

COMP 410 – Data Structures



Lectures: MW 8h00–9h15, at SN 014

Instructor: Sanjoy Baruah (baruah@cs.unc.edu). **Office:** FB-134. **Office Hours:** Immediately after class, or by appointment

Teaching assistant: TBD

Text: Mark Allen Weiss, *Data Structures and Algorithm Analysis in Java*, Third Edition.

Prerequisites:

- COMP 401 (Foundations of Programming)
- An ability to program in Java — if you do not already have this ability, it is your responsibility to obtain it ASAP.
- Basic math: exponentiation and logarithms; series sums; familiarity with proof techniques (including proof by induction).

Course Description: We will study several widely-used data structures from two perspectives: how to use them, and how to implement them in an efficient manner. We will learn terminology for expressing the efficiency of implementations, and techniques for evaluating algorithms to determine their efficiency. We will implement these data structures using the Java programming language. We will learn how we choose data structures to achieve desired programming goals, and how to design and implement new data structures if necessary.

Target Audience: A sound knowledge of data structures is a basic competency in computer science and a prerequisite to a deeper understanding of most topics in computer science. This course is therefore a prerequisite for many higher-level courses in computer science, and is hence targeted at students that intend to take such courses.

This course assumes that the students have some programming experience, and adequate background in mathematical concepts and formal reasoning.

Goals and Key Learning Objectives: Upon completing this course satisfactorily, you will be able to use a wide range of common data structures in an appropriate manner. You will be able to identify the data structure that best matches your needs for a particular programming context, and to design and implement your own if necessary. You will be familiar with the concepts and terminology that are used to describe and compare the efficiency of algorithms that implement data structures.

Course requirements:

- Several (3-5) homework assignments, each of which includes a programming component in which you implement some data structure and evaluate your implementation, and an analytical component in which you answer some questions regarding your implementation. These programming assignments are to be done individually by each student — no collaboration is permitted.
- Several closed-book exams that are administered in class, testing your knowledge of the material being covered in the lectures.
- A comprehensive final exam. This, too, will be closed-book.

Grading criteria: The breakup of grades is tentatively set as follows

- Several homework assignments: 30%
- Several in-class exams: 40%
- Final exam: 30%

Class Participation: This class will be far more enjoyable for everyone if all students come to class ready and willing to discuss the material to be covered. I plan to reward those who participate in class by increasing their final grade by up to half a letter grade. I also reserve the right to add a similar negative “reward” to those who do not observe appropriate etiquette in class.

Course Policies:

- **Assignments:** No collaboration is permitted upon assignments. Please do not accept anyone's assistance (other than the instructor's or the TA's) - all submitted work must be your own, and you must include a signed honor statement with each submission explicitly stating that all submitted work is the result of your own effort only. Each assignment will contain submission instructions; assignments are due in class on the date specified. Late assignments are not accepted.
- **Grading policy:** Assignments are graded by the TA under the instructor's supervision. Graded assignments are returned in class. If you wish to dispute the score assigned to you, it is your responsibility to initiate negotiation via email on this, within one week of the date that the assignments are returned in class. No complaints will be considered after this one-week deadline.
- **Class Etiquette:** You are expected to maintain proper etiquette in class. This includes:
 - Not making a habit of arriving late, or leaving in the midst of class. If you must be late once or twice, take an aisle seat quietly; likewise if you must leave early. If this becomes habitual, you should drop the course.
 - Keeping cell-phones, pagers, etc. off during class. Not talking in class. Private discussion between students, even whispers, carry surprisingly well and are a real distraction to those seating near you and to the instructor.
 - Not using your laptop to browse the web

We will try to be courteous to you and we ask that you be courteous to others as well. Thank you.

- The **course final** is given in compliance with UNC final exam regulations and according to the UNC Final Exam calendar. It will be held during the 8am–noon time-slot on **Monday, May 5th** .

Honor Code: No collaboration is permitted in assignments or exams. All exams are closed-book; all code submitted in assignments must be the students own. No code obtained from other sources may be used without the explicit (email or written) permission of the instructor or TA.

Collaboration in assignments or exams, or the use of code not the students own, constitutes an honor code violation.

The Honor Code and the Campus Code are in effect for this course. The following is adapted from a memo from the Chancellor.

The Honor Code prohibits lying, cheating, or stealing when these actions involve academic processes or University, students, or academic personnel acting in an official capacity. The Campus Code requires students to conduct themselves so as not to impair significantly the welfare or the educational opportunities of others in the University community. As a student at UNC-CH, you have accepted a commitment to the Honor Code and the Campus Code, and the principles of academic integrity, personal honesty, and responsibility on which they were founded more than 100 years ago.

Academic dishonesty in any form is unacceptable, because it circumvents the purpose of the University. The instructor and teaching assistant have a responsibility to report any possible Honor Code violations to the Student Attorney General. Please join us in supporting the Honor Code by signing the Honor Pledge on all written work, and consult us if you are uncertain about your responsibilities within this specific course.

Disclaimer: The professor reserves the right to make changes to the syllabus, including project due dates and test dates. These changes will be announced as early as possible.

TOPICS TO BE COVERED

- Course introduction: **abstraction** and the trade-off between *correctness* and *efficiency*
- Linear data structures — stacks and queues
 - Implementation using arrays and linked lists
- A brief introduction to run-time analysis: Big-Oh notation
 - Illustration via sorting (insertion/ bubble/ /merge-sort) and searching
- The priority queue ADT
 - Implementation using heaps
 - Heapsort
- Some more sorting
 - quicksort
 - linear-time sorts (bucket sort and radix sort)
 - external sorting
- Dynamic Dictionaries
 - binary search trees (BSTs),
 - balanced BSTs,
 - hash tables
- Disjoint set union-find
- Graphs representation; top-sort