a Java development environment. Additional resources may be required based on particular assignments.

Collaboration

Students must solve the problem sets individually and submit their own work. Graduate students working in a team will deliver a co-authored report and presentation.

ECE's academic honesty statement

Faculty in the ECE Department are committed to detecting and responding to all instances of scholastic dishonesty and will pursue cases of scholastic dishonesty in accordance with university policy. Scholastic dishonesty, in all its forms, is a blight on our entire academic community. All parties in our community—faculty, staff, and students—are responsible for creating an environment that educates outstanding engineers, and this goal entails excellence in technical skills, self-giving citizenry, an ethical integrity. Industry wants engineers who are competent and fully trustworthy, and both qualities must be developed day by day throughout an entire lifetime. Scholastic dishonesty includes, but is not limited to, cheating, plagiarism, collusion, falsifying academic records, or any act designed to give an unfair academic advantage to the student. The fact that you are in this class as an engineering student is testament to your abilities. Penalties for scholastic dishonesty are severe and can include, but are not limited to, a written reprimand, a zero on the assignment/exam, re-taking the exam in question, an F in the course, or expulsion from the University. Don't jeopardize your career by an act of scholastic dishonesty. Details about academic integrity and what constitutes scholastic dishonesty can be found at the website for the UT Dean of Students Office and the General Information Catalog, Section 11-802.

Students with disabilities

Students with disabilities may request appropriate academic accommodations from the Division of Diversity and Community Engagement, Services for Students with Disabilities (Tel: 512-471-6259; online: http://www.utexas.edu/diversity/ddce/ssd/).

Calendar (tentative)

Week 1	8/28	Introduction, course overview, Java/JUnit basics
Week 2	9/4	Graph theory, logic, and discrete math basics
Week 3	9/9	Graph theory, logic, and discrete math basics
	9/11	Chapter 2 (1): Basic software testing principles and concepts
Week 4	9/16	Chapter 7 (2): Graph coverage Criteria
	9/18	Chapter 7 (2): Graph coverage Source code
Week 5	9/23	Chapter 7 (2): Graph coverage Designs/Specifications and Use cases
	9/25	**Mid-term Exam 1**
Week 6	9/30	Chapter 8 (3): Logic coverage Criteria
	10/2	Chapter 8 (3): Logic coverage Source code
Week 7	10/7	Chapter 8 (3): Logic coverage Specifications/finite-state machines
	10/9	Chapter 6 (4): Input space partitioning Input domain modeling
Week 8	10/14	Chapter 6 (4): Input space partitioning Combination strategies criteria and constraints among partitions
	10/16	Chapter 9 (5): Syntax-based Testing Criteria
Week 9	10/21	Chapter 9 (5): Syntax-based Testing Program-based and specification-based grammars
	10/23	Chapter 9 (5): Syntax-based Testing Input space grammars
Week 10	10/28	**Mid-term Exam 2**
	10/30	Chapters 13, 14 (6): Practical considerations Regression testing
Week 11	11/4	Chapters 10, 11 (6): Practical considerations Test process and test plans
	11/6	Implementing test automation tools
Week 12	11/11	Model checking and automated test input generation
	11/13	Combinatorial testing and constraint-based testing
Week 13	11/18	Symbolic execution
	11/20	Declarative models
Week 14	11/25	**Mid-term Exam 3**
Week 15	12/2	Project presentations
	12/4	Project presentations
Week 16	12/9	Project presentations