

CS122 Algorithms and Data Structures

Syllabus – Fall 2004

MW 11:00 am - 12:15 pm, MSEC 101

Instructor: Xiao Qin

Phone/Office: 835-5902 / Speare
Building 146

Office Hours: MW 1:00 pm-2:00 pm **Email:** xqin@cs.nmt.edu

TA: TBA

Class Web Page

Homeworks and announcements are posted on the class web page:
<http://www.cs.nmt.edu/~xqin/courses/cs122>

Prerequisite: CS111 with a grad of C or better

Objectives

CS122 aims to:

- present fundamental techniques in the design and analysis of data structures that lie at the heart of computer science; (Example fundamental data structures include: lists, stacks, queues, trees, priority queues, hashing, graphs, and search trees.)
- introduce algorithm design and analysis techniques such as recursion and formal methods for analyzing the time and space requirements of programs; and
- provide programming projects that require students to apply the concepts introduced in classes in the development of rather large programs.

On completion of the course, the student should be able to:

- identify the specifications for the abstract data types (ADT) introduced in the course;
- use classes for implementation of each ADT;
- develop object-oriented software engineering skills such as software design methods, user specifications, implementation, and testing;
- choose appropriate data structures and algorithms design techniques to implement programs for problems of substantial complexity; and
- demonstrate ability to use the big O-notation to describe the order of the running time of a program;

Textbook

Adam Drozdek, “Data Structures and Algorithms in C++,” by Brooks/Cole
Thomson Learning, 2nd Edition, 2001, ISBN: 0-534-37597-9.

Additional Optional Reading

Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++," Addison-Wesley, 2nd Edition, 1999.

If you are not fluent in C++, you should also have access to a good reference book. Any one of the following books is recommended:

- *The C++ programming language* by Bjarne Stroustrup (3rd or Special edition)
- *Absolute C++* by Walter Savitch
- *C++: How to program* by Deitel & Deitel (4th edition).

Topics Covered

Topics covered include: introduction, list, stacks, queues, trees, hashing, sorting, sets, complexity analysis, recursion, and graph algorithms. (These topics may change.)

Exams and Grading

Quizzes	10%	There will be about four unannounced quizzes
Mid-term	15%	Oct. 13, 2004
Final Exam	25%	During finals week
Written homework	25%	
Programs	25%	

Scale

Letter grades will be awarded based on the following scale. This scale may be adjusted upwards if it is necessary based on the final grades.

A+ ≥ 97 A ≥ 93 A- ≥ 90 B+ ≥ 87 B ≥ 83 B- ≥ 80 C+ ≥ 77 C ≥ 73 C- ≥ 70
D+ ≥ 67 D ≥ 63 D- ≥ 60 F < 60

Reading

Students are expected to read the appropriate sections of the book before each lecture.

Homeworks

Most problems involve some mathematical reasoning, and some problem sets will consists of programming assignments. C++ is an official programming language. You are encouraged to use *gcc*, the GNU project C++ compiler. Please type "man gcc" for more information concerning *gcc*. You may use the UNIX/Linux system in the departmental workstation lab or any other system.

Late Homeworks

Homeworks are due at the beginning of class. There is a 20% deduction for late homework. The deduction becomes 50% if the homework is two days late. No credit is given after three days. Any exceptions must be arranged at least a day before the homework is due.

Cheating

Homework assignments have to be done individually. No “group projects” are allowed. However, students may discuss with their friends about general approaches to solving problems and writing programs. Students should not share program code. All code for programs must be written from scratch and you are allowed to use the libraries discussed in class. Please do not attempt to recycle code from the Internet (plagiarism). Any instance of suspected cheating or plagiarism will be referred to student judicial affairs.