University at Buffalo
Department of Computer Science & Engineering
338 Davis Hall – (716) 645-3180

Syllabus

Please read this sheet carefully, and save it for future reference.

The syllabus is subject to change based on the needs of the course and will be communicated with you as appropriate.

Instructors

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

Course Information

CSE 250 Introduction to Data Structures – 4 credits

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 Intesive Computing Certificate students only

Course 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.

Course 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.

Textbooks

- "Object-Orientation, Abstraction, and Data Structures Using Scala, 2nd Edition." by M. Lewis, L. Lacher (ISBN: 978-1498732161)
    - Required by Feb 6
- "Scala Cookbook" by A. Alexander (ISBN: 978-1449339616)
    - Many chapters available freely online: http://scalacookbook.com/
- "Functional Programming in Scala" by Paul Chiusano, Runar Bjarnason (ISBN: 978-1617290657)

Computing 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.

Assignments

Homework assignments for this course will be composed of:

- Programming Assignments
- Written Assignments

There are two categories of assignments which will be given alongside one another. Written assignments will generally be given on a shorter basis, with around 1 week to complete. Programming assignments will generally be longer and assigned over a larger time period, with around 2 weeks to complete. It is possible that not completing one assignment may disqualify you from completing another related one (for example, a programming assignment may ask you to implement something and then a follow-up written assignment may ask you to explain and analyze your code but if you do not have code, you cannot complete this work). Due to the nature of the content of the course, you may be required to analyze code that you have written, in addition to providing correct solutions to the problems. This will likely be separated between the written and programming assignment, though there is no requirement that this is the case. Expectations will be clearly noted when the assignments come out as well as the duration of the assignment. Please pay attention to the amount of time that each assignment provides and begin early. There will be roughly 4 written assignments and 4 programming assignments, though the exact number may vary slightly. **You must score a 100% on the Academic Integrity assignment or you will be given an F in the course.**

For homework assignments, we will only be accepting electronic submissions. These will be accepted via Autolab. Assignments will be submitted as described in the write-up. Written submissions are generally accepted in the form of a PDF. There are two ways to complete your written homework:

- 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.
- You may handwrite your written submission and then scan it. An easy way to scan is using your phone along with the Scan feature in Google Docs (for Android). If you prefer to scan your documents, you may do so with your own scanner or on campus, the libraries provide scanning services.

Make sure to double check your assignments before and after submission to ensure that part of your writing wasn't chopped off or distorted, as **the integrity of your submission is your responsibility**. Also, if you handwrite an assignment, make sure that you write legibly. **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 total points. For example, if the homework is worth 100 points and you submit it one day late, you will receive the maximum of your score earned minus 50 points and 0 points. **No grace days may be applied to written assignments.**

Late Policy for Programming Assignments

The policy for submissions on programming assignments is as follows. You will always be graded on your **latest submission**. If your latest submission was submitted:

- On or before the deadline: 100% of what you earn.
- Late penalty per day (per 24 hours): 25% point penalty (lower your earned points by 25% of the possible points for the **entire** assignment).
- >2 days late: 0 points.

From this policy, you will see that all assignments must be submitted within two days past the assigned deadline. Each submission receives a grade based on the date it was submitted. Your final score is the score earned by your most recent submission. For assignments with multiple component submissions, only 1 penalty will be assessed based on the file submitted latest. If a staggered deadline is given (e.g., two components due one week apart), the earlier deadline will be a hard deadline and no late submissions will be accepted for the first component.

You will have the ability to use three grace days throughout the semester, and at most two per programming assignment (since assignments are not accepted beyond two days late). Using a grace day will negate the 25% point penalty for one day of late submission, but will not allow you to submit more than two days late.

Please plan accordingly. **You will not be able to recover a grace day if you decide to work late and your score was not sufficiently higher**. Grace days are automatically applied to the first instances of late submissions, and are non-refundable. For example, if an assignment is due on a Friday and you make a submission on Saturday, you will automatically use a grace day, regardless of whether you perform better or not. Be sure to test your code before submitting, especially with late submissions, in order to avoid wasting grace days.

Keep track of the time if you are working up until the deadline. Submissions become late after the set deadline, even by 1 minute. Keep in mind that submissions will close 48 hours after the original deadline and you will no longer be able to submit your code after that time.

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.

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.

Exams

There will be one in-class 45-minute midterm exam and one 3-hour final exam. The midterm is worth 20% of your grade. The final exam is worth 40% of your grade. If you have taken the midterm exam and score higher on the final exam, then the final exam score will replace your midterm score. We reserve the right to change the scaling of the exams.

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

Grading Policy

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
Assignments	30%
Attendance	10%
Midterm Exam	20%
*Final Exam	40%

**The final exam grade will replace the midterm grade if it is higher.*

Percentage	Letter grade	Percentage	Letter grade
90 - 100	A	65-69	C+
85 - 89	A-	60-64	C
80 - 84	B+	55-59	C-
75-79	B	50-54	D
70-74	B-	0-49	F

Regrading

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.***

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 **Thursday, March 2, 2023**.

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

What Constitutes a Violation of Academic Integrity?

These bullets should be obvious things not to do (but commonly occur):

- Turning in your friend's code/write-up (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 forums asking someone to solve the problem. Note: Aggregating every [stack overflow answer|result from google|other source] because you "understand it" will likely result in full credit on assignments (if you aren't caught) and then failure on every exam. Exams don't test if you know how to use Google, but rather test your understanding (i.e., can you understand the problems to arrive at a solution on your own). Also, other students are likely doing the same thing and then you will be wondering why 10 people that you don't know have your solution.

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.

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.**

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 tour of Scala (https://docs.scala-lang.org/tour/tour-of-scala.html)
- Any code from Scala Collections (https://docs.scala-lang.org/overviews/collections-2.13/introduction.html)
- Any code from Scala API (https://www.scala-lang.org/api/2.13.0/)

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.

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 me an email **BEFORE THE VIOLATION IS DISCOVERED BY ME.** The email should take the following format:

> Dear 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.

Critical Campus Resources

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.

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.

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

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.