

# CSCE 3444.003 – Software Engineering

Fall 2025 – 3 Credit Hours

University of North Texas – Department of Computer Science and Engineering

## Course Information

Class Timing: **Tuesday and Thursday 10:00 am – 11:20 am**  
Location: **NTDP K110**

---

Instructor: Hamed Jalali (preferred name: **Hans** — you may call me Hans)  
Office Hours: TBD  
Email: [Hamed.Jalali@unt.edu](mailto:Hamed.Jalali@unt.edu)  
Location: NTDP F267  
Zoom Link: <https://unt.zoom.us/j/8961265792>

---

TA: Ahmed Mshaal N Aljohani  
Email: [AhmedAljohani@my.unt.edu](mailto:AhmedAljohani@my.unt.edu)  
TA Office Hours: Thursday 12:00 pm – 2:00 pm  
Location: NTDP F288

---

Course Documents: All lecture notes, assignments, and announcements will be available on Canvas  
Class Textbook: Software Engineering: A Practitioner's Approach 9th Edition By Roger Pressman and Bruce Maxim. (supplementary)

---

## Course Description

This course introduces students to the principles and practices of software engineering, with an emphasis on the complete software development lifecycle. Students will learn how to gather and document requirements, design software systems using modern tools such as UML, apply project management techniques, implement and test software, and create user and system documentation. The course integrates collaborative group work on a large-scale project, giving students hands-on experience with agile methodologies, accessibility considerations, and real-world software engineering challenges.

## Why This Course Matters

Software engineering is more than just writing code. It's about building reliable, maintainable, and user-centered systems that solve real problems. This course will give you the skills and teamwork experience needed to design, develop, and deliver software in a professional setting.

## Learning Objectives

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

1. Gather and refine user functional and non-functional requirements for a large-scale software system and create a software requirements specification (SRS).
2. Perform software analysis and design tasks using recognized methods to produce a preliminary design specification based on requirements.
3. Apply project management principles, skills, and tools to create schedules, budgets, and production plans for software projects.
4. Use configuration management, project management, and design tools effectively throughout a project.
5. Implement accessibility and usability best practices in software design and evaluation.
6. Apply modern testing techniques, including unit, integration, and usability testing, to ensure software quality.
7. Maintain and evolve software systems by updating documentation, refactoring code, and addressing defects.






## Prerequisite

CSCE 3110 with a grade of C or better.

## Grading Criteria

<b>Team Project &amp; Presentation</b>	<b>40%</b>	Semester-long group project with midterm and final presentations
<b>Group Assignments</b>	<b>25%</b>	Collaborative tasks and project deliverables
<b>Individual Assignments</b>	<b>20%</b>	Individual contributions to SRS, Design, and Test Plan
<b>Peer Evaluations</b>	<b>5%</b>	Teammate feedback on contributions
<b>Attendance</b>	<b>10%</b>	Regular attendance and active participation

Grade Assignments are made based on the following:

	<b>A = 90-100%</b>
	<b>B = 80-89.9%</b>
	<b>C = 70-79.9%</b>
	<b>D = 60-69.9%</b>
	<b>F = less than 60%</b>



## Course Schedule: (subject to change; updates will be posted on Canvas)

Week	Class Topic	Expected Team Work	Deliverables
1	Course introduction, syllabus review, team formation, project brainstorming	Form teams, exchange contact info, set up Discord/GroupMe, brainstorm 3 project ideas	<b>Team formation</b>
2	Introduction to SDLC, Project Management basics	Choose project, research similar apps, pick tech stack (languages/frameworks), watch tutorials	<b>Project proposals</b>
3	AI in Software Engineering, GitHub Setup	Set up GitHub repo, create README, initialize project structure with AI help, install development tools	<b>Project Plan &amp; GitHub repo</b> <b>Individual Assignment 1</b>
4	Requirements gathering techniques, SRS writing	create UI mockups, start basic frontend	<b>SRS document</b> Status Report #1
5	Software design principles & UML diagrams	Code core features (1 per member), create class diagrams for existing code	<b>Individual Assignment 2</b> Peer Evaluation #1
6	Design Patterns	Refactor code using patterns, continue feature development	Status Report #2 <b>Design Document</b>
7	Code Quality, Testing basics, Code Review practices	Integrate all features, fix bugs, prepare prototype & presentation	<b>Test Plan outline</b> Peer Evaluation #2
8	Midterm project progress presentations with working prototype (live demos in class)	Reflect on presentation feedback, plan improvements for final	<b>Midterm report</b>
9	Test Plan, Usability testing concepts & planning	Write unit tests, document bugs found	<b>Complete Test Plan</b> Peer Evaluation #3 <b>Individual Assignment 3</b>
10	Accessibility in software development	Add alt text, keyboard navigation, color contrast fixes, test with accessibility tools	<b>Accessibility audit</b> Status Report #3
11	Software testing methods, Usability and Accessibility testing	Write more unit tests, test each other's features, create test checklist	<b>Testing coverage report</b> Peer Evaluation #4
12	Heuristic Evaluation Overview	Heuristic Evaluation	<b>Heuristic Evaluation Report</b> Status Report #4
13	Software maintenance & evolution	Complete final features, write user guide, clean up code, add comments	<b>User manual</b> <b>Feature freeze</b> Peer Evaluation #5
14	Thanksgiving – NO CLASS		
15	Final project presentations	Presentations and Final Demo	<b>Final code &amp; slides</b>

## Project Deliverables:

1. **Project Plan** – A 2-3 page document outlining your problem statement, team roles, development timeline, and tech stack. Include your GitHub repo link and when/how your team will meet weekly.
2. **Software Requirements Specification (SRS)** - A detailed document describing WHAT your software will do using the provided template. Include functional requirements, user stories, use cases, and UI mockups (5-8 pages).
3. **Design Document** - Technical blueprint showing HOW you'll build the software with architecture and UML diagrams. Include system architecture, class diagrams, database schema, and API endpoints (4-6 pages).
4. **Test Plan** - Document outlining your testing strategy with at least 20 specific test cases. Include what you'll test, how you'll test it, and your target code coverage goals.
5. **Final Presentation & Demo** - 15-minute presentation with 8-minute LIVE demo of your working software. Submit slides, final code on GitHub, and be ready to answer technical questions. (All team members have to participate in the presentation)
6. **Status Reports (Bi-weekly)** - One-page team update on what you completed, current work, blockers, and individual contributions.
7. **Peer Evaluations (Bi-weekly)** - Confidential rating of each teammate's contribution, effort, and collaboration. Your project grade may be adjusted  $\pm 10\%$  based on peer feedback.

## Attendance:

- Attendance is required and will be taken periodically.
- Notify me or the TA in advance if you must miss class.
- Participation in group work and presentations requires your presence.

## AI Usage Policy:

AI tools (GitHub Copilot, Claude, ChatGPT) are encouraged as development aids and must be cited using code comments. You must understand and be able to explain all submitted code. Include AI usage in your status reports. AI cannot be used for peer evaluations or reflection writing. The goal is to become an AI-augmented engineer, not AI-dependent. Uncited AI use or submitting AI-generated code without understanding constitutes academic dishonesty.

## Project Guidelines:

### THE PROJECT CAN:

- Be done in a language of your choosing (approval required if other than C/C++, C#, Python, Java, JavaScript)
- Be web-based or mobile app (you supply your own hardware)

### THE PROJECT MUST:

- Have appropriate scope for 3-4 team members over the term
- Have a Graphical User Interface (GUI)
- Have some portion that is used by a person (hence needing the UI)
- Have at least one customer/user group you can gather requirements from

### THE PROJECT MUST NOT:

- Be trivial (no simple calculators or to-do lists)
- Be static web pages or screens
- Be a game (Serious/educational games are OK with approval)
- Already exist (no cloning existing repositories)
- Harm the safety, security, or privacy of users
- Be primarily used by minors

## Communication Expectations:

- Email is the primary way to contact me or the TA.
- Expect a reply within 24 hours on weekdays.
- Keep messages short and clear. No need for overly formal AI-generated text.
- For grade questions, contact the TA first; follow up with me if needed.

## Late Submission:

- Assignments are due on the posted deadline.
- Late work loses 20% per day, up to 2 days late.
- After 2 days, the assignment will not be accepted and will receive a zero.
- Technical issues (e.g., internet outage, computer crash) are not valid excuses; plan ahead.

## Project Grade Scaling:

- Group project grades will be scaled based on peer evaluations
- Speak up and volunteer – "my team didn't give me work" is not an excuse
- Every member must participate in coding
- Failure to participate in coding = 0 for project grade
- Contact the instructor or TA early if team issues arise

## Academic Dishonesty/Plagiarism:

Students must follow the UNT Policy on Student Academic Integrity (Policy 06.003) and the CSE Department Cheating Policy. Academic dishonesty — including cheating, plagiarism, unauthorized collaboration, or using prohibited materials — will result in a failing grade for the course and a report to the Office of Academic Integrity.

## Student Perceptions of Teaching (SPOT)

SPOT is your opportunity to provide feedback on this course and instructor. It will be available near the end of the semester. Your responses help improve teaching and learning at UNT.

## Syllabus Revisions

This syllabus may be modified as the course progresses should the instructor deem it necessary. Notice of changes to the syllabus shall be made through Canvas and/or in-class announcements.

## Disability Accommodation

The University of North Texas complies with Section 504 of the 1973 Rehabilitation Act and with the Americans with Disabilities Act of 1990. The University of North Texas provides academic adjustments and auxiliary aids to individuals with disabilities, as defined under the law. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring accommodation, please see the instructor and/or contact the Office of Disability Accommodation at 940-565-4323 during the first week of class.

## My Perspective on Learning and Growth

---

*University is not just a place to gain knowledge. It is also a space for professional growth. You learn how to communicate, work in teams, solve problems, manage your time, stay disciplined, and take responsibility. Place to explore and discover things about yourself that you may not have learned elsewhere. I respect all teaching methods and believe in diversity in how we learn, but my approach focuses on hands-on activities, group projects, and applying what we learn directly to real tasks. There are no exams in this course, but your involvement in projects, assignments, and activities is key. Let's make this a collaborative and interactive class, and most importantly, let's make the best use of the time we have together. Time is your most valuable resource; use it wisely.*

---