CSc 620

Debugging, Profiling, Tracing, and Visualizing Programs

0 : Administrivia

Christian Collberg

collberg+620@gmail.com

Department of Computer Science

University of Arizona

Contact Information

Class	:	620
	:	Debugging, Profiling, Tracing, and Visualizing Programs
Lecturer	:	Christian Collberg
Email	:	collberg+620@gmail.com
IM	:	uacsc620 (AIM)
WWW	:	http://www.cs.arizona.edu/~collberg/Teaching/620/2005
Office	:	Gould-Simpson 758
Office Hours	:	Open door policy
Phone	:	621-6612
Lectures	:	MW 3:00-4:15, GLD-S 906

Course Outline

In this seminar class we will study various ways of learning about a program. We're both interested in how to find bugs (logical bugs as well as performance bugs) in a program and how to present this information to the programmer. Topics include:

1. how do debuggers work?
2. how to profilers work?
3. how do do you visualize the static structure of a program?
4. how do you visualize the dynamic behavior of a program?

Course Outline. . .

The course will consist of

1. Reading and presenting research papers in class.
2. Assignments to learn how to write debuggers, profilers, tracers, and visualizers.
3. Programming projects where the entire class will collaborate to build a debugger/profiler/visualizer.

Necessary prerequisites:

1. A strong background in compilers.
2. Knowledge of graphics and systems programming
3. Very strong programming skills in one or more of C/C++/Java.

Syllabus

You are responsible for reading

and understanding this syllabus.

If you have any concerns or issues

about the information in this document

you should bring them up during the

first week of class.

Course Description

Description of Course — 1st Half

How do debuggers work?

- Read papers/books about debuggers and present to the class.
- Learn Java bytecode.
- Learn about the Java Debugging Interface (JDI).
- Learn about debugging in Linux (the ptrace system call).
- Learn how to manipulate Java bytecode using SandMark BCEL.

Description of Course — 1st Half. . .

- Write a small tool to trace/debug Java programs using JDI and/or bytecode manipulation.
- Learn how to manipulate x86 binaries.
- Write a small tool to trace/debug C programs using <code>ptrace</code> and/or binary modifications.
- Investigate current debuggers and present them to the class.

Description of Course — 1st Half...

How do profilers work?

- Read papers/books about profilers and present to the class.
- Learn about the Java Profiling Interface.
- Learn about profiling in Linux .
- Write a small tool to profile C programs.
- Write a small tool to profile Java programs using the Java Profiling Interface.
- Investigate current profilers and present them to the class.

Description of Course — 1st Half...

How do program visualizers work?

- Read papers/books about visualizing software and present to the class.
- Write a small tool to visualize C and/or Java programs.
- Using computer game engines for visualization.
- Investigate current code visualizers and present them to the class.

Description of Course — 2nd Half

- Pick a group to work with!
- Pick a project!
- Implement the project!
- Do a presentation to the class!
- Read more papers to get a more in-depth view of particular topics that interest you.

Exam-Schedule

- There will be no written exams.

Course Objectives

At the end of the course you should be familiar with

- how to manipulate Java bytecode and x86 binaries;
- how debuggers and profilers work;
- various techniques for visualizing the behavior of programs;
- the use of computer game engines for visualization.

Required Texts

- Various web resources.
- Lecture notes.

Course Methodology

- Read and present papers to the class.
- Investigate tools and present to the class.
- Work programming assignments.
- Implement and present a team project.

Course Methodology. . .

- This course requires much individual initiative and work!
- We will essentially be engaging in a novel and creative research team project, much as you might do during the thesis phase of your Ph.D.
- Don't expect everything to be laid out nice and easy for you — you are expected to take the initiative to read, investigate, and learn on your own.
- I might, for example, suggest that you use a particular tool (computer game engine, debugging library, etc.) to solve a particular task in your project. You are then expected to download, install, read the documentation for, experiment with, this tool, until you're the expert.
- If you don't feel ready for this level of responsibility, then this is not the right class for you.

Course Methodology. . .

- This is a collaborative class.
- This means that during the project phase you are encouraged to share information, data, and code with your classmates.
- For example, if one teams writes a visualization library that another team needs, go ahead and share it!
- During the first half of the class when we do assignments, no such sharing is acceptable, of course!

Required extracurricular activities

- None.

Special materials required for the class

- None.

Assignment Format

- Assignments will be in the form of programming problems.

Prerequisites, Required Knowledge

- A strong background in compilers.
- Knowledge of graphics and systems programming
- Very strong programming skills in one or more of C/C++/Java.

Assessment Scheme

Tests, Quizzes, and Assignments

You will be graded on

1. five assignments, worth a total of 25%;
2. two-to-four presentations of papers or tools, worth a total of 25%;
3. quizzes on presented papers and tools, worth a total of 20%; and
4. two project presentations, worth a total of 30%.

Late Assignments

- Assignments handed in no more than 24 hours late will incur a 10% penalty.
- Assignments handed in more than 24 but no more than 48 hours late will incur a 20% penalty.
- Assignments handed more than 48 hours after the deadline will receive a grade of 0.
- Presentations have to be given at the agreed-upon time, or a grade of 0 will be awarded.
- Quizzes cannot be taken late.

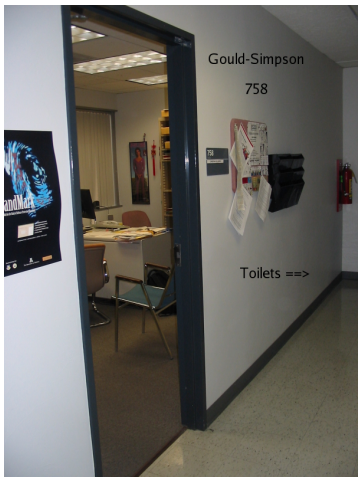
Incomplete work policy

- Except under exceptional circumstances I will not assign incomplete grades.
- I decide what is an exceptional circumstance.

Policies

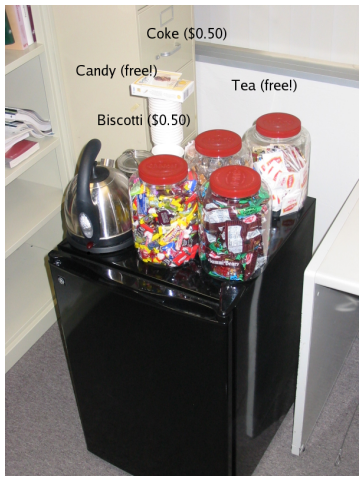
Office hours

- I use an open door policy:



Collberg's Café

- Please come and see me to chat, ask questions, or snack:



Attendance Policy

- My goal is to keep class attendance high so that we can get good discussions going in the class.
- You are not required to attend lectures, but. . .

  you cut class at your own risk.
- To encourage class attendance and participation there will be quizzes on presented papers and tools throughout the semester.

Subject to Change Policy

- The information contained in this course syllabus, other than the grade and absence policies, may be subject to change with reasonable advance notice, as deemed appropriate by the instructor.
- The instructor reserves the right to
  1. may add, drop, or change topics;
  2. change exam or homework dates, etc;
- We may
- Changes will be announced in class and on the class web site!
  You are responsible for checking this site regularly.
- You should also check the course news group `cs.course620` for announcements.

Notification of Objectionable Materials

- There will be no objectionable material in this class.

Computer Access/Setup

Computers will be provided.

Handicapped Accessibility

Students with disabilities who require reasonable accommodations to fully participate in course activities or meet course requirements must register with the Disability Resource Center. If you qualify for services through DRC, bring your letter of accommodations to me as soon as possible. See http://www.salt.arizona.edu/.

Student Code of Academic Integrity

- Assignments in this course require individual attention and effort to be of any benefit. All work is expected to be that of each student alone. You may not consult with others, except in ways specifically authorized by the course instructor. You also may not plagiarize another person's work or copy another person's code.

Student Code of Academic Integrity. . .

- Students are responsible for understanding and complying with the University's Code of Academic Integrity. A synopsis of the Code is attached; the full text is available from the Office of the Dean of Students in Room 203 Old Main. Among other provisions, the Code demands that the work you submit is your own, and that graded papers and exams will not subsequently be tampered with. Copying of another student's programs or data, or writings is prohibited when they are part of a published class assignment; it is immaterial whether the copying is by computer, xerox, pen or other means. Witting collaboration in allowing such copying is also a Code violation.

Student Code of Academic Integrity. . .

- Assignments in this course require individual attention and effort
- Violations of the Code will, at minimum, result in loss of credit for a graded item. An egregious first violation or any second violation will minimally result in failure of the entire course.
- See also http://studpubs.web.arizona.edu/policies/cacaint.htm the University of Arizona Code of Academic Integrity.

I take academic integrity seriously! I will report *every* violation!

Expected classroom behavior

- Be courteous and treat others in the class with respect.
- Please be courteous to other students by refraining from talking, playing loud music in your headphones, silencing cell phones, pagers, etc.
- We come to class to learn: don't read the newspaper, solve cross-word puzzles, etc.

Policies against threatening behavior

- Read and abide by the following link:
  `http://policy.web.arizona.edu/~policy/threaten.shtml`.

Now What?

Let's Have Fun!!!