Syllabus

CMP-338: Data Structures & Algorithms

Lehman College, City University of New York

Course Information

Semester	Class Section	Class Hours	Room Number
Fall 2024	01-LEC - 39449	MW 12:00 pm - 1:40 pm	GI 231

Instructor Information

Instructor	Email	Office Number	Office Hours
Steven Fulakeza	steven.fulakeza@lehman.cuny.edu	GI 232	Mon & Wed 1:40 pm - 3:40 pm

---

Course Description: (*4 hours, 4 credits*)

Abstract characterizations of data structures such as arrays, stacks, queues, linked lists, binary search trees, heaps, hash tables and graphs; analysis and implementation of algorithms for sorting, searching, and memory management. Algorithms will be analyzed for their asymptotic behavior in terms of time and space complexity. Implementation issues will be considered and students will write programs that embody these data structures and algorithms.

Prerequisites:

- **CMP 168** - In this course we will do extensive programming in Java. It is assumed that all students are capable of reading and writing object-oriented Java code.
- **CMP 232** - In this course we will demonstrate correctness and efficiency of many of the algorithms presented via mathematical arguments and proofs. It is assumed that students are capable of following the logic of such when presented.

Pre or Corequisites:

- CMP 269

Course Objectives:

At the end of the course students should:

1. Improve skills in object-oriented programming
2. Improve understanding of recursive methods
3. Understand a core group of basic data structures as enumerated in topics below
4. Be able to conceptualize many programming issues at a higher level through data structures
5. Know the tradeoffs of each studied data structure so as to employ the appropriate one for a given situation
6. Be able to write parameterized data structures using generics
7. Be able to design algorithms that incorporate data structures for efficient handling of data
8. Be able to code algorithms involving data structures using an object oriented programming language
9. Be able to analyze new data structures and their algorithms for asymptotic behavior
10. Achieve a level of maturity in the subject so that further study of data structures can be pursued independently

For each algorithm, verification of its correctness and analysis of its efficiency will be considered.

Expectations:

Students will be expected to do extensive programming in Java. It is assumed that at the start of this course, all students are capable of reading and writing object-oriented Java code. Students are expected to learn the material covered in class, the material in the textbook and other assigned readings. Completing homework is an essential part of the learning experience. Students should review topics from prior courses as needed using old notes and books.

Honor Code:

You are encouraged to work together on discussing and planning the overall design of the programs and homework. However, for specific programs and homework assignments, all code written must be your own creation. All submissions must be your own independent work. You are responsible for knowing and following Lehman's academic integrity code (available from the Undergraduate Bulletin, Graduate Bulletin, Office of Academic Standards and Evaluations, or the Smart Catalog). All incidents of cheating will be reported to the Vice President of Student Affairs.

You must complete all your homework assignments, projects, labs, and exams individually. This presents the best way to learn and absorb the material. Feel free to consult the textbook, the instructor, and the CS Tutoring Lab to help you solve problems.

You are allowed to discuss problems with classmates, but only in general terms, and you must specifically avoid discussing any solutions. If we find you plagiarizing and cheating, we will not accept "I didn't know" as an excuse.

You must also resist the urge to copy code from the web. Therefore, students in this course may not use any websites that enable cheating, such as by uploading or downloading material for this purpose. Use of these websites including uploading proprietary materials constitutes a violation of the academic integrity policy. Below are some sources which you are not allowed to use or even consult while taking this class:

- https://www.chegg.com/
- https://www.coursehero.com/
- https://www.bartleby.com/
- https://www.answersaccess.com/
- https://stackoverflow.com/
- Code generation tools (AI-based or otherwise), such as Github CoPilot, ChatGPT, Google Bard
- Submissions or work in progress from other students, past or present, in full or in part
- Solutions or solution fragments you may find online or elsewhere.
- Course materials from previous semesters, regardless of how you obtained them.
- Outside tutors or "work-for-hire" services.
- This list is not exhaustive; ask the instructor before using any source not explicitly listed above.

A note for students retaking the class: resubmitting or reusing your work from earlier attempts at the class is also considered plagiarism. You must do all the work from scratch, otherwise you will not learn as much as you should.

Posting, sharing or making available your own solutions (in full or in part) or any course materials such as homeworks, tests, quizzes, projects, exams, solutions is also a violation of our academic integrity policy. This extends even after the semester ends.

Communication:

We will be communicating with you on a regular basis throughout the semester using **Blackboard** for this course. You are required to make sure that the email address on Blackboard is your current Lehman email address and you must check it on a regular basis. **There will be no acceptable excuse for missing an announcement**.

Grading Policy:

The grading for the course will be based on:

- Participation Activities From zyBooks Textbook: 10%
- Homework: 30%
- Midterm Exam: 30%
- Final Exam: 30%

Homework:

Programming assignments are due most weeks. Assignments will be submitted via your zyBooks textbook. These programming problems reinforce concepts covered in class. To receive full credit for a program, it must be completed by the specified due date and the program must perform correctly. You will be allowed to submit your solution multiple times; the submission with the highest grade will count as your grade. All homework assignments have a deadline, **No late homework will be accepted.**

Also, Participation and Challenge Activities via the online textbook zyBooks will be assigned for every topic covered in class. Your Participation Activities are submitted by answering the questions in your zyBooks as you read each chapter. Completion of these activities is expected by the specified due date.

All your assignments: Participation and Homework have strict due dates. Late assignments are **NEVER** accepted.

All assignments allow for unlimited attempts to submit via zyBooks prior to the deadline. For each assignment, the highest earned submission score will be recorded.

Grading Scale:

Letter Grade	Ranges %
A	93 - 100
A-	90 - <93
B+	87 - <90
B	83 - <87
B-	80 - <83
C+	77 - <80
C	73 - <77
C-	70 - <73
D	60 - <70
F	< 60

Exams:

- Midterm Exam date: 10/30/2024 (During class time)
- The Final Exam will be on 12/16/2024 from 11:30 am to 01:30 pm

The final exam is comprehensive. Since the final exam is comprehensive, if you do better on the final exam than the midterm exam, the final grade can replace the midterm grade. This will be done automatically when your final grade is calculated. **Please note that there are no make-up exams. Note: Missed final exam = Unofficial Withdraw (WU).**

Exams will be done in person. No notes or electronics (laptops, calculators, tablets, phones, smart watches, etc.) will be allowed during exams

The last date to withdraw from a course with a W is November 6th by 11:59 pm.

Materials and Resources:

Textbook: https://learn.zybooks.com:

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code: **CUNYCMP338FULAKEZAFALL2024**
3. Subscribe

Suggested Additional Textbooks:

- Data Abstraction and Problem Solving with Java: Walls and Mirrors by Frank M. Carrano and Janet J. Prichard (3rd Edition) ISBN 978-0-13-212230-6.
- Data Structures & Algorithms by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser (6th Edition) ISBN 978-1-118-77133-4.

Technology:

- Access to personal computers with Eclipse IDE, and JDK 8

Computer Access:

Access to a computer capable of running the necessary software packages will be required. The college has computer laboratories available with the necessary configurations. These machines are for work related to this course only and a code of conduct applies to computer use in the department and on-campus. Misusing university computers could result in losing your computer access for the rest of the term, making it exceedingly difficult to complete this course.

Additional Online Resources:

Oracle Documentation: https://docs.oracle.com/javase/tutorial
Oracle JavaDoc: https://docs.oracle.com/javase/8/docs/api
GitHub Repository https://github.com
Additional Book: Data Structures and Algorithms in Java, 6th Edition Wiley ISBN: 978-1-118-77133-4

Accommodating Disabilities:

Lehman College is committed to providing access to all programs and curricula to all students. Students with disabilities who may require accommodations are encouraged to register with the Office of Student Disability Services located in Shuster Hall, Room 238. http://www.lehman.edu/student-disability-services
Telephone: 718-960-8441 Email: disability.services@lehman.cuny.edu

---

Topics:

Intro Topics:

- Abstract Data Types (ADTs) - Specification and Implementation
- Asymptotic Analysis and Notation: “Big-O”
- Sorting: Merge Sort, Quick Sort, Radix Sort

Linked Lists:

- A simple List ADT
- Implementing a List using Linked Nodes
- Implementing a List using an Array
- Implementation Issues
- The use of dummy nodes

Stacks:

- The Stack ADT
- Array Implementations: Fixed size and resizable.
- Reference Based Implementation
- Comparisons of efficiency for Array and Linked List implementations
- Application: Evaluation of Algebraic Expressions

Queues:

- The Queue ADT
- Circular Array Implementation
- Reference Based Implementation
- Comparing Implementations: Fixed size, resizable

Binary Search Trees:

- Definitions and Properties for Binary Tree and Binary Search Tree
- Implementing Binary Trees using Linked Nodes
- Implementing Binary Trees using Arrays
- Full, Complete, Balanced Binary Trees
- Preorder, Inorder, Postorder tree traversal
- Using a BST to Implement Treesort
- Additional methods for manipulating Binary Tree Data
- Using the definitions to determine correctness of Binary Tree Algorithms

Heaps:

- Definitions of Max-Heaps and Min-Heaps
- Implementing a Heap
- Using a Heap to Implement Heapsort

Graphs:

- Graph ADT and Definitions
- Data Structures and Implementation issues for Graphs
- Graph Traversal: Breadth First and Depth First Traversal (BFS), (DFS)
- Greedy Algorithms
- Shortest Path: Dijkstra's Algorithm
- (Time Permitting: Euler Circuit, Hamilton Circuit)
