# CMP 569 - Data Structures & Algorithms for Educators
# Spring 2024 Syllabus

## Course Information:

| Semester | Class Section | Class Hours | Room Number |
|---|---|---|---|
| Spring 2024 | OA-01 | Online Asynchronous | |

## Instructor Information:

| Instructor | Email | Office Number | Office Hours |
|---|---|---|---|
| Steven Fulakeza | steven.fulakeza@lehman.cuny.edu | GI 232 | Mon & Wed 12:50 pm - 1:50 pm<br>and<br>Mon & Wed 03:50 pm - 4:50 pm |

## Course Description: *3 hours, 3 credits*

Abstract characterizations as well as the design and implementation of data structures such as arrays, stacks, queues, linked lists, binary search trees, heaps, hash tables and graphs along with algorithms that make use of such structures including algorithms for sorting, searching, will be studied. 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.

## Prerequisite:

CMP 568 or departmental permission.

## 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.

## Grading Policy:

- Participation Activities from zyBooks: 10%
- Homework Problems: 30%
- Quizzes: 8%
- Midterm Exam: 22%
- Final Exam: 30%

## 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 |

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).**

## 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 reading. 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.

**Communication:**
We will be communicating with you on a regular basis throughout the semester using the email address listed on 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 email announcement.**

## Homework:

Participation Activities via the online textbook zyBooks will be assigned for every topic covered in class. Completion of these activities is expected by the specified due date.

Programming assignments are due most weeks. Assignments will be submitted to the online textbook zyBooks. 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. **No late homework will be accepted.**

**Materials and Resources:**

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

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code: CUNYCMP569Spring2024
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, JDK 8, Java 8 Documentation

## 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

## Delivery Method

This course will be delivered online in an asynchronous (non-real-time) format using Blackboard. It will be accessible on or before January 25th, 2024. The course is done in an independent study format.

**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)