

63.271: Data Structures SYLLABUS

Catalog description

An in-depth presentation of recursion, collections and iterators, fundamental techniques in graphics, and threading. Students implement linked lists, stacks, queues, trees, heaps, graphs, hash tables, and related algorithms. Students implement a significant programming project.

Prerequisites

43.200 Precalculus

63.252 Computer Science II

Textbook

Michael T. Goodrich and Roberto Tamassia, *Data Structures and Algorithms in Java, 4th Ed.* (Wiley, 2005; ISBN 0-471-38367-8).

Reading textbook material is considered essential to college-level academic work.

To contact instructor

Office hours(Hemenway Hall 318A):

M 11:30-12:30 p.m., W 9:30-10:30 a.m.,

Th 2:30-3:20 p.m., others by appointment

Telephone: (508) 626-4724

Email: dkeil@frc.mass.edu

URL: www.framingham.edu/faculty/dkeil

Course overview

This course presents fundamental concepts in data structures and algorithm analysis. We will look at why some arrangements of data lend themselves to efficient handling and how to implement these arrangements. Data structures are *collections*: dynamic sets, stacks, queues, and graphs, implemented using arrays, linked lists, and trees.

We will begin by reviewing some data types presented in CS II: numeric values, strings, classes, arrays, and linked lists. These provide the basis for implementing the other data structures discussed.

Three classic data structures are the *stack*, the *queue*, and the *priority queue*, ordered collections whose elements are accessed in a particular sequence only.

We will present a general-purpose collection, *binary search trees*, for speedy access, and the very

fast *hash-table* structure. We will discuss algorithms to find paths between vertices of a *graph*, such as routes between cities on a map.

Two features of the Java language are support for concurrent *threads* of control, and support for *graphics* display. We will introduce multithreaded and graphical programming via these features.

Much of your work will be on a computer through programming projects. You will improve your skills in algorithm design, program documentation, and program testing. Much classroom time will be allocated to group work.

Data Structures is a rigorous course that is an essential prerequisite to several junior- and senior-level courses. It has its own serious prerequisites and will demand a strong commitment of time and effort outside class, 8-10 hours per week.

The course material is mostly language-independent; examples will be in pseudocode or Java.

Basic skills

All students will demonstrate *all* the following during in-class quizzes or presentations:

1. Proficiency with method calls and definitions, including the use of parameters;
2. Strong skills with nested loops such as those found in search and sorting algorithms;
3. String and array manipulation skills in Java, including arrays of objects;
4. Use of dynamic allocation, including linked lists;
5. Strong testing, tracing and debugging skills;
6. Identifying $O(1)$, $O(n)$, and $O(n^2)$ algorithms;
7. Identifying linked-list, heap, tree, hash table, and graph structures.

Performance objectives

1. To be prepared to argue effectively that an algorithm works, using invariants, preconditions, and postconditions;
2. To recognize the efficiency trade-offs of using arrays, hash tables, linked lists, heaps, and trees;
3. For each data structure presented, to state in big-O notation the running times associated with the standard algorithms discussed;

4. To recognize when a general collection, stack, queue, priority queue, or graph structure is required to solve a problem;
5. To be prepared to code the insert, delete, and search operations on all the structures presented;
6. To be ready to use the concepts presented here when taking the Algorithms and Theory of Computing courses;
7. To code sufficiently well to do the work required in the Computer Architecture, Software Engineering, Networking, and Operating Systems Internals courses.

Classroom format

Format will center on lecture (with slides), discussion, and group problem solving. Your questions and participation are important. Part of our responsibility is to express doubt about each other's claims.

We will write some program code on the board; other code we will compile, run and display, using a projector.

Classroom is a focused work environment governed by attention to business, free inquiry, and mutual respect.

Group work

For each topic, in-class teams will solve assigned problems.

Based on individual or group work, each student will lead a discussion on two problems related to group work or the semester project.

Group work is to be submitted via the Discussion Board. For each course topic, each group will post a set of solutions to several problems. Groups may revise and add to solutions submitted. Names of all group members who did their share of the work, and only those members, should be included with *each* solution posted.

Grading

There will be one short multiple-choice quiz per topic and three major quizzes. Make-ups for quizzes will be available at the time of the following quiz or at final-exam time.

Group work and the project are graded for documentation of specification, documentation of code, correctness of code, and testing.

Semester grade:

Major quizzes (3)	30
Basic skills	20
Mini quizzes	10
Final exam	10
Group work assignments	10
Project	10
Presentations	5
Participation	5
	100 %

Accommodations

Students who seek accommodations during the semester because of disabilities should meet with the instructor after class or during office hours early in the semester.

Course Plan

Dates	Topic	Textbook chapters
9/8	<i>Introduction</i>	1-2 (review)
9/8, 9/15	1. Numeric, class, and array data	3.1, 4, 6, 11, 12.1
9/22	2. Linked lists	3.2-3.4
9/29	3. Stacks and queues	5
10/6	<i>Major quiz (Topics 1-3)</i>	
10/6, 10/13	4. Priority queues, heaps	7-8
10/20, 10/27	5. Trees	10
11/3	6. Hash tables	9
11/10	<i>Major quiz (Topics 4-6)</i>	
11/10, 11/17	7. Graphs	13
11/24	8. Multithreading	Handouts
12/1	9. Graphics programming	Handouts
12/8	<i>Pre-final quiz (Topics 1-9)</i>	
12/8	<i>Review</i>	
12/15	<i>Final exam (multiple choice, Topics 1-9)</i>	