CSE 2050: Data Structures and Object-Oriented Design

Term: Fall 2023
Instructor: Prof. Jake Scoggin (jake.scoggin@uconn.edu)
Office location: EII 316
Office Hours and Availability: TBD by second week of class. We will do our best to respond to Discord posts within 24 hours and emails within 48 hours Monday – Friday.

Course Descrition

Data Structures and Object-Oriented Design is a foundational course that delves into the world of data structures and algorithms. In this course, students will embark on a journey to understand the core principles and best practices for designing efficient data structures, while also exploring the fundamental concepts of object-oriented programming (OOP).

This course was created with the aim of acknowledging and embracing the unique thinking and learning styles of neurodiverse students. We strongly believe in fostering an inclusive and empowering learning environment that focuses on individual strengths. Throughout the course, we will introduce various innovative teaching methods, such as collaborative learning with fellow students, stimulating activities such as coding challenges and worksheets, and different assessments like exams, coding assessments, labs, and homework assignments. These methodologies are geared towards not only enriching your comprehension but also fostering active engagement and offering effective problem-solving opportunities. In addition to traditional in-person lectures, we provide supplementary resources like recorded videos and module-specific slides to accommodate various learning styles. Our lectures will also be live streamed to ensure accessibility. We strongly encourage collaboration with other students, promoting group discussions on homework assignments, and fostering a supportive learning environment.

Course Objectives

1. Write programs in python using imports, functions, and object-oriented programming.
2. Compare data structures and algorithms based on time and space complexity and choose the correct ones for a given problem.
3. Implement abstract data types (stacks, queues, deques, mappings, priority queues) using various data structures (lists, linked lists, doubly linked lists, heaps, trees, graphs) and algorithms.
4. Use recursive algorithms to solve problems.

Schedule

Dates	Module	Book
8/28 - 9/1	Mod 1 - Basic Python	Ch. 1-4
9/4 - 9/8	Mod 2 - Object-Oriented Programming and Testing	Ch. 3-4
9/11 - 9/15	Mod 3 - Running Time Analysis	Ch. 5
9/18 - 9/22	Mod 4 - Linear Data Structures	Ch. 6-8
9/25 - 9/29	**Programming Assessment 1 (Thursday lab) and Exam 1 (Friday lecture)**	
10/2 - 10/6	Mod 5 - Recursion and Dynamic Programming	Ch. 9-10
10/9 - 10/13	Mod 6 - Searching and Sorting	Ch. 11-12
10/16 - 10/20	Mod 7 - Divide-and-Conquer	Ch. 13-14
10/23 - 10/27	Mod 8 - Mappings and Hashing	Ch. 15
10/30 - 11/3	**Programming Assessment 2 (Thursday lab) and Exam 2 (Friday lecture)**	
11/6 - 11/10	Mod 9 - Trees	Ch. 16-18
11/13 -11/17	Mod 10 - Priority Queues and Heaps	Ch. 19
11/20 - 11/24	Fall Recess	
11/27 - 12/1	Mod 11 - Graphs (Part I)	Ch. 20-21
12/4 - 12/8	Mod 12 - Graphs (Part II) and **Programming Assessment 3 (Thursday lab)**	Ch. 20-21
12/11 - 12/15	**Exam 3**	

Each module has a lab and a homework. Labs are due at 11:59 PM EST Sunday (e.g. Lab 1 is due on 9/3). Homework is typically due at 11:59 PM EST Tuesday (e.g. the Module 1 homework assignment is due on 9/5).

Materials, Platforms, and Software

- **Textbook:** A First Course on Data Structures in Python. (open source pdf)
- Computer and internet connection
- HuskyCT
- Gradescope (accessed through HuskyCT)
- Discord (See HuskyCT for server invite)
- Python 3

  We code in Python 3 in this course. You can code however you're comfortable for labs and homework but will need to use a provided virtual machine for the three programming assessments.

  Your virtual machine supports terminal (text-based) coding via vim, emacs, or nano, as well as GUI coding via VS Code in browser, so it is best to use one of these methods for all assignments.

Grades

The grading breakdown for the course is as follows:

Course Components	**Weight**
Formative	
Lecture Participation	5%
Lab Participation	5%
Labs	10%
Homework assignments	10%
Summative	
Programming Assessment 1	10%
Programming Assessment 2	15%
Programming Assessment 3	15%
Exam1	10%
Exam2	10%
Exam3	10%

Detailed descriptions of each course component are below.

Participation

We want you to actively participate in labs and lectures for 2 main reasons:

1. Effective communication and the ability to work with others are part of the Computer Science program objectives (full list here). These are great opportunities to build those skills.

2. The single greatest predictor of student performance is class attendance. From a meta-analysis of 52 published articles and 16 unpublished dissertations:

   "These relationships make class attendance a better predictor of college grades than any other known predictor of academic performance, including scores on standardized admissions tests such as the SAT, high school GPA, study habits, and study skills."

Our course schedule consists of three sets of four modules: Set 1 (Mods 1-4), Set 2 (Mods 5-8), and Set 3 (Mods 9-12). We calculate participation discretely for each set: you can earn up to one-third of the total 5% allocated to lecture and lab participation during each set.

In all cases, credit is given for active participation - attendance is not enough.

Lecture Participation (5%)

- You can earn 1 point for each lecture you attend by participating in class activities.
- There will be a weekly quiz completed during class on Friday worth up to 1 point.
- This means you can accumulate a total of 4 points every week, or up to 16 points for each set of modules (ignoring lectures that are cancelled due to weather or instructor illness).
- You need at least 12 lecture points for each set to get maximum credit for that set.
- That means you can miss about 25% of lectures during each set, no questions asked.

While we encourage actively participating in as many lectures as possible, there is no extra credit for going above and beyond 12/16 points in each set. We want to avoid any pressure to attend when you are mentally or physically unwell, and feel extra credit here might lead towards decisions that are unhealthy for you and your classmates.

Points do not roll over, since we want to encourage participation during what are traditionally the toughest modules; e.g. you **cannot** accumulate 32 points during the first 2 sets of modules, and use those to skip all but 4 lectures during the final set.

Lab Participation (5%)

We have a total of 11 labs (we will hold a programming assessment covering modules 9-11 during lab on the week of module 12):

- You can earn 4 points for each lab you attend, so...
- You can accumulate 16 points during each of the first two sets of modules and 12 points in the last.
- Full credit for lab participation requires 12/16 points during each of the first two module sets and 8/12 points during the final set.

As with lectures, there is no extra credit for surpassing the lab participation thresholds, and points do not carry over between module sets.

Labs (10%)

Weekly collaborative assignments. Feel free to share code with your partner, but partners must each submit work individually. We design labs to be completable in 75 minutes, but they are due a few days after lab to give you time to troubleshoot if things go poorly.

Homework (10%)

Homework assignments consist of problems tailored to help you evaluate your understanding of each module. Homework assignments may use any mix of manual and automated grading; exact weights will be tailored to each problem set. The standard rubric we will use for manually grading coding assignments can be found here.

Homework Collaboration

You can collaborate with up to 3 classmates on each homework assignment, for a total of at most 4 students in each such group. You must note any such collaborators in your submission (a comment at the top of the main script in that homework is fine). Discussion is encouraged, but you must fully complete each assignment independently - any code you submit must be written solely by you, and sharing your code (by sharing files, screen-sharing a text editor with the file open, or telling someone what to write line-by-line) is forbidden.

You may also discuss homework problems within the appropriate channels on our Discord server. As above, keep these discussions conceptual - do not post actual homework code. Small snippets of related problems to help convey your point are fine.

Using Code from the Internet

The textbook and external sites like GeeksForGeeks are excellent resources, but copying large chunks of code from them undermines our assessment abilities. We therefor do not allow copy-/pasting or importing code in any assignments in this course unless otherwise specified - you should write all code you submit yourself.

Should you find a function online that you think is appropriate to solve a problem, follow the following protocol to ensure that the work you submit is your own:

- Copy/paste the code locally. Run it, experiment with it, extend it.
- Delete your local copy.
- Go do something else for at least 2 hours.
- Come back and write the code from scratch without referencing the source.

This will ensure you really understand any code you are submitting to the point of mastery - the ability to write such code from scratch.

Similarly, do not use code written by AI tools such as ChatGPT. If using these tools helps you understand concepts, that's fine, but any code you submit must be synthesized and written by you.

Do not post new questions on forums (e.g. Chegg, StackOverflow, Reddit, other Discord servers). Referencing old questions is fine, but remember you have to write all your code from scratch.

Programming Assessments (40%)

While most of your labs will be formative (designed to help you learn), we will have 3 summative (designed to measure mastery) programming assessments during lab:

- Programming Assessment 1 (Mods 1-4) - 10%
- Programming Assessment 2 (Mods 5-8) - 15%
- Programming Assessment 3 (Mods 9-11) - 15%

Exams (30%)

We will have 3 exams to assess conceptual mastery. The first 2 exams will be held during the same weeks as the programming assessments; the third will be during finals week:

- Exam 1 (Mods 1-4) - 10 %
- Exam 2 (Mods 5-8) - 10 %
- Exam 3 (Mods 9-12) - 10 %

We do not offer makeup exams outside of exceptional circumstances. If you cannot take an exam, contact your professor as soon as possible. The earlier you contact us, the more likely we can provide a better alternative than a 0.

Late Work Policy

This course is too large for us to handle late work requests on a case-by-case basis with the appropriate compassion and nuance. Additionally, request-based late policies are subject to implicit bias of the instructor (see e.g. Chin et. al 2020) and may be underutilized by those who feel they will be judged for making such requests (see e.g. Whillans et. al 2021).

Instead, we use the following no-questions asked late policies:

1. We will drop the lowest lab and homework from the first 10 modules. The final homework and lab are not eligible to be dropped.
2. You can submit any lab up to 48 hours late, no questions asked. Note that if you find yourself regularly submitting labs late, you are likely falling behind in the course, and should speak to an instructor or TA about getting caught up ASAP.

3. You will have 5 late day tokens to spend on homework as you see fit. You can use 1 late day token to submit up to 24 hours late, or 2 to submit up to 48 hours late.

   To use a late token, just submit your homework within the 48-hour period after the deadline. We will manually deduct any late tokens used after the late window closes.

4. We do not accept any late work after the 48-hour window, and will not grade homework assignments submitted late without sufficient late tokens. You can see how many late day tokens you have left in HuskyCT under My Grades.

In the event of a major life event that warrants additional extensions (it happens), contact the dean of students. They will provide you with additional resources, including contacting your instructors for additional extensions.

Thresholds

We will determine exact floors and half letter grades at the end of the semester according to the standards of performance below. Instructors reserve the right to modify the floors up or down to match the appropriate level of mastery.

Grade	Letter Grade	GPA
93-100	A	4.0
90-92	A-	3.7
87-89	B+	3.3
83-86	B	3.0
80-82	B-	2.7
77-79	C+	2.3
73-76	C	2.0
70-72	C-	1.7
67-69	D+	1.3
63-66	D	1.0
60-62	D-	0.7
<60	F	0.0

Academic Misconduct

The penalty for academic misconduct is an F in the course. Academic misconduct includes but is not limited to:

- Submitting any code you did not write yourself
- Sharing any code with classmates
- Sharing a video feed of your code
- Discussing exams before all grades are posted
- Posting questions on forums like Reddit, Stack- Overflow, or Chegg
- Using AI-tools like ChatGPT to write code for an assignment

Student Responsibilities and Resources

As a member of the University of Connecticut student community, you are held to certain standards and academic policies. In addition, there are numerous resources available to help you succeed in your academic work. Review these important standards, policies and resources, which include:

- The Student Code
  - Academic Integrity
  - Resources on Avoiding Cheating and Plagiarism
- Copyrighted Materials
- Credit Hours and Workload
- Netiquette and Communication
- Adding or Dropping a Course
- Academic Calendar
- Policy Against Discrimination, Harassment and Inappropriate Romantic Relationships
- Sexual Assault Reporting Policy

Students with Disabilities

The University of Connecticut is committed to protecting the rights of individuals with disabilities and assuring that the learning environment is accessible. If you anticipate or experience physical or academic barriers based on disability or pregnancy, please let me know immediately so that we can discuss options. Students who require accommodations should contact the Center for Students with Disabilities, Wilbur Cross Building Room 204, (860) 486-2020 or http://csd.uconn.edu/.

Blackboard measures and evaluates accessibility using two sets of standards: the WCAG 2.0 standards issued by the World Wide Web Consortium (W3C) and Section 508 of the Rehabilitation Act issued in the United States federal government." (Retrieved March 24, 2013 from Blackboard's website.

Help

Technical and Academic Help provides a guide to technical and academic assistance. This course is completely facilitated online using the learning management platform, HuskyCT. If you have difficulty accessing HuskyCT, you have access to the in person/live person support options available during regular business hours through the Help Center. You also have 24x7 Course Support including access to live chat, phone, and support documents.

Evaluation of the Course

Students will be provided an opportunity to evaluate instruction in this course using the University's standard procedures, which are administered by Office of Institutional Research and Effectiveness (OIRE). Additional informal formative surveys may also be administered within the course as an optional evaluation tool.

Excluding materials for purchase, syllabus information may be subject to change. The most up-to-date syllabus is linked in Discord.