

CS 6340: Software Analysis and Testing

Syllabus for Fall 2016

(Last updated: 19 August 2016)

Instructors: Mayur Naik, Chris Pryby

Teaching Assistants: Ben Taylor (Head TA), Aditya Kamath, Lu Ma, Ravi Teja

Outline

- [1. Course Overview](#)
- [2. Course Objectives](#)
- [3. Prerequisites](#)
- [4. Required Texts](#)
- [5. Lessons](#)
- [6. Course Support](#)
- [7. Assignments](#)
- [8. Exams](#)
- [9. Grading Policy](#)
- [10. Deadlines and Extensions](#)
- [11. Technical Requirements](#)
- [12. Academic Integrity Policy](#)
- [13. Disability Services](#)
- [14. Course Calendar](#)

1. Course Overview

Aspects of software development besides programming, such as diagnosing bugs, testing, and debugging, comprise over 50% of the cost of software development. Modern technology has come a long way to aid programmers with these aspects. At the heart of this technology lies software analysis: a body of work that concerns discovering facts about a given program. Many diverse software analysis approaches exist, each with their own strengths and weaknesses. The course will teach the principles underlying these approaches as well as impart hands-on experience with applying them to automate testing, debugging, and finding bugs in complex real-world programs.

2. Course Objectives

Upon successfully completing the course, you will be able to:

- Describe the abstract properties of debugging and analysis techniques.
- Compute the outcome of debugging and analysis techniques on concrete examples.
- Evaluate the advantages and disadvantages of several debugging and analysis techniques for a given piece of software and/or set of constraints.
- Run debugging and analysis tools on actual software and interpret their results.

3. Prerequisites

The course presumes familiarity with the following topics:

- Mathematical and logical reasoning (e.g. set theory, boolean algebra, and probability)
- Programming in Java, C/C++, or equivalent (most assignments will use Java)
- Using virtual machines (with e.g. VirtualBox, VMWare)
- Scripting on Linux-like operating systems

4. Required Texts

There is no required text for this course. Supplementary readings are posted in the “Instructor Notes” sections underneath relevant lecture videos on Udacity.

5. Lessons

The lessons are available in two formats: videos and slides. Lesson videos are available to view or download from the course’s Udacity website, at udacity.com/course/viewer#!/c-ud333. Lesson slides in PPT and PDF formats are available on T-Square in the Resources tab under the Lecture Slides folder. Transcripts of lesson videos are provided in the Notes section of the corresponding PPT slides. The Instructor Notes accompanying the lesson videos provide links to optional further readings in the form of technical papers, books, and online articles.

Lesson #	Topic	Suggested Study Period	Week #
1	Introduction to Software Analysis	22–28 August	1
2	Introduction to Software Testing	29 August–4 September	2
3	Random Testing	5–11 September	3
4	Automated Test Generation	12–18 September	4
5	Dataflow Analysis	19 September–2 October	5-6
6	Pointer Analysis	3–16 October	7-8
7	Constraint-Based Analysis	17–30 October	9-10
8	Type Systems	31 October–13 November	11-12
9	Statistical Debugging	14–20 November	13
10	Delta Debugging	21–27 November	14
11	Dynamic Symbolic Execution	28 November–4 December	15

6. Course Support

We will use Piazza (accessible via a link on T-Square) for all course-related discussion. For questions to the instructors and TAs, please make a private post to “Instructors”.

Important announcements (including updates or clarifications to assignments) will be made on Piazza and pinned to the top of the feed. You are responsible for checking Piazza daily for announcements about the course.

Office hours will be held by the TAs. Their schedule will be announced on Piazza.

7. Assignments

There are 8 assignments. Each assignment is of one of the following two kinds:

- Using a software analysis tool based on a technique taught in lectures, reporting your findings, and answering conceptual questions.
- Implementing a software analysis algorithm in a high-level programming language (usually Java), sometimes with the assistance of Chord.

A VM image compatible with both Virtualbox and VMWare will be provided for relevant assignments. (It should work also with other virtualization software, but only these two are officially supported.)

In the schedule below, each assignment's deadline is 8:00 a.m. Eastern time on the date listed.

Title	Deadline	Associated Lesson
Dafny	5 September	Lesson 1 (Intro to Software Analysis)
Monkey	19 September	Lesson 3 (Random Testing)
Korat	26 September	Lesson 4 (Automated Test Generation)
Randoop	3 October	Lesson 4 (Automated Test Generation)
Dataflow	10 October	Lesson 5 (Dataflow Analysis)
Constraints	31 October	Lesson 7 (Constraint-Based Analysis)
CBI	28 November	Lesson 9 (Statistical Debugging)
Delta	5 December	Lesson 10 (Delta Debugging)

8. Exams

Two proctored exams (a midterm exam and a final exam) will be conducted to evaluate conceptual understanding of the course material. The exams will be based on the material covered in the lessons. (In particular, they will not be based on the assignments.) The quizzes interspersed in the lessons should be used as warm-up exercises to prepare for the exams. The exam questions will be similar in nature to, but perhaps a bit harder than, the quiz questions.

In the schedule below, the open and closing times are 8:00 a.m. Eastern time on each of the dates listed.

	Lessons Covered	Open Date	Deadline
Midterm Exam	Lessons 1–6	14 October	17 October
Final Exam	Lessons 6–11	9 December	12 December

9. Grading Policy

Grades will be assigned as follows: A for scores $\geq 90\%$, B for scores 80%–89%, C for scores 70%–79%, D for scores 60%–69%, and F for scores $< 60\%$. The instructor may relax this policy towards the end of the course. Individual assignments will not be “curved.”

Your overall course average will be computed from the following components:

- Assignments: 50% (6.25% per assignment)
- Midterm Exam: 25%
- Final Exam: 25%

Grades will be posted in the Gradebook section on T-Square. You have two weeks after a given grade has been released to contest it by posting a message to “Instructors” on Piazza. Do not email the instructors or TAs to contest grades. **The grade will be considered final after two weeks.** It is possible that if the regrading reveals errors that the graders overlooked initially, the regrading will result in a lower grade.

10. Deadlines and Extensions

Assignments and exams are due at 8:00 a.m. Eastern time on their respective due dates. **No late submissions will be accepted** except in extraordinary circumstances, so please plan ahead in order to submit the assignments on time.

Due to the large-scale nature of this course and to be fair to all students, it is not possible for us to make exceptions to deadlines and exam dates except for an emergency situation (e.g. illness or death in the family) or an officially sanctioned Georgia Tech absence (usually not applicable for online students).

If you have an emergency involving personal illness or family illness or death, you should file a [care request](#) with the Office of the Dean of Students so the office can verify your situation. If you have an emergency situation not covered by the above, send a private message to the instructors on Piazza with details of your request. Your request must be made in a timely manner in order to be considered. Moreover, if you are requesting an extension of N days for an assignment, you must turn in the assignment within N days of the regular deadline, or else the assignment will still be considered late.

11. Technical Requirements

To use the Udacity platform, you need to meet these [minimum requirements](#).

Exams will be administered using the Proctortrack proctoring system. Make sure to view the [technical requirements](#) for using Proctortrack. You should also complete the “Orientation Exam” on your testing computer at least 48 hours before your first exam for this course.

Additionally, Georgia Tech's Office of Student Computer Ownership issues [minimum hardware requirements](#) to incoming undergraduates. You are recommended to meet or exceed these guidelines.

12. Academic Integrity Policy

All Georgia Tech students, including students in the OMSCS program, are expected to read and uphold the [Georgia Tech Academic Honor Code](#). Honest and ethical behavior is expected at all times. All incidents of suspected dishonesty will be reported to the Office of Student Integrity.

In particular, in this course *all assignments must be done individually*. You are not allowed to collaborate with other students or people outside the class on any of the assignments, including taking code from online sources (except those specifically allowed). We take this rule very seriously, and we run a plagiarism detector on all submissions.

Additionally, all exams must be taken individually. You may use blank paper and pencil/pen for computations and scratchwork, and you may use a PDF reader to have the official course notes available during your exam. No internet searches, calculators, or other reference materials are permitted.

If you are found to have violated this policy, you can expect at a minimum to receive a 0 on the assignment or exam in question and to have a warning posted to your academic record. Repeat offenders are subject to increased penalties such as automatic course failure, academic suspension, or expulsion.

13. Disability Services

Georgia Tech is an ADA-compliant educational institution. If you have a disability that requires accommodations, contact [Disability Services](#). To receive accommodations, ask Disability Services to [forward the instructor](#) a letter specifying the accommodations you should receive. Do so as soon as possible in the semester, as it can take up to 15 business days for your initial application to be processed.

14. Course Calendar

The course calendar will be maintained as a Google calendar which can be accessed via the links below. Changes to the calendar will be announced on Piazza.

- HTML: <http://tinyurl.com/zb2acs2>
- ICAL: <http://tinyurl.com/znzbdq6>