Syllabus
CICS 160: Object-Oriented Programming
Fall 2022

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. This course also provides experience with the development and analysis of recursive algorithms and programs. 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.

Prerequisites: CICS-110, or INFO-190S, or COMPSCI-121, or a score of 4 or higher in the CS AP exam.
Credits: This course is worth four credits.

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 others by using their ADT specifications.
- Correctly reason about and explain the complexity of the algorithms and programs seen and developed in class, including recursive algorithms and programs.
- 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
Office: LGRC A137, and https://umass-amherst.zoom.us/j/2581898802 Office hours:

Mondays 11 AM – 12 noon
Tuesdays 9-10 AM
Thursdays 2:30-3:30 PM
Fridays 9-10 AM or by appointment

Class Meetings:

Both the lecture sessions (Mondays and Wednesdays 2:30-3:45 PM) and the lab sessions (Mondays 10:10-11 AM) meet at the Computer Science building, room142.

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.

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 four programming assignments, eight weekly labs (which all take place during our Mondays 10:10-11 AM meetings) and five short exams. The exams will all take place during selected Monday labs.

Programming assignments	60%
Labs	10%
Exams	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 prepared, 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.

Inclusivity Statement

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.

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

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.

To find the Piazza page for our course, please follow the link you can find in our course's Moodle page.

Homework

- **Homework submissions will be via Gradescope**
  - We will be using gradescope to upload homework and provide feedback. You can find a link and instructions on our course's main Moodle page.
  - 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. Because of this, the general rule is that late homework will not be accepted. 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/ .

Attendance Policy

Attendance to lab meetings (Mondays at 10:10 AM) is required. 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 let me know 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.

Attendance to lecture meetings (Mondays and Wednesdays 2:30-3:45 PM) 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.

Course Topics

The following schedule might need to be modified slightly, depending on how things progress during the semester.

Week	Dates	Topics	Work due	Notes
1	September 6-10	• Introductions to the course. • The VSCode IDE		No lab this week. We will only meet on Wednesday September 7
2	September 11-17	• Python review ◦ input/output ◦ conditionals ◦ lists ◦ loops ◦ perhaps new: dictionarie • Textbook sections: 1.4.1-1.4.5 s		
3	September 18-24	• Classes and Objects ding, ◦ Basics (encapsulation, methods, information hi Abstract Data Types) • Textbook sections: 1.4.6 • Textbook chapter 1	Exam #1 (on I/O, conditionals, loops, and arrays)	
4	September 25 – October 1	• Inheritance • Testing (with unittest) • Type hints. ◦ type hints • textbook chapter 1.6: inheritance tedtbook chapter 13.2: testingh wit unittest • Textbook chapter 2.2: Introducing type hints	HW#1 (Python review)	
5	October 2-8	• Recursion • program correctness • textbook sections: 4.1-4.2		
6	October 915	• Sorting: ◦ searching in a sorted list	HW#2 (Lists)	No lab this week. We will only meet on

		◦ bubble sort • textbook sections: 5.1-5.2.2; • 5.3.1 Texbook chapter 8: Lists		Wednesday October 12
7	October 1622	• Introduction to complexity analysis • Sorting ◦ selection sort ◦ insertion sort • textbook sections: 2-1 – 2.3, 5.3.2 – 5.3.3 •	Exam #2 (up to and including lists)	
8	October 2329	• Sorting recursively	HW#3 (sorting)	

Week	Dates	Topics	Work due	Notes
		◦ merge sort ◦ quick sort • textbook sections: 5.3.5 – 5.3.6		
9	October 30-November 5	• The implementation of lists • textbook section: 3.6	Exam #3 (Sorting)	
10	November 6-12	• Implementing linked lists. • textbook section: 3.6		
11	November 13-19	• Further exploration of object oriented programming ◦ nodes ▪ textbook section: 3.6.2.1 ◦ information hiding	Exam #4 (Linked lists)	
12	November 20-26	• Ordered lists. • Abstract classes in Python • Texbook chapter 7		No Wednesday class. We will only meet on Monday November 21
13	November 26-December 3	• Implementing lists in Java ◦ Basics ◦ inheritance and multiple inheritance ◦ abstract classes ◦ interfaces	HW#4 (lists in Java)	
14	December 410	• Queues and stacks	Exam #5 (Java, inheritance, abstract	

			classes, interfaces)	
15	December 11-12	• Review		Last day of classes is Monday December 12