

# Seminar on Program Analysis

## ECS 289C (Programming Languages and Compilers)

### WINTER 2015

### Syllabus

## General Information

**Instructor:** Cindy Rubio-González

**Time and Location:** TR 1:40p-3:00p, 146 Robbins

**Contact E-mail and Phone:** crubio@ucdavis.edu, (530) 752-7069

**Office Hours:** TBD, Kemper Hall 3039

**Credits:** 4 credits

**Course Website:** <http://web.cs.ucdavis.edu/~rubio/289c>

## Course Overview

This course will explore research in the area of programming languages, with particular emphasis on program analysis. Program analysis is the process of automatically analyzing the behavior of computer programs to discover facts. Topics include static analysis (dataflow analysis, model checking, type inference), dynamic analysis (software testing, automated debugging, automatic program repair), and hybrid approaches. We will cover a wide variety of application domains, and languages. This course is intended for graduate students interested in starting research in the area of programming languages, or graduate students whose ongoing research could benefit from program analysis.

## Course Format

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

## Paper Reviews

We will discuss two papers per class, but you only need to write a review for one of those two papers. You can choose what paper to write the review for. In general, each student is required to write a paper review per class, except for presenters. If you are a presenter in a given lecture, then you will not be required to write a review for either paper presented that day. Furthermore, you 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 6pm**. Reviews will be available to all students in the class.

## **Paper Presentations**

Students will present at most twice during the quarter depending on the number of students enrolled in the class. The instructor will provide a preliminary reading list in the first lecture. Students are required to e-mail the instructor their top 4 topic choices. Optionally, students can indicate what papers they would like to present within the selected topics. Based on the students topic (or paper) preferences, the instructor will finalize the reading list, assign the papers to be presented, and schedule the presentations. Except for the first couple of presentations, presenters will meet with the instructor outside class about a week before the presentation to receive early feedback.

In general, presentations should follow a conference talk format. Presentations will be 20 minutes long, followed by 15 minutes of discussion. It is up to presenters to use available slides, make their own slides, or use other resources such as the whiteboard. At least for the first round of presentations, the instructor will provide students with a questionnaire to provide the presenter with valuable feedback on his or her presentation. Such feedback will be counted towards class participation.

## **Course Project**

A main component of the course is a term project. The project is an opportunity for students to (1) propose novel program-analysis techniques to solve interesting problems, (2) build upon existing analysis 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 may be completed individually or in groups of two students. Students are free to choose their team partners.

## **Project Proposal**

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.

## Project Milestones

There will be two project milestones before the full 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.

**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.

## 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 March 19th, 1pm-3pm** (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.

## 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 may also be necessary in order to successfully complete the course project.

## Materials

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

## Grading

Grades will be based on paper reviews, in-class paper discussions, and paper presentations (50%) as well as the course project (50%).

|                                      |     |
|--------------------------------------|-----|
| Paper Reviews:                       | 20% |
| Paper Presentations:                 | 20% |
| Class Participation:                 | 10% |
| Project Proposal:                    | 10% |
| Final Project Report and Milestones: | 30% |
| Final Project Presentation:          | 10% |

## Late Policy

Late paper reviews or project reports 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.