

CIS 22 Data Structures

Instructor Scott Dexter
Email sdexter@sci.brooklyn.cuny.edu
Office 0113N
Telephone 951-3125

Office Hours W 12:00–1:30pm
Th 1:05–2:30pm and by appointment

Website <http://www.sci.brooklyn.cuny.edu/~sdexter/>

Required Text *Data Structures Using C and C++*, Tenenbaum, Augenstein, and Langsam.

Other Sources *Practical Debugging in C++*. Ford and Teorey.
The ANSI C Programming Language. B. W. Kernighan & D. M. Ritchie.
Effective C++. Meyers.
C++ for Java Programmers. Budd.

Grading Your grade will be determined by your performance on 2 in-class examinations, quizzes, homework (i.e. programming) assignments, and one cumulative final examination. These will contribute to your grade as follows:

Homework	20%	Exam I	20%
Quizzes	10%	Exam II	20%
		Final Exam	30%

Reading This is a reading-intensive course. You should do the assigned readings *before* lecture on the day they are assigned — and you should probably read them again after the lecture.

Homework You will be given 5 or 6 homework assignments throughout the semester; these will be comprised mostly of programming exercises. The details of homework submission will be discussed in class. **Late homework will not be graded.**

Quizzes I will give two or three 20-30 minute quizzes. Most will be announced at least one class period in advance, but some may be unannounced, particularly if class participation is low.

Exams Exams I and II will not be cumulative. The Final Exam *will* be cumulative. **Cheating on exams and quizzes will not be tolerated.**

Attendance You are responsible for everything that is done in class. While the textbook is quite comprehensive, much additional material will be covered in lecture. Although attendance is not mandatory, I will take attendance most class periods. Class participation will affect the assignment of final grades in "borderline" cases.

Lateness I am in general very intolerant of lateness when it comes to handing in assignments (or missing exams!). As noted above, I usually will simply not grade late work. Exceptions to this are *documentable* family/medical emergencies, in which case I can be fairly flexible. If a more complicated situation is causing you to fall behind, *please* talk to me as soon as possible so we can make appropriate arrangements.

Calendar

All dates are subject to change.

Date	Topic	Reading
Feb 2	Introduction; Review: <code>typedef</code> , <code>struct</code> , <code>malloc</code> , ...?	
Feb 4	Abstract Data Types and Object-Oriented Programming Intro to C++; An Array class	1
Feb 9	Node-based dynamic allocation; A Node class; friend-ship	4.2-4.3
Feb 11	A List class	4.6
Feb 16	College Closed	
Feb 18	The Stack ADT; implementing the Stack	2
Feb 23	Stack applications	4.6
Feb 25	Templates	
March 1	The Queue ADT	4.1-4.3
March 3	Simulation; priority Queues	4.4
March 8	Review	
March 10	Exam I	
March 15	Lists revisited	4.5
March 17	Variations on Lists	
March 22	Recursion: algorithms and structures	4.5
March 24	Intro to trees and tree algorithms	5.1-5.5
March 29	Tree implementations and applications; Binary Search Tress (BSTs)	5.1-5.5
March 31	Balanced BSTs	(handout)
April 5	No Classes (Spring Recess)	
April 7	No Classes	
April 12	No Classes	
April 14	Review	
April 19	Exam II	
April 21	Intro to searching and sorting; time complexity	6.1
April 26	Bubble, insertion, selection sorts	6.2-6.4
April 28	Quick sort	
May 3	Heap sort	
May 5	List Searching	7.1
May 10	Tree Searching	7.2
May 12	Hashing	7.1,7.2
May 17	Hashing/Intro to Graphs	8.1, 8.3
May 19	Review	