CICS 210 Data Structures

Syllabus

Version 4.0 Fri, 30 Aug 2024 @ 3:38 PM

Table of Contents

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.

Technical material will be presented in lecture and lab. We do an accelerated Java review in the first few weeks of the semester, it is expected that you have the familiarity with Java that the prerequisite provides. If you do not, you are responsible for self-studying to the expected level.

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.

Time, Location, & Staff

This course has two instructors teaching two sections that assemble for 75-minute lectures on different days and times. It is important that you know which section you are enrolled in and attend the corresponding lecture. This course also has a 50 minute lab that meets once a week on Mondays. Make sure you know which lab you are enrolled in and attend that lab.

Lectures

Lecture 01 (Mo/Wd) 5:15-6:45 PM Hasbrouck Lab Add Room 20

- Instructor: Mordecai Golin
- Email: mgolin@umass.edu

Lecture 02 (Tu/Th) 4:00-5:15 PM Goessmann Laboratory Room 20

- Instructor: Tim Richards
- Email: richards@cs.umass.edu

Labs

All labs meet on **Mondays** unless otherwise specificied in the detailed schedule.

Lab 01LL (28469) Mo 9:05AM - 9:55AM Lederle Grad Res Tower Rm 145

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 01LM (28470) Mo 10:10AM - 11:00AM Lederle Grad Res Tower Rm 145

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 01LN (28471) Mo 11:15AM - 12:05PM Lederle Grad Res Tower Rm 147

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 02LM (36344) Mo 11:15AM - 12:05PM Engineering Laboratory rm 325

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 01LQ (28472) Mo 12:20PM - 1:10PM Lederle Grad Res Tower Rm 143

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 01LR (37680) Mo 12:20PM - 1:10PM Computer Science Bldg rm 140

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 02LN (36345) Mo 12:20PM - 1:10PM Engineering Laboratory rm 325

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 02LL (36343) Mo 1:25PM - 2:15PM Engineering Laboratory rm 325

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 02LQ (36346) Mo 1:25PM - 2:15PM Lederle Grad Res Tower Rm 143

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Lab 02LR (37681) Mo 4:00PM - 4:50PM Computer Science Building Room 140

- Teaching Assistant: TBD
- Undergraduate Course Assistant: TBD

Office Hours

Office hours for all course staff **are posted on the Canvas home page**. The post is pinned to the top of the Piazza feed and is subject to change. This means it is important for you to frequently visit that post to decide which office hours you can attend. The course staff will announce on Piazza the start of office hours to remind you. We will also announce on Piazza if an office hour is cancelled or changed.

Attending office hours is a key resource for your success in this course. During office hours, you have the opportunity to receive personalized assistance tailored to your specific questions and challenges. This one-on-one help can clarify complex concepts and provide targeted feedback on your assignments. Engaging with the material in this setting allows for a deeper understanding of the course content, enabling you to explore topics more thoroughly and discuss ideas not fully covered during lectures. Regular attendance at office hours can positively impact your grades by addressing areas of confusion promptly, helping you keep up with the course material and perform better on assignments and exams.

Office hours also foster critical thinking and problem-solving skills through discussions that explore different approaches to problems and analyze case studies. Additionally, building a rapport with your instructor, teaching assistants, and undergraduate course assistants during office hours can be beneficial for your academic and professional growth. Attending office hours demonstrates your commitment to the course and your willingness to take an active role in your education. It shows that you are proactive in seeking help and dedicated to mastering the course material.

Moreover, office hours are the appropriate time to address individual concerns such as your progress, personal challenges affecting your studies, or any other issues, allowing you to discuss these matters confidentially with your instructor. To make the most of office hours, come prepared with specific questions or topics you need help with, be open-minded to exploring different perspectives and solutions, and follow up on any advice or resources provided. Remember, office hours are a valuable resource designed to support your learning. Take advantage of this opportunity to enhance your understanding, improve your performance, and succeed in this course.

NOTE: Office hours end after the last day of classes! Unless you hear otherwise, we won't be holding regularly-scheduled office hours after the last lecture of this course.

Required Material

Computer

A computer capable of running the most recent LTS version of the Java Development Kit (JDK21) 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.

Book

There is one required textbook for this course, available at zyBooks. If you prefer physical texts, this book is "Data Structures and Algorithms in Java," 6th Edition, by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser. Here are the instructions to gain access to the book:

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code: UMASSCICS210Fall2024
3. Subscribe

Optional Material

For students who want additional Java references, we 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, we suggest the free online textbook Open Data Structures. The Java version is written in a pretty terse style that we don't find super helpful (we 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 characteristics.

You are expected to read and abide by the UMass Guidelines for Classroom Civility and Respect.

Communication policy

Email

Per the University Email Policy, you are expected to check your University email regularly—at least once a day. We will use your UMass email address as the primary point of contact across all online tools we use, such as Piazza, Gradescope, and Canvas, and for official communications outside of class.

While your UMass email will remain a key point of contact, please note that Canvas Inbox/Messages will be our primary method for communicating with you regarding grade performance, missing submissions, and other course-related issues.

We will generally not use email for routine communication. Instead, you should reach out to instructors and course staff via Piazza (see more details below). If you need to send a private message, please post to Instructors. If you send an email, it may go unanswered, or you may receive a brief reply directing you to post your question on Piazza.

DO NOT EMAIL THE COURSE STAFF. USE PIAZZA!

Piazza

Piazza is an online discussion forum designed to facilitate communication and collaboration among students and instructors in an academic setting. It provides a platform where students can post questions about course material, assignments, and projects, and receive answers from their peers, teaching assistants, and instructors. Piazza is structured to support a dynamic and interactive learning environment, allowing for real-time collaboration and the sharing of knowledge.

We (course staff) can use Piazza to post announcements, share resources, and create discussion threads on various topics relevant to the course. The forum is organized in a way that makes it easy to navigate through different discussions, with questions and answers often grouped together to maintain clarity and coherence. Students can post their queries anonymously if they prefer, encouraging participation from those who might be hesitant to speak up in a traditional classroom setting.

One of the key features of Piazza is its ability to facilitate detailed and structured responses. Students can contribute by editing questions and answers, similar to a wiki, which helps in refining the information and making it more comprehensive. Additionally, Piazza supports multimedia posts, allowing users to include images, code snippets, and LaTeX for mathematical notation, making it a versatile tool for a wide range of subjects.

Overall, Piazza enhances the learning experience by providing a collaborative space where students can engage with the course material outside of regular class hours. It fosters a sense of community, enables peer learning, and helps students receive timely help, thereby supporting their academic success.

For course-content related questions, especially questions that other students might benefit from seeing the answers to, **please use Piazza**. For other questions and unusual logistics stuff, like extensions requests, exam rescheduling, and so on, use the "course contact forms," linked to on Piazza. If you do need to send the Instructors email because of something particularly confidential, please include "CICS 210" in the subject line to make sure we 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 us to manage by email.

Course staff typically respond to Piazza questions and other requests within two business days, but we 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* your 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.

Piazza Guidelines

Please follow these guidelines in your use of Piazza:

- You should use Piazza to ask questions and get advice on assignments. But you should not use Piazza to step through each and every tiny problem you encounter in an assignment.
- You may not post assignment solutions to Piazza, 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 and provide a link to your submission to help us get to your code as easily and quickly as possible.
- 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 Piazza 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 Piazza, 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 Piazza will be archived and closed after our final exam.

Time Management

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 assignments and readings (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 expected and required 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 us 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 us as soon as possible.

- If you are absent or miss deadlines for health reasons, we expect you to notify us as soon as possible – generally before, not after, the missed course element, and to provide documentation from your health care provider.

- 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., we expect you to notify us as soon as possible and provide written documentation (again, if you seek excusal or extensions).

Things such as other deadlines or exams, vacations, weddings, visiting family, unexpected traffic, mild cold-like illness (e.g., sore throat, headaches, cough, runny noses, etc.), supporting a friend or family member through their tough times, and other very reasonable challenges that we all face, are not generally reasons for class absence excusal, missing an exam, or missing a homework deadline. Note that if you are in isolation or quarantine due to COVID, the University Health Services will provide you documentation on request.

If you miss a quiz or exam – *especially without prior notice*, we will require an explanation and clear written documentation to judge whether the absence is excusable. We do not grant these very often.

If you must miss a quiz or exam for an excusable reason, we will work with you to find an acceptable time for you to take a makeup. Quizzes must be made up within three business days (by the end of the following Friday, typically), 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, we will excuse you from missed work, but you are responsible for both notifying us 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 70% of the course up through the time of the incomplete request with a passing grade. Otherwise, withdrawal is the recommended course of action.

Schedule

Weekly Lecture Topics

The approximate weekly 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. Iteration/traversal
5. Big-O analysis, Amortization
6. Generics, Recursion Review
7. Maps, Hash Tables
8. Trees (binary search trees)
9. Balanced Trees (AVL and B-trees)
10. Priority Queues (heaps and treaps) / Heapsort
11. Sorting and Selection
12. Graphs (representation; BFS and DFS search algorithms)
13. Advanced Structures: Union-Find, Bloom filters, Tries
14. FINAL exam

Weekly Lab Activity and Quizzes

The tentative scheduled lab meetings will be as follows, with either quizzes or lab activities scheduled for each meeting:

Day	Date	Event	Topic
Mo	09/09	Activity	
Mo	09/16	Quiz 1	
Mo	09/23	Activity	

Day	Date	Event	Topic
Mo	09/30	Quiz 2	
Mo	10/07	Activity	
Mo	10/14	Holiday – Indigenous Peoples Day	
Tu	10/15	(Monday Schedule) No Lab	
Mo	10/21	Quiz 3	
Mo	10/28	Activity	
Mo	11/04	Quiz 4	
Mo	11/11	Holiday - Veterans' Day	
Mo	11/18	No Lab	
Mo	11/25	Activity	
Mo	12/02	Quiz 5	
Mo	12/09	Activity	

Exams

Exams play a crucial role in computer science courses, serving as a valuable tool to assess and reinforce your understanding of key concepts. These exams are carefully designed to gauge your grasp of the material, ensuring that you have a solid foundation in essential topics such as arrays, linked lists, stacks, queues, trees, and graphs. They help both you and your instructors identify areas where you are excelling and areas where you might need additional practice or support.

One of the primary reasons exams are given is to provide you with an opportunity to demonstrate your knowledge and skills in a structured environment. They encourage you to review and consolidate your learning, which can deepen your comprehension and retention of the material. By preparing for exams, you engage in a process of active learning that reinforces your ability to solve problems, think critically, and apply theoretical concepts to practical scenarios.

Exams also serve as a benchmark for your progress throughout the course. They offer a clear indication of how well you are meeting the learning objectives and help you track your development over time. This feedback is invaluable, as it allows you to adjust your study habits and seek additional resources or assistance if needed.

Moreover, exams are an integral part of the academic process, ensuring that the knowledge you gain is robust and comprehensive. They help maintain academic standards and provide a measure of accountability for both students and instructors. By evaluating your performance through exams, instructors can tailor their teaching strategies to better support your learning needs.

It is important to approach exams with a calm and positive mindset. Remember that they are just one aspect of your learning journey and are designed to support your growth and success. By staying organized, managing your time effectively, and seeking help when needed, you can approach exams with confidence and use them as a valuable opportunity to showcase your hard work and dedication to mastering the subject.

In short, exams in a computer science data structures course are given to assess your understanding, reinforce your learning, and provide valuable feedback on your progress. They are a key component of the educational process, helping you build a strong foundation in data structures that will serve you well in your academic and professional endeavors.

The exams and associated dates and times in this course are as follows:

Mid-Term Exams

- Midterm 1: DATE AND TIME TO BE DECIDED
- Midterm 2: DATE AND TIME TO BE DECIDED

Final Exam

- Lecture 01: DATE AND TIME TO BE DECIDED
- Lecture 02: DATE AND TIME TO BE DECIDED

You are responsible for clearing your schedule at the beginning of the semester to take exams. If you cannot commit to taking exams as scheduled, **you should drop this class immediately**. Makeup exams will be offered **only** in those special cases where required by university policy. You know the scheduled final exam days before the start of the semester. Do not schedule departures from campus until after the final exam schedule. There will be no makeups.

Grading

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

Weight	Category
5%	in-person lab exercises
20%	programming assignments
15%	quizzes
15%	midterm 1
15%	midterm 2
30%	final exam

The numerical cutoff for final course letter grade assignment will be finalized 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, we will point you at this line in the syllabus.

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.

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

You are responsible for monitoring your grades, typically visible on Gradescope and Canvas. If you encounter any issues with your grades, you will have **three days** past the first posting of a particular assignment’s grade 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 the three days, so you must be prompt.

Assignments

We will post assignments – programming and the like – about once every two weeks, and you will typically have about two weeks to complete them. The first programming assignment is an exception as it is only to get you started, so 1 week is sufficient. Assignments will be available on Canvas, announced on Piazza, and must be submitted through Gradescope. Assignments may be completed individually or with a partner. See the course honesty policy, below, for more details.

Each assignment is worth the same amount of credit toward the final grade – if we have 7 assignments, each will be worth (20% / 7 = about ~2.85) percentage points toward your course grade, regardless of how many “points” each assignment is listed as having in Gradescope.

You are responsible for submitting your work to Gradescope. 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 auto graded assignments, you are responsible for submitting code that compiles and runs on the auto grader; if you submit code that does not compile or that gets stuck in an infinite loop on the auto grader, you will receive no credit for the auto graded portion of that assignment. In other words, you should expect to receive the grade that Gradescope produces for auto graded 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.

Attempts to manipulate, game, or otherwise incorrectly use the auto grader 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, must be completed in a group in lab, and must be submitted by the end of your lab session – they will not be accepted late.

In other sessions (about every other, see the “Schedule” section) you will take a short quiz. Each will consist of a few programming and short-answer questions. 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.

IMPORTANT: Quizzes assume you have submitted prior programming assignments by the assigned due date, not the late due date. If you are consistently late on submitting programming assignments, it can also potentially hurt your success on quizzes.

Midterms and Final Exam

There are two midterm exams and one 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, listed in the “Schedule” section. 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 the instructors for details.

The final exam will be given at the University-scheduled time and place, as listed in the “Schedule” section.

For conflicting Midterm/Final Exams you must submit an online form to the registrar. They will inform us (the instructors) of which class must provide a makeup. **The forms should be submitted two weeks prior to the exam date**. See

https://www.umass.edu/registrar/i-have-two-midtermfinal-exams-scheduled-same-time-what-do-i-do

> For conflicting exams, please submit an exam conflict form through your SPIRE account for Registrar's Office review and approval. The ***Evening Exam Conflict*** and ***Final Exam Conflict*** forms are located in the Class & Finals Schedule section of the Manage Classes tile in your SPIRE account. The Registrar's Office will review your exam schedule and notify you and your instructor when a make-up exam is warranted based on the policy as outlined in the academic regulations.

The relevant policies are in the Academic Rules and Regulations: Section XI:

For midterm exams:

> In addition, it is University policy that if a student is scheduled to take two evening examinations at the same time, the faculty member teaching the course with the higher final digit (or digits) in its class number (the unique 5-digit number which represents a particular section in **the Schedule of Classes**) is required to offer a make-up examination.

For Final Exams

> …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.

Late Policy and Assignment Extensions

Submission Deadlines

Assignments must be submitted by the assigned due date as specified by the course LMS assignment and/or the submission platform (e.g., Gradescope). This due date is the expected date by which students should complete and submit their work.

Automatic Extension: Late Days

Each assignment is allotted a specific number of "late days," which serve as an automatic extension beyond the initial due date. Typically, this allotment is five (5) late days. During this period, students may submit their assignments **without penalty**. Students with accommodations requiring additional time on assignments automatically fall under this policy. No further action is required by the student to utilize these late days.

End of Late Day Period

Upon the expiration of the allotted late days, submissions will no longer be accepted. The system will automatically close, and any attempt to submit the assignment thereafter will be denied. Students are encouraged to manage their time effectively to avoid missing the final submission window.

Late Penalty Policy

The late submission policy is designed to be fair and flexible, allowing students to accumulate up to 10 late days across all assignments without penalty. Any late days beyond this threshold will result in a deduction of 0.5% per day from the final grade, with a maximum penalty of 10%. However, students with accommodations are exempt from this penalty. This approach encourages timely submission while accommodating occasional delays. The following is exactly how we compute the late penalty.

Policy Outline:

- **Due Date**: The initial deadline for all programming assignments.
- **Late Due Date**: A grace period where students can submit their assignments late without an immediate penalty.
- **Threshold**: A student can accumulate up to `10` late days without any penalty. Beyond this threshold, penalties will be applied.
- **Penalty per Day**: For each day an assignment is submitted late beyond the threshold, a small percentage (e.g., 0.5% per day) is deducted from the final grade.
- **Penalty Cap**: The maximum penalty that can be applied to the final grade is capped at a certain percentage (e.g., 10%).
- **Accommodation Exemption**: Any student with accommodations is exempt from this penalty.

Formula:

Let:

- `di` = Number of days late for the `i`-th assignment (beyond the late due date)
- `T` = Total number of late days allowed without penalty (threshold) = 10
- `D` = Total number of late days across all assignments
- `pd` = Penalty per day (e.g., 0.5%)
- `P` = Total penalty applied to the final grade

The formula for the total penalty is:

$P=min(Cap,pd\times max(D-T,0))$

Where `Cap` is the maximum penalty that can be applied (e.g., 10%).

Implementation in Java

Here is the Java implementation of this policy:

```
public class LatePenaltyCalculator {

    public static double calculateLatePenalty(int[] lateDays, int threshold,
                                              double penaltyPerDay, double cap,
                                              boolean hasAccommodation) {
        // Exempt students with accommodations from penalties
        if (hasAccommodation) {
            return 0.0;
        }

        // Calculate total late days
        int totalLateDays = 0;
        for (int days : lateDays) {
            totalLateDays += days;
        }

        // Calculate the penalty for days beyond the threshold
        double totalPenalty = 0.0;
        if (totalLateDays > threshold) {
            int excessDays = totalLateDays - threshold;
            totalPenalty = excessDays * penaltyPerDay;
        }

        // Apply the cap
        return Math.min(totalPenalty, cap);
    }

    public static void main(String[] args) {
        int[] lateDays = {2, 3, 1, 4, 2}; // Example late days for assignments
        int threshold = 10;
        double penaltyPerDay = 0.005; // 0.5%
        double cap = 0.10; // 10%
        boolean hasAccommodation = false; // Example: no accommodation

        double penalty = calculateLatePenalty(lateDays, threshold, penaltyPerDay, cap,
                         hasAccommodation);
        System.out.printf("The penalty to be applied to the final grade is: %.2f%%%n",
                          penalty * 100);
    }
}
```

Explanation of Implementation

- **Threshold**: Set to 10 days by default, allowing up to 10 total late days across all assignments without penalty.
- **Penalty per Day**: Set to 0.5% (0.005 as a fraction of the final grade).
- **Cap**: Set to 10% of the final grade.
- **Accommodation Exemption**: Students with accommodations are exempt from any late penalties.

The `calculateLatePenalty` method computes the penalty by summing the total late days, checking if they exceed the threshold, and applying the appropriate penalty. If the student has accommodations, the method immediately returns a penalty of 0. The `main` method provides an example of how this method can be used.

Compiling and Running the Late Day Calculator Java Program

If you exceed the 10-day threshold and would like to determine the penalty that will be applied to your final grade, do the following:

15. Save the code in a file named `LatePenaltyCalculator.java`.

16. Open a terminal or command prompt.

17. Navigate to the directory where the file is saved.

18. Compile the program with the following command:

    ```
    javac LatePenaltyCalculator.java
    ```

19. Run the compiled program with the following command:

    ```
    java LatePenaltyCalculator
    ```

This will execute the program and display the penalty to be applied to the final grade based on the input late days and policy parameters which you will need to configure.

Extenuating Circumstances

If a student is unable to submit an assignment by the end of the late day period due to extenuating circumstances, the student must submit documented evidence supporting their claim. The extension request must be submitted through our extenuating circumstances form available as a survey in the quizzes section of Canvas. A link to the correct form will also be available as a link from the assignment, quiz, or exam in Canvas. The form must be submitted before the expiration of the late day period unless the nature of the extenuating circumstances makes this impossible.

Examples of extenuating circumstances include, but are not limited to:

Medical Emergencies: A note from a licensed healthcare provider detailing the medical condition and the period during which the student was incapacitated.

Family Emergencies: Documentation such as a death certificate or a note from a licensed counselor or clergy member confirming the family crisis.

Legal Obligations: A court summons or subpoena indicating that the student was required to attend a legal proceeding that conflicted with the assignment deadline.

Natural Disasters: Proof of residence in an area affected by a declared natural disaster, along with documentation of the event from a relevant authority (e.g., FEMA notice, local government notification).

Multiple Extension Requests

Students are advised that multiple requests for extensions beyond the automatic late day extensions may result in a recommendation to withdraw from the course. In cases where withdrawal is not feasible, the instructor reserves the right to impose a penalty on the student's final grade, commensurate with the frequency and severity of the requests. This policy ensures fairness to all students and maintains the integrity of the course schedule.

Summary

Students are expected to adhere to the submission deadlines and utilize the provided late days responsibly. The outlined procedures for extensions due to extenuating circumstances are designed to accommodate genuine needs while maintaining academic rigor and fairness across the course.

Missing Labs and Quizzes (Extensions/Exemptions)

Labs

The lab sessions must be done in person (usually as group work) at the Monday lab and cannot be extended. If you need to miss a lab session due to extenuating circumstances, e.g., sickness, you must submit a request form for that specific lab. As with late assignments, these forms are available as surveys in the quizzes section and a link from the assignment in Canvas. The form should be submitted before your missed lab although documentation can be provided afterwards. We will not require documentation for medical issues (we understand that you can wake up feeling sick the day of a lab and not have time or need to go to a doctor before the lab) but, if you make multiple such requests during the semester, we will require that documentation.

If your request is approved, then you will be exempted from that lab and it will not count towards your grade.

Quizzes

If you will miss a Monday quiz due to extenuating circumstances, you must submit a request form for **delaying** that quiz. Again, this form is available as a survey in the quizzes section and a link from the assignment in Canvas. Also again, we will not require documentation for medical issues but, if you make multiple such requests during the semester, we will require that documentation.

If your request is approved you will still need to make up that quiz by Friday of the same week by sitting it at one of the TA's office hours. The names, office hours and contact information of the TAs are available on Canvas. It will be your responsibility to contact the appropriate TA whose schedule fits yours and request to take the quiz at their office hours. You must contact them **by email** at least two days before the day you want to take the quiz.

If, due to extenuating circumstances, you are unable to take the quiz and to arrange a makeup, you should request an exemption using the form. This will require documentation.

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.

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, midterms 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 *both* collaboration is explicitly permitted (as it is on in-person lab exercises and 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 (who is not your partner) 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: We're not asking you to Google solutions, we're asking you to write them yourself, and to ask for help if you need it.

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 instructors (Marc Liberatore, Mordecai Golin, Tim Richards); exceptions are clearly noted. These works are protected by U.S. copyright laws and by university policy. The creator of the work is the exclusive owner of the copyright in materials they 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 our 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 ours!), making or selling audio, video, or still recordings of course material, is not allowed without express written permission from the instructors.

Recording

The lecture halls we are in have 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 to assist with the completion of out-of-class assignments is frowned upon. Write your code yourself, lest you be unable to write it on an exam!

Other use of AI tools (for example, to generate review questions and answers, or to more thoroughly understand a concept) is permitted in the same way you might, say, search the web for help or go to review sessions or talk with a friend.

Acknowledgments

Some material taken from the Rust Code of Conduct.

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

This version (Fall 2024) is a revision of a previous version written by Marc Liberatore.