

CS 3500: Software Practice

Course Details and Objectives

Fall 2023

Course Information

Description of CS 3500

Large software systems are some of the most complex artifacts ever created. The field of software engineering encompasses the study and application of effective tools and techniques for organizing the efforts of teams of people towards designing, developing, deploying, and maintaining software systems. CS 3500 is an introduction to software engineering. Although you will learn about all the steps involved in creating a system, we will focus primarily on the software construction step by building on the programming background that you acquired from CS 1410 and CS 2420. CS 3505 will in turn build on this course by treating the rest of the software development process in more depth. In CS 3500, we will bridge the gap between introductory programming and full-blown software engineering.

Instructor

Travis Martin, *Email:* travis@cs.utah.edu *Office:* MEB 3122

Lectures

Lectures are Tuesdays and Thursdays from 12:25pm - 1:45pm in S BEH AUD.

Labs

When enrolling in this class, you also enrolled in a lab/discussion section. These are smaller, weekly meetings led by the TAs designed to give you practice with the concepts discussed in class and other necessary instruction. Attending and completing labs is required for credit. Attend the lab section you are enrolled in.

Laptop Requirement

Per School of Computing policy, students enrolled in a CS class with a lab/discussion component are required to use their own laptop for the lab. Students are responsible for administering their own laptops, such as installing Windows and Visual Studio. Windows is strongly recommended, as it is used by most of the course staff and therefore there are more ways for you to get help if you get stuck with IDE problems. Macs, including those with M series chips, are also supported, but with minimal options for IDE-specific help, so you might be required to get yourself unstuck. For recommendations and further information, please see the official policy: https://handbook.cs.utah.edu/2023-2024/CS/Academics/laptop_policy.php

Class Website

The class website is on Canvas at <https://utah.instructure.com>. It will contain all pertinent course info and materials such as lectures, announcements, updates, corrections, and grades. Students are required to check their email and Canvas regularly until final grades are posted.

Coursework

Grading

Your grade for this course will be determined by the following:

Assignments	40%
Midterm exam	20%
Final exam	25%
Labs	8%
Quizzes and In-class polling	7%

If X is your overall course score, letter grades will be assigned using the below scale. Scores will *not* be rounded.

$100 \geq X \geq 93$	A	$90 > X \geq 87$	B+	$80 > X \geq 77$	C+	$70 > X \geq 67$	D+	$60 > X \geq 0$	E
$93 > X \geq 90$	A-	$87 > X \geq 83$	B	$77 > X \geq 73$	C	$67 > X \geq 63$	D	$63 > X \geq 60$	D-
		$83 > X \geq 80$	B-	$73 > X \geq 70$	C-				

Assignments

Roughly the first half of the semester will consist of weekly software development assignments using C# in Visual Studio. These assignments will all build towards one larger software system. The specifications and due dates will be posted online each week.

The second half of the semester will consist of a team project implementing a multiplayer networked game. Development of this project will be mostly from scratch.

You will hand in your software through your *Github* repository (info on this forthcoming). Assignments will be graded on documentation, comments, design of the code, as well as correctness in execution. Partial credit may be given for incorrect or incomplete assignments, but it must be clear that a strong attempt was made. If your software does not compile or run according to the specifications, no credit will be given.

Exams

The midterm exam will be given during the regular class time in the regular class room on Thursday, October 5th. The final exam will be held on Wednesday, December 13 from 10:30am - 12:30pm in the regular class room. All exams are written exams.

Quizzes

There is a short quiz on Canvas associated with each lecture. In general, there will be two quizzes due each week, and they are designed to not take very long. Quizzes will not be accepted late. The lowest two quiz scores will be dropped.

The purpose of these polls is to give students a chance to apply what they're learning, not to judge understanding, and so the grading is lenient. For any particular class period, you will get full credit if you get at least 50% of the poll questions correct. And several scores will be dropped.

Getting Help

See the “Getting Help” page on Canvas for information about my office hours, TA help hours, discussion boards, etc.

Course Guidelines

Piazza

Piazza is used for questions and discussions related to the course. Students must use their first and last names (as they appear in Canvas) in their Piazza profile, such that the correct name is visible to the instructor and TAs on posts. Note that students may select to post anonymously, such that their name is not visible to classmates.

Working Together

Some of the work in this class will be completed in pairs with another student. When pair work is required, students must adhere to the techniques of pair programming. Partners are required to contribute equally to the work. Students are encouraged to discuss high-level solution strategies with fellow classmates, but each student is responsible for writing their own answer.

Cheating is: sharing (outside of a partnership) written or electronic work either by copying, retyping, looking at, or supplying a copy. Cheating is not: discussing concepts, answering questions about concepts or clarifying ambiguities, or helping someone understand how to use the class tools and software.

The rules for using Large Language Models or other AI assistants for help in this class are the same as the rules for getting help from other classmates. That is, you may ask an AI assistant to help teach you fundamental concepts, or to remind you the particulars of an API, but you must ultimately write your own solution, and submitting non-trivial amounts of code written by an AI is academic misconduct.

There must be no collaboration during quizzes or exams. See the class web page for the cheating policy for this course. See the University of Utah Student Code for a detailed description of the University policy on cheating. Any student found cheating will fail the course. Supplying cheated materials is considered cheating just as using them is.

Late Work

Assignments may be submitted up to 3 days late, with a substantial penalty. Late assignment submissions will incur a penalty of 20% of the assignment's max value (no matter how many days late it is). Work submitted more than three days late will not receive credit. An assignment is considered late if submitted any amount of time past the deadline, as measured by the submission system. Any delays caused by the submission system or corrupt/lost files is not an excuse for lateness. **Do not risk submitting at the last minute.** Late days apply to programming assignments only; other work, such as labs and quizzes, will not be accepted late.

Each student is allowed one free late submission – this will be automatically applied to your first late submission. This is designed to accommodate any number of unexpected emergencies students may face, but an even better way to handle emergencies is to work to always finish your assignments well before the deadline.

Assignment Correction

Each student may fix *one* assignment and resubmit it to earn half of the missing points back. So, if the student got a 20/40 on the autograder tests, and a 20/60 on other parts of the

assignment, and fixed everything they were deducted for, they would get 10 points back for the autograder and 20 points back for the other part, for a final score on that assignment of 70/100. Late penalties cannot be corrected.

A corrected assignment *must* be committed to GitHub less than 7 days after the assignment grade is released. A student may correct multiple assignments, but at the end of the semester they must choose a single assignment they want to submit to the TAs for a new grade.

Github

Your code for this class will be stored on a private Github repository owned by the course staff. Your repository may be deleted after the end of the semester. Make sure you backup your work if you want to keep it after the end of the semester. Note that your repository may be deleted some time after the semester ends.

College of Engineering Guidelines

For information on withdrawing from courses, appealing grades, and more, see:
<https://www.coe.utah.edu/semester-guidelines>

Students with Disabilities

The University of Utah seeks to provide equal access to its programs, services, and activities for people with disabilities. If you need accommodations in this class, reasonable prior notice needs to be given to the Center for Disability Services, 162 Olpin Union Building, 581-5020 (V/TDD). CDS will work with you and the instructor to make arrangements for accommodations.

Safety

The University of Utah values the safety of all campus community members. To report suspicious activity or to request a courtesy escort, call campus police at 801-585-COPS (801-585-2677). You will receive important emergency alerts and safety messages regarding campus safety via text message. For more information regarding safety and to view available training resources, including helpful videos, visit safeu.utah.edu.

Violence and harassment based on race, national origin, color, religion, age, disability, sex or gender (which includes sexual orientation and gender identity/expression) is a civil rights offense and will not be tolerated. If you or someone you know has been harassed or assaulted, you are encouraged to report it to the Title IX Coordinator in the Office of Equal Opportunity and Affirmative Action, 135 Park Building, 801-581-8365, or the Office of the Dean of Students, 270 Union Building, 801-581-7066. For support and confidential consultation, contact the Center for Student Wellness, 426 SSB, 801-581-7776.

Positive COVID-19 Tests

Any student who tests positive for COVID-19 must self-report via coronavirus.utah.edu

Course Objectives

Upon completion of CS 3500, students will be able to:

- Design and implement large and complex software systems (including concurrent software) through the use of process models (such as waterfall and agile), libraries (both standard and custom), and modern software development tools (such as debuggers, profilers, and revision control systems)
- Perform input validation and error handling, as well as employ advanced testing principles and tools to systematically evaluate software
- Apply the model-view-controller pattern and event handling fundamentals to create a graphical user interface
- Exercise the client-server model and high-level networking APIs to build a web-based software system
- Utilize a modern relational database to define tables/objects, as well as store and retrieve data
- Appreciate the collaborative nature of software development by discussing the benefits of peer code reviews