Syllabus

COSC 211: Data Structures

Will Rosenbaum
Updated: February 6, 2022

Department of Computer Science
Amherst College

1 Basic Information

Instructor Will Rosenbaum, SCCE C216

Lectures T/Th 1:00–2:20, SCCE A131

Office Hours TBD

Course Website willrosenbaum.com/teaching/2022s-cosc-211/

2 Official Course Description

A fundamental problem in computer science is that of organizing data so that it can be used effectively. This course introduces basic data structures and their applications. Major themes are the importance of abstraction in program design and the separation of specification and implementation. Program correctness and algorithm complexity are also considered. Data structures for lists, stacks, queues, dictionaries, sets, and graphs are discussed. This course will provide advanced programming experience.

3 Expected Background

You should have significant experience programming in an object oriented language, such as Java (preferred), C++, or Python. In particular, you should be comfortable with objects, defining and using classes, object inheritance, and polymorphism. This course will be taught in the Java language. While prior experience with Java is not required for the course, you should be prepared to port your knowledge of another language to Java.

4 Course Aims

In this course, you will experience the full life-cycle of problems in computer science from formal task specification to implementation, application, and execution. Emphasis will be given to fundamental problems of organizing and manipulating data. Given a computational task, you will learn to

1. formally specify the task as an *abstract data type* (ADT),
2. devise a *data structure* that represents the ADT,
3. reason about the correctness and performance of the operations supported by the data structure,
4. implement the data structure in the Java programming language, and
5. measure the performance of your implementation.

Thus, you will learn to synthesize mathematical reasoning, programming, and empirical analysis to solve fundamental problems in computer science. By analyzing many different data structures, you will learn to identify which data structure is appropriate for a given application.

5 Resources

There is no required text for this course. Much of the material we will cover is contained in the following texts:

- *Open Data Structures* by Pat Morin, freely available at https://opendatastructures.org. *Recommended text.*
- *Data Structures and Algorithms in JAVA* by Goodrich and Tamassia. Covers much of the material in the course, and includes a concise introduction/review of relevant features of the Java programming language. 2 copies on reserve at Keefe Science Library. (If you wish to purchase this text, note that used copies of older editions are significantly less expensive than the newest edition.)
- *Data Structures and Algorithms* by Aho, Hopcroft, and Ullman. Classic, but dense and somewhat outdated text. 1 copy on reserve at Keefe Science Library.

Links to additional resources and lecture notes will be posted to the course Moodle site.

6 Coursework and Evaluation

Coursework will consist of weekly coding assignments, weekly quizzes, and exams. You will also be expected to actively engage in discussions during lectures. Final grades will take into account performance in three categories:

Coding Assignments. You will have weekly programming assignments. Typically, these assignments will ask you to implement and test a data structure, or apply a data structure to solve a problem. Most coding assignments will be completed individually, though some assignments may ask you to work in small groups. Assignments may also include extensions that may be completed for additional credit. *Each coding assignment will be due on Thursday by 5pm on the week after it is assigned.* Submission will be through Moodle.

Quizzes and Exams. You will have have weekly short quizzes, typically consisting of one or two conceptual questions. Additionally there will be two midterm exams during the semester. All quizzes and midterm exams will be take-home assignments with no time limit. You should expect to spend at most 30 minutes to complete each quiz, and 2 hours to complete the midterm exams. There will also be a take-home final exam during finals week. *All quizzes and midterm exams will be handed out at the end of class on Tuesday and due at the beginning of the following class on Thursday.*

7 Topics and Approximate Schedule

The course material is divided into four conceptual modules, each of which will take approximately three weeks.

Part 1 ADTs and Linear Data Structures (Weeks 1–4)

- Abstract data types and program specification (stacks, queues, lists, sets, maps, priority queues)
- Arrays, linked lists, and variations
- Analysis of programs: correctness and runtime

Part 2 Tree-based Data Structures (Weeks 5–7)

- (Binary) search trees
- Heaps
- Balanced trees

Part 3 Randomized Data Structures (Weeks 8–10)

- Probability basics
- Skip lists
- Hash tables

Part 4 Selected Topics and Applications (Weeks 11–13)

- Graphs (adjacency lists and adjacency matrices)
- Data compression and Huffman codes
- Data hashing and sketching

Midterm examinations will be scheduled for Week 05 (March 7–11) and Week 10 (April 18–22).

8 Course Structure and Support

The class will meet for two 80 minute periods each week. This time will be used for a combination of lecture, discussion, and small group work. While the class is currently planned with in-person meetings, we may need to switch to remote meetings as dictated by the ongoing pandemic. You should plan to attend and particpate in class regularly, though you will not be penalized for missing classes.

You are encouraged to collaborate with your peers, but all submitted work is expected to be yours and and yours alone. For coding assignments, this means you should talk about your work with your peers, but you should not share your code with anyone, nor should you ask anyone else—currently taking the class or otherwise—to show you their code or to directly help you write code. As a heuristic for what constitutes encouraged collaboration, if it is something you can discuss without sharing your computer screen or writing Java syntax, then you are probably alright.

If you are stuck on an assignment or are struggling with the material, there are many opportunities for help:

- ask during lecture;
- start a discussion on Moodle;
- ask a TA during evening TA sessions (schedule TBD);
- ask during the professor's office hours.

Often, if you have a question about an assignment or course material, there will be many other students with similar questions. Thus, it is beneficial to everyone to discuss these questions in a shared venue such as the lecture or discussion forums. For questions that are specific to you or require sharing code (e.g., debugging), you should ask a TA during an evening session or me during my office hours.

If you find that you are struggling with the course and would like more one-on-one help, please contact me about individual tutoring through Amherst's peer tutoring program.

Unsolicited Advice. I have found that often the most productive way to make progress on any problem is to give it time. Step away from your computer. Take a few deep breaths. Get some physical activity, preferably surrounded by trees and/or dogs. Almost without fail, I find that I have a better perspective on things when I get back to work.

9 Course Policies

Class participation. While attendance is not strictly mandatory, you are expected to actively participate in class on a regular basis. Participation includes both preparing for class by engaging with the assigned readings as well as contributing to classroom discussion.

COVID protocols & absences. You are expected to follow college-wide COVID precautions. In particular, you should wear appropriate masks ((K)N-95 or KF-94) at all times in class, and there is no eating or drinking in the classroom. **If you are under isolation/quarantine or are feeling ill, you should not attend class.** Class/lecture notes will be made available so you will have an idea of what was discussed in class.

Collaboration. You are encouraged to collaborate with your peers, but all submitted work is expected to be yours and and yours alone. For coding assignments, this means you should talk about your work with your peers, but you should not share your code with anyone, nor should you ask anyone else—currently taking the class or otherwise—to show you their code or to directly help you write code. As a heuristic for what constitutes encouraged collaboration, if it is something you can discuss without sharing your computer screen or writing Java syntax, then you are probably alright.

Academic Integrity. All students will be expected to uphold the Amherst College Honor Code. In particular, you should not:

- share your code for assignments with other students in the class or ask others to share their code with you;
- discuss material appearing on a quiz or exam before you or others have submitted the test;
- submit code written by someone else or derivative of someone else's code without crediting the original author/source.

Failure to comply with these terms may result in loss of credit for the assignment or course, as well as possible disciplinary action by the college.

If you would like to use code that you did not write, be sure to attribute the source of the code in the comments of your code. Attribution is required even if you modify the original code.