

CS3300 Syllabus

Course Information

Course Prefix and Number: CS3300 A/GR

Course Name: Introduction to Software Engineering

Instructor: Roy, Nimisha

Course Description

CS 3300 introduces fundamental software engineering processes and practices, including requirements engineering, software architecture and design, implementation, testing, refactoring, and deployment. The course emphasizes hands-on application of these concepts through two team-based software development projects. Throughout the semester, students will also explore the use of AI-assisted and agentic software development tools across the software development lifecycle. Students will learn not only how these tools can support software engineering tasks, but also how to critically evaluate their effectiveness, limitations, and appropriate use. Particular emphasis is placed on verification, human oversight, and maintaining responsibility for AI-assisted work. Through project-based activities, students will compare conventional and AI-assisted approaches, experiment with emerging development workflows, and reflect on when AI improves - or does not improve - the software engineering process.

Some pre-requisites for success in this course include:

- Proficiency in Java programming
- Proficiency in using GIT - clone, add, commit, push, pull, branches, merge conflict, pull requests
- Basic client-side programming (JS, CSS, HTML)
- Basic introductory experience of using LLMs like Chat GPT
- Teamworking skills (being able to collaborate effectively with your teammates)
- Basic knowledge of UML diagrams and design patterns
- Cloud computing knowledge is a plus

Course Learning Outcomes

By the end of this course, students will be able to:

- Design, implement, and deliver **two substantial software systems** in a team environment
- Apply the **software development lifecycle (SDLC)** from requirements to deployment
- Design software using **architecture patterns, UML, and design patterns**
- Use modern tools and workflows, including **Git, code reviews, APIs, and cloud platforms**
- Develop and evaluate **software testing strategies** (black-box, white-box, TDD)
- Use and critically evaluate **AI-assisted and agentic software development tools and workflows** across different stages of the software development lifecycle, including their effectiveness, limitations, failure modes, and need for human oversight.
- Collaborate effectively and communicate work through **demos, reports, and presentations**

Grading Policy:

Letter grades are assigned according to the following convention:

- A = [90, 100) points (or more if bonus points are awarded)
- B = [80, 90) points
- C = [70, 80) points
- D = [60, 70) points
- F = [0, 60) points

Point values (fractional or otherwise) are NOT rounded to the next grade level. For example, 89.99 will be reported as "B." 90.00 (or higher) will be reported as "A." There are no exceptions.

Weights

Scores are calculated using the following assignment category percentages:

Assignments/Lecture Activities	20%
Project 1	30%
Project 2	30%
Team Collaboration	7%

Weekly Quizzes	10%
Surveys	3%
Extra Credit- Quizizz	1.5%
Extra Credit- Survey submissions	0.4%
Extra Credit in Project 1 and Project 2	TBD

Quizzes

We will have regular quizzes throughout this course. The weight of each individual quiz is calculated by dividing the total category weight (10%) by the total number of quizzes administered. All quizzes are mandatory. Quizzes are open notes and will be proctored using Honorlock. However, quizzes must be completed independently. Use of generative AI tools (e.g., ChatGPT, Copilot, or other AI assistants) is not permitted during quizzes.

To take a quiz, you must have a stable broadband internet connection, webcam, microphone, valid photo ID, and Chrome as your web browser. Please complete Quiz 0 (Practice Quiz), released at the end of Week 1, to familiarize yourself with Honorlock before the first graded quiz. Consult the [Honorlock Student Guide](#) for technical requirements.

Quizzes will primarily consist of conceptual questions assessing your understanding of topics covered in class that week. Each quiz will have a 7-minute time limit. Quizzes will generally be released on Fridays at 5:00 PM ET and will be due the following Tuesday at 11:59 PM ET. Please check the course website for the topics covered before beginning each quiz. Course staff availability is limited on Saturdays and Sundays, so students are encouraged not to wait until the weekend or deadline to raise technical or content-related questions.

Quiz answers will be released after all students, including students with ODS accommodations, have completed the quiz. Please do not post or discuss quiz questions or answers on Ed before solutions have been released. Quiz questions are randomly selected from a question bank; therefore, students may receive different questions.

Course Policies

Attendance and Class Participation

Engagement with the course is required to be successful in the course. Any in-class quizzes or individual assignments will be excused for absences that are verified by the Dean of Students Office (DoS). If a student misses class and feels it is excusable, they may submit their documentation to the Dean of Students office here: [Public CARE Report | Georgia Tech Student Conduct Management System](#) Please do not send me your documentation directly though you may notify me of your absence; I will not excuse any submission until I receive notification from the DoS Office. Participation in the projects is mandatory. Before the end of the course, student grades will be affected by peer evaluations from CATME.

Academic Integrity

Georgia Tech aims to cultivate a community based on trust, academic integrity, and honor. Students are expected to act according to the highest ethical standards. Review [Georgia Tech's Honor Code](#) and the student [Code of Conduct](#).

Any student suspected of cheating or plagiarism on a quiz, exam, or assignment will be reported to the Office of Student Integrity, who will investigate the incident and identify the appropriate penalty for violations.

CATME Peer Evaluations & Team Policies

CATME peer evaluations contribute to your individual grade; students are expected to contribute equitably to their team projects.

Team Accountability & Conflict Resolution

Teams are expected to collaborate professionally and address issues early.

- Concerns about team contribution should be raised during mentor check-ins or anytime during office hours.
- If issues persist, teams may request a group change before the end of Sprint 2
- Requests must include documented evidence (e.g., CATME results, reflections, Git activity)
- Final decisions on grade adjustments or team changes are made by the instructor

Artificial Intelligence Policy

Generative AI tools (e.g., ChatGPT, GitHub Copilot, and agentic coding tools) are intentionally integrated into this course as part of learning modern software engineering practices. You may use AI tools when permitted or directed by an assignment, including for activities such as software design, implementation, testing, debugging, refactoring, and agentic development workflows. The goal is not simply to use AI, but to learn how to critically evaluate, verify, and take responsibility for AI-assisted work.

Guidelines for Responsible AI Use

- Follow assignment-specific AI instructions. Different assignments may require, permit, limit, or prohibit particular uses of AI. The instructions for each assignment take precedence.
- Your reasoning and analysis must be your own. Reflections, evaluations, comparisons, interpretations of results, and explanations of your decisions must represent your own thinking. Do not use generative AI to write these portions of your submissions unless explicitly permitted.
- Evaluate AI-generated outcomes. When an assignment asks you to use an AI tool, you are expected to critically assess its output rather than accept it automatically. This may include checking generated code, testing functionality, identifying errors or limitations, revising outputs, and documenting your evaluation when requested.
- Maintain human oversight. AI-generated code or other artifacts should not be incorporated into your project without appropriate review and verification. You should understand what is being submitted and be able to explain and defend the decisions made.
- Document AI use when required. Some assignments will explicitly ask you to report the AI tools, prompts, workflows, agent interactions, revisions, failures, or human interventions involved. Provide this information accurately and completely.

Ownership and Responsibility

Anything you submit - including work created with the assistance of generative AI - is considered your work and carries your ownership and responsibility. Using an AI tool does not transfer responsibility for correctness, quality, security, academic integrity, or understanding to the tool. Always review, verify, and understand AI-generated output before incorporating it into submitted work. If you are unsure whether a particular use of AI is permitted for an assignment, ask the instructional team before using it.

Student-Faculty Expectations Agreement

At Georgia Tech, we believe that it is important to strive for an atmosphere of mutual respect, acknowledgement, and responsibility between faculty members and the student body. This

summarizes my expectations for you and what you can expect from me. In the end, simple respect for knowledge, hard work, and cordial interactions will help build the environment we seek. Therefore, I encourage you to remain committed to the ideals of Georgia Tech while in this class.

Expectations of Advisors and Advisees