Syllabus
CICS-160: Object-oriented Programming
Fall 2023

Course Description

This course will expose students to programming practices beyond the introductory level, concentrating on Object Oriented Programming techniques and an introduction to Data Structures. Students will also study and analyze the complexity of both the algorithms presented in class and of the algorithms they develop. Before taking this course, students are expected to have been exposed to the following concepts through a college-level course or equivalent in some high level computer programming language: input and output operations, conditional statements, loops, arrays, recursion, and functions/methods. The course places an emphasis on the careful design and testing of programs, as well as in the analysis of said programs. Students should typically expect this course to require between six and ten hours of hour a week.

Prerequisites: CICS-110, or INFO-190S, or COMPSCI-121, or a score of 4 or higher in the CS AP exam.

Course goals:

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

- Explain key concepts of the object oriented programming paradigm, including data encapsulation, information hiding, inheritance, polymorphic methods, and other.
- Solve programming tasks by decomposing them into smaller tasks through the careful development of algorithms.
- Make use of object oriented practices in order to solve programming problems of medium to advanced complexity.
- Appropriately test the code they develop.
- Explain the concept of Abstract Data Types (ADT), specify the ADT of objects and classes they design, and interact with data structures designed and implemented by other by using their ADT specifications.
- Correctly reason about and explain the complexity of the algorithms and programs seen and developed in class.
- Understand and make use of basic data structures such as lists, stacks, queues, and trees.
- Implement linear data structures such as different types of lists.

Instructor:

Jaime Dávila, jaimedavila@cs.umass.edu, 413-545-8929
Office: LGRC A137
Check Moodle web page.

TAs and UCAs:

Listed on our Moodle page.

Text books:

Students often ask if purchasing the books is mandatory. This is what I have to say about that: no one is going to check to see if you bought the book or not, and your assignments and tests are not taken from these books. Nevertheless, the two books listed here are excellent, to the point that in our schedule (listed further down) I mention which sections of the books you can use as references. Both books are available in electronic and in printed form. You might have a favorite format, but I personally prefer to have a printed book, since I find it easier to jump to a particular page/section. Below I list the books in the colors I refer to them in the schedule further down. I will do my very best to provide you with enough materials in class and in our Moodle web page as to allow you to be successful in this course without necessarily having the listed textbooks. The textbook, however, will often expand and provide additional examples that might accentuate in your learning. As we embark on a continuing journey in learning about computer science, do consider the value of building your own personal reference library.

Textbook on Python and Data Structures: Problem Solving with Algorithms and Data Structures using Python, by Brad Miller and David Ranum. ISBN:9781590282571

Textbook on Python and Object Oriented Programming: Python Object-Oriented Programming - Fourth Edition by Steven F. Lott, Dusty Phillips. ISBN:9781801077262

Grades

Your grade in this course will be based on five programming assignments, nine labs (which all take place during Friday discussion lab meetings) and a four short exams. The exams will all take place during selected Friday labs.

Programming assignments	60%
Labs	10%
Exam	30%

- Programming assignments will always be due by 11:59 PM on the day they are due.

The final letter grade will be as follows, after rounding the numeric grade to the closest integer:

- 93 or higher: A
- 90 to 92 -> A-
- 87-89: B+
- 83-86: B
- 80 to 82: B-
- 77-79: C+
- 73-76: C
- 70 to 72: C-
- 60 to 69: D

Class philosophy:

I strive to create a learning community among all of us. We are not in competition with each other. You are not in competition with each other. I measure success by how many of you do great in the course. I commit to coming to class prepare, to putting my best effort towards your learning, and to develop an environment where all of us can achieve their maximum. I hope you do the same. For all of us, this means coming to class prepared, being ready to work, being respectful of everyone, and working hard in order to do our best work. If there is something you think I can do that would create a better learning environment, let’s talk about it.

Classroom Norms

We celebrate the diversity in our community and actively seek to include and listen to voices that are often silenced in the computing world. We welcome all 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.

Everyone has the right to be addressed by the name and pronouns that they use for themselves. You can indicate your preferred/chosen first name and pronouns on SPIRE, which appear on class rosters. I am committed to ensuring that I address you with your chosen name and pronouns. Please let me know what name and pronouns I should use for you if they are not on the roster. Please remember: A student’s chosen name and pronouns are to be respected at all times in the classroom.

This course is geared towards you working in groups. As such, we expect that you will observe social decorum at all times when interacting with peers. Please consult the UMass Guidelines for Classroom Civility and Respect: http://www.umass.edu/dean_students/campus-policies/classroom

Title IX

UMass is committed to fostering a safe learning environment by responding promptly and effectively to complaints of all kinds of sexual misconduct. 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. If you experience or witness sexual misconduct and wish to report the incident, please contact the UMass Amherst Equal Opportunity (EO) Office (413-545-3464 | equalopportunity@admin.umass.edu) to request an intake meeting with EO staff. Members of the CICS community can also contact Erika Lynn Dawson Head, director of diversity and inclusive community development (erikahead@cics.umass.edu | 860-770-4770).

Piazza

Our course will make use of Piazza for class discussions. I encourage you to post your questions on Piazza. This will allow other students to benefit from your questions, your answers, and other people answers. Seeing questions on Piazza also allows me know what material I might need to go over again. Students can find instructions on how to subscribe to Piazza on our course’s Moodle web page. Typically, your questions on Piazza will be answered in 48 hours or less.

HOMEWORK

- **Homework submissions will be via Gradescope**
    - We will be using gradescope to upload homework and provide feedback. An account on Gradescope.com will be created for you, and you will receive instructions on how to use Gradescope the week before the semester begins.
    - Most of the exercises in assignments will be automatically graded by gradescope. That means that you will be able to submit your answers, receive feedback, see if your answers as correct, and resubmit new answers, if you want. As such, the earlier you start working on assignments, and submitting your work to gradescope, the better.
- **Late Homework Policy**

    - Turning homework in late helps no one. When students turn homework in late, they fall behind. Late homework also keeps me from being able to give you feedback on time, from detecting which material I need to re-emphasize, etc.
    - Nevertheless, I do understand that personal issues can keep any of us from being able to meet particular deadlines. Because of this, you have a total of four “late without penalty” days, which you can use for whichever assignment(s) you might need to. You can use one late day for one assignment, two for another, and so forth, up to a total of four. Or you can use no late days at all, which is what I recommend if at all possible, since it will allow you to keep appropriate pace with the course material. If you do use these late days, it is your responsibility to keep track of them. If you submit an assignment late, and you have late days still available, they will be automatically used.
    - The only exception to this is justified medical or personal situations that fall outside the ordinary. If you have a medical situation that keeps you from turning an assignment in, please ask for and provide documentation from a medical professional. If you have a personal life situation that keeps you from submitting an assignment on time, let me know as quickly as humanly possible. I might ask you for documentation in those cases too. I’m not trying to be stricter than I need to be, I just want to avoid assignment submissions from turning into a free-for-all.
    - If you have an accommodation need officially documented with UMass, please provide me with an official letter, and I will, of course, provide whatever accommodations are needed, in terms of assignments, exams, or anything else.
- I have designed assignments to be a central part of your learning. As such:
    - you should expect a great amount of learning to come from your homework. You will see that, while recorded videos and slides will provide you with a lot of information, some material will only become clearer as you work on assignments. Assignments will engage you in the exploration of important concepts. As such, you should not expect to “know how to do everything” before you engage in the assignments. This is normal. This is how assignments have been designed.
    - Because of this, I recommend that you start working on assignments as soon as the material is available.

Academic Honesty and Collaboration Policy:

You may collaborate with other students on homework assignments, provided that (a) you indicate anyone with whom you worked and (b) the materials you submit are entirely your own. As a guideline, to distinguish permitted collaboration from plagiarism, feel free to discuss problems verbally or via temporary written means (e.g. whiteboard), but do not share written files, printouts, or pages, or take screenshots. It is OK, and even encouraged, for you to talk about algorithms and general ideas, but you should not be discussing specifics of code with others, except for code presented in class or discussed in our textbooks. You should then, though, write code into your own files, and submit only those files. If you have questions about this matter, please ask.

Please, please observe this academic honesty policy. Having to handle academic dishonesty cases is an unpleasant experience for everyone involved. If you feel like you are falling behind with material, are concerned by your grade, or there’s anything else that keeps you from engaging as might be needed with material, I’m here to help. I am successful if you are all successful. Let’s get to our learning goals, together, and honestly.

We follow the university's Academic Honesty Policy and Procedures. You can find those at https://www.umass.edu/honesty/ .

Important note on chatGTP and other AI-driven code generation systems.

The use of chatGPT and other AI-driven code generation systems in this course is forbidden, and will be considered academic dishonesty. There are moments, times, and places where such systems could be useful. This course is not one of them. This has at least two important reasons:

- chatGPT bases the code it generates on other people's code. As such, it is not able to develop completely original code. I want students to be able to create original code when needed. That should also be one of your goals. No amount of reading and studying code generated by someone/something else will fully prepare me to develop fully original code. The same thing applies to other longer-standing sources of code, like stack overflow, github, chegg, friends, etc.
- chatGPT generates code/prose/images/whatever based on things it has seen before. If it has seen garbage before, it will every once in a while generate garbage, but will carry on as if its output were wonderful, not knowing that it is indeed incorrect. This is dangerous in a world where good software is essential and can affect safety and fairness.

Attendance Policy

Attendance to Friday discussion lab meetings is mandatory. 10% of your grade will be based on work you do during lab meetings. If you cannot make it to a lab meeting because of an important unavoidable reason (medical or personal/family emergency, official UMass Amherst trip, etc.), you need to inform your lab leader as soon and as early as you can. You will still need to submit the work completed during that lab by working on your own. These labs are designed to expose you to important course concepts, and to allow you to practice important concepts in a supportive environment. There are many important course topics that will be introduced during labs, so attending labs and engaging with lab exercises is an essential part of the course.

Attendance to lecture meetings (Tuesdays and Thursdays) is not mandatory, but is strongly suggested. You will find copies of slides on moodle, and our lecture meetings will be recorded to echo360. However, based on my 23 years of teaching at the college level, I can guarantee to you that your learning will benefit from coming to class, if at all possible.

Accommodation statement

It is my firm commitment to provide each student, to the best of my abilities, with equitable access to educational opportunities. In addition, The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented accommodation on file with Disability Services (DS), you will be provided with reasonable deadline extensions that might be needed for you to succeed in this course. If you have a documented disability that requires an accommodation, please notify me during the first week of the course so that we may make appropriate arrangements.

Additional Resources

Supplemental Instruction

This course has Supplemental Instruction (SI) available. The Moodle page for that service, with information about meeting times and other resources, is at https://umass.moonami.com/course/view.php?id=35809

Course Topics

Week	Topics	Work due
1	• Introductions to the course. • The VSCode IDE	• Lab 1 worksheet, on Friday February 10 by 8 PM.
2	• Python review ○ input/output ○ conditionals ○ lists ○ loops ○ perhaps new: dictionaries • Textbook sections: 1.4.1-1.4.5	• Lab 1 reflections, on Sunday February 12 • Short exam/quiz, during Friday lab (February 17)
3	• Classes and Objects ○ Basics (encapsulation, methods, information hiding, Abstract Data Types) • Textbook sections: 1.4.6 • Textbook chapter 1	• HW#1 due on Sunday February 19 at 11:59 PM • Lab 2 worksheet due on Friday February 24 by 8 PM.
4	• Inheritance • Testing (with unittest) • Type hints. • textbook chapter 1.6: inheritance • textbook chapter 13.2: testing with unittest • Textbook chapter 2.2: Introducing type hints	• Lab 2 reflections due on Sunday February 26. • Worksheet for lab 3 due Friday March 3 by 8 PM
5	• Introduction to complexity analysis • Searching in a list • Textbook chapter 8: Lists	• Reflections for lab 3 due Monday March 6 • Lab 4 worksheets due Friday March 10 by 8 PM.
6	• Searching • Introduction to complexity analysis • Textbook chapter 8: Lists	• Reflections for lab 4 due Sunday March 12 • Lab 5 worksheet due Friday March 24 by 8 PM.
7	• Implementing linked lists. • textbook section: 3.6	• Reflections for lab 5 due Sunday March 26. • Assignment 2 due on Sunday March 26 • Short exam/quiz, during Friday lab, March 31
8	• Sorting: ○ bubble sort ○ selection sort • textbook sections: 2.1-2.3;5.1-5.3.3; 5.3.1	
9	• Recursion • Sorting recursively ○ merge sort ○ quick sort • textbook sections: 5.3.5 – 5.3.6	• Reflections for lab 6 due Sunday April 9. • Short exam/quiz during Friday lab, April 14.
10	• The implementation of lists	• Assignment #3 due on Sunday April 16.

Week	Topics	Work due
	○ with arrays vs. with linked lists • textbook section: 3.6	• Lab 7 worksheet due Friday April 21 by 8 PM.
11	• Introduction to Java ○ inheritance ○ abstract classes ○ interfaces • Ordered lists.	• Reflections for lab 7 due on Sunday April 23. • Worksheet for lab 8 due on Friday April 28
12	• Implementing lists in Java	• Reflections for lab 8 due on Sunday April 30 • Short exam/quiz during Friday lab, Friday May 5
13	• Abstract classes in Python • Queues and stacks • Texbook chapter 7	• Assignment #4 due on Sunday May 7 • Worksheet for lab 9 due on Friday May 12
14	• The SOLID programming principles	• Reflections for lab 9 due on Sunday May 14 • HW #5 due on Wednesday May 17