Data Structures Syllabus

CSCI-UA.0102-40

Jan 19th 2026

Lecture

- **Time**: Monday / Wednesday 11:00am - 12:15pm
- **Location**: Room 317. Warren Weaver Hall (WWH), 251 Mercer St.
- **Instructor**: Romain Cosson (rc6142@nyu.edu)
- **Office Hours**: Wednesday 9:00-10:00am, (Room 412 WWH)

Recitation

- **Time**: Friday 9:30-10:45am
- **Location**: Room LL138. Bobst Library.
- **Recitation Leader**: Kevin Peter (kevin.peter@nyu.edu)
- **Office Hours**: TBD

Tutoring

Tutoring for all is organized at the department level for all CS 102 courses. Details will be provided later.

Prerequisite

A grade of C or higher at Introduction to Computer Science (CSCI-UA 101) or a successful placement exam. Contact your advisor if you need to switch to a CS101 class.

Topics

The goal of this class is to understand the design of data structures, and their role to obtain efficient and reliable algorithms. Topics include:

- Abstract data types (lists; stacks; queues; priority queues; dictionaries)
- Implementations (arrays; Linked lists; doubly linked lists; resizable arrays; binary heaps; binary search trees; AVL trees;)
- Algorithms (sorting; searching; recursion)
- Java and object oriented programming (encapsulation; types; inheritance; polymorphism)

Students who are already familiar with the material taught in this course can test out of the course.

Material

The class is supported by three components:

- **Slides**: Used to cover the material for the class (theory).
- **Whiteboard**: Used to go through classic examples and exercises (practice).
- **Code**: Used to implement algorithms seen in class (code).

Much of the material used in this class has been graciously provided by Prof. Evan Korth and Prof. Joanna Klukowska, who have been teaching the topic for years.

More material

- **Books**: *Data Structures and Algorithms in Java, 6th edition, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser* (recommended) and *Data Structures and Algorithm Analysis in Java (third edition), Mark Allen Weiss* (optional).
- **Coding environment**: All programming work in this course will be done in Java. We recommend using Visual Studio Code (VS Code) rather than Eclipse or other IDEs. install the Java Development Kit (JDK) from the official Oracle website and verify the installation by running `java -version` and `javac -version` from a terminal. Configure your workspace so that VS Code recognizes the installed JDK.
- **Brightspace**: Brightspace will be used to post class and recitation content, exercises, grades, and announcements. There will be two distinct Brightspace websites: one for the lecture and one for the recitations.

Assignments: Homework, Quizzes, and Exams

- **Exams**: There will be one midterm and one final. The midterm will take place during class. The final exam will take place during finals week. Time and location will be announced via Brightspace. You should come to me at the beginning of the semester if you have foreseeable constraints (e.g., observed religious holiday) that would prevent you from taking the midterm on some date. If you must miss the midterm or final for an emergency reason (e.g., medical), contact me before the start of the exam.
- **Quiz**: Quizzes will consist in a (10-15) minutes evaluation on paper, usually at the end of recitations. There will usually be a mock quiz at the end of the Wednesday lecture. There will be 5 to 10 quizzes in the semester.
- **Homework**: Homeworks will consist of (mostly) programming assignments, which will be submitted via Brightspace usually on Fridays. There will be 5 to 10 homeworks. You are allowed one late homework (4 days max) with no justification needed. Homeworks will be presented and initiated during the recitations. Students are also expected to complete the weekly exercise sessions at home.

Collaboration policy and AI

See the Computer Science Department's Academic Integrity statement. Oral collaboration between the students is permitted (and encouraged) but code should be your own. The use of AI is allowed to help you understand concepts covered in class, and even to help you if you are stuck with your homework. But your code should not be copy-pasted from a conversation with AI. During the semester, the recitation leader or I will/may test you by asking you questions about your code. You are expected to be able to explain every line of code that you write.

Grading

The overall grade (G) will be a weighted combination of final (F) midterm (M) quizzes (Q) and homeworks (H). The **tentative** weights are:

$$G = 0.4F + 0.3M + 0.2Q + 0.1H.$$

Conclusion

Your ideas to to promote a better understanding and learning in class are welcome. Fell free to ask questions, to engage in discussions.