

# CS 6340: Software Analysis

Syllabus, Summer 2026

## Change Log

### **Instructional Staff**

Instructors: Mayur Naik and Chris Poch

Head TAs: Kelly Parks and Joel Cooper

Teaching Assistants: John Daniel, Joseph Dougherty, Joseph Knaack, Susie Lin, Samantha Llewellyn, Alex Nazarov, Jared Summers, and Kevin Windbigler.

### **Table of Contents**

[1. Course Overview](#)

[2. Course Objectives](#)

[3. Course Schedule](#)

[4. Prerequisites](#)

[5. Required Text](#)

[6. Lessons and Lesson Quizzes](#)

[7. Course Forum](#)

[8. Office Hours](#)

[9. Lab Assignments / Development Environment / Gradescope](#)

[10. Quizzes and Exams](#)

[11. Grading Policy](#)

[12. Ed Discussions Participation](#)

[13. Regrade Requests](#)

[14. Deadlines and Extensions](#)

[15. Academic Integrity Policy](#)

[Collaboration](#)

[Plagiarism](#)

[Penalties](#)

[Use of Generative Artificial Intelligence \(AI\) Tools](#)

[Use of Source Control](#)

[Use of Non-GT Forums for Course Discussion](#)

[16. Technical Requirements](#)

[17. Disability Services](#)

[18. Expected Student Conduct](#)

## **1. Course Overview**

Have you ever wondered how your development environment can determine that you have an unused variable before you compiled the code? How do strongly-typed programming languages prevent bugs in your code? How can we be ahead of our users in finding bugs in complicated systems that may not have been caught by our testing efforts? In this class, we will explore these topics along with several others that together make up the field of software analysis.

The point of software analysis is to determine whether software is correct. Today, the cost of software development is less than 50% programming, with testing, debugging, security assessments, and similar tasks taking more resources than developing the software itself. As a result, there is an increasing focus in the software industry on using tools to write better software. These tools can take the form of testing tools that help find bugs but they can also take the form of analysis techniques that have the goal of building a stronger code foundation with fewer areas where defects can emerge, the end result being less risk in the software development lifecycle. While testing is frequently part of software analysis, the approach to software testing presented in this class is directly tied to analysis and is frequently different than the testing usually performed as part of quality assurance in a typical software development lifecycle. A past student put it well: “why test software when you can make software test software?”

In this class, we will investigate multiple techniques for analyzing software and the limits of what software analysis can tell us. Each lecture covers a different area of software analysis and many lectures have a corresponding lab where you will gain hands-on experience with the technique covered in the lecture. The labs will evaluate your practical use of the techniques, while the exam will evaluate your understanding of the lessons not covered by labs. Quizzes will evaluate all lessons, including the theories of software analysis techniques.

Applications of techniques covered in this class relate to performance, cyber security, testing, and debugging. In this course, you will study the underlying principles of software analysis and these approaches, and gain hands-on experience applying them to automate testing software and finding bugs in complex, real-world programs. You will explore techniques including dataflow analysis, constraint-based analysis, type systems, automated test case generation, symbolic execution, and more.

## **2. Course Objectives**

After successfully completing the course, you will be able to do the following:

- Evaluate the suitability of different analysis and testing techniques for different types of programs and under a given set of constraints
- Implement and run analyses to determine useful facts about a given program
- Apply various testing techniques to find bugs in a given program
- Describe qualitative properties of different techniques for analyzing and testing programs
- Compute the outcome of a given analysis or testing technique on a program

### **3. Course Schedule**

The schedule for this course will be made available through Canvas under the **Syllabus** and **Calendar** sections. It includes a pacing schedule for covering course material, completing labs, and taking exams, as well as important dates throughout the semester.

In particular, it is rare for work in our control to be due at any day or time other than Mondays at 8am ET, so you should figure out when Monday at 8am ET is in your time zone and adjust your Canvas time zone so you can see due dates in local time if you are not in Eastern Time ([Canvas Settings](#) -> click Edit Settings on the right).

### **4. Prerequisites**

We expect you to be fluent in a C-family language, such as C, C++, Java, or C#. Most classwork will be in C++, and the effort required will be greatly increased if you are not already fluent in a C-family language. We have created a [C++ Primer](#) to highlight the language features used in our labs.

In this course, you will use a variety of tools implemented in different programming languages to generate useful facts about a program's behavior and analyze them. You will learn about the tools and their theoretical underpinnings in this class, however it is assumed that you are already familiar with basic concepts in computer science and mathematics, as well as the use of various tools/commands in a UNIX-like environment.

The instructors and teaching assistants have only a limited capability to assist students with learning these tasks, and they will not be explicitly covered in video lessons. We have created a [Readiness Assessment](#) that you can use to determine prerequisite tasks that you may wish to learn or re-familiarize yourself with before enrolling in this course.

### **5. Required Text**

There is no required text for this course but we recommend several books that cover various sections of the course, listed at <https://rightingcode.github.io/books.html>. Many past students who have struggled with discrete math notation commonly used in computer science have found Chapter 0 of Michael Sipser's classic text *Introduction to the Theory of Computation* helpful.

### **6. Lessons and Lesson Quizzes**

Lesson videos are available for viewing or download through Ed Lessons, and slides used in the lessons (including transcripts) are available in both PPTX and PDF format through Canvas in the **Lesson Notes** section.

Each lesson has a graded quiz that corresponds with it. These quizzes are a few questions that are based on the topic of the lesson and are time limited to 30 minutes. These quizzes are automatically graded and may contain different types of questions such as fill in the blank,

multiple choice, matching, and entering a number. These quizzes are open book, open note, and open internet searching; however they are closed internet posting and may not be discussed with your classmates. You will get one attempt on each quiz except we will allow two attempts on the first quiz. You can practice for these quizzes using the ungraded quizzes in the lesson and the practice questions available on Canvas in the navigation sidebar.

Please note that the practice quizzes embedded into the lessons in Ed were designed for a previous platform we used that had different testing capabilities. Many of the quizzes have fewer answers accepted than we'd like there to be given what is presented in the lesson. However, we're stuck with what our current tools allow us to build. These quizzes are ungraded, so we ask for your understanding with what is presented as a correct answer. No need to panic about a wrong answer as long as the answer explanation video says yours is fine.

## **7. Course Forum**

**We will use Ed Discussions for all course-related official communication**, unless otherwise required by GT policy. For questions to the instructional team, please make a private question (post will be available to all instructors and TAs but no students). Please do not use Canvas, Teams, or email to message the instructors as we like to keep all communication in one place and Canvas frequently does not alert us to new messages.

We will make important announcements (including updates or clarifications to labs) on Ed Discussions and pin them to the top of the feed. **We expect you to check Ed daily for announcements about the course.** You are responsible for reading all messages pinned to Ed.

## **8. Office Hours**

Office Hours will be held for multiple hours a week via Slack at <https://omscs6340.slack.com/>. You sign in to this Slack instance using your GT login. In the Slack instance, you will find a **#office\_hours** channel. We have a [calendar](#) showing the times each week when Office Hours are being held. During times when a TA or instructor is available for office hours, post in this channel if you would like to receive one on one help. The person hosting office hours will direct message you to offer help. So that you get the fastest help from the correct person, please do not initiate direct message conversations with course staff on Slack. These sessions are a good way to get another set of eyes on things that would be inappropriate to share publicly, such as your code for a lab. We have a large class and all course staff in this class are part time and may work at different hours, so we as a policy do not respond to student-initiated direct messages. We will happily follow up with you via direct message when we are holding office hours if you have a private question.

Questions that may be relevant to the whole class should be asked on Ed so the answer is available to all students. Questions specific to your work on a lab should be asked in Office Hours, as sharing your work publicly can result in an academic integrity violation.

If you try to sign into Slack and you get a notice that your account is not active, you must contact the GT Slack administrator by going to <https://gatech.service-now.com/technology> and

submitting a help request. Please give them your GTID, such as cpoch3, and tell them that you need your Slack account to be enabled.

## **9. Lab Assignments / Development Environment / Gradescope**

There are seven graded labs in this course plus an introductory lab that is about setting up your environment. Labs in this course will be made available through Canvas in the **Assignments** section. Labs 0 and 1 will be available at the start of the course and the remaining labs will be released on the Monday morning after registration closes, at 8am ET. This semester, all labs except for the introductory lab will be weighted at 8% of the class grade each.

In general, labs involve using a software analysis or testing tool based on a technique taught in the lessons or implementing a software analysis algorithm using a software analysis library.

To reduce student effort required to set up labs, we provide a Docker container (preferred) and a virtual machine (VM) preloaded with tools and libraries required to complete labs. Lab instructions assume you are using one of these two development environments. All of your labs will be graded through Gradescope. Additionally, unless instructed otherwise, all labs must complete execution and terminate in a reasonable amount of time to receive credit. You can submit your lab an unlimited number of times, but you will not be able to see the result of all test cases until we release lab grades. Please note that we discourage trying to run any of the labs directly on your computer and do not provide any support for students who do not use the official development tools.

Most labs include some test cases distributed with the lab. You should expect to be graded on these test cases along with other ones not provided to you. We recommend that you monitor the output of Gradescope as it will typically run the provided test cases for you so that you do not have an avoidable grading surprise. Some labs have many public tests available on Gradescope. Other labs have tests where you can see the results, but not the details other than pass/fail. Other labs just check to see that your code compiles. While Gradescope is a helpful tool, we do not want you to exclusively use Gradescope for validating your work. We want you solving the problem, not coding to Gradescope so we intentionally hide some of the results for all labs.

While it is generally fine to install development tools like an IDE on the course VM, you should not run software updates or upgrade installed packages on the Docker image or VM as it could render a lab inoperable.

The username for the VM is **cs6340** and the password is **student**. If you choose to use the VM, we recommend keeping an unused copy of the VM download or taking a snapshot of the clean VM so that you can easily go back to a known good state if you run into any issues on the VM.

Lab 0 is about setting up your environment for lab 1. Links to download the Docker container and VM are available in the class Canvas from the **Docker/VM** section.

## **10. Quizzes and Exams**

Our goal is for all course concepts to matter equally to your final grade. To that end, there are seven main labs and eleven lessons. We will have a single exam this semester, and it will cover the four lessons without a main lab. Regardless of if your knowledge is tested through labs or exams, all concepts will have roughly equal weight.

Logistical details for the exam will be posted on Ed, typically a week prior to the exam. You are permitted to use the course lesson PDFs (on screen only) and one PDF page of your own notes, using our provided template, plus one blank sheet of physical scratch paper during exams. No other resources are permitted. We will provide quiz and exam preparation resources (including solutions) for you to use to prepare for the exams this semester. They are available through Canvas in the **Quiz and Exam Preparation** section.

Exams are delivered using the Honorlock proctoring system. During the first week of the course, we will give you a quiz on the syllabus. This quiz will be administered exactly like an exam so that you become familiar with exam procedures. Unlike an exam, the syllabus quiz will be completely multiple choice and you may take it multiple times. However, you must score 100% on the syllabus quiz to receive credit.

Note that we require a mirror scan of your testing area as part of all exams. If you do not have a mirror at least 200mm x 200mm (8in x 8in) in size, with larger mirrors being better, that you can use to show us your working area when taking an exam, you should acquire one now so you are prepared to take your first exam. Without a room scan that meets the requirements given in the exam logistics Ed post, your exam will not be considered valid and will not receive credit. In particular, based on past issues with exams, we need to see your work surface, monitor, keyboard, and mouse, the area surrounding them, and that front and back of your scratch paper are blank clearly in the room scan.

If you have been previously enrolled in the class, please note that you may have a different exam format than your classmates to avoid giving you an unfair advantage.

## **11. Grading Policy**

We will post grades in the **Grades** section of Canvas. We will post an announcement on Ed when grades have been released. Your final course average will be computed as follows:

- 60% Labs
  - Lab 0 (4%)
  - All seven other labs (8% each)
- 20% Course Exam, covering
  - Introduction to Software Analysis
  - Introduction to Software Testing
  - Automated Test Generation
  - Pointer Analysis
- 20% Quizzes, Surveys, and Participation
  - Beginning of class, mid-course, and end of class surveys, (0.5% each)
  - Syllabus quiz (2%)
  - Quiz corresponding to each lesson (1.5% each)
  - Ed participation, 0.05 points for each meaningful contribution (post) - maximum of 5% of total grade possible\*

\* We reserve the right to adjust the participation cap depending on how the semester goes but any changes will be in your favor (more points possible)

**Individual lab, exam, or quiz scores will not be “curved”.** The final assignment of letter grades is at the sole discretion of the instructors. After the exam, the instructors will release a set of total score ranges for each letter grade. These score ranges are how we scale the class to account for changes in lab or exam difficulty. Our goal is that for the same level of effort and preparation, you would receive the same grade regardless of which semester you take the class. Since we are not having a final exam this semester, we will release updated grade brackets around the same time we release the last lab grade.

We cannot offer “make up” assignments if you are not happy with the grade that you received. Additionally, we will not be offering extra credit outside of what is described in this syllabus, as giving additional opportunities to improve your grade that are not offered to the entire class would not be fair to your classmates. We also are not permitted to give “pity points”; being on academic probation or in danger of losing financial aid are not valid reasons to request a grade change.

Since it is different from most other US universities, please note that Georgia Tech does not assign plus or minus grades. The only grades that you can receive in OMSCS or any Georgia Tech class are full letter grades (A, B, C, D, F, W, I). The only thing that is recorded at the end of the semester is the letter grade you earned, so a student with an overall score of 91% and another student with 98% would both receive the same “A” on their transcript.

Here is the scale used to determine final grades in this class:

- A:  $\geq 90\%$
- B:  $\geq 80\%$
- C:  $\geq 70\%$
- D:  $\geq 60\%$
- F < 60%

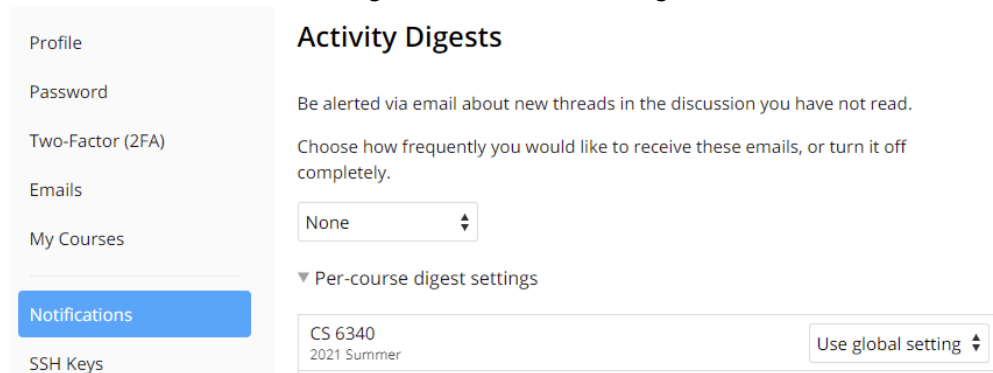
## **12. Ed Discussions Participation**

The place where the most valuable learning frequently takes place is between students. The class forum on Ed is our “classroom”. Ed is where we have most of our class communication, including between students, and is the “system of record” for questions and clarifications. Because we want to reward helpful students, we have a participation grade based largely on your meaningful participation on Ed. Since these kinds of participation scores can be gamed, we do not release our exact participation formula, but we give the following guidelines to help you know how to get a good participation score:

- You will receive 0.05 points for the following non-anonymous activities
  - Non-trivial posts and comments (for example, “+1” or “I agree” will not earn you any credit as these are rarely helpful in a large class)
  - Additional 0.05 points for posts a large number of users found helpful (pressed the heart)
  - Additional 0.05 points for particularly insightful posts
- At our discretion, we may provide an additional bonus at the end of the semester to all students with an exceptional number of “days active” or a very high percentage of all

posts read. If we offer this bonus, it will be available to all who qualify, but we will only be checking if it applies if it could increase your final grade

- Note that we cannot track anything that Ed emails to you so you will only get credit for reading posts through the Ed site
  - We recommend the following Ed notification settings:



The screenshot shows a user interface for Ed notification settings. On the left is a navigation menu with options: Profile, Password, Two-Factor (2FA), Emails, My Courses, Notifications (highlighted in blue), and SSH Keys. The main content area is titled 'Activity Digests' and contains the following text: 'Be alerted via email about new threads in the discussion you have not read. Choose how frequently you would like to receive these emails, or turn it off completely.' Below this is a dropdown menu currently set to 'None'. A section titled 'Per-course digest settings' is expanded to show 'CS 6340 2021 Summer' with a 'Use global setting' button.

- You may lose participation points for
  - Trolling
  - Any posts that do not contribute to a positive class atmosphere
  - Posts with information that is an Academic Integrity violation
- We will give updates on current participation points earned several times during the semester
- Participation will become final when the last assignment is due

### **13. Regrade Requests**

You have **one week** after the release of a grade to request that it be regraded. To request a regrade, make a private question on Ed tagged as “Graded Work Review” and answer the questions explaining why you believe an error was made in grading your work. We will not process regrades or increase scores on labs or exams for reasons other than errors in grading - do not submit regrade requests asking for a higher score in order to meet OMSCS program requirements or because you believe the rubric was too harsh. Regrades are specifically limited to the stated grading criteria being incorrectly applied to your work. If the regrading process reveals an error the grader overlooked initially, you may receive a lower score. Please note that regardless of what Canvas says, we do not receive notifications from Canvas or Gradescope for comments posted there and so any grading issues must be addressed via Ed private post - we will not see any comments left for us on Canvas or Gradescope with respect to your submission or a grade.

Regrade requests are typically handled by the Teaching Assistant that assigned the original grade. If necessary, a head TA or instructor will also review the request. The instructors are the final authority on grades assigned in this course.

If you feel your issue has not been resolved after an instructor review, you may file a complaint via email to [oms-advising@cc.gatech.edu](mailto:oms-advising@cc.gatech.edu). In your email, request that your ticket be assigned to Jay Summet.

## **14. Deadlines and Extensions**

Labs and exams are due at 8:00 a.m. Eastern Time on their respective due dates, typically Monday mornings. We **do not** accept late submissions. If you are not located in Eastern Time, you should determine what 8:00 a.m. Eastern Time is in your local time zone now so that you do not miss any deadlines.

Lesson quizzes have a “due date” of the week after the lesson is assigned; however this due date is not strictly enforced. The firm due date for lesson quizzes is 4 May at 8am ET.

We do not offer a late penalty. Work is either accepted or not. If Canvas accepts it through the assignment’s submission (NOT as a comment on the assignment or your gradebook entry), we will accept it. If it is past when Canvas is set to allow submissions, we will not accept it without a qualifying extension as explained below.

We **will only** grant extensions to labs/exams deadlines for students with documented, excused absences. These absences must be verified and approved either by an instructor or the Office of the Dean of Students. Typical situations where an absence will be excused include illness, hospitalization, death in the family, or a military deployment. In most approved cases, you will be given an extension equal in length to the amount of work missed. For example, a three day illness would permit up to a three day extension.

To get an extension, you must notify the instructors of your qualifying absence within one day of being able to resume class. For situations where you know an approximate time frame in advance, such as pregnancy or military deployment, you may reach out to the instructors as soon as you are aware of a need for a qualifying absence. Students absent from more than a week of class may need to take an “Incomplete” for this semester and make up the missing work in a future semester. An “Incomplete” puts the semester on pause, allowing you to resume in the future although you must resolve all outstanding incompletes by the end of the next semester you are enrolled or within a calendar year. Please note that an incomplete does not let you re-do work where you wish you had received a better grade.

To start the process of verifying an absence, please notify the instructors via a private question on Ed Discussions of category “Extension Request”, making sure to include the dates of your absence. It is important that requests are filed in a timely manner; please do not wait until a deadline has passed to start this process. At the latest, you must start the process within a day of being able to resume classwork if it is after the deadline. If you prefer to have the Office of the Dean of Students perform the absence verification, please additionally file a [care request](#) with the Office of the Dean of Students. Excused absences **will not** be granted for personal travel, social events, weddings, or work conflicts, and as such are not justification for an extension. All classwork will be available at the start of week two, so if you have upcoming events that may interfere with your school work, we suggest working ahead.

## **15. Academic Integrity Policy**

Our goal in CS 6340 is to provide you with an opportunity to learn more about software analysis. We find that the best learning takes place when students teach each other, so we see most collaboration as a good thing. On the other hand, cheating is considered a very serious offense

and is vigorously prosecuted. To ensure that we are fair, we want to make sure that we are clear on the differences between collaboration and cheating. If you obtain help of any kind on a submitted assignment, always include comments (in source code) or footnotes (in reports) indicating the source of the help in your submission.

## **Collaboration**

Collaboration is essential to both the learning experience in this class as well as in the real world. Collaboration is working with other people cooperatively to enhance understanding. We encourage you to:

- Share ideas
- Explain concepts in lessons and practice problems to each other
- Share resources relating to software analysis that you come across
- Explain your high level approach to a problem. The level of detail should not include implementation specifics or your code
- Pseudocode is generally safe to share, although it should be at a high enough level where if two people implemented it in real code they would certainly have different implementations

## **Plagiarism**

Plagiarism is using someone else's work instead of doing your own work.

- We expect you to complete all labs, exams, and quizzes by yourself
- We consider using AI such as ChatGPT or code writing plugins like GitHub Copilot to not be doing your own work (see "Use of Generative Artificial Intelligence (AI) Tools" below)
- Never share your lab code or reports with current, future, or even past students
- Never use someone else's code or text in your submitted work
- You may quote outside resources in a lab report, but make sure your quotes are clearly attributed
- Never consult project code or text that you find anywhere, such as the internet and specifically sites designed for sharing of code or assignments, unless specifically permitted in the lab instructions
- Do not share any part of the course materials, including the lab descriptions and code anywhere including GitHub, Dropbox, StackOverflow, Chegg, or Pastebin
- Use of unauthorized resources on exams
- If in doubt, ask the course staff

## **Penalties**

Sadly, we sometimes find students cheating. While cheating may take on many forms, we do not tolerate it because it is not fair to your classmates and it is also not fair to you. Our goal is to help you learn the course material and cheating does not achieve that goal. Therefore the worst penalty of cheating is depriving yourself from learning the material. Additionally, cheating is considered scholarly misconduct. The actual penalty for confirmed cheating for a first offense

will consist of not receiving any credit for the work in question along with the inability to receive an A as a final grade in the class. If you are found guilty of cheating on a lab, that lab's 0 grade *must* be included in your final grade calculation (it is not eligible to be dropped for any reason). Subsequent violations will result in additional penalties and will include a failing grade in the course.

All Georgia Tech students, including students in the OMSCS program, must read and uphold the [Georgia Tech Academic Honor Code](#). Georgia Tech expects honest and ethical behavior of you at all times. We will report all incidents of suspected dishonesty to the Office of Student Integrity (OSI). If OSI finds you responsible for violating the Honor Code you can expect, at minimum, to receive a score of zero on the lab or exam in question and to receive a warning posted to your academic record; additional grade penalties; academic probation or dismissal; and prohibition from withdrawing from the class. Any grade penalty imposed by OSI means that the grade in question must be factored into your final grade; it will not be dropped for any reason.

Since it has been a past issue, you should be extra cautious if you have a close associate (spouse, boyfriend/girlfriend, sibling, roommate, close coworker) who is also in this class. Prior to 2023, we found that around 25% of the students with academic integrity issues fall into one of those categories, which is much higher than the expected rate of close associates in the class. We have found that it is easy for close associates to inadvertently have inappropriate discussions leading to academic integrity issues about sensitive class material such as lab or exam answers, in particular if you share a home with someone in the class. Therefore, you should question whether it is the right time for you to take this class if you have a close associate also enrolled this semester.

### **Use of Generative Artificial Intelligence (AI) Tools**

While there are many great uses of generative AI in industry, we see generative AI much like we see a calculator. It's a useful tool, but it's even more useful if you understand the underlying concepts of what it does well on your own. What we have found over the past several semesters when we allowed some limited use of AI is that many students who claimed they followed our policies to the letter ended up submitting work that was clearly not their own and frequently couldn't explain why their submissions worked. We could switch to "AI proof" assignments, but there is huge value in learning how to implement some of the foundational algorithms in our field, and we can't AI proof well known algorithms.

As a result, our course policy now is simple: **do not use any AI tools, such as ChatGPT, Gemini, Claude, GitHub Copilot, Cursor, or similar for any aspect of your course work.** Under no conditions should you put any course materials, concepts, or code that you have written for this class into an AI system. You may NOT ask AI to explain any graded work, lab materials, or help you implement an algorithm.

You should not develop any code for submission to class using an IDE or other program that has GitHub Copilot, Claude, or any other AI assistant enabled in any way where it is offering you suggestions on your code. You also may NOT use an AI tool to debug your code in any way.

Generative AI derives its output from previously created sources. Per GaTech's Honor Code, you may not submit any work generated by an AI program as your own.

### **Use of Source Control**

If you choose to store your work in an online version control system such as Git, you may only use Georgia Tech's GitHub at <https://github.gatech.edu> and must verify that your repository is set to **Private**. We have had previous academic integrity issues arise from students saving their work to repositories that were set to public and in these cases, both the author and the users of the work may be held responsible. We will consider the use of any other source code repository or posting your work to the internet to be an academic integrity violation and you should expect to not receive credit for any of your lab assignments found on that system. We highly recommend using GT Git and pushing your work frequently to avoid heartache if you have an issue with your computer.

### **Use of Non-GT Forums for Course Discussion**

Due to past academic integrity issues arising from students inappropriately discussing course work through non-GT channels, we require that all online discussion of graded course work take place on either Ed Discussions or the official CS 6340 Slack. Creating, using, or promoting chats through platforms such as WeChat, Whatsapp, Discord, other Slack instances, or similar where graded course work is discussed between current, former, or future students will be considered academic misconduct on whatever lab is due next. We have this requirement because we have had past issues and we can help keep discussions from becoming inappropriate on the official channels. If you become aware of other course discussions, please create a private post on Ed with the information.

*Academic Integrity Policy modeled after CS 6200 course policies and research by Tony Mason, David Joyner, and Angela Smiley. Used with permission. Used with permission.*

## **16. Technical Requirements**

OMSCS issues [technical requirements](#) needed for the program; you should meet or exceed these guidelines as well. In particular, make sure you have at least 8GB of RAM for a decent experience. If you have a computer with a non-x86-64 processor (a M-series Mac or some Surface models), you will likely need to use Docker as our VM is x86-64 only. Please note that we only support compiling the code in our labs on a Docker image or course VM and are not able to provide assistance if you try to run our labs directly on your hardware. In particular, LLVM is difficult to get running correctly.

Finally, you must also meet the [technical requirements](#) for taking exams using Honorlock. In particular, you will need to be using a Chrome browser with a relatively current version running on macOS or Windows.

## **17. 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 this as soon as possible, as it can take up to 15 business days for the office to process your initial application. Even if you are not sure if you need accommodations, you should arrange for any potential accommodations at the start of the course as we are unable to retroactively give accommodations. Disabilities covered by the ADA include ADHD, dyslexia, and pregnancy.

## **18. Expected Student Conduct**

As members of the Georgia Tech community, we are committed to maintaining an environment of mutual respect, professionalism, and academic excellence. By enrolling in CS 6340, you agree to the following expectations:

- **Professional Communication:** We expect all interactions on Ed Discussions and Slack to be professional, respectful, and constructive
- **Proactive Engagement:** Students are responsible for staying informed by checking Ed Discussions daily for important announcements
- **Integrity and Responsibility:** Students are expected to uphold the Georgia Tech Academic Honor Code
- **Collaboration Boundaries:** While we encourage cooperative learning, students must respect the boundary between constructive discussion and academic dishonesty as discussed in Section 15
- **Instructor Support:** The instructional staff is committed to providing support through official channels (Ed and Office Hours). In return, we expect students to utilize these official channels rather than messaging or emailing staff through other systems