

Instructor's Course Syllabus

CSC-242-01 – Fall 2025

Data Structures

Course Number: CSC-242-02
Date/Time: M/W 1:00 PM – 2:15 PM
Room: D102A

Instructor: Ali Azhari
Telephone: 617-228-3221
Email: ali.azhari@bhcc.edu

Office Hour: M/W: 2:30 AM - 3:30 OM
T/TH: 10:30 AM – 11:30 AM
Office Location: D-122-F

<https://bhcc.webex.com/join/ali.azhari>

Course Title: Data Structures

COURSE DESCRIPTION:

This course prepares students to understand the fundamentals of data structures with an emphasis on software engineering. Topics include multidimensional arrays, records, dynamic memory allocation, stacks, queues, lists, trees, graphs, and algorithms.

PREREQUISITE:

Writing Skills II (ENG-095), College Algebra-STEM (MAT-194) and Java (CSC-239) with grade C or better or equivalent experience with permission of department chairperson.

ESSENTIAL QUESTIONS:

- What are the types of Data Structures?

- What is the difference between static and dynamic memory?
- Why is it important to think abstractly about concrete things?
- When and how to implement certain sorting algorithms?

PROCEDURES AND OBJECTIVES:

This course is taught with a combination of lecture, quizzes, and individual and group projects. Examples of each programming feature that we will learn will be given in class, and exercises (both theoretical and programming) similar to the ones appearing in the exams will be covered in class.

The classroom will be run in an informal manner, and I encourage students to ask questions if the subject of the lecture is not clear. Remember that if you do not understand something it is probably not clear to many other students as well and should be discussed further. Again -- please ASK QUESTIONS! Try to keep good notes so you can review the examples at a later time. You may record the lecture if that is a useful learning aid to you.

Questions are encouraged during the discussion and the solution to selected problems is presented and analyzed. The lecture will be in interactive style whenever possible.

Homework assignments will be posted on the class Moodle.

STUDENT LEARNING OUTCOMES:

Upon completion of this course students should know and be and feel comfortable using the following material, concepts, and techniques:

1. Understand and apply the OO paradigm and its key concepts: data abstraction, encapsulation, inheritance, and polymorphism.
2. Understand the use of recursion in methods and classic data structures.
3. Understand and implement as Java classes the following classical data structures: lists, stacks, queues, trees, graphs, dictionaries, maps, and hash tables.
4. Understand Sorting and Binary Search Tree.

REQUIRED TEXT

Introduction to JAVA Programming, 12th edition. Author: Y. Daniel Liang Publisher: Pearson, 2020.

ISBN-13: 978-0-13-651935-5

REQUIRED READING OUTSIDE OF TEXT(S): articles (printed and internet-based) as assigned.

SUPPLEMENTARY READING RECOMMENDED BUT NOT REQUIRED: TBD

CONTACT HOURS (WEEKLY): 1 session remotely and one session in-person

ATTENDANCE POLICY:

The instructor considers attendance as a form of class participation. If you miss a class, you are responsible for all the material and assignments presented in class which you should obtain from your fellow classmates.

If you must miss a class, simply email the instructor at ali.azhari@bhcc.edu before the start of class. Be sure that you get the missed material from a classmate.

This is a difficult and highly conceptual course. You cannot expect to pass this course without attending class and doing the assignments. Although this course is taught in hybrid mode,

THIS IS NOT AN ONLINE COURSE! ATTENDING CLASS IS IMPORTANT, EVEN THOUGH CLASS IS both IN-PERSON and REMOTE (HYBRID). You are responsible for everything covered and assigned in class. "I was not there" is NOT an acceptable reason for being unprepared.

GRADING POLICY:

Students are graded on the basis of correctness of concepts as evidenced by oral participation in class, completion of lab exercises and homework, and the degree of success achieved in programming projects and examinations in the following manner:

Midterm	10%
Final Exam and/or Interview	20%
Quizzes	20%
Programming Projects	50%
Total	100%

General Grading Rubric: Data Structures and Algorithms Projects (Total: 100 points)

1. Correctness & Functionality (30 points)

- **30–27 (Excellent):** Program runs correctly on all test cases, produces expected results, no runtime errors.
 - **26–22 (Good):** Minor issues; most functionality correct, occasional errors.
 - **21–15 (Fair):** Partial implementation; significant errors or missing features.
 - **14–0 (Poor):** Program does not run or produces mostly incorrect results.
-

2. Implementation of Data Structures & Algorithms (25 points)

- **25–23 (Excellent):** Appropriate and efficient data structures/algorithms used; demonstrates strong understanding.
 - **22–18 (Good):** Mostly appropriate choices, some inefficiencies.
 - **17–12 (Fair):** Limited or somewhat incorrect use of structures/algorithms.
 - **11–0 (Poor):** Incorrect or missing implementations.
-

3. Efficiency & Complexity Analysis (15 points)

- **15–14 (Excellent):** Code is efficient, avoids redundancy, correct use of time/space complexity, Big-O analysis included.
 - **13–11 (Good):** Mostly efficient, minor inefficiencies; complexity analysis mostly correct.
 - **10–7 (Fair):** Inefficient approach; weak or missing analysis.
 - **6–0 (Poor):** No attention to efficiency or complexity.
-

4. Code Quality & Style (15 points)

- **15–14 (Excellent):** Clean, readable, consistent formatting, meaningful variable/method names, proper documentation.
 - **13–11 (Good):** Mostly clear; some formatting or naming issues.
 - **10–7 (Fair):** Readability problems; limited documentation.
 - **6–0 (Poor):** Unreadable, messy, no documentation.
-

5. Testing & Validation (10 points)

- **10–9 (Excellent):** Comprehensive test cases (normal, edge, extreme inputs) provided and explained.
 - **8–7 (Good):** Adequate testing; some cases missing.
 - **6–4 (Fair):** Limited testing; only basic cases.
 - **3–0 (Poor):** No testing or validation.
-

6. Report / Presentation / Explanation (5 points)

- **5 (Excellent):** Clear report or in-class explanation of design, approach, and results.
 - **4 (Good):** Adequate explanation, minor gaps.
 - **3–2 (Fair):** Weak or unclear explanation.
 - **1–0 (Poor):** No report or explanation provided.
-

Total = 100 points

DUE DATES: If you cannot meet a due date, it is expected that you will discuss a plan of action toward completion with the instructor.

STORAGE MEDIA: You will want at least a 4GB flash drive to store your projects, or be prepared to use your OneDrive account (free) or a Google Drive account or GitHub to store your work. You will be sharing your projects with the instructor and demonstrating in class, so you must be able to access them!

PROFESSIONAL BEHAVIOR: You are reminded that the college expects each student to exhibit the virtues of honesty and integrity. Students who violate this policy by submitting the work of another as their own or who collaborate on examinations or programming assignments will be required to redo the work or take an alternate assessment. Continued offenses will be referred to the Dean of Science, Engineering, and Mathematics for possible additional disciplinary action.

GROUP PROJECTS: Working as a part of a team is an important employment and life skill. *Group projects, if any, are designed to help you learn some of the outcomes for the course.* A group project is a collaborative effort - and you will each be responsible for learning the key outcomes for the project. After the project is finished, there will be an

assessment of these learning outcomes. Your project grade is based on the assessment, not on the group project itself.

DISABILITY SUPPORT SERVICES: The disability support services office is a student-focused department dedicated to assisting members of the BHCC community with documented disabilities. Students may be eligible for services that include testing and classroom accommodation. For more information or to request an accommodation, contact the disability support services office at disabilitysupport@bhcc.edu or 617-228-2327. Students are encouraged to request accommodations as early as possible, ideally before the start of the semester. For information about programs and services please visit <https://www.bhcc.edu/disabilitysupportservices>.

OTHER INFORMATION

MEETING ETIQUETTE: Please sign in on time. If you are not going to turn on your video camera, then please make sure your full name is posted. Please make sure your microphone is working but muted when you enter the meeting space. A webcam is not required, but a working microphone will be very helpful. If you share your workspace with others, a set of noise-canceling headphones is recommended.

COMPILER: In class we will be using either the Visual Studio Code, IntelliJ or the NetBeans IDE. If you prefer another IDE you may, but all projects must be turned in as requested by the instructor. It is important that you organize your disk (create a folder for each homework project, containing all related files, including input and output).

COMPUTER WORK: There probably going to be both individual and group projects in this class. For projects you will be working in groups of three to five, with a group leader simulating a programming team with a project manager. The group is expected to meet frequently and keep a detailed log of the activities of the group and each member. (A good way to do this is using an app such as Slack or Discord.) More details about the inner workings of the groups will be given in class if and when there is a group project.

Important: Each student is expected to work on his/her own for the **individual projects**, and not as a team with other students. Students found to have copied from each other or worked side-by-side to the extent of handing in the same result will have to redo the assignment or be given an alternative assessment. **See the details under the cheating section.**

It is important to spend time in front of a computer trying to solve as many of the book's exercises and laboratories, even the ones that are not assigned as official homework. It is highly recommended that you read the corresponding chapters from the textbook before class and before attempting the assignments. This is far more detailed and contains more step-by-step examples.

COURSE REQUIREMENTS:

To successfully complete this course and receive a grade you must:

- Read all assigned material before class
- Do the quizzes/ assignments
- Have a passing grade on and turn midterm and final exam/interview
- Have a passing grade on all individual projects.

EXAMINATIONS: Exams will be projects or tests of concepts. The final project will cover all course material and is due and will be presented **Dec 18, 2025**.

- All class lecture material, homework assignments, and lab projects will be considered as "fair game" for assessment purposes. You are expected to take the exams/assessments at the time scheduled unless prior arrangements have been made.

Reading: It is strongly recommended that you read thoroughly all material in the text, in particular any and all pages assigned for reading **before** the class. Do not attempt to rely solely on material covered in classroom lectures. There is no substitute for time spent outside of class reading the text for a thorough understanding of every topic presented in class. Please refer to the course schedule for chapters to be covered in class.

Office hours/getting assistance: You must make an appointment to meet with me using the tool in the Moodle. Office hours are to be posted. Or you can e-mail me at ali.azhari@bhcc.edu to set up another appointment.

CHEATING:

Policy regarding cheating: I feel **very** strongly about cheating. Individuals caught cheating (copying/sharing work / turning in work that is not their own) **will get a zero**. Individual projects are individual work. You are expected to think for yourself and come up with your own solutions to problems and programs.

Contract Cheating: "The term contract cheating describes the form of academic dishonesty where students get academic work completed on their behalf, which

they then submit for academic credit as if they created it themselves."
(<http://www.contractcheating.com>).

Any assignments found posted to sites such as Chegg.com, CourseHero.com, FreeLancer.com, or other sites that purport to be tutoring sites, but in fact support contract cheating will be investigated, and the student posting the assignment, and students downloading the "expert answer" can expect both of the following consequences:

- A replacement assignment/assessment to be done with camera on during scheduled time with the instructor.
- Referral to the Dean of Students office for academic discipline

Any students found sharing the answers to an assessment or test (paid for or not) with another student can expect both of the following consequences:

- A replacement assignment/assessment to be done with camera on during scheduled time with the instructor.
- Referral to the Dean of Students office for academic discipline

Note: if your final project/exam is posted to one of these sites or shared with another student, you will receive a failing grade for the project/exam and mathematically would not be able to pass the course. Your grade would possibly be changed after final grades were posted, depending on the length of time of the investigation.

ChatGPT and other AI: Using tools that are available to you is expected but you **MUST** reference the use of any AI tool that you have used by including the AI used, the query used, and the AI's response, as well as how you have modified the response for your use.

Always evaluate the AI response - as of this writing, the AI is not 100% accurate, and often contains significantly incorrect information in code, explanation, or both. The incorrectness can be subtle!

When allowed (**and ONLY when allowed**), you may use generative AI to complete assignments under the following conditions:

1. You must clearly identify the use of AI-based tools in your work. Any work that utilizes AI-based tools must be clearly marked as such, including the specific tool(s) used. For example, if you use ChatGPT-4, you must cite "ChatGPT-4. (YYYY, Month DD of query). "Text of your query." Generated using OpenAI. <https://chat.openai.com/>"
2. You must be transparent in how you used the AI-based tool, including what work is your original contribution. An AI detector such as GPTZero (<https://gptzero.me/>) may be used to detect AI-driven work.

3. You must ensure your use of AI-based tools does not violate any copyright or intellectual property laws.
4. You must not use AI-based tools to plagiarize - you may not submit AI-generated work as your own.
5. Total AI content must not make up more than 5% of the assignment.

To be completely clear, you may use AI to help you complete Reading Assignments (assignments graded as 'homework'). You may not use AI to complete quizzes, assessments, tests, discussions, reflections, or labs, or learning community assignments.

Internet research: It is expected that you will research solutions to problems on the internet, and you may find code that does exactly what you need. If you understand the code, you should be able to write your own code that accomplishes the task at hand. However, if you do use someone else's code, you **MUST** give credit to the original author/website, **and** you **MUST** give a detailed explanation of the code. Please note: copy and pasting are cheating.

If I can use the internet for research, what is cheating? Simply put, cheating is turning in work that **is** not your own as if it is your own. Copying open source code and turning it in as yours is plagiarism, which is cheating. Copying open source code (or pseudocode or flowchart) and editing it to change the variable or function names and turning it as yours is also plagiarism, which is cheating. Typing someone else's code in is still copying, and is still cheating. And just to be clear, paying someone to write your code for you is also cheating. **GITHUB and StackOverflow:** GITHUB is an important tool that you should be using to build your portfolio, but it is too easy to copy and paste other's work and pass it off as your own. Similarly, many questions can be answered through sites such as StackOverflow. It is good practice to review these sites, but do not plagiarize programs! Plagiarism includes, but is not limited to copying another programmer's code, and changing the variables, class names, and/or method names. If you do not write your own code, you will not learn the language. You will simply learn to copy. You will be caught, and you will have to do far more work, with a lot less time to get it done..

You may use GitHub to store your programs, but all programs must be turned in as requested by the instructor.

Notes/Tips: The DUE date is not the DO date! Begin your work in each assignment the day you receive it. Failure to do so will, in all likelihood, force you into a situation where

time constraints will become a severe problem. That will reduce your ability to design and implement clean code. It is unlikely that I will be able to answer complex code questions the night before the assignment is due.

This course tends to be more time consuming than many. Since it is a programming course, understand that debugging can take a long time. Expect to spend on average 12 hours a week outside of class.

If at any point during the course you find yourself falling behind, DO NOT WAIT. Get tutoring help. Come and see me immediately and ask for help. Ask your classmates for help. It is difficult to catch up when you fall behind. Do not panic. Do not give up. Just get help. Try the office hours and/or the TASC.

Date	Week	Topics	Major Assignments	Reading BEFORE Class
09/03	1	Setting up tools (IDE, GitHub...etc.), Getting started with Java, Review		Chapters 1, 2, 3, 4, 5, 6, 7,
09/08 09/10	2	More Review of programming principles OO design, fundamental data structures		Chapters 8, 9, 10, 11, 19
09/15 09/17	3	Array and ArrayList	Indiv. Project 1 Due Quiz1	Chapter 7, 20
09/22 09/24	4	LinkedList		Chapter 10
09/29 10/01	5	List / Stack	Indiv. Project 2 Due	Chapter 11
10/06 10/08	6	List / Queue		Chapter 14
10/13 10/15	7	Priority Queues	Indiv. Project 3 Due Quiz 2	Chapter 15
10/20 10/22	8	Review Midterm	Midterm	Chapter 8 – Chapter 15
10/27 10/29	9	Sets and Maps	Indiv. Project 4 Due	Chapter 16

11/03 11/05	10	Trees	Quiz 3	Chapter 12
11/10 11/12	11	Sorting / Searching Algorithms		Chapter 13
11/17 11/19	12	Binary Search Tree	Practice	
11/24 11/26	13	Graphs		Chapters 17 & 18
12/01 12/03	14	Graphs	Quiz 4	Chapter 19
12/08 05/10	15	Final Exam, Interview	Indiv. Project 5 Due	
12/15 12/17		Final correction		
Schedule & assignments subject to change at the instructor's discretion.				