CICS 210

Data Structures

General Information

Semester:	Spring 2023 (this is an example semester with course staff for Fall 2022)
Start Date:	September 6, 2022
End Date:	December 12, 2022
Credits:	4
Lectures:	**01-LEC (44871)**: TueThu 04:00PM-05:15PM, GsmnAdd 64, Gordon Anderson **02-LEC02 (44945)**: TueThu 04:00PM-05:15PM, MOR1N375, Cheryl Swanier
Labs:	**LAB01LL (44872)**: Mon 10:10 AM-11:00 AM, HasbAd104B, Staff **LAB01LM (44898)**: Mon 10:10 AM-11:00 AM, HasbAd104A, Staff **LAB01LN (44868)**: Mon 11:15 AM-12:05 PM, HasbAd104B, Staff **LAB01LQ (44869)**: Mon 11:15 AM-12:05 PM, HasbAd104A, Staff **LAB01LR (44908)**: Mon 12:20 PM-01:10 PM, HasbAd104B, Staff **LAB01LS (44909)**: Mon 12:20 PM-01:10 PM, HasbAd104A, Staff **LAB01LU (44924)**: Mon 01:25 PM-02:15 PM, HasbAd104A, Staff **LAB01LV (44925)**: Mon 01:25 PM-02:15 PM, HasbAd104B, Staff **LAB02LL (44961)**: Wed 10:10 AM-11:00 AM, HasbAd104B, Staff **LAB02LM (44962)**: Wed 10:10 AM-11:00 AM, HasbAd104A, Staff **LAB02LN (44963)**: Wed 11:15 AM-12:05 PM, HasbAd104B, Staff **LAB02LQ (44964)**: Wed 11:15 AM-12:05 PM, HasbAd104A, Staff **LAB02LR (44965)**: Wed 12:20 PM-01:10 PM, HasbAd104B, Staff **LAB02LS (44966)**: Wed 12:20 PM-01:10 PM, HasbAd104A, Staff **LAB02LU (44967)**: Wed 01:25 PM-02:15 PM, HasbAd104A, Staff **LAB02LV (44968)**: Wed 01:25 PM-02:15 PM, HasbAd104B, Staff
Prerequisites:	CICS 160 (or INFO 190T) with a grade of C or better.
Instructors:	Gordon Anderson, Cheryl Swanier
Staff	TBD

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.

Statement of Inclusivity

We honor UMass's commitment to embrace diverse people, ideas, and perspectives to create a vibrant learning and working environment. In this course, each voice in the classroom has something of value to contribute and each voice in the classroom is valued. You are welcome regardless of age, background, citizenship, disability, education, ethnicity, family status, gender identity, geographical origin, language, military experience, political views, race, religion, sexual orientation, socioeconomic status, and work experience.

We respect everyone's right to be addressed by the name and pronouns that they choose. You can indicate your preferred/chosen first name and pronouns on SPIRE, which appear on class rosters. A student's chosen name and pronouns are to be respected at all times in the classroom.

Please read the UMass Guidelines for Classroom Civility and Respect and take care to respect the different experiences, beliefs, and values expressed by students and staff involved in this course.

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.

Course Material

Textbook

The following textbook is required:

- *Programming with Data Structures*, an interactive textbook from zyBooks designed specifically for this course. This book includes state-of-the-art learning material, proven effective, and designed to maximize learning while respecting student time. The online textbook has embedded exercises and assignments with due dates.

To activate your zyBook subscription:

- Go to: https://learn.zybooks.com
- Create an Account. Make sure to sign up with your **@umass.edu** email address.
- When prompted for the zyBook code, enter: XXXXXXXXXXXXXXXXXXXX
- You can purchase the Zybook online with a credit card. A subscription is $$$. You may begin subscribing on DDDDDDDD, and the cutoff to subscribe is DDDDDDDD. Subscriptions will last until DDDDDDDD.

Laptop Computer

It is highly recommended that you have a laptop computer. We will be writing code both in and out of class, so a portable computer capable of installing software (not a Chromebook) is valuable for this class. Most in-class and lab programming activities will be group-based, so if you do not own a laptop, you can easily work with another student in class.

Software Platforms and Tools

Moodle

We will use the Moodle Learning Management System (LMS) as the primary hub for course content. You will be able to access readings, lecture material, assignments, and any other important material pertaining to this course. We may use Moodle to submit some assignments. You will be able to access your latest grades and comments for assignments using the Moodle Gradebook.

zyBooks

The textbook for this course is available from zyBooks. Not only is zyBooks the book for the course, but it also includes visual and interactive content that increases your understanding of the material. It provides participatory content and challenge exercises giving you an improved learning experience and helping you solidify your understanding of the material.

Gradescope

We use Gradescope for grading your programming projects and lab assignments. Gradescope allows us to provide fast and accurate feedback on your work. You will be added to Gradescope at the start of the semester and will receive an invitation to join, providing you with your account details. If you do not receive it, please contact the professors immediately.

Projects are graded automatically using Gradescope autograder. Before the deadline you can submit as many times as you need, so submit early and often to ensure you have something in before the deadline. Become familiar with Gradescope and verify that your submission has been properly uploaded before the deadline. Use OneDrive, DropBox, Google Drive, or some other backup software to ensure that your work is not lost in the event of a computer failure

The Gradescope autograder will provide you with some limited feedback on your submissions: does it compile, does it pass the public and private tests, what your score is, etc. The autograder does not provide detailed feedback. If your code does not pass all tests it means you have more work to do in your VSCode environment, which provides the tools for code debugging and development. Purely relying on Gradescope feedback is not a time- efficient way to develop your project code. We want you to develop the proficiency to test and debug your programs with the right tool, in this case, VSCode.

VSCode

We use the VSCode Development Environment for developing, debugging, and testing programming projects. This is free software and you will be given installation instructions and training in its use. There are many excellent integrated development environments (IDE), editors, and tools that exist, however, we recommend VSCode as it is easy to learn, and we will be using it in class for code demonstrations. You can read more about VSCode here and we encourage you to try out the “Getting Started with Java in VSCode” as preparation for the class.

Java

We will be using the Java programming language in this course. It is an excellent language to learn to explore data structures and is pervasive in industry. It is a statically and strongly typed language with an emphasis on safety, design, and reliability. It provides language constructs that will enable you to learn programming techniques and data structure design more deeply and provide you with a tool that you can use in future projects. We will be using the latest version of the Java Development Kit (JDK) which is version 19 as of this writing.

Communication

Email

Email should not be used to communicate to instructors or course staff. **Please post privately to Instructors on Piazza**.

In the unlikely event that you are unable to post to Piazza, please send an email to the instructor teaching your course section.

We will occasionally use your UMass email account to communicate important information, such as exam access times and passwords. Every student enrolled in a UMass course has a UMass email account. Therefore, you are responsible for reading all messages we send via your UMass email account. We do not want anyone to miss an exam because they did not read their UMass email!

You might want to look into setting up message forwarding to your personal email so that you do not miss any messages.

Piazza

We will be using Piazza for all other communication. This online discussion forum should be your first choice for asking questions. You should check the discussion forum before asking your question to see if the same question has already been posted. We will not answer questions that have already been answered in the discussion forum. Think before you post. We expect you to do a reasonable amount of thinking to try to solve your problems before posting for help. Make sure you are articulate and clear with your post (i.e., think before you post). You should post questions related to assignments early rather than wait until the last minute. Questions that are posted very near an assignment deadline may not be answered. Course staff are expected to answer questions Monday through Friday. Do not expect prompt answers on Saturday, Sunday, and scheduled holidays and breaks.

Please post with respect and kindness. Posts that are disrespectful, crude, inappropriate, or mean will not be tolerated and will be reported and result in your immediate removal from the course and a failure for the course.

Attendance

1. **Lectures:** Attendance at lectures is optional. Note that class time is used for active learning and live coding. You can download starter code and develop on your own or along with the professors during the class time. This coding work is completely optional and no credit is assigned.
2. **Labs:** Attendance at labs is required. We will excuse one lab attendance, though you still have to complete and submit the lab for a grade. In the case of illness, religious or funerary events, university-related events (athletic event, field trip, or performance), or extenuating non-academic reasons (military obligation, family illness, jury duty, automobile collision), you may be excused if you fill in the Attendance Excusal form on Moodle. You still have to complete and submit the lab for a grade.

Your Success is Important

Your success in this class is important to us. We all learn differently and bring different strengths and needs to the class. If there are aspects of the course that prevent you from learning or make you feel excluded, please let us know as soon as possible. Together we'll develop strategies to meet both your needs and the requirements of the course. There are also a range of resources on campus, including:

- Academic Calendar
- Learning Resource Center
- Center for Counseling and Psychological Health (CCPH)
- English as a Second Language (ESL) Program

Assessment and Grading

You can track how you are learning by monitoring your grades and activity completion tracking in Moodle. Grades will be available through Moodle and you should check them regularly and review any provided feedback. If you encounter any issues with your grades, please let us know 1 week past the return of your grades, so that we can investigate. Make sure you address grading issues promptly within a week.

Each assessment component is worth a fixed number of points. At any point during the course you can easily calculate your current grade by the number of points you have achieved with respect to the total number of points you can attain in the course. Note that missing and late assignments can have a dramatic impact on your final grade and it is important that you submit all work in a timely manner. The grading is categorized and weighted as follows:

- **10% Textbook Assignments**
- **10% Lab Assignments**
- **10% Quizzes**
- **30% Exams**
- **40% Programming Projects**

You should schedule approximately 6-8 hours per week outside class to complete project and zyBooks chapters. Although there is no opportunity for extra credit in this course, don't hesitate to take advantage of the resources listed in the Plan for Success section below.

The final numerical cutoff for final course letter grade assignment will be made after all grading is completed. Here is the approximate grade thresholds that usually apply: A (93-100), A- (90-92), B+ (87-89), B (83-86), B- (80-82), C+ (77-79), C (73-76), C- (70-72), D+ (67-69), D (60-66), F (0-59).

Please note that we will not make adjustments to any individual student's grade once the final grades have been calculated. Any rounding is done before the final grades are calculated. While there is no extra credit or any other additional work available outside of the assigned material, don't hesitate to ask us for help in completing assignments.

Textbook Assignments

This course uses zyBooks' "Programming with Data Structures", state-of-the-art learning material, proven effective, and designed to maximize learning while respecting student time. The online textbook has embedded exercises and assignments with due dates.

Lab Assignments

Labs allow for the opportunity to explore topics in more detail and develop code that reinforces the topics covered in the text and in lecture. You will complete an activity in groups and attendance is expected. You are to attend the lab session you are registered for. Starter code is provided for lab work. A TA and UCAs will be present to answer any questions you may have about the lab.

Quizzes

"Topic Quizzes" will be available during the semester. These quizzes are designed to provide you with an opportunity to test your knowledge of a subset of the course topics and to give you some test-taking practice. The quiz questions are designed to be similar to exam questions you might see on the midterm and final exams. These quizzes are given in Moodle.

The quizzes consist of various types of questions and are automatically graded. We will provide answers to the coding questions. You will receive credit for taking a quiz. You get 3 attempts to complete the quiz and your highest score is counted. Each quiz tests a subset of the material (see Moodle for schedule). We recommend that you take a quiz shortly after you finish the block of material that it covers. You may continue to review the quiz after they finish.

Exams

There is a midterm and final exam for this course. These exams will take place in-person.

Programming Projects

This course assigns a number of programming projects during the semester. These projects provide you with an opportunity to apply the skills and concepts you learn in the course to more involved coding and testing scenarios. The projects are completed individually and you may attend office hours to seek help. Writing and debugging software is much more time consuming than most people (including professionals) expect. Therefore, this course strongly recommends that you begin work on projects as soon as they are assigned. That way, you have time to get help if you encounter difficulties. After all, the course staff are here to support student learning, and projects are key to learning this material.

Late Submissions

It is your responsibility for maintaining your own schedule and being prompt with your submissions. We expect that you become familiar with the course submission software and verify that your submission has been properly uploaded. Instructors will not grant late submissions due to lack of checking on this. The course staff assumes:

- You are an adult and have the ability to check and verify your work has been received properly.

- You are capable of using OneDrive, DropBox, Google Drive, or some other backup software to ensure that your work is not lost in the event of a computer failure.
- You have a back-up plan in place in the event that your computer fails or your internet connection is unavailable. Make sure you have a plan B and C if your computer crashes or your internet is unavailable. This is your responsibility.

To ensure that you submit assignments on time you should begin them early and not wait until the last minute to submit. For some assignments you may be able to submit multiple times so submit early and often to ensure you have something in before the deadline.

Exemptions

There may be situations when you cannot take an exam, or submit an assignment by the due date. You may ask for an extension by filling in the Request for Extension form in Moodle in the case of illness, religious or funerary events, university-related events (athletic event, field trip, or performance), and extenuating non-academic reasons (military obligation, family illness, jury duty, automobile collision). For religious reasons, you must provide us with a written list of such dates within one week of your enrollment in the course. In all other instances, please provide us with written documentation as soon as possible.

If you add the class late, you are responsible for notifying us when you join the course and arranging for any deadline extensions with the instructors.

We understand that sometimes you may be unable to complete course requirements within the allotted time because of severe medical or personal problems. In these cases, you may request a grade of Incomplete from the instructor of the course. Otherwise, the recommended course of action is to withdraw and retake the course in the future. Please see the Academic Regulations Section IV Grading System and Credit Guidelines for further details.

Incompletes

Typically, a course is completed after the last class, final exam, and/or final project or assignment. In rare cases, extenuating circumstances may prevent you from completing a course by that time. As part of the University Regulations, we may issue an Incomplete (INC) for a course, rather than a course grade, if you submit a request to the instructor(s). The criteria for granting an INC request are determined by the course instructors. The following is an excerpt from Section VI D in the Academic Regulations:

> *"Students who are unable to complete course requirements within the allotted time because of severe medical or personal problems may request a grade of Incomplete from the instructor of the course. Normally, incomplete grades are warranted only if a student is passing the course at the time of the request and if the course requirements can be completed by the end of the following semester. Instructors who turn in a grade of "INC" are required to leave a written record of the following information with the departmental office of the academic department under which the course is offered: (1) the percentage of work completed, (2) the grade earned by the student on the completed work, (3) a description of the work that remains to be completed, (4) a description of the method by which the student is to complete the unfinished work, and (5) the date by*

which the work is to be completed. In the case of an independent study where the entire grade is determined by one paper or project, the instructor should leave with the department information pertaining to the paper or project, which will complete the course. To avoid subsequent misunderstanding, it is recommended that the student also be provided with a copy of this information."

Granting of an Incomplete is entirely up to the discretion of the instructor. Incomplete grades are warranted only if you are passing the course at the time of the request and have completed a significant percentage of the assigned work. If the Incomplete is granted, you and the instructor agree upon a plan for completing the remaining assignments for the course with specific deadlines. It is your responsibility to complete the remaining material according to the plan. The remaining work is done individually, as it is not possible to offer one-on-one teaching of the material past the course completion date.

Criteria

The incomplete criterion for this course requires that you have:

- At least 60% of the course must be completed with a passing grade.
- A valid reason for requesting an INC that relates to a severe medical or personal problem.

Requests

Towards the end of the semester a notification will be posted about incomplete requests. You will follow the instructions provided to submit an incomplete request. After we review the request, we will make one of the following determinations:

1. We **approve** the request upon which you will be notified by email and a separate incomplete agreement document will be sent to you to read through and sign no more than 48 hours after receiving the incomplete agreement document. This document will include what remains to be completed for the course and a deadline. After you sign and return this document, we will open extensions for the missing work. After the course has ended, we do not provide any additional help or support regarding the specifics of the course material. You are expected to complete the work using the material and online platforms that were available to you when the course was active.
2. We **deny** the request and submit a grade based on your performance at the end of the course.

Course Support

Office Hours

Office hours are times when we provide real-time access to the instructor, TAs, and UCAs. You do not need an appointment to attend office hours, attendance is optional, and all questions you have about the course are welcome. These sessions will be held at different times during the week. Office hours will be posted on the course website. Office hours will be held both in person and on Zoom.

Learning Resource Center

Supplemental resources available to you via the Learning Resource Center in the UMass DuBois Library include: Supplemental Instruction, ExSEL Group Tutoring, and 1:1 Tutoring. Visit the Learning Resource Center website for more details.

Your professors and TAs will also contact Student Success & academic advisors regarding your progress in the course. If you are contacted, please consider scheduling appointments such as tutoring or academic advising. Please email academicalert@umass.edu if you have any questions or need assistance connecting with resources.

Accommodations

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 your instructor as soon as possible so that we may make appropriate arrangements. For further information, please visit Disability Services (https://www.umass.edu/disability).

Title IX

If you have been the victim of sexual violence, gender discrimination, or sexual harassment, the university can provide you with a variety of support resources and accommodations. UMass is committed to providing these resources with minimal impact and costs to survivors on a case-by-case basis. Resources are available to survivors with or without them filing a complaint. No upfront costs are charged to any currently enrolled students for University Health Services or the Center for Counseling and Psychological Health, and no fees exist for services in the Dean of Students Office, the Center for Women and Community, Student Legal Services, or by live-in residential staff.

Plan For Success

Your success in this class is important to us. We all learn differently and bring different strengths and needs to the class. If there are aspects of the course that prevent you from learning or make you feel excluded, please let us know as soon as possible. Together we'll develop strategies to meet both your needs and the requirements of the course.

Successful Learning Tips

1. Check your UMass email every day to receive announcements and reminders in a timely manner.
2. Check our Moodle course site frequently.
3. Plan ahead! Create a study calendar for the whole semester.
4. Space your learning throughout each week.
5. Start working on projects as soon as they are assigned. Use office hours to ask questions when they occur.
6. Engage with the materials in the presented order to allow the most beneficial learning outcome.

7. Track upcoming deadlines and due dates for assignments with reminders to yourself.
8. Set up an inspirational and quiet learning environment either in your home or another space, like a library, that is conducive to focused and uninterrupted learning.
9. If you encounter any technical issues, resolve them as quickly as possible using Tech support.
10. Attend office hours for help.
11. Speak up! Asking for help is part of learning. Refer back to the section in the syllabus on communication.

General Education Requirements

Statement

CICS 210 is a 4-credit General Education course that satisfies the R2 (Analytic Reasoning) general education requirements for graduation. The General Education Program at the University of Massachusetts Amherst offers students a unique opportunity to develop critical thinking, communication, and learning skills that will benefit them for a lifetime. For more information about the General Education Program, please visit the GenEd webpage.

General Education Learning Outcomes

The General Education Program has four common objectives that pervade all designations. CICS 210 satisfies the following General Education objectives:

- **Content:** Students will know the design, implementation, and methods of analysis of data structures. In particular, students will learn how to design and implement fundamental data structures using a modern programming language, apply programming and problem solving to real world problems, and analyze the efficiency of the data structures they study.
- **Critical Thinking:** Students will apply and demonstrate creative, analytical, quantitative, & critical thinking through inquiry, problem solving, & synthesis. Students will use critical thinking skills to solve problems requiring the use of efficient data structures. As part of the problem-solving process students will use logical reasoning to create algorithms and design and build data structures to develop programs that put their solutions into action. Furthermore, students will investigate aspects of performance and comparisons of equivalent algorithms to draw conclusions on efficiency. Lastly, students will explore and ask questions about real world problems and determine the best data structure to use in particular circumstances.
- **Communication:** Students will develop their writing skills through various assignments that require an articulation of their solution. They will also practice their oral communication by working in groups during labs (i.e., explaining a data structure, algorithm, or programming technique).
- **Connections:** Students will connect the material in this course to real world problems such as using specialized data structures to represent information about population size and how it relates to pollution or how graphs are implemented to represent geographic locations and relationships between them (distance) to search for optimal routes based on fuel efficiency. The goal of this course is to intentionally provide a connection between

the concepts that are covered and how it impacts the world. This course is not only about learning how to program, it is about how to design and construct data structures, analyze and evaluate their efficiency, and use that information to solve problems in a reasonable amount of time.

R2 Critical Thinking

The course uses zyBooks' "Programming with Data Structures", a state-of-the-art interactive textbook. The textbook includes embedded exercises and assignments including coding exercises. Labs allow for the opportunity to explore topics in more detail and develop code that reinforces the topics covered in the text and in lecture. Students complete an activity either individually or in groups. Starter code is provided. Each lab is led by a TA with a supporting UCA. Students are able to work through the problems in the lab assignment and ask questions during that time. The programming projects provide students with an opportunity to apply the skills and concepts learned in the course in more involved coding scenarios. Quizzes include a mix of analysis and coding questions and are delivered via Moodle.

Examinations include a variety of questions that test different kinds of thinking including: (1) a vocabulary section to test recall of definitions, (2) a section assessing familiarity with programs written in the class by having students explain how to modify programs to solve new tasks, (3) questions where students must determine what a particular program fragment will do, (4) questions where students must find errors in program fragments, (5) questions where students must apply the analysis techniques they have learned to determine the efficiency of program fragments, and (6) questions where they must write program fragments to solve newly introduced problems.

In terms of grading, the zyBooks platform includes automated completion tracking of embedded questions in the interactive text. Students are graded on their lab work and programming projects. Students submit lab and programming project material to Gradescope, an online service that runs "autograding" tests against student work to assess correctness of implementations. Quizzes and Exams include a mix of automatically graded questions and coding questions that are manually graded.

The course runs with one (or two) lecture sections of approximately 300-400 students per fall/spring semester. Each section has an instructor assigned. The typical supporting course staff include four full time TAs, two half-time TAs, and a total of 10 UCAs (varies depending on enrollment). The course staff participate in both leading labs and in manual grading of code and analysis questions.

R2 Learning Outcomes

This course will satisfy the R2 learning outcomes. In particular, it will advance student's formal reasoning skills beyond the basic competency level by having them solve programming challenges on a weekly basis using a programming language to design and implement data structures. This will increase a student's sophistication as a consumer of numerical information as they must have a fundamental understanding of how a computer represents information (numerical or otherwise) in a discrete environment and how to use that environment to represent and store information. Students will also gain familiarity with the analysis of

algorithms, a branch of mathematics that studies the resources (both time and space) used by computations. Computer literacy is established in this context and naturally the limits of formal methods and the abuse of numerical arguments will be covered as part of developing programs and solving problems using data structures in general.

Schedule

The following is a schedule of topics for this course. This schedule may change as the course progresses based on uncontrolled circumstances, student progress and understanding, or any other reason the instructor determines is a valid reason to make a change.

1. Introduction to Data Structures and Algorithms (Java, abstract data types, generics)
2. Linear Structures Review: Stacks and Queues (array and linked implementations)
3. Big O Analysis, Searching and Algorithm Analysis (linear, binary search)
4. Amortization and Extendable Arrays
5. MID-TERM exam
6. Trees (binary search trees)
7. Balanced Trees (AVL and B-trees)
8. Priority Queues (heaps and treaps)
9. Union-Find / Merge Sort
10. Hash Tables
11. Bloom Filter
12. Graphs (BFS and DFS search algorithms)
13. Review week
14. FINAL exam

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, shape, race, ethnicity, age, religion, nationality, or other similar characteristics.
- 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. 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). 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.

Extra Resources for Data Structures

Below are some helpful texts in case you are interested:

1. Introduction to Java Programming and Data Structures, David Liang. This book has great code and visualizations for data structures. The first part of the book contains a thorough review of introductory java concepts.
2. Object-Oriented Data Structures Using Java, We have used this textbook for many years, but no longer require it.
3. Data Structures and Algorithms in Java, Michael Goodrich, Roberto Tamassia, and Michael Goldwasser. A bit more technical and complete than the previous book, but we haven't thoroughly reviewed it.
4. Data Structures and Algorithms in Java, Robert Lafore. This book has great code and visualizations.
5. Java Precisely, Peter Sestoft. This book is intended more as a reference for people who already know the language, but it has both all the details you aren't likely to memorize and a very clear definition of exactly what the language is and what it does.
6. Learning Java, Patrick Niemeyer and Daniel Leuck. A learn-by-example book about Java. You can read it online for free, or buy the e-book.
7. OpenDSA Data Structures and Algorithms - An on-line textbook with good animations.
8. Algorithms and Data Structures - An online chapter with sample code.

Academic Honesty

It is very important in all courses that you be honest in all the work that you complete. In this course you must complete all assignments, quizzes, exams, etc. on your own unless otherwise specified. If you do not, you are doing a disservice to yourself, the instructors for the course, the Manning College of Information and Computer Sciences, the University of Massachusetts, and your future. We design our courses to provide you the necessary understanding and skill that will make you an excellent computer scientist. Assignments and exams are designed to test your knowledge and understanding of the material. Plagiarism and academic honesty of any sort may seem like an easy way to solve an immediate problem (which it is not), however, it can have a substantial negative impact on your career as a computer science student. There are many computing jobs out there and many more people working hard to get those positions. If you do not know your stuff you will have a very difficult time finding a job. Please take this seriously.

We will carefully review your submissions automatically and manually to verify that "cheating" has not taken place. If you are suspected of plagiarism we will follow an informal path to determine if academic dishonesty has taken place. If you are found guilty you will receive an F for the course and it will go on your permanent record at UMass. This will disrupt your schedule

for completing courses and may lead to you not completing your degree in a timely fashion. You should carefully review the Academic Honesty Policy, Avoiding Plagiarism, and the Academic Honesty Flowchart to understand what academic dishonesty is, how you can avoid it, and the procedure we will follow if you are under suspicion. In general, you should review all documentation described by UMass' Academic Honesty Policy and Procedures.

Specifics for this course

Projects:

1. Projects in this course are individual. Discussing the projects with your peers is allowed, but you must write every line of code on your own and implement solutions on your own. Copying from other students, either digitally or manually, is strictly prohibited.
2. We check all project submissions for similarity to other student submissions and to solutions posted on external websites.
3. If you plan to use version control software, such as Github, GitLab, or Bitbucket to manage your projects, you must make sure your repositories are private and not publicly available.

Course Content:

1. You may not reproduce, distribute, upload, or display lecture slides, lecture recordings, projects, or labs without the faculty member's permission. These activities are a violation of the faculty member's copyright protection.
2. Use of materials from previous offerings of this course, no matter the source, and even if you are retaking the course, is prohibited.
3. Posting materials from this course on any websites either during or after this course is prohibited.

Collaboration:

While we support learning from your peers, the rule of thumb is that any learning should be in your head. Therefore you should not leave an encounter with another student (in person or electronic) with anything written down (or electronically recorded) that you did not have before. Thus, giving or receiving electronic files is specifically considered cheating.

University statement on academic honesty:

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

Visit http://www.umass.edu/dean_students/codeofconduct/acadhonesty for more information.

Copyright Policy

The College of Information and Computer Sciences explicitly forbids any redistribution (including publicly available posting on an internet site) of any CICS course materials (including student solutions to course assignments, projects, exams, etc.) without the express written consent of the instructor of the course from which the materials come. Violations of this policy will be deemed instances of “facilitating dishonesty” (since a student making use of such materials would be guilty of plagiarism) and therefore may result in charges under the Academic Honesty Policy.

Wellness and Success

The professors and teaching assistants are eager to help you learn and to work through any difficulty. Please contact your professors for any assistance that you need.

You are not alone at UMass – many people care about your well-being and many resources are available to help you thrive and succeed.

You have resilience and are already using effective strategies to help you achieve your educational goals. Take stock of these and consider what new steps or resources could be helpful. Getting enough sleep, exercising, eating well, and connecting with others are all antidotes to stress. If you are struggling academically, reach out to us prior to deadlines and before the demands of exams, papers, and projects reach their peak.

Connect with one or more of the many supportive resources on campus that stand ready to assist. You matter to us.

Acknowledgement

We are grateful for the material in this document that has been shared in syllabi in other courses at CICS, UMass.