Syllabus
CSE 250: Data Structures

University at Buffalo, Computer Science and Engineering

Spring 2026

Instructor

Name	Email	Office	Office Hours
Dr. Eric Mikida	epmikida@buffalo.edu	208 Capen Hall	TBD

1. Course Information

CSE 250 Data Structures – 4 credits

1.1. Prerequisites

CSE 116 and (CSE 191 or MTH 311) and (MTH 141 or MTH 131 or MTH 121). Engineering, Bioinformatics, Computational Physics, or Math Majors, Computer Science Minors, and Data Intensive Computing Certificate students only

1.2. Description

Provides a rigorous analysis of the design, implementation, and properties of advanced data structures. Topics include time-space analysis and tradeoffs in arrays, vectors, lists, stacks, queues, and heaps; tree and graph algorithms and traversals, hashing, sorting, and data structures on secondary storage. Surveys library implementations of basic data structures in a high-level language. Advanced data structure implementations are studied in detail. Illustrates the importance of choosing appropriate data structures when solving a problem by programming projects in a high-level language.

1.3. Learning Outcomes

Upon completion of the course, students will be able to...
1. compute, compare, and analyze runtime and function growth using asymptotic notation.
2. identify functionality of basic data structures.
3. identify the tradeoffs of different data structures, given their implementation. This also includes recognizing which situations benefit or suit the use of one data structure over another.
4. use data structures in programming.
5. implement and analyze basic algorithms such as searching and sorting, as well as recursive algorithms, tree traversal algorithms, and graph traversal algorithms.

2. Required Resources

2.1. Textbooks

There is no textbook for the course. All needed materials will be provided via the course's website.

2.2. Other Resources

You will be using various free on-line tools for this course – links will be posted on the course website. Course-related communications should be via the Piazza forum linked from the course website. Piazza posts can be either public to the class or private to instructors. Any email communications must come from your UB email account and include [CSE 250] in the subject line. All communications with course staff are expected to be professional. Graded work will be both submitted and returned via Autolab.

3. Course Requirements

3.1. Assignments

Homework assignments for this course will be composed of:
- Programming Assignments
- Written Assignments

You can expect three Programming Assignments and three Written Assignments over the course of the semester. Assignments are submitted via Autolab, and all assignments have an equal weight. General expectations for each assignment are described below, but every assignment will clearly state all deadlines, expectations, and the submission process in the handout for the assignment.

In addition to the normal assignments there is a short Academic Integrity quiz and a short assignment for setting up your programming environment at the start of the semester. **You must get a 100% on both the Academic Integrity quiz and setup assignment to pass the course.**

Programming Assignments
Programming assignments assess your ability to apply concepts from lecture to both implement and use various data structures. Programming assignments consist of two parts: a testing phase and an implementation phase. Generally speaking you will have two weeks to complete a programming assignment, with one week dedicated to the testing phase and one week dedicated to the implementation phase. Programming assignments are graded automatically by AutoLab, and students may submit multiple times to address autograded feedback. **Your grade on each phase of each programming assignment is based on your LAST submission for that phase**.

Late Policy for Programming Assignments
Since the implementation phase of each programming assignment depends on the testing phase, **no late submissions are accepted for the testing phase of each programming assignment**.

For the implementation phase, each programmng assignment remains open for submission for two days after the stated deadline.

- If your last submission is submitted on or before the stated deadline, your grade will be 100% of the points your submission earns.
- If your last submission is submitted after the stated deadline, a 25% per day penalty will be applied to the points your submission earns.
- No submissions are accepted for any reason more than two days after the stated deadline.

Each student receives 3 grace days at the start of the semester. Grace days are **automatically** applied by AutoLab to negate the 25% penalty for late submission. For example, if a students last submission is two days late, but they have at least two grace days, those grace days will be spent and no grade penalty will be applied. If they have one grace day, that grace day will be spent, and only a 25% penalty will be applied. If the student has no grace days remaining, the full 50% penalty (25% per day late) will be applied.

Please plan accordingly. **You will not be able to recover a grace day if you decide to submit late and your score was not higher.** It is important that you test your code before submitting, especially with late submissions, in order to avoid wasting grace days. Grace days will not allow you to submit more than two days late, nor will grace days allow you to make a late submission for the testing phase.

Written Assignments
Written assignments will assess your understanding of the theoretical concepts taught in the course, as well as your conceptual understanding of what you implemented in prior programming assignments. Written assignments will generally be given on a shorter basis, with around 1 week to complete. Submissions must be in the form of a PDF. When creating a PDF for submission you generally have two options:

1. You may typeset your written submissions using any word processor you wish (Microsoft Word and LaTeX are good options). Overleaf is a convenient web-based options for using LaTeX. If you use Word, make sure to export your submission as a PDF.

2. You may handwrite your written submission and then scan it using your phone, your own scanner, or an on campus scanner. The campus libraries provide scanning services as well. If you handwrite your assignment, ensure that you write legibly.

Make sure to double check your assignments before and after submission to ensure that your submission looks as you expect, and all pages are present. **You will not receive credit if your submission is invalid/corrupt/wrong file format or if your submission is illegible. It is fully your responsibility to determine if your submission is valid.**

Late Policy for Written Assignments
Late written assignments will be accepted up to 1 day late for a penalty of 50% of the earned points. **No grace days may be applied to written assignments.**

3.2. Attendance and Participation

Attendance in lecture is not mandatory, but highly encouraged. You are, after all, paying us to deliver an interactive experience. If you don't understand something, wouldn't it be nice to just raise your hand and get to the bottom of it then and there? Lectures will typically be recorded, but recordings have been known to get corrupted; do not count on lecture recordings.

3.3. Recitations

We will be meeting for recitation to go over homework assignments and any questions you have about the material. In addition, the recitations may review or extend lectures and are an excellent environment to ask more individual questions regarding the course material. Attendance in recitation is mandatory. Questions posed in recitation may be graded for participation or correctness. You may miss 3 recitations without penalty for any reason.

3.4. Exams

There will be two in-class 45-minute midterm exams and one 3-hour final exam. The midterms are worth 15% of your grade each. The final exam is worth 30% of your grade. We reserve the right to change the scaling of the exams.

No makeup exams will be given except in provably extreme circumstances.

4. Grading

The following indicates the grade breakdown which will be used in assigning grades in the course. It is possible that these ranges may be adjusted at the end of the semester to address inconsistencies or hardships that arise. Grades will not be curved/adjusted during the semester.

Requirement	Weight	Percentage	Letter Grade	Percentage	Letter Grade
Assignments	30%	$90 \leq x \leq 100$	A	$65 \leq x < 70$	C+
Participation	10%	$85 \leq x < 90$	A-	$60 \leq x < 65$	C
Midterm Exams	15% x 2	$80 \leq x < 85$	B+	$55 \leq x < 60$	C-
Final Exam	30%	$75 \leq x < 80$	B	$50 \leq x < 55$	D
		$70 \leq x < 75$	B-	$x < 50$	F

4.1. Regrades

Any questions about the grading of a piece of work must be raised within one week of the date that the graded work was returned to you. Instructions for how to submit a regrade request will be given when the work is returned.

4.2. Incomplete (I) Grades

A grade of incomplete ("I") indicates that additional coursework is required to fulfill the requirements of a given course. Students may only be given an "I" grade if they have a passing average in coursework that has been completed and have well-defined parameters to complete the course requirements that could result in a grade better than the default grade. An "I" grade may not be assigned to a student who did not attend the course.

Prior to the end of the semester, students must initiate the request for an "I" grade and receive the instructor's approval. Assignment of an "I" grade is at the discretion of the instructor.

The last day to resign the course is **Tuesday, April 14, 2026**.

5. Academic Integrity

Academic integrity is a fundamental university value. Through the honest completion of academic work, students sustain the integrity of the university while facilitating the university's imperative for the transmission of knowledge and culture based upon the generation of new and innovative ideas. Please refer to the university Undergraduate Academic Integrity Policy https://catalog.buffalo.edu/policies/academic_integrity_2019-20.html for additional information.

As an engineer or computer scientist, you have special ethical obligations. As per the NSPE Code of Ethics, "engineers shall avoid deceptive acts" and "shall conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession https://www.nspe.org/resources/ethics/code-ethics. Similar sentiments of honesty, integrity, fairness, and responsibility are fundamental to the ACM Code of Ethics https://www.acm.org/code-of-ethics.

A violation in this class generally results in an F for the entire course. The Computer Science and Engineering department's policy on academic integrity can be found here:

https://engineering.buffalo.edu/computer-science-engineering/information-for-students/undergraduate-program/cse-undergraduate-academic-policies/cse-academic-integrity-policy.html

5.1. What Constitutes a Violation of Academic Integrity?

The work you submit for assessment must be your own work, developed by yourself using only the allowable resources stated below.

These bullets should be obvious things not to do (but commonly occur):
- Turning in your friend's code/write-up (obvious).
- Turning in code creating using any sort of generative AI, including any present in your IDE of choice (obvious).
- Turning in solutions you found on Google with all the variable names changed (should be obvious). This is a copyright violation, in addition to an AI violation.
- Turning in solutions you found on Google with all the variable names changed and 2 lines added (should be obvious). This is also a copyright violation.
- Paying someone to do your work. You may as well not submit the work since you will fail the exams and the course.
- Posting to/searching forums (ie Stack Overflow) asking someone to solve the problem.

Other violations that may not be as obvious:
- Working with a tutor who solves the assignment with you. If you have a tutor, please contact me so that I may discuss with them what help is allowed.
- Sending your code to a friend to help them. If another student uses/submits your code, you are also liable and will be punished.
- Joining a chatroom for the course where someone posts their code once they finish, with the honor code that everyone needs to change it in order to use it.
- Reading your friend's code the night before it is due because you just need one more line to get everything working. It will most likely influence you directly or subconsciously to solve the problem identically, and your friend will also end up in trouble.

5.2. What Collaboration is Allowed?

Assignments in this course should be solved individually with only assistance from course staff and allowed resources. You may discuss and help one another with technical issues, such as how to get your compiler running, etc.

There is a gray area when it comes to discussing the problems with your peers and I do encourage you to work with one another to solve problems. That is the best way to learn and overcome obstacles. At the same time you need to be sure you do not overstep and not plagiarize. Talking out how you eventually reached the solution from a high level is okay:

"I used a stack to store the data and then looked for the value to return."

but explaining every step in detail/pseudocode is not okay:

"I copied the file tutorial into my code at the start of the function, then created a stack and pushed all of the
data onto the stack, and finished by popping the elements until the value is found and use a return statement."

The first example is OK but the second is basically a summary of your code and is not acceptable, and remember that you shouldn't be showing any code at all for how to do any of it. Regardless of where you are working, you must always follow this rule: Never come away from discussions with your peers with any written work, either typed or photographed, and especially do not share or allow viewing of your written code.

5.3. What Resources are Allowed?

With all of this said, please feel free to use any [files|examples|tutorials] that we provide directly in your code (with proper attribution). Feel free to directly use anything from lectures or recitations. You will never be penalized for doing so, but should always provide attribution/citation for where you retrieved code from. Just remember, if you are citing an algorithm that is not provided by us, then you are probably overstepping.

More explicitly, you may use any of the following resources (with proper citation/attribution in your code):
• Any example files posted on the course webpage (from lecture or recitation).
• Any code that the instructor provides.
• Any code that the TAs provide.
• Any code from the Java API (https://docs.oracle.com/javase/8/docs/api/)

Omitting citation/attribution will result in an AI violation (and lawsuits later in life at your job). This is true even if you are using resources provided.

5.4. Amnesty Policy

We understand that students are under a lot of pressure and people make mistakes. If you have concerns that you may have violated academic integrity on a particular assignment, and would like to withdraw the assignment, you may do so by sending us an email **BEFORE THE VIOLATION IS DISCOVERED BY US**. The email should take the following format:

```
Dr. Mikida,

I wish to inform you that on assignment X, the work I submitted was not
entirely my own. I would like to withdraw my submission from consideration to
preserve academic integrity.

J.Q. Student
Person #12345678
UBIT: jqstuden"
```

When we receive this email, student J would receive a 0 on assignment X, but would not receive an F for the course, and would not be reported to the office of academic integrity.

6. Critical Campus Resources

6.1. Accessibility Resources

If you have any disability which requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources in 60 Capen Hall, 716-645-2608 and also the instructor of this course during the first week of class. The office will provide you with information and review appropriate arrangements for reasonable accommodations, which can be found on the web at: http://www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html.

6.2. Sexual Violence

UB is committed to providing a safe learning environment free of all forms of discrimination and sexual harassment, including sexual assault, domestic and dating violence and stalking. If you have experienced gender-based violence (intimate partner violence, attempted or completed sexual assault, harassment, coercion, stalking, etc.), UB has resources to help. This includes academic accommodations, health and counseling services, housing accommodations, helping with legal protective orders, and assistance with reporting the incident to police or other UB officials if you so choose. Please contact UB's Title IX Coordinator at 716-645-2266 for more information. For confidential assistance, you may also contact a Crisis Services Campus Advocate at 716-796-4399.

6.3. Mental Health

As a student you may experience a range of issues that can cause barriers to learning or reduce your ability to participate in daily activities. These might include strained relationships, anxiety, high levels of stress, alcohol/drug problems, feeling down, health concerns, or unwanted sexual experiences. Counseling, Health Services, and Health Promotion are here to help with these or other issues you may experience. You can learn more about these programs and services by contacting:

Counseling Services:
- 120 Richmond Quad (North Campus), 716-645-2720
- 202 Michael Hall (South Campus), 716-829-5800

Health Services:
- 4350 Maple Rd, Amherst, NY 14226, 716-829-3316

Health Promotion:
- 114 Student Union (North Campus), 716-645-2837

7. Diversity

The UB School of Engineering and Applied Sciences considers the diversity of its students, faculty, and staff to be a strength, critical to our success. We are committed to providing a safe space and a culture of mutual respect and inclusiveness for all. We believe a community of faculty, students, and staff who bring diverse life experiences and perspectives leads to a superior working environment, and we welcome differences in race, ethnicity, gender, age, religion, language, intellectual and physical ability, sexual orientation, gender identity, socioeconomic status, and veteran status.