CICS 210 Syllabus, Fall 2023

Table of Contents

Welcome

In this course, each voice in the classroom has something of value to contribute. Please take care to respect the different experiences, beliefs and values expressed by students and staff involved in this course. My colleagues and I support UMass's commitment to diversity, and welcome individuals regardless of age, background, citizenship, disability, sex, education, ethnicity, family status, gender, gender identity, geographical origin, language, military experience, political views, race, religion, sexual orientation, socioeconomic status, and work experience.

View this syllabus as a guide to the course. It provides important information regarding the course, its assignments, policies, grading, and available university resources.

You should read it once, thoroughly, at the start of the semester. However, this document should be considered a working document. It is possible throughout the semester that a topic may take more time than expected, topics or assignments may change, or some material may be canceled or delayed due to a snow day or another emergency. If that is the case, the syllabus and schedule will be updated and a revised version will be distributed to the class.

Course description

An introduction to the design, analysis, and implementation of data structures. This course teaches you how to build, test, debug, document, and evaluate objects that encapsulate data and their associated operations using programming constructs and data abstractions of a modern programming language. Concepts and techniques covered include linear and non-linear structures, recursive structures and algorithms, traversal algorithms, binary search trees, balanced trees, priority queues, union-find, hash tables, Bloom filters, and graphs. We will also informally compare and contrast the run time efficiency of algorithms and their performance characteristics including the concept of worst-case running time analysis and the classification of algorithms in terms of constant, logarithmic, linear, log linear, quadratic, and exponential time using Big-O notation.

Prerequisites: CICS 160 or INFO 190T with a grade of C or better.

Learning outcomes: At the completion of this course you will be able to:

- Design, implement, and analyze fundamental abstract data types and data structures such as lists, stacks, queues, priority queues, trees, sets, hash tables, union-find, heaps, Bloom filters, and graphs;
- Define and implement recursive structures and algorithms over those structures;
- Demonstrate an understanding of iteration and traversal to implement iterators for the aforementioned data structures;
- Define and implement the operations and algorithms associated with fundamental data structures;
- Compare data structure tradeoffs to select the appropriate implementation for an abstract data type;

- Informally explain, compare, and contrast the run time efficiency of algorithms and their performance characteristics including the concept of worst-case running time analysis and the classification of algorithms in terms of constant, logarithmic, linear, log linear, quadratic, and exponential time;
- Explore and use various programming abstraction techniques including object-oriented and functional approaches to implement data structures;
- Identify and remedy flaws in a data structure implementation that may cause its behavior to differ from the intended design through debugging and testing;
- Increase your proficiency in writing code including designing, documenting, writing, testing, and debugging.

What, when, where, who

CICS 210: Data Structures
Tuesdays and Thursdays, 1:00–2:15 in Marcus Hall room 131
Labs meet Mondays (see SPIRE for your lab section time and location)

Instructor: Marc Liberatore (please call me "Marc")
Email: liberato@cs.umass.edu (though see note below about Campuswire)
Phone: 413-545-3061 (on campus: 5-3061)
Office: Computer Science, Room 318
Office hours: Thursdays 2:30 (with occasional announced exceptions)

Graduate Teaching Assistants (TAs): Uma Pal, Mashrur Rashik, Matthew Weiner, Atif Abedeen, Ryan Hilton

Undergraduate Course Assistants (UCAs): Gavin O'Neil, Zeina Zahoori, Yana Djotova, Lana Vu, Victor Chen, Sowrathi Somasundaram, Ian D'Souza, Vani Gupta, Anshul Gupta, Mariana Jaramillo

Please see CampusWire for TA and UCA office hours, it's a big list and subject to change.

Note that office hours end after last day of classes! Unless you hear otherwise, we won't be holding regularly-scheduled office hours after December 8th.

Required and optional material

A computer capable of running the most recent LTS version of the Java Development Kit (JDK17) is required. The ability to use Visual Studio Code and its Java development environment is also required. Finally, we strongly recommend a laptop computer you can bring to class and lab. As time permits, course staff may be able to help with minor technical issues, but we are not IT support staff; we cannot generally solve installation or configuration issues, especially remotely.

There is one required (online) textbook for this course, available at zyBooks using subscription code: UMASSCICS210LiberatoreFall2023

Technical material will be presented in lecture. Note that while we do an accelerated Java review in the first few weeks of the semester, it is expected you have the familiarity with

Java that the prerequisite provides. If you do not, you are responsible for self-studying to that level.

For students who want additional Java references, I suggest several *optional* resources:

- Java for Python Programmers may be helpful for those of you coming from a Python background.
- Similarly, Teach Yourself Java in 21 Minutes may be helpful.
- Past students have reported both CodingBat and PracticeIt to be helpful resources.
- The Java Tutorials, are (old) guides to the Java language.
- Similarly, the Java Platform API provides a comprehensive description of all classes Java Platform; we'll make extensive use of some of them, and they are fully documented by Oracle.
- Java Precisely, 3rd edition, by Peter Sestoft. If you want to know something about the Java language – syntax or semantics – this book is a great reference. Note that it is not a textbook or a how-to manual, but a reference book that explains what specific part of the language mean. It also provides some explanation of important parts of the Java standard library (also known as the class library or Java Platform API).
- There are many other Java references online; if you find one that you feel is particularly helpful, please recommend it to the class!

For students who want additional data structures references, I suggest the free online textbook Open Data Structures. The Java version is written in a pretty terse style that I don't find super helpful (I prefer the pseudocode version) but it's easy enough to bounce between them. You might also find the OpenDSA project helpful; in particular, their CS2 and CS3 texts.

Code of conduct
- The course staff are committed to providing a friendly, safe and welcoming environment for all, regardless of level of experience, gender identity and expression, sexual orientation, disability, personal appearance, body size, race, ethnicity, age, religion, nationality, or other similar characteristic.
- Please be kind and courteous. There's no need to be mean or rude.
- Respect that people have differences of opinion and that differing approaches to problems in this course may each carry a trade-off and numerous costs. There isn't always a single right answer to complicated questions.
- Please keep unstructured critique to a minimum. Criticism should be constructive.
- We will informally warn you, once, if you insult, demean or harass anyone. That is not welcome behavior. After that we will report your behavior to the Dean of Students office. We interpret the term "harassment" as including the definition in the Citizen Code of Conduct under "Unacceptable Behavior"; if you have any lack of clarity about what might be included in that concept, please read their definition and then ask us for clarification. In particular, we don't tolerate behavior that excludes people in socially marginalized groups.

- Private harassment is also unacceptable. No matter who you are, if you feel you have been or are being harassed or made uncomfortable by a member of this class, please contact a member of the course staff immediately (or if you do not feel safe doing so, you should contact the Chair of the Faculty of CICS, currently Prof. Erik Learned-Miller, elm@cs.umass.edu, or the Dean of Students office). Whether you've been at UMass for years or are a newcomer, we care about making this course a safe place for you and we've got your back.
- Likewise any spamming, trolling, flaming, baiting or other attention-stealing behavior is not welcome.

Communication policy

Per the University Email Policy, you are expected to check your University email regularly – at least once a day. I will use your UMass email address as your primary point of contact in all online tools we use (such as Campuswire and Gradescope) and as my primary means to contact you individually outside of class. Group announcement will be posted to Campuswire, which you can configure to send you notifications whenever an instructor makes a post.

For course-content related questions, especially questions that other students might benefit from seeing the answers to, please use Campuswire. For other questions and unusual logistics stuff, like extensions requests, exam rescheduling, and so on, use the "course contact form," linked to on Campuswire. If you do need send me email because of something particularly confidential, please include "CICS 210" in the subject line to make sure I see them in a timely fashion. *Do not* use email for questions about course content or routine requests – the volume of such requests is too high for me to manage by email.

Course staff typically respond to Campuswire questions and other requests within two business days, but I (Marc) do not typically respond to communications after about 5pm or on weekends. Course staff tend to get a higher volume of messages when a deadline is approaching. If you contact the course staff (that is, at least one TA/UCA *and* the instructor) at least two full business days *before* a deadline, you are guaranteed a reply before the deadline. Otherwise we'll do our best, but no guarantees.

Campuswire

Campuswire is a online discussion management system. It will be used as the main communications hub for this class. Please follow these guidelines in your use of Campuswire:

- You should use Campuswire to ask questions and get advice on assignments. But you should not use Campuswire to step through each and every tiny problem you encounter in an assignment.
- You may not post assignment solutions to Campuswire, either in questions or in answers to others' questions.

- If you post code you are working on so that we can help you, you should do so only through private posts (visible only to the course staff). But it's usually best for us if you upload your assignment to Gradescope so we can see the whole thing!
- Try hard to ask a good question. Please do not post code and ask only, "what is wrong with my code?" See, for example, https://jvns.ca/blog/good-questions/ for constructive advice on asking questions.
- You are encouraged to help other students by answering their questions!

The course staff will monitor Campuswire and answer your questions in a timely manner (generally within a business day). But please do not expect us to provide real-time answers on Campuswire, especially in the last few hours before an assignment is due.

If a question has already been answered in a previous post we may not respond to you, instead directing you to the previous answer. If a question does not follow the guidelines above we may not answer it. If we find that a private question is relevant to a larger audience, we may make mark it public to help others in the course.

The course Campuswire will be archived and closed after our final exam.

Time management and what to expect

As a general guideline, the university suggests that students spend three to four hours of time on a class per credit hour. This is a four-credit course, therefore you should plan to spend twelve to sixteen hours a week on this class.

In a typical week, you will:

- read any assigned readings, and perhaps any suggested / optional readings
- attend lab where either there may be an individual or group activity, further discussion of the week's topics, or a quiz
- attend lecture, participate in any in-class exercises, and ask questions
- complete programming and zyBooks assignments (these two are where you will spend the bulk of your time in this course)
- optionally, attend office hours

You will also attend two out-of-class evening midterms and a final exam.

Attendance

This is an in-person course; attendance is expected and required in both lectures and lab. You are also required and expected to complete assignments by their due date, though see below for details on the class late policy.

- If you will be absent or miss deadlines due to religious reasons, you must provide me with a written list of such dates within one week of your enrollment in the course or the start of the course, whichever is later.
- If you will be absent or miss deadlines for a University-related event, such as an athletic event, field trip, or performance, you must notify me as soon as possible.

- If you are absent or miss deadlines for health reasons, I expect you to notify me as soon as possible – generally before, not after, the missed course element.
- If you are absent or miss deadlines for other extenuating non-academic reasons, such as a military obligation, family illness, jury duty, automobile collision, etc., I expect you to notify me as soon as possible and provide written documentation (again, if you seek excusal or extensions).

If you miss an assignment deadline for health reasons, I may excuse you from the assignment or grant you permission to submit late. Everyone gets sick from time to time and I will not generally require documentation for one or two missed deadlines. More frequent missing of deadlines for health reasons implies a multi-week illness or chronic condition, for which I will require documentation. Note that if you are in isolation or quarantine due to COVID, the UMass contact tracers will provide you documentation on request.

If you miss a quiz or exam *without prior notice*, I will require an explanation and clear written documentation in order to judge whether the absence is excusable.

If you must miss a quiz or exam for an excusable reason, I will work with you to find an acceptable time for you to take a makeup. Quizzes must be made up within three business days, unless there are documented exceptional circumstances (such as a hospitalization or extended jury duty), in which case they will be excused.

If you add the class late, I will excuse you from missed work, but you are responsible for both notifying me when you add in a timely fashion, and for completing the work on your own.

Incompletes

Incompletes will be granted only in exceptional cases, and only if you have completed at least half the course up through the time of the incomplete request with a passing grade. Otherwise, withdrawal is the recommended course of action.

Schedule

The approximate schedule for the course is as follows:

1. Introduction to Data Structures and Algorithms (Java review)
2. Object orientation; Java Development (IDEs, testing, etc.)
3. Linear Structures Review (arrays, Array Lists, Linked Lists)
4. Generics, Recursion Review
5. Big-O analysis, Amortization
6. Iteration/traversal
7. Trees (binary search trees)
8. Balanced Trees (AVL and B-trees)
9. Priority Queues (heaps and treaps) / Heapsort
10. Maps, Hash Tables, Skip Lists

11. Sorting and Selection
12. Graphs (representation; BFS and DFS search algorithms)
13. Advanced Structures: Union-Find, Bloom filters, Tries
14. FINAL exam

Grading

We expect the breakdown for the final course grade to be as follows:

5% zyBooks problems
5% in-person lab exercises
15% programming assignments
15% quizzes
15% midterm 1
15% midterm 2
30% final exam

The numerical cutoff for final course letter grade assignment will be made after all grading is completed. Expect to require at least a 93 to get an A, a 90 to get an A-, an 87 to get a B+, an 83 to get a B, an 80 to get a B-, and so on.

Individual grade items are not typically curved.

There are no unannounced opportunities for extra credit in this course; please do not ask. If you do ask I will point you at this line in the syllabus and deduct 1% from your final course average. (OK, not the first time you ask – I'm not a complete monster!)

Also: It's 2023. Storage and bandwidth are virtually free. Back your work up, store it in the cloud, whatever. "My computer crashed" won't be acceptable as an excuse in this class for missing deadlines.

I will retain all graded materials for this course until the end of next semester. If you wish to review them, please come to see me during office hours (or make an appointment).

You are responsible for monitoring your grades, typically visible on Gradescope or within zyBooks for those problems. If you encounter any issues with your grades, you will have one week past the first posting of a particular assignment's grade to Gradescope to contact the course staff so that we can investigate. Please contact us via the regrade request system in Gradescope. We will not generally accept questions about an individual assignment's grade beyond this one week, so you must be prompt.

Assignments

I will post assignments – programming, zyBooks problem sets, and the like – about once a week, and you will typically have about one week to complete them. They will be announced in on Campuswire and must be submitted through Gradescope and/or zyBooks. Assignments may be completed individually or with a partner. See the course honesty policy, below, for more details.

Each week's programming assignments are worth the same amount of credit toward the final grade – if we have 13 assignments, each will be worth (15% / 13 = about 1.15) percentage points toward your course grade, regardless of how many "points" each assignment is listed as having in Gradescope.

Similarly, each week's zyBooks assignments will be equally weighted, regardless of the point value listed per assignment.

You are responsible for submitting your work to Gradescope and zyBooks. Email submissions, whether late or on time, will not be accepted.

You are responsible for verifying that you have submitted the intended versions of your files and answers. Similarly, you are responsible for ensuring that the final submission you make is the one you wish to have graded. Requests to substitute another version may be granted entirely at the discretion of the course staff.

For autograded assignments, you are responsible for submitting code that compiles and runs on the autograder; if you submit code that does not compile or that gets stuck in an infinite loop on the autograder, you will receive no credit for the autograded portion of that assignment. In other words, you should expect to receive the grade that Gradescope produces for autograded items; in particular, "But it runs fine on my computer" will not serve as the basis for an appeal. If you are seeing a discrepancy, you must ask us about it *before* the assignment deadline to guarantee a chance of consideration – after the deadline, it's entirely at our discretion.

You are responsible for completing and uploading your work before the deadline. The deadline for an assignment is not the time by which you must finish working on your computer; it is the time by which you must successfully upload your answers and confirm the system has recorded those answers. We recommend that you upload your answers at least one hour before the deadline. File system timestamps on your local hard drive or in a Github repository or the like are never acceptable as evidence of existence of a file prior to the deadline, because you are under complete control of that timestamp. Requests to submit after the late submission deadline will be handled as described in "Extensions and late policy."

Attempts to manipulate, game, or otherwise incorrectly use the autograder will be treated as academic dishonesty.

Labs and quizzes

Some lab sessions will function as discussions or as opportunities to complete exercises in a less formal setting. These exercises will typically be pass/fail or attendance based.

In other sessions (about every other) you will take a short quiz. These will be announced, generally by the lecture before the lab where the quiz will take place. Each will consist of a few programming questions, and they may include a few other short-answer questions as well. The quizzes are designed to be completed in 25 minutes, though you will have the entire 50 minutes to complete them. Like assignments, the quizzes are equally-weighted in regards to your course grade, regardless of how many "points" are listed on each quiz.

Unlike assignments, these quizzes **must be completed on your own**, without collaborating with your peers or performing Internet searches or the like (they are closed-book, closed-note, closed-computer, etc.). It's critical that you (and we) be able to evaluate your *own* learning progress.

Midterms and the final exam

There are two midterm exams and a final exam. Each is cumulative.

Like the quizzes, these **must be completed on your own**, without collaborating with your peers or performing Internet searches or the like (they are closed-book, closed-note, closed-computer, etc.).

The midterm exams will be given at a University-scheduled time and place. They are tentatively scheduled for October 19th in Hasbrouck 20 and November 29th in Marcus 131 at the standard time of 7–9pm. There are also proctored double-time exams starting at 5pm those days – if you have a DS accommodation for 1.5x or double time, or for a distraction-reduced environment, you may take those exams. Contact me for details.

The final exam will be given on our scheduled exam day, currently listed by the Registrar as 10:30am on Thursday, December 14th. **You must achieve a passing grade on the final exam to pass the class.**

Please note (from the [Academic Rules and Regulations]):

> …it is University policy not to require students to take more than two final examinations in one day of the final examination period. If any student is scheduled to take three examinations on the same day, the faculty member running the chronologically middle examination is required to offer a make-up examination if the student notifies the instructor of the conflict at least two weeks prior to the time the examination is scheduled. The student must provide proof of the conflict. This may be obtained from the Registrar's Office, 213 Whitmore.

You are responsible for clearing your schedule at the beginning of the semester to take exams. The Registrar announces the final exam period before the beginning of classes, and I will announce scheduled mid-term exams as soon as the Registrar provides me with that information. In particular, if you cannot commit to taking the final exam during the exam period, you should drop this class immediately. Makeup exams will be offered only in those cases where required by university policy.

Extensions and late policy

zyBooks assignments must be submitted on time to be accepted for credit.

Programming assignments are due at the time specified in Gradescope, and we strongly suggest that you always submit before that deadline, even if your solution is incomplete. That at least shows us that you are making progress.

But life happens: you miss the deadline because of other courses, illness, events in your personal life, and so forth. Rather than the burden of you having to ask for an extension in those circumstances, we will configure Gradescope to accept late submissions, usually up to 48 hours after the deadline. **There will be no penalty for these late programming assignment submissions, up to the 48-hour "late" deadline.** This is automatic extension you may choose to grant yourself. We trust you to do so wisely – often, making use of them will result in you having less time to work on the next assignment. The purpose of automatic extensions is to give you a tool to manage the demands of life, including the following:

- routine illness
- minor injury
- travel
- job fairs
- job interviews
- large workloads in other courses
- extra-curriculars

Be aware that after Gradescope decides that the late submission deadline has passed – even one second past it – no further submissions will be accepted without course staff intervention.

Beyond those automatic extensions, we will grant exceptional extensions in truly exceptional circumstances. Contact us *before the regular deadline* to request such an exceptional extension, and be prepared to provide documentation supporting your inability to work on the assignment for the time before the regular deadline as well as the late deadline. In particular, "I had other big assignments to work on / exams to study for" is not generally a sufficient reason for an extension. Learning to manage deadlines is part of the learning process.

Academic honesty

General academic honesty statement

Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst. Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to: cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair. Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent.

In addition, you should read the UMass Academic Honesty Policy, as ignorance of the policy is not an excuse for violating it.

Course-specific academic honesty policy

Investigating academic dishonesty is an unpleasant experience for both the instructor and the student. Please help me by avoiding any questionable behavior.

Academic dishonesty is usually the result of other problems in school. Please come see me or the other course staff if you are unable to keep up with the work for any reason and we will do our best to work something out. The course staff want to see you succeed, but we cannot and will not tolerate academic dishonesty.

Be aware that if something looks like academic dishonesty to us, we will treat it as such, unless you can provide strong evidence to the contrary. When in doubt, it is your responsibility to contact the course staff about whether a potential action would be considered academic dishonesty.

What is permitted and what is not?

You may discuss material with others, but when collaboration is forbidden (specifically: on the quizzes and the exams), your work (code and prose) must be entirely your own. The quizzes and final exam are entirely closed-book, and you may not communicate with your peers during the exam period, nor consult notes, nor use electronic devices, etc.

You may not provide your solutions to others, either directly or via some sort of public or private posting, except when collaboration is explicitly permitted (as it is on all take-home assignments) *and* when both you and the other person – who must be your partner on that assignment – are currently enrolled in this course.

You may not copy code/paste from online sources – except for the current semester's course site. You may not use third-party online forums such as StackOverflow to ask for specific help on assignments, nor third-party course "notes" sites that traffic in solutions to assignments, nor may you search for solutions online. Copying and pasting code from another student (whom you are not submitting with) or from a third party is a violation of academic honesty. We may endeavor to detect this by any means available to us, including automated similarity analysis of submitted assignments. It's the nature of introductory courses that there is "nothing new under the sun." Be that as it may, I'm not asking you to Google solutions, I'm asking you to write them yourself, and to ask for help if you need it.

Group work policy

Assignments permit a single partner (a partner means one other person, besides yourself). Having a partner is optional. If you work with a partner, please submit only one copy of the assignment, and identify your partner using the Gradescope interface so you both receive credit.

You are free to work with different partners on each assignment. Make sure to acknowledge any past partners whose work might have influenced your own as appropriate (either in comments in code, or in the text of written answers).

If you work with partner(s), you must do all the work together. It is against the rules to split up the work, or to have one person do it and another person "check" it, or to have one person write the code and another person write the tests, etc. When you submit as a partnership or group, you are asserting that all the work was done together.

The above rule has implications for academic honesty policy violations. Consider that carefully. If your partner went off and implemented a lot of code "on their own," how do you know they wrote it? Once in a while, it turns out somebody actually copied the code or solution from elsewhere. You will be culpable if your partner does this, because by submitting as a partnership you are claiming the solution to be jointly written by both of you.

Accommodation statement

The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented physical, psychological, or learning disability on file with Disability Services (DS), you may be eligible for reasonable academic accommodations to help you succeed in this course. If you have a documented disability that requires an accommodation, please notify me within the first two weeks of the semester so that we may make appropriate arrangements.

Title IX statement

In accordance with Title IX of the Education Amendments of 1972 that prohibits gender-based discrimination in educational settings that receive federal funds, the University of Massachusetts Amherst is committed to providing a safe learning environment for all students, free from all forms of discrimination, including sexual assault, sexual harassment, domestic violence, dating violence, stalking, and retaliation. This includes interactions in person or online through digital platforms and social media. Title IX also protects against discrimination on the basis of pregnancy, childbirth, false pregnancy, miscarriage, abortion, or related conditions, including recovery. There are resources here on campus to support you. A summary of the available Title IX resources (confidential and non-confidential) can be found at the following link: https://www.umass.edu/titleix/resources. You do not need to make a formal report to access them. If you need immediate support, you are not alone. Free and confidential support is available 24 hours a day / 7 days a week / 365 days a year at the SASA Hotline 413-545-0800.

Other academic regulations

The Office of the Registrar publishes Academic Regulations yearly. You should be familiar with them. Particularly relevant are the policies on attendance, absences due to religious observance, and examinations.

A word about putting your solutions on GitHub, GitLab, BitBucket, etc.

Per the course-specific academic honesty policy, you are not permitted to make your solutions to the assignments in this class available to others. This includes reposting them to public GitHub repositories (or any service where another student might plausibly see them).

A word about copyrights

Some of the material (lecture notes, lectures, some assignments, and so on) in this course is original work created by the instructor (Marc Liberatore); exceptions are clearly noted. These works are protected by U.S. copyright laws and by university policy. I am the exclusive owner of the copyright in materials I create. As noted by the Office of the Provost, usage of notes or in-class recordings without the faculty member's permission is a violation of the faculty member's copyright protection.

You may take notes and make copies of course materials for your own use in this class. You may also share those materials with another student who is registered and enrolled in this course.

You may NOT reproduce, distribute, upload, or display any lecture notes or recordings or course materials in any other way – whether or not a fee is charged – without my express written consent. If you do so, you may be subject to disciplinary action under the UMass Code of Student Conduct or other applicable rules and laws.

While you are welcome to use the material for your own personal and educational use, you may not redistribute them to others outside the class. In particular, selling or otherwise redistributing your notes (or mine!), making or selling audio, video, or still recordings of course material, is not allowed without express written permission from me.

Recording

The lecture hall we are in has both cameras and microphones, and you may be recorded during your lecture attendance.

AI tools

The use of AI tools on in-person graded assessments (quizzes, midterms, and the final exam) is prohibited.

The use of AI tools on out-of-class assignments is frowned upon. Write your code yourself, lest you be unable to write it on an exam!

Acknowledgments

Some material taken from the Rust Code of Conduct.

Some material taken from the Cornell CS 3110 syllabus and related policies.