TENTATIVE SYLLABUS

CS 560 AUTOMATED DEBUGGING

Lecture Hours: Mondays 13:40 – 14:30 (FENS L067)
Tuesdays 08:40 – 10:30 (FENS L063)

Instructor Cemal Yilmaz
E-mail: cyilmaz@sabanciuniv.edu
Office: FENS G019

Zoom https://sabanciuniv.zoom.us/j/98223640931

DESCRIPTION

Program debugging is a process of identifying and fixing bugs. Identifying root causes is the hardest, thus the most expensive, component of debugging. Developers often take a slice of the statements involved in a failure, hypothesize a set of potential causes in an ad hoc manner, and iteratively verify and refine their hypotheses until root causes are located. Obviously, this process can be quite tedious and time-consuming. Furthermore, as software systems are getting increasingly complex, the inefficiencies of the manual debugging process are getting magnified.

Many automated approaches have been proposed to facilitate program debugging. All these approaches share the same ultimate goal, which is to help developers quickly and accurately pinpoint the root causes of failures.

This course will cover state-of-the-art automated debugging approaches from both practical and research perspectives and will consist of two main parts. The goal of the first part is two folds: 1) To turn program debugging from a black art (as many believe) into a systematic and well-organized discipline; and 2) To provide students with enough background information to read and understand the scientific literature. The topics which will be covered in the first part are: How Failures Come To Be, Tracking Problems, Making Programs Fail, Reproducing Problems, Simplifying Problems, Scientific Debugging, Deducing Errors, and Mining and Detecting Anomalies. The second part of the course will survey the related literature by dividing it into four broad categories, namely static-analysis-based, dynamic-analysis-based, model-based, and empirical approaches.

TENTATIVE PROGRAM

week 1 Introduction
week 2 Tracking Problems
week 3 Making Programs Fail
week 4 Reproducing Problems
week 5 Simplifying Problems
week 6 Deducing Errors
week 7 Observing Facts I
week 8 Observing Facts II
week 9 Tracking Origins
week 10 Asserting Expectations
week 11 Detecting Anomalies I
week 12 Detecting Anomalies II
week 13 Paper and Project Presentations I
week 14 Paper and Project Presentations II

GRADING POLICY

	Contribution (%)
Project	50
Paper Presentation	20
Final Exam (take home)	25
Participation	5

IMPORTANT NOTICE

Until a further notice by YÖK or Sabanci University, all the lectures and presentations will be streamed in a synchronous manner. During these online sessions, you may be asked to turn on your cameras and the sessions may be recorded. Consequently, your voice, camera images, and chat messages captured during the sessions, may be included in the recordings.

The syllabus may also be revised according to the reassessment to be made YÖK. The content to be delivered is certain, but the method of course delivery as well as all other details governing the course are subject to change.

COLLABORATION POLICY

Project groups may discuss ideas about their projects with other groups, but they should not share any project artifacts with others (e.g., requirement documents, design documents, source code, etc.) Each group is responsible in making sure that their artifacts are well protected from others.

MAKE-UP POLICY

It's simple. Do NOT miss the final exam!

If you do miss it, no makeup exams will be granted unless you have a documented emergency situation and notify the instructor within 48 hours after the exam date.

TEXTBOOK

No textbook is required, but the following is suggested:

Andreas Zeller, "Why Programs Fail: A Guide to Systematic Debugging", Morgan Kaufmann, ISBN: 1558608664