

Topic Notes: Introduction and Overview

Welcome to Data Structures!

Why Take Data Structures?

You have programming experience, at least some in Java.

In this course, you will become a more sophisticated programmer and problem solver, as you learn about designing correct and efficient algorithms and data structures for use in your programs. Along the way, you will:

- hone your problem solving skills,
- gain experience in programming in general, Java in particular,
- learn how to implement algorithms and data structures in Java,
- learn how to evaluate and visualize data structures and algorithms,
- learn how to understand (and prove) some properties of data structures and algorithms,
- learn how to consider the relative merits of different structures and algorithms, and
- learn how to design large programs (in an object-oriented way) so that it is easy to modify them

We focus mostly on the relatively simple textual interface often used by advanced programmers as opposed to graphics. But the algorithms and data structures may be used in (and are often essential to) those graphical programs, and some of our programs will use a graphical user interface. Your additional programming experience will allow you to understand and make use of the extensive base of reusable code, Java and otherwise, that is available to today's programmers, even though we will use only a limited subset of those tools here.

Sample Problems

Here are some examples of the kinds of problems we are aiming to be able to solve, some of which you should be able to solve by the time we finish.

1. Find the shortest path from Loudonville to Albuquerque on the national highway system (and do it efficiently).

2. Develop a game decision tree to allow a computer player for a game such as chess.
 3. Design and implement a scientific calculator.
 4. Design and implement a simulator that lets you study traffic flow in a city or airport.
 5. Design and implement a pattern matching system to find a particular sequence of nucleotides in the sequenced DNA of a given organism.
 6. Design and implement a simulation for some physical phenomenon (e.g., fluid flow).
-

Administrative Tasks

(See syllabus and course web site.)

Advice

Important: check your school email regularly. That will be my primary way of getting you important information. If you don't check it regularly, forward it somewhere that you do check regularly.

General important note: you will probably get confused at some point, but *don't remain confused*. If you do:

- come to my office hours or make an appointment
- e-mail any time
- see the tutors
- head off the problems before they get severe
- make me work – that's why you're here

I expect that I will see many of you in my office on a regular basis.

Coming to office hours will never be seen as a sign of weakness. On the contrary, asking for help outside of class will be seen as a sign of dedication to the course and a commitment to your success.