

ECS 260 Software Engineering

SPRING 2015

Syllabus

1 General Information

Lecture Time and Location: TR 9:00a-10:20a, 1342 Storer Hall.

Course Website: <http://web.cs.ucdavis.edu/~rubio/courses/ecs260-s15>

Credits: 4 credits.

Contact Information and Office Hours:

Instructor	Teaching Assistant
Cindy Rubio-González	Ching-Lan (Alan) Chen
crubio@ucdavis.edu , (530) 752-7069	clchen@ucdavis.edu
TR 10:30a-11:30a, 3039 Kemper Hall	MW 1:00p-2:00p, 55 Kemper Hall

2 Course Overview

This course will explore concepts and research in the area of software engineering, with particular emphasis on software testing and debugging. Topics include black-box testing (boundary value analysis, random testing, differential testing), white-box testing (structural testing, mutation testing, symbolic execution), model checking, automated debugging, and agile development, among others.

3 Course Format

In this course, students will read both classic, and recent state-of-the-art papers in the area of software engineering. Students will write paper reviews, participate in in-class discussions, and complete programming assignments. Students will also propose and complete a substantial term project. The following sections provide more details on each of these course components.

3.1 Paper Reviews

The reading material for some lectures will consist of conference papers. For those lectures, students will be required to write *one* paper review per lecture. If more than one paper is discussed in a given lecture, students can choose what paper to write the review for. Furthermore, students are allowed to skip two paper reviews in the quarter.

Paper reviews will be no more than one page long. The reviews should consist of four short paragraphs: (1) a brief paragraph summarizing the paper (problem solved, technical

approach, and experimental results), (2) a paragraph discussing one or two strengths of the paper (key contributions), (3) a paragraph discussing one or two weaknesses, and (4) a paragraph discussing an idea for how to extend this work. Paper reviews are **due the day before lecture by 6:00p**. Reviews will be available to all students in the class.

3.2 Class Participation

Students are required to participate in in-class discussions. For a student to receive full credit in participation, he or she must participate at least 5 times during the quarter. In order to keep track of student participation, the instructor will ask students who participated in a given lecture to write their names and date of participation in a participation card at the end of each lecture.

3.3 Homeworks

We will have 3-4 homeworks during the quarter. All homeworks will require programming. The tentative plan is to have a small homework on black-box testing, and three larger homeworks on (1) symbolic execution, (2) fuzz testing and delta debugging, and (3) program transformation using LLVM. Homeworks are to be completed individually.

3.4 Course Project

A main component of the course is a term project. The project is an opportunity for students to (1) propose novel software testing techniques, (2) build upon existing testing techniques to solve new problems, (3) apply the techniques discussed in class to new problem domains, or (4) conduct user studies to evaluate existing techniques or tools. Other project ideas are welcome. Please discuss them with the instructor before the project proposal is due. Projects will be completed in groups of 3 or 4 students. Students are free to choose their team partners.

Project Proposal and Presentation

A project proposal will be required for each term project. The proposal will be 1-2 pages long, and it must include the following:

- Project title, and team members.
- Problem description and motivation.
- Technical approach.
- Evaluation methodology.

Each team will give a short presentation of their proposals to the instructor and other students in the class on **Tuesday April 21st, 2015**. The length of the proposal presentation will be determined later during the quarter depending on the number of proposals to be presented.

Project Milestones

There will be two project milestones before the final project report is due.

Milestone #1 The first milestone will consist of a first draft of the project report that must contain the following sections: introduction, motivating example, high-level description of the technical approach, *complete* related work, and high-level description of the evaluation methodology. Also include a section that briefly describes the contributions of each team member towards this milestone, and a section briefly describing the tasks to be completed by the milestone #2 deadline.

Milestone #2 The second milestone will consist of a 20-minute early demo (to be scheduled outside class) along with a second draft of the project report. This second draft must include a *complete* technical approach section, and initial results for the experimental evaluation. Additionally, any comments made on the first draft must be addressed in the second draft. As in milestone #1, please include a brief description of the contributions of each team member towards milestone #2, and a list of tasks to be completed by the time the final report is due.

Final Report

The final project report must be in 2-column conference style format with a maximum of 11 pages excluding bibliography. If you submitted the two project milestones in a timely fashion, the new material in the final report will simply be a complete experimental evaluation section, updated introduction section, and new abstract, and conclusions.

Final Presentation

All term projects will be presented to the rest of the class. Students outside the class, and professors in the department will also be invited to attend the project presentations. Final project presentations will be held on **Thursday June 4th, 9:00a-10:20a** (the last day of classes) and **Saturday June 6th, 3:30p-5:30p** (when the final exam for the class is scheduled). The length of the presentation will be determined later during the quarter depending on the number of projects to be presented.

4 Requirements

There are no formal requirements for this course, but background in programming languages and software engineering will be helpful to better understand the papers to be discussed in class. Programming experience in languages such as C, C++ or Java will also be necessary in order to successfully complete the homeworks and course project.

5 Materials

This course has no required textbook. Relevant materials will be made available on the course website.

6 Grading

Grades will be based on paper reviews, in-class participation, homeworks, and the course project.

20%	Paper Reviews
10%	Class Participation (5+)
35%	Homeworks (3-4)
5%	Project Proposal and Presentation
30%	Project Milestones, Final Project Report and Presentation

7 Late Policy

Late submission of paper reviews, project proposal and final project report will **not** be accepted. Please inform the instructor as soon as possible if you will not be able to make a deadline due to extenuating circumstances. Homeworks and project milestones will be accepted at most *two* days after the deadline with a 10% grade penalty applied per day.