CMPS 102 Syllabus and Information
Spring 2006

Welcome to CMPS 102, Introduction to Analysis of Algorithms. The information on this sheet and on the course home page will be updated as the course progresses.

Prerequisites: All students must have successfully completed CMPS 101. Transfer students **must** have credit for this course approved by the CIS/CE board office.

Main text: Introduction to Algorithms, 2nd Edition, by Cormen, Leiserson, Rivest and Stein.

We might also have supplemental reading assignments from:

- **Fundamentals of Algorithmics** by Brassard and Bratley
- **Computer Algorithms** by Baase and Van Gelder

Another **great** resource is the book-in-preparation by Dasgupta, Papadimitriou, Vazirani available at `www.cse.ucsd.edu/~dasgupta/algorithms/` While a bit terse, motivation and intuition are superb.

Instructor: Dimitris Achlioptas, Email: `profcs102@gmail.com`, Office: E2 343A

TA: Colin Southwood, Email: `cfs@cse.ucsc.edu`

Office hours: MW 12:30-13:30 at E2 343.

Sections: (tentative) Wednesdays at 3:30 to 4:45 tentatively in Jack's lounge in Baskin Engineering; and on Friday afternoons at the same location at a time to be decided (but after 2:30pm).

Course Work: The course will be taught like a math course, not like a programming course. You will have regular written assignments, two in-class midterms, and a final examination. Your get to **choose** between one of the following two weighing schemes:

Default		**Homework-heavy**	
Homeworks	15	Homeworks	35
First Midterm	20	First Midterm	15
Second Midterm	25	Second Midterm	20
Final	40	Final	30

Your choice should be indicated on the front page of you first assignment and will be final (no indication will be taken as choosing the Default option).

Important Rule: If you chose the Homework-heavy option you will be subject to the following rule. On each of the three exams (two mid-terms, final) there will be one question from a past homework. For that question, your exam-answer should be "as good" as your homework answer or better (allowances will be made for presentation and minor details). If that's not the case, then you will get a score of 0 for that question on the exam *and you will be placed back on the Default grading scheme.*

I will give an incomplete only if there has been medical/family emergency **and** the student has been doing at least average work. To gauge the difficulty of the exams and homeworks, I will average the top three overall scores to get a target percentage. Anyone getting at least 90% of that target percentage will get an A or A-, anyone getting at least 80% of that target percentage will get a B (or B+ or B-) and anyone getting at least 70% of that target percentage will get at least a C.

Assignments: Do not cheat.

Remember: if you don't cheat, the worst that can happen is you'll get a lower grade; compare that to losing your dignity. If you believe that you deserve to cheat, e.g., because you "shouldn't have to take this class", please do the honorable thing and bring up this point in class for discussion.

In any case: the Computer Science Department of UCSC has a zero tolerance policy for 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, every case of academic dishonesty will be 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 elses work as your own. Examples would include copying another student's written homework assignment, or allowing your own work to be copied. Although you may discuss problems with fellow students, your collaboration must be at the level of ideas only.

At a minimum, what you turn in as your homework should be something that you could reproduce given nothing but pen, paper and a copy of the textbook.

And, certainly, 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. When you do collaborate (legitimately) or receive help from anyone, you must credit them by placing their name(s) at the top of your paper.

Class Accounts: Out of class-communications will take place over web and (mostly) email. You can use any email address you chose, but you must be able to read PDF files.

Syllabus and Reading (tentative): The following lists the various topics I will cover. You should read the appropriate sections at least once before lecture, even if it is difficult to understand. Later, go back and read carefully to ensure that you fully understand the material. Those topics marked with an "(R)" represent review material will be covered only briefly in class, you are expected to review these topics in more detail on your own.

Topic	**CLRS**	B&B	Baase and VG
introduction and asymptotic notation (R)	1,2,3	1,2,3	1
Proof Techniques (R)	–	1	3.3, 3.4
Summations and Recurrences (R)	4, ap. A	1.4,4	3.6
Counting and Probability (R)	5, ap. C	1.7	–
Divide and Conquer	7,8,9,28.2	7	3
Backtracking, Branch and bound	–	9	–
Dynamic Programming	15, 25.2	8	10
Greedy Methods	16, 23, 24.3	6	8
Lower bounds	8	12	5
String Matching	32	–	11

Chapter 10 and appendix B the text is material you should know from 101. Other topics, such as amortized analysis, NP-completeness, and approximation algorithms will be covered if time is available.