

CMPS 101

Abstract Data Types

Summer 2004

Eight Week Session: June 22 – August 12

Description: Studies basic algorithms and their relationships to common abstract data types. Covers the notions of abstract data types and the distinction between an abstract data type and an implementation of that data type. The complexity analysis of common algorithms using asymptotic (big O) notation is emphasized. Topics include sorting and searching techniques, basic graph algorithms, and algorithm design techniques. Abstract data types covered include priority queues, dictionaries, disjoint sets, heaps, balanced trees, and hashing. Familiarity with C, Java, and Unix is assumed.

Prerequisites: CMPS 012B and CMPE 016; and either MATH 019B or MATH 011B or ECON 011B; and either MATH 021 or MATH 022 or MATH 023A.

Time and Place: TTh 6:00 – 8:25pm Baskin Engineering 372

Class Webpage: <http://www.soe.ucsc.edu/classes/cmcs101/Summer04/>

Class News Group: ucsc.class.cmcs101

Instructor: Patrick Tantalo (<http://www.cse.ucsc.edu/~ptantalo/>)

Office: Baskin Engineering 181

Office Hours: TTh 2:00 – 5:00pm, and by appointment.

Email: ptantalo@soe.ucsc.edu

Phone: 831-459-3898

MEP Tutors: Meeting times and places to be announced

Michi Mutsuzaki <mmutsuza@ucsc.edu>

Steven Chhay <schhay@cats.ucsc.edu>

Required Text: *Introduction to Algorithms*, second edition, by Cormen, Leiserson, Rivest, & Stein. McGraw-Hill, 2001. The following reading schedule is a rough guide to what we will discuss and when. It is subject to change. I expect that the material from appendices A.1-A.2, B.1-B.3, C.1 is already familiar.

<i>Week</i>	<i>Sections</i>	<i>Topics</i>
1	1.1-1.2, 2.1-2.3	Introduction, ADTs, Design and Analysis of Algorithms
2	3.1-3.2, 4.1-4.3	Growth of Functions, Asymptotic Notation, Recurrences
3	6.1-6.5, B.5	Heaps, Priority Queues, Trees
4	B.4, 22.1-22.4	Graphs, Graph Algorithms, BFS, DFS
5	22.5, 23.1, 23.2	Connected Components, Minimum Weight Spanning Trees
6	21.1-21.3, 24.1-24.5	Disjoint Sets, Single-Source Shortest Paths
7	12.1-12.3, 13.1-13.4	Binary Search Trees, Red-Black Trees
8	7.1-7.4, 8.1-8.4	Sorting Algorithms

Coursework and Evaluation:

Written homework will consist of assignments taken from the exercises in the book. Its purpose is to prepare students for the exams, and will be graded only as to its completion, not its correctness. We will have five **Programming Assignments**, due dates to be announced. It is a requirement of this course that each student have a CATS (Communication and Technology Services) computer account. All programming assignments will be submitted through that account. In particular, no work of any kind will be accepted via email. **Midterm Exam 1** will be held **Thursday July 8**, and **Midterm Exam 2** will be held **Thursday July 29**. The **Final Exam** will be held on the last day of class **Thursday August 12**,

6:00 – 9:00 pm. Please make arrangements now to be available at the appropriate times. Coursework will be weighted as follows:

Homework	5%
Programming Assignments	30%
Midterm Exam 1	20%
Midterm Exam 2	20%
Final Exam	25%

The grading scale for the class will be approximately: A+::97%-100%, A::93%-96%, A-::90%-92%, B+::87%-89%, B::83%-86%, B-::80%-82%, C+::76%-79%, C::70%-75%, D::60%-69%, F::0%-59%. Letter grade boundaries may be lowered at my discretion in order to eliminate some borderline cases.

Academic Honesty:

In recent years, there has been an increased number of cheating incidents in many UC campuses, and unfortunately, UCSC is no exception. The Baskin School of Engineering has a zero tolerance policy towards any incident of academic dishonesty. If cheating occurs, consequences within the context of the course may range from getting zero on a particular assignment, to failing the course. In addition to these sanctions, every case of academic dishonesty is referred to the students' college Provost, who sets in motion an official disciplinary process. Cheating in any part of the course may lead to failing the course and suspension or dismissal from the university.

What is cheating? In short, it is presenting someone else's work as your own. Examples include (but are not limited to) copying another student's written homework assignment, or program, allowing your own work to be copied, or in any way facilitating the cheating of others. Although you may discuss problems with fellow students, your collaboration must be at the level of *ideas* only. Legitimate collaboration ends when you "lend", "borrow", or "trade" *written solutions* to problems, or in any way share in the act of *writing* your answers. You may freely give and receive help with the computer facilities, editors, the UNIX operating system, and the proper use and syntax of the C and Java programming languages; but you may not copy, paste, email, or in any way share *source code*. If you do collaborate (legitimately) or receive any form of help from anyone, you must credit them by placing their name(s) at the top of your paper, or in the case of programming assignments, in your README file.

Please go to http://www.ucsc.edu/academics/academic_integrity/ to see the full text of the University's policy on Academic Integrity.