

CSC 4351: Capstone I

Georgia State University, Fall 2022

Disclaimer: The course syllabus provides a general plan for the course; deviations may be necessary.

1. Logistics

1.1. When and Where

- Fridays, 10:00-11:40am, Classroom South 325

1.2. Who

Instructor: **John Martin**

- Engineer in Residence (EIR), as part of Meta-GSU partnership to bring industry insights to the classroom
- Email: jmartin191@gsu.edu
- Office Hours:
 - Location: 25 Park Place, Room 742
 - Class office hours: TBD
 - Career office hours: 4-5pm, every day, see <https://discord.gg/QebAG4fJDv> for details

1.3. COVID-19

Please stay home if you have been exposed to COVID or are otherwise feeling unwell.

2. Course Information

2.1. Prerequisites

- CSC 2720: Design and Analysis of Algorithms, and at least one 4000-level course (can be taking concurrently)
- You must be graduating in Spring or Summer of 2023 to enroll in this course.

2.2. Textbook

There are **no** required textbooks for this course. Online documentation for the technologies we use will aid you in best practices, styling, and samples for reference.

2.3. Why should I take this course?

2.4. Course Goals

Upon successful completion of this course, you will be able to:

- Collaborate and effectively communicate with engineers you have never worked with.
- Enumerate, analyze, and evaluate the implementation of different solutions to an issue.
- Design an elegant solution to a real-life problem, including technical specs.
- Organize weekly scrums to accomplish a shared vision with large moving components.
- Employ tools and technologies that are relevant (as of 2022) to the industry.

2.5. Course Non-Goals

This course will **not** prepare you to:

- Learn how to code.
- Become an expert on how specific technologies, libraries, or frameworks work.
- Master algorithms-style interview questions.

2.6. Disability Accommodation

Students who wish to request accommodation for a disability may do so by registering with the Access and Accommodation Center. Students may only be accommodated upon issuance by the Access and Accommodation Center of a signed Accommodation Plan and are responsible for providing a copy of that plan to instructors of all classes in which accommodations are sought.

2.7. Basic Needs Statement

Any student who faces challenges securing their food or housing and believes this may affect their performance in the course is urged to contact the Dean of Students for support.

Furthermore, please notify the professor if you are comfortable in doing so. This will enable us to provide resources that we may possess. The Embark program at GSU provides resources for students facing homelessness.

3. Course Philosophy

Everyone in this classroom has the capacity for greatness, whether greatness means finishing your degree, landing an incredible job, building technology that changes the world, all of the above, or something completely different. For all of us to ensure that we can achieve these things, we must embrace certain core philosophies in this class.

3.1. Respecting **All** Voices

It is my responsibility, as well as yours, to treat one another with the highest regard and make space for everyone to have an equal voice and opportunity to learn. Everyone should feel comfortable sharing a differing perspective. As your instructor, I will place the voices of my

students above mine, and I hope to learn from them and use them to guide the direction of our course. If you don't agree with something I am doing, then I ask that you share this with me in any way you feel comfortable so we can work to address it.

3.2. Beyond the Classroom

This class will ask you to visualize yourself not as a student with all the scaffolding of a classroom, but as a real engineer navigating the turbulent waters of industry, which will require a larger set of skills. Learning to give and receive constructive, critical feedback, asking for help when you need it, and collaborating effectively on open-ended projects are all crucial for future success, and we'll practice all of those here. Every class should be demonstrably valuable. If you feel like you don't understand why a concept is important in either an academic, industry, or interview-style environment, then please let me know and I will work to address your feedback.

3.3. Making Mistakes

I do not expect anybody to master 100% of the content in this course on their first attempt. The open-endedness of the course content means that there are often no perfect answers, and I would be disappointed if this course did not push every student's limitations and challenge their capabilities. I hope to create an environment where people understand that they will be supported in these moments, both by me and other classmates. There will be ample time to try, fail, learn, and try again before grades are on the line.

4. Course Tools

4.1. Development Environments

VSCode: VSCode is an integrated development environment (IDE) where you can write and execute code on your machine or on a remote VM. It is free, and is also recommended for this course.

In addition to an IDE, you will need some kind of Linux system for installing Python packages and other software. If you are on a Mac, one of these is already built in to your computer - the Terminal app will do everything you need. If you're on a PC, read on.

Windows Subsystem for Linux (WSL): If you are on a PC, this is the recommended way of running an Ubuntu instance on your computer. There is extensive documentation online that you can consult to get up and running (more details will be shared in an early homework assignment).

Azure VM: If you don't want to develop on your local machine, GSU students all receive \$100 of Microsoft Azure credit as part of a partnership with Microsoft. You can apply this credit toward provisioning a **virtual machine (VM)**, which you can use for remote development. I will post instructions for doing this setup on iCollege. This should be a last resort - VMs are workable but

have proven unreliable for a significant number of people, and you should only take this route if you are absolutely unable to do work on your own machine.

You are welcome to go your own way with choosing an IDE and development environment, but you will have much less support if you don't use tools the instructors are familiar with. Homework assignments and projects will assume you have root access to an Ubuntu machine (i.e. you can install packages). There are certainly equivalent procedures for doing so from your own computer, but you will have to figure them out yourself if you take that path.

4.2. Assignments

Gradescope: We'll be using a platform called Gradescope for submitting most assignments. We're venturing outside of iCollege because doing so will allow me and the TAs to leave more detailed feedback on your individual work, especially on code you've written. I'll post instructions on setup and the general assignment submission flow separately.

GitHub organization: ALL of your programming assignments will be turned in on GitHub in a specific folder (organization).

4.3. Communication

- **Discord:** This is your best bet for quick questions and answers, from myself and from other students. I'll also post announcements here.
- **iCollege:** Please don't use my iCollege email. I don't know how to check it as of the writing of this syllabus. However, iCollege is where I'll post the bulk of class materials: assignments, grades, lecture recordings, and the like. I will cross-post announcements on Discord and iCollege.
- **Email:** If you ever prefer, you are also welcome to email me. Please include the course name in the subject line so I realize it's about class.

5. Academic Integrity

Please review the [Policy on Academic Honesty](#) for details on relevant procedures. Please remember that it is my professional obligation to report any academic misconduct to the Dean of Students. If I suspect academic dishonesty has occurred, I may take any necessary actions, ranging from giving you a 0 on the assignment up to filing notices of academic dishonesty and seeking additional punitive measures.

5.1 My Standard

I know many students find it difficult to define the line between academic dishonesty and getting help. As such, I will hold you to the following standard: **submitting code as your own work that you cannot explain or reproduce in a controlled environment is academic dishonesty**. Here are some concrete guidelines to help.

- **OK:** Discussing the problems ("what does this mean?") and discussing solution approaches in general terms ("try using a Subscription model").
- **OK:** Looking up syntax online as long as you have written the code yourself first.
- **OK:** Posting on Discord or StackOverflow for feedback is okay as long as you are not sharing code that reveals answers to other students.
- **NOT OK:** Posting or copying answers to any assignments or assessments (for example, on Chegg or GitHub) is never okay. This will automatically result in punitive measures.
- **NOT OK:** Blindly following instructions is never okay i.e. if a classmate tells you "Do X, then do Y, then Z" without understanding why, even if you implement it yourself.
- **NOT OK:** In general, copying any code from anywhere that you do not fully understand.
- **NOT OK: It is also academic dishonesty if you are on the other side of the transaction** i.e. giving a classmate instructions or sharing your code for a classmate to copy.

If you are still unsure about whether a specific behavior constitutes academic dishonesty, don't do it - instead, ask me, and I will be happy to clarify for you!

5.2 Cheating is not in your interest on any level

I am here to help you, not to catch you, and I do not wish to enforce academic integrity through fear. I much prefer to provide you with the necessary resources so that you do not feel compelled to resort to academic dishonesty. Should that temptation arise, please consider the following:

- I did not come here to fail people and stop them from graduating. If you are struggling with course content and feel like you can't catch up, **please** reach out to me and the TAs and we will work with you to get you back on track. No one failed this class in the fall semester except those who gave up before the final project. *It's going to be okay.*
- This course is about preparing you for industry jobs. Grades are one measure of how successfully you've internalized the lessons, but the bigger one is whether companies will hire and retain you. Even if you find ways to game the system here, it won't help you in the long run, and will in fact hurt your chances in the real world.
- Software mistakes can be enormously costly, both in money and in human life. Boeing 737 Max crashes and Therac-25 lethal radiation doses are both directly attributable to shoddy software construction. Habits matter, and taking shortcuts on the way to learning how to engineer properly could come back to haunt you more than I can say.

6. Grading Policy

6.1 Grading Scale + Breakdown

A+	A	A-	B+	B	B-	C+	C	C-	D	F
?	≥ 93%	≥ 90%	≥ 87%	≥ 83%	≥ 80%	≥ 77%	≥ 73%	≥ 70%	≥ 60%	≥ 0%

(A+'s are given at the instructor's discretion. They are intended to recognize truly distinctive work, and so there is no numerical cutoff to receive one.)

6.2 Grade Breakdown

- Individual Assignments (40%)
- Group Assignments (10%)
- Participation/Attendance (10%)
- Group Project (30%)
- Career Prep (10%)

6.3 Late Policy

Late work will receive 10% off per day, rounding up (so 1 hour late = 10% off, 25 hours late = 20% off).

However, **everyone** in the class will be given **10 free late days**, which can be used at any time throughout the semester. Here's how it works: suppose you turn in a homework assignment three days late, and you haven't used any late days yet. Then I will subtract three of your late days, but you will receive full credit on the assignment. In other words, **there are no late penalties until you run out of late days**. Each **project group** will receive **5 free late days**, to be used on any group assignment or aspect of the project.

Regardless of your late day balance, the last day to turn in anything for credit is **Dec 4** at 11:59 PM.

If you have truly extenuating circumstances, please let me know as early as possible and we can reach a resolution. Completing assignments on time does not come before your physical or mental health.

6.4 Individual Assignments (40%)

The majority of your grade will come from ~5-7 individual assignments, which will all involve some code or software system.

6.5 Group Assignments (10%)

A major objective of this course is getting you ready to work on software in teams, because that is *the* fundamental task of technical professionals. You will be assigned a project team in the first few weeks, and will work through some smaller assignments with them during this semester.

6.6 Group Project (30%)

Later in the course, your group will work on documenting and specifying your project, which you will begin in earnest in Capstone II. A substantial part of your grade will be how well you set yourselves up for success in the second semester.

6.7 Participation/Attendance (10%)

Attendance is required* and will account for 5% of your grade (I promise I will make classes worth your time - see above!). Another 5% will come from completing in-class exercises, when applicable.

*With attendance, a similar policy to late days exists - **you can miss 3 days before there are any repercussions** (so you lose points on your fourth absence, and then on all subsequent ones). In addition, I will honor legitimately extenuating circumstances if you let me know in advance.

6.8 Career Prep (10%)

My assumption is that most of you in this class are to some extent interested in working in the tech field at some point. Even if you have other plans like graduate school, there is a good chance you could find yourself applying to jobs now or later. If you are resolute on substantially different plans, you will have the opportunity to discuss that with me through a survey and when we meet for office hours so that we can make sure to support you with whatever your future career plans may be. Otherwise, I will assume you are at least somewhat interested in tech, even if unsure or if not immediately.

Companies are *currently* recruiting for summer interns and full-time employees for Summer 2023. A common misconception is that you should wait until you graduate to apply for jobs / internships, but actually, you can (and likely should) apply soon! That being said, applying to internships and jobs can be really daunting, and it is easy to have it deprioritized with everything going on in your life. I hope to help with this challenge by incorporating career preparation as part of this class. This may include assignments like:

- Crafting and polishing a resume
- Applying to the next steps of your career (whether that be an internship, job, or something else)
- Mock Interview practice (during Career OH, with Meta engineers, peers, etc)

Note that I want to push you and help you with the next steps of your career. If you are not planning to ever work or go to graduate school for this field or otherwise have a special

circumstance, please share with me so we can work something out, but I hope these assignments will still be a great learning experience.

6.9 Failure Policy

CSC 4351 and 4352 run together as a sequence, the completion of which fulfills the same graduation requirement as CSC 4350. If you fail CSC 4351, you will not be able to enroll in CSC 4352, and you will have to enroll in CSC 4350 in the Spring or Summer to graduate on time.*

*Disclaimer: This is just my understanding of where things stand. I am not a GSU advisor; this is not legal advice.

7. Course Schedule

Fall 2022

Date	Topic (Subject to Change)	Milestones (Tentative)
Aug 26	Syllabus + Intro	
Aug 26	<i>Add/Drop Deadline</i>	
Sep 2	Git 1: Commits, Branches	Groups assigned
Sep 9	Git 2: Pull Requests	Custom project proposals due
Sep 16	Python	Projects finalized
Sep 23	Flask/HTML/CSS; Client-Server Architecture	
Sep 30	Career Day: Resume Tips; Guest Speaker	
Oct 7	APIs; HTML Forms	
Oct 11	<i>Last Day to Withdraw</i>	
Oct 14	Review; Databases 1	
Oct 21	Databases 2	
Oct 28	Career Day: Applying to Jobs	
Nov 4	Product Lifecycle	Individual Project due
Nov 11	Requirements Engineering; System Specification	
Nov 18	Frontend Design	
Nov 25	<i>No class - Thanksgiving Break</i>	
Dec 2	Presentations	Project plans due

Spring 2023

(Assuming we continue to meet on Fridays)

Date	Topic (Subject to Change)	Milestones (Tentative)
Jan 13	Intro, Project Logistics	
<i>Jan 13</i>	<i>Add/Drop Deadline</i>	
Jan 20	Agile; Teamwork, Code Review	
Jan 27	JavaScript 1	Sprint Plan Due
Feb 3	JavaScript 2	
Feb 10	Career Day: Interviewing	Sprint 1 Due
Feb 17	Unit Testing	
Feb 24	Unit Testing 2; Continuous Integration	Sprint 2 Due
<i>Feb 28</i>	<i>Last day to withdraw</i>	
Mar 3	Career Day: Understanding Compensation, Negotiating Jobs, When to Leave	
Mar 10	Code Quality	Sprint 3 Due
<i>Mar 17</i>	<i>No class - Spring Break</i>	
Mar 24	System Architecture	
Mar 31	Deployment; Ethics	Sprint 4 Due
Apr 7	Software Security	
Apr 14	Career Day: Industry Success Tips; Guest Speaker	
Apr 21	Presentations	Project Due