

Fall 2019
EE379K: Engineering Dynamic Program Analysis
TTH 9:30am to 11:00am, EER 1.528

Instructor

Milos Gligoric
POB 7.886
gligoric@utexas.edu
Office hours: Tuesday and Thursday 10:30am to 11:30am (but may change depending on other service)

Teaching Assistant

Jaeseong Lee
Jason.Lee27@utexas.edu
Office hours: On demand

Website

We will use Canvas (<http://canvas.utexas.edu/>) for the course website.

Catalog entry

Dynamic program analysis: code instrumentation; dynamic dependency analysis; program transformation; incremental analysis; autotuning; generative programming; and concurrency analysis.

Prerequisites

Electrical Engineering 360C Algorithms, 422C Software Design & Implementation II, and 461L Software Engineering and Design Laboratory with a grade of C- or higher in each. It is also recommended that students take 360T Software Testing prior to taking this course. Students can get a permission from the instructor to take the course if they demonstrate sufficient background even if they did not take all required courses. Students are expected to have considerable programming experience.

Description and Topics

Dynamic analysis is commonly used to detect errors in software, including memory errors, concurrency errors (e.g., data races), performance issues, etc. Although valuable, dynamic analysis can be costly because the program execution needs to be (continuously) monitored to collect necessary data for the analysis. Additionally, naively engineered dynamic analysis can interfere with the program being analyzed, which can impact the conclusions of the analysis. The main goal for this course is to provide motivation for various

Topic	Assignment
Code instrumentation	Code coverage (ASM)
Regression test selection	Regression test selection
Test case prioritization	Test case prioritization
Test-suite reduction	Test suite reduction
Automated debugging	Delta debugging
Random test generation	Randoop
Bounded-exhaustive test generation	Korat
Monitor-oriented programming	JavaMOP
Generative programming	Jumble
Program transformations	JIT
Test and program repair	ReAssert
Continuous delivery	CIS and build systems
Design patterns	Visitor and Fly
Concurrency analysis	State space exploration
Compilers overview	-
Lexing and parsing	yacc & flex
Interpreters	JPF
IR	Java bytecode
Data-flow analysis	CFG
IR manipulation	ASM
Compiler optimizations	Constant propagation
Dynamic instrumentation	Java agents
JIT	JVM
Fuzz testing	Fuzz JIT

Figure 1: List of potential topics and assignments.

dynamic analysis techniques, introduce popular tools that are frequently used to implement an efficient and effective dynamic analysis, and provide hands-on experience in developing dynamic analysis techniques.

Table 1 shows the list of potential topics. The exact list will vary from year to year; this will depend on the students that are in the class and prior experience with the course. Clearly, we will not cover all the topics from the table but only a subset (or some variant of a subset). The list of topics may change at any point in the semester.

Textbooks—Recommended (none required)

The following text provide some basic material that would help students understand material presented in the class.

- *Design Patterns: Elements of Reusable Object-Oriented Software* by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Addison-Wesley Professional. ISBN: 0201633612.
- *Introduction to Software Testing* by Paul Amman and Jeff Offutt. Cambridge University Press. ISBN: 0521880386.
- *Effective Java* by Joshua Bloch. Addison-Wesley. ISBN: 0321356683.
- *Compilers: Principles, Techniques, and Tools* by Alfred Aho, Jeffrey Ullman, Monica S. Lam, and Ravi Sethi. Pearson. ISBN: 0-201-10088-6.
- *Modern Compiler Implementation in Java* by Andrew Appel and Jens Palsberg. Cambridge University Press. ISBN: 052182060X.

- *Engineering a Compiler* by Keith D. Cooper and Linda Torczon. Morgan Kaufmann Publishers Inc. ISBN: 012088478X.

Course Grades

Course grades are based upon the total quantity of points earned during the semester. Points are available to be earned according to the following schedule:

Component	Date	Points
Exams 1	TBD	39 points
Exam 2	TBD	39 points
Exam 3	TBD	50 points

At the end of the semester, the total number of points earned by each student will be converted into a letter grade. (The number of points will be multiplied by a number between [0,1], which reflects the effort on the assignments.) The precise conversion formula is subject to change if, based on the sole discretion of the professor, the awarding of points during the semester has been too low to be consistent with grades awarded to EE379k students in past semesters. In addition, some students may receive a grade higher than that prescribed by the formula if, in the professor's sole discretion, the student's actual performance in EE379k was considerably better than the performance implied by the scores. Such cases are extremely rare. In no case will a student be assigned a grade lower than that determined by the formula.

- 165 points or higher – grade of A
- 145 points or higher – grade of B
- 125 points or higher – grade of C
- 110 points or higher – grade of D
- Less than 110 points – grade of F

Examinations

Exams will cover material from lecture and the assignments. Exams will be cumulative, although they will be more heavily weighted towards material not yet tested. All exams will be given during the semester. If your work or a personal situation forces you to unexpectedly miss exams, you should expect to get a zero on those occasions. If you miss an exam because of illness, you are expected to provide a statement from a doctor stating that, in his/her opinion, it was impossible for you to attend because of illness. A slip showing you visited the UT Health Center or your personal doctor is not sufficient. Bring your student ID to the exam; it may be requested for proof of identity. Nobody can leave the exam room within 45 minutes from the start of the exam and 15 minutes prior to the end of the exam (to avoid disturbing those that are still working).

Programming Assignments

The assigned class work in this course will consist of several programming assignments. Programming is a discipline that you learn by doing, not by listening to a lecturer. Therefore, doing the programming assignments is crucial to performing well in this class. Assignments will be given most of the weeks. Each assignment will have a clearly stated due date and time. **NO LATE ASSIGNMENTS WILL BE ACCEPTED.** It is the students' responsibility to ensure that completed assignments are received by the TA through the proper submission procedure. If you complete an assignment but fail to submit it (or submit it incorrectly), you are likely to receive zero credit for that assignment.

Submitting Programming Assignments

Programming assignments will be submitted for grading using the Git version control system. Email submissions will not be accepted. You should open a GitLab account, and create a private repository that you will share with us. All the assignments should be done inside this repo unless stated otherwise. Please share your repository with milos.gligoric and TBD.

Final Examination

There will be no final exam.

Grade Disputes and Corrections

If you discover an error in a grade assigned to you, you must submit your complaint, along with supporting evidence or arguments, to me (or to your TA or grader) within one week of the date that I (or your TA or grader) first attempted to return the exam or assignment results to you. For programming assignments, the dispute period starts with the posting of your score on the class Canvas page.

Complaints about grades received after the one-week deadline will be considered only if there are extraordinary circumstances for missing the deadline (e.g., student hospitalization). Exams submitted for re-grading will be completely re-graded.

Attendance

You choose if you want to attend the lectures or not. Whether you come to lectures or not, you are responsible for keeping up with what happens in lectures. If you miss a lecture, it is not reasonable for you to expect me to repeat the material that was covered in the lecture. This applies both to the content of the lecture as well as to announcements about course policies, events, deadlines, or whatever.

Use of Email

You cannot expect to get detailed answers to technical questions by email. Students are encouraged to discuss important matters with the teaching team in person, typically during office hours.

Other Course Related Policies

Academic Dishonesty (cheating)

The University and the Department are committed to preserving the reputation of your UT degree. Every piece of work that you turn in with your name on it must be yours and yours alone. No co-working is allowed on any test, project, or programming assignment. As an honest student, you are responsible for enforcing this policy in several ways:

1. You must not turn in work that is not yours. *Specifically, you are not allowed to copy someone else's program code.* This is plagiarism.
2. You must not enable someone else to turn in work that is not his or hers. Do not share your work with anyone else. Make sure that you adequately protect all your files. Even after you have finished a class, do not share your work or published answers with the students who come after you. They need to do their work on their own.
3. You must not allow someone to openly violate this policy because it diminishes your effort as well as that of your honest classmates. *Providing the questions or answers on an exam that you took earlier to another student who will take it later is cheating.*

Students who violate University rules on scholastic dishonesty in assignments or exams are subject to disciplinary penalties, including the possibility of a lowered or 0 grade on an assignment or exam, **negative (-50) points on an assignment or exam**, failure in the course, and/or dismissal from the University. Plagiarism detection software will be used on the programs submitted in this class. If cheating is discovered, a report will be made to the Dean of Students for all involved in the incident. Incidents of cheating become a permanent part of your academic record. The Dean of Students office may take assign additional penalties, up to and including dismissal from The University.

Learning Disabilities

If you have a learning disability that requires special attention, either during class or during an exam, please give me a letter from the Dean of Students describing what needs to be done. You should do this during the first week of classes. (The University of Texas at Austin provides upon request appropriate academic accommodations for qualified students with disabilities. For more information, contact the Office of the Dean of Students at 471-6259, 471-4641 or the Cockrell School of Engineering, Director of Students with Disabilities at 471-4321.)

Religious Holy Days

A student who is absent from an examination or cannot meet an assignment deadline due to the observance of a religious holy day may take the examination on an alternate day, submit the assignment up to 24 hours late without penalty, or be excused from the examination or assignment, *ONLY* if proper notice of the planned absence has been given to the instructor at least fourteen days prior to the classes scheduled on dates the student will be absent. For religious holy days that fall within the first two weeks of the semester, notice should be given on the first day of the semester. The notice must be personally delivered to the instructor and signed and dated by the instructor, or sent by certified mail, return receipt requested. Email notification will be accepted if received, but a student submitting such notification must receive email confirmation from the instructor. A student who fails to complete missed work within the time allowed will be subject to the normal academic penalties.

Online Privacy

Web-based, password-protected class sites are associated with all academic courses taught at The University. Syllabi, handouts, assignments and other resources are types of information that may be available within these sites. Site activities could include exchanging e-mail, engaging in class discussions and chats, and exchanging files. In addition, electronic class rosters will be a component of the sites. Students who do not want their names included in these electronic class rosters must restrict their directory information in the Office of the Registrar, Main Building, Room 1. For information on restricting directory information see: <http://www.utexas.edu/student/registrar/catalogs/gi06-07/app/appc09.html>.

Classroom Evacuation for Students

All occupants of university buildings are required to evacuate a building when a fire alarm and/or an official announcement is made indicating a potentially dangerous situation within the building. Familiarize yourself with all exit doors of each classroom and building you may occupy. Remember that the nearest exit door may not be the one you used when entering the building. If you require assistance in evacuation, inform your instructor in writing during the first week of class. For evacuation in your classroom or building:

1. Follow the instructions of faculty and teaching staff
2. Exit in an orderly fashion and assemble outside
3. Do not re-enter a building unless given instructions by emergency personnel