CMPS 201
Analysis of Algorithms
Spring 2010

Description:
Rigorous analysis of the time and space requirements of important algorithms, including worst case, average case, and amortized analysis. Techniques include order-notation, recurrence relations, information-theoretic lower bounds, adversary arguments. Analysis of the key data structures: trees, hash tables, balanced tree schemes, priority queues, Fibonacci and binomial heaps. Algorithmic paradigms such as divide and conquer, dynamic programming, union-find with path compression, augmenting paths. Selected advanced algorithms. Introduction to NP-completeness. **Prerequisites:** To take this course you must have completed an undergraduate course in data structures (such as CMPS 101), an undergraduate course in algorithms (CMPS 102), or an undergraduate course in graph theory (CMPE 177).

Time and Place: MW 5:00 – 6:45 Lockheed Martin San Jose
Class Webpage: http://ic.ucsc.edu/~ptantalo/cmps201/Spring10/

Instructor: Patrick Tantalo (http://www.cse.ucsc.edu/~ptantalo/)
Email: ptantalo@soe.ucsc.edu
Office: UCSC E2 257
Office Hours: MW 11:00-2:00 pm, and by appointment
Phone: 831-459-3898

Required Text:
Introduction to Algorithms by Cormen, Leiserson, Rivest, & Stein; 2nd edition, McGraw Hill (2001).

Optional Texts:
Fundamentals of Algorithmics, Brassard and Bratley, Prentice Hall (1996).
Algorithmics: Theory and Practice, Brassard and Bratley, Prentice Hall (1988).
Computer Algorithms, Baase and van Gelder, 3rd ed, Addison-Wesley (2000).
Computer Algorithms: Introduction to Design and Analysis , Sara Baase, 2nd ed, Addison-Wesley (1988).
Algorithm Design Manual, Steven S. Skiena, Telos/Springer-Verlag (1997).

Coursework and Evaluation:
Homework assignments and due dates will be announced in class and on the webpage. The **Midterm Exam** will be held in class on **Wednesday May 5.** The **Final Exam** will be held on **Monday June 7, 5:00-8:00 pm.** Course work will be weighted as follows:

Homework	**10%**
Midterm Exam	**45%**
Final Exam	**45%**

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:
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/or 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 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" *source code*, or in any way share in the act of *writing* your programs. 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) 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.