EE382C-16 Software Testing (Option III) Syllabus
Spring 2020

Instructor

Sarfraz Khurshid
EER 7.880
x1-8244
khurshid@ece.utexas.edu
Office hours: By appointment

Teaching Assistant

Chenguang Zhu
cgzhu@utexas.edu
Office hours: By appointment

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 benefical 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.

Website

We will use Canvas (`courses.utexas.edu`) for the class website.

Textbook (recommended)

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

Deliverables and grading

There will be three to six problem sets, and three in-class exams. In addition, the students will work in teams with 2 or 3 students each on a 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 exams for 60%, and the project for 20% of the grade.

The (tentative) dates for the three exams are: March 14, April 4, and May 1.

Collaboration

Students must solve the problem sets individually and submit their own work. 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)

January	Fri 1/24	Introduction, course overview, Java/JUnit basics
		Graph theory, logic, and discrete math basics
		Chapter 2 (1): Basic software testing principles and concepts
	Sat 1/25	Chapter 7 (2): Graph coverage – Criteria
		Chapter 7 (2): Graph coverage – Source code
		Problem Set 1 out; due: 2/14
February	Fri 2/14	Chapter 7 (2): Graph coverage – Designs/Specifications/use-cases
		Chapter 8 (3): Logic coverage – Criteria
		Chapter 8 (3): Logic coverage – Source code
		Project – proposal due: 3/10
	Sat 2/15	Chapter 8 (3): Logic coverage – Specifications/finite-state machines
		Chapter 6 (4): Input space partitioning – Input domain modeling
		Chapter 6 (4): Input space partitioning
		Combination strategies criteria and constraints among partitions
		Problem Set 2 out; due: 3/13
March	Fri 3/13	Chapter 9 (5): Syntax-based Testing – Criteria
		Chapter 9 (5): Syntax-based Testing
		Program-based and specification-based grammars
		Chapter 9 (5): Syntax-based Testing – Input space grammars
		Project proposal feedback
	Sat 3/14	**Exam 1**
		Chapters 13, 14 (6): Practical considerations
		Regression testing
		Chapters 10, 11 (6): Practical considerations
		Test process and test plans
		Problem Set 3 out; due: 4/3
April	Fri 4/ 3	Implementing test automation tools
		Constraint-based testing
	Sat 4/ 4	**Exam 2**
		Symbolic execution and model checking
		Project – final report due: 5/1
May	Fri 5/ 1	**Exam 3**
		Project presentations
	Sat 5/ 2	Project presentations