

# CSE 5239: Dynamic Analysis of Software

## Autumn 2012

### Basic Information

- Instructor: Atanas (Nasko) Rountev
- Contact: DL 685, tel. 292-7203, email: rountev@cse.ohio-state.edu
- Credits: 2 semester credits
- Class meetings: Dreese Labs 317, Wednesday and Friday 11:30 am – 12:25 pm
- Course page: <http://www.cse.ohio-state.edu/~routtev/5239>
- Office hours: Wednesday 12:30 pm – 1:30 pm, Friday 10:30 am – 11:30 am, or by appointment

### Prerequisites

The prerequisites for this course are (1) CSE 755 or CSE 756, and (2) working knowledge of C++ or Java. If you are not sure whether you have the necessary background for this course, please contact me immediately – we will cover advanced graduate-level and research-oriented material which requires strong background in programming languages.

### Course Overview

The goal of this course is to introduce students to basic foundations and current advances in the area of **dynamic (run-time) program analysis**. Recent years have seen a great increase in the use of such analysis for performance optimization (e.g., generating profiling information and identifying performance bottlenecks), run-time checking (e.g., protecting against run-time errors and vulnerabilities), and debugging (e.g., identifying the root causes of incorrect behavior). In this course we will consider dynamic analyses for imperative languages (e.g., C) and object-oriented languages (e.g., Java). The primary goals are (1) to provide entry-level knowledge of foundational analysis techniques, and (2) to apply some of these techniques to real-world code. We will also consider some **static (compile-time or link-time) analyses** that are required for subsequent dynamic analysis (e.g., in order to decide what instrumentation to insert in the program code). The covered material will be useful for students working in compilers, systems, programming languages, software engineering, and software security.

### Course Organization

#### *Lectures*

In the first few course meetings I will present an overview of some basic dynamic analyses and the related static analyses. The goal of these lectures is to establish some background for the subsequent paper presentations and for the project. No previous knowledge of program analysis is expected.

### ***Paper presentations***

For part of the semester, students will present papers from our reading list. The goal is to gain more experience understanding other people's work, as well as to develop and improve presentation skills. I will help with the harder papers. We will spend a fair amount of time on the foundational papers, in order to get a firm grasp on the basic techniques. Depending on the available time, we will also cover some more advanced topics.

### ***Project***

There will be several projects to choose from. Depending on the student's preferences, the focus can be on analysis of C, C++, Java, or Fortran. We will use popular frameworks (e.g., LLVM, Jikes RVM, Soot) that allow the creation of instrumented code and the deployment of run-time analysis agents. Each student will implement a small dynamic analysis tool. I will have some predefined projects, but you will be free to define your own project if you want. Even though you cannot build a real tool in one semester, we will try initial exploration of ideas that may be useful for your ongoing research, or for future publications.

### **Grading**

Project 80%, paper presentations 20%. If a student presents more papers than other students, the grading scheme will be adjusted accordingly.

### **Students with Disabilities**

Any student who feels he or she may need an accommodation based on the impact of a disability should contact me privately to discuss his or her specific needs. Please contact the Office of Disability Services at (614) 292-3307, or visit 150 Pomerene Hall, to coordinate reasonable accommodations for students with documented disabilities.

### **Religious Obligations**

I will do my best to accommodate any religious obligations you may have. Please contact me privately, at least a week in advance, to work out any relevant details.