

CMPS 102 Introduction to Analysis of Algorithms, Syllabus and Course Information Jan. 07, 2009

- Instructor: Allen Van Gelder, 355 Engr. II, 459-4611, avg@cs.ucsc.edu
Office Hours: Mon., Wed. 3:30–4:30, and appt./drop in
- Teaching Asst.: Krishna Vuppala, vrk@soe.ucsc.edu, office , ext. Office Hours:
Section/Lab Hours: ???
- Text: Baase and Van Gelder, *Computer Algorithms, 3rd Edition* (3rd printing preferred)
Supplemental materials at <http://www.cse.ucsc.edu/~avg/Supplements/>
including errata and some sample Java code. See class locker also.
- Optional: Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms* 2nd ed.
Aho, Hopcroft, and Ullman, *Design and Analysis of Computer Algorithms*, 1979 (“the bible”).
- More Optional: Sedgewick, *Algorithms in C* (numerous variations).
Aho, Hopcroft, and Ullman, *Data Structures and Algorithms*, 1983 (“baby algorithms”).
- Programming: Gehani, Narain, *C: An Advanced Introduction, ANSI C Ed.*, 1989(?).
Roberts, Eric S., *The Art and Science of C*.
Hodges, *Introduction to Berkeley Unix and ANSI C*.
Rosen, Ken et al., *Unix: The Complete Reference*, 2nd Ed.
A Discipline of Data Abstraction using ANSI C, . . . /Supplements/adt-c.{ps,pdf}.
- Papers: To be announced and handed out.
- Web Page: <http://www.cse.ucsc.edu/classes/cmcs102/Winter09>.
- Course Directory: [/cse/classes/cmcs102/Winter09/](#)
- Newsgroup: `ucsc.class.cmcs102`; readable from UCSC only.
On ICL Solaris machines (e.g. `unix.ic` or BE105 workstations), do:
`trn -q +x ucsc.class.cmcs102`
-q is important! Otherwise, you have to discuss subscribing to 100’s of other newsgroups.
+x is important! Otherwise, you can miss messages due to “threading.”
Disable threads somehow if you use other readers.
- ICL servers: `ssh unix.ic.ucsc.edu`
- Optional texts are on reserve at the library. Look before you leap.

Learning Goals of the Course

Analysis of Efficiency, Analysis of Correctness, Recursion, Induction, Elements of NP-completeness theory, Algorithmic Paradigms: Divide and Conquer, Dynamic Programming, Graph Searching, Greedy Strategies.

Prerequisites: The prerequisites for this class are: CMPS 101, and its pre-requisites, especially CMPE 16 and Math 19B (or Econ 11B). Transfer students **must** have credit for these courses approved by the School of Engr, BE225.

Other Background: You are expected to be able to program in C and/or Java and to understand Unix and have some experience using your Instructional Computing Labs (ICL) account.
Visit <http://ic.ucsc.edu/students>.

Course Work: This course will have weekly problem sets, which are read but not graded. Evaluation will be based on a take-home review exam and 3 midterms. Some changes to the evaluation scheme might be announced early in the quarter. Students should plan to be present during the scheduled final-exam time unless it is announced otherwise.

| | |
|----------------------------------|-----|
| Review Exam (take home) | 10% |
| Midterms (in class, closed book) | 90% |

Students may be requested to show student-ids at the exams.

Policy on Academic Dishonesty: Any instance of academic dishonesty (cheating) is grounds for failing the course, regardless of how the student performed on other parts of the course. In general, academic dishonesty is the submission of any class work as though it were your own work, when in fact it is not. All assignments are to be done *individually*, although they are not graded and you might engage in discussions before writing them up.

Plagiarism: Plagiarism is academic dishonesty (see above), and will not be tolerated. Copying another person’s program, written assignment, exam answer, or any other work is **plagiarism**. Copying information from the internet and presenting it as your own work is **plagiarism**. Plagiarism is also considered unprofessional conduct by all computer professional societies, such as ACM and IEEE.

Principle of Disclosure: If in doubt, **acknowledge and give credit** for anything that you did not create yourself, and then you cannot be accused of plagiarism or dishonesty. If you disclose what you did (copying, got help, or whatever) in the work you turn in, then you clearly are not trying to conceal the facts. You might get a zero if you were not supposed to do that, but it’s an “honest zero”. If in doubt about any issue, ask the instructor.

File copying prohibited: *In no case* should you copy another student’s class-related computer files, nor should you permit another student to copy your class-related files. Files in the class locker, on the class “web” page, supplements to the main text, and so on, **may be** and often should be copied. Files elsewhere on the Internet in general **may not be copied**, because that will not help you achieve the learning goals of any assignment.

30 Minute Rule: After studying together in a group, do not take away notes on specific homework or program assignments, and wait at least 30 minutes before writing up or typing in your solutions. This helps to ensure that what you write reflects your understanding and not just your memory of what someone else told you. The same applies if the TA in section works on a specific homework problem. More often the TA will work on similar problems, not the exact homework.

Written Assignments: will be turned in at class **within the first 5 minutes**, or else directly to the TA outside class. Do not email any work to the instructor.

Tentative Schedule

Topics marked *Review* were supposed covered in CMPS 101 and/or math pre-requisites, and I will mainly cover newer material in class. You are expected to be already familiar with them, and/or read them on your own to brush up.

| Week | Chapter | Topics |
|---------------|------------------|---|
| 1 J/07 | 1,2 | <i>Review:</i> Asymptotic Notation, Sums, Logs, other basics, Three Great Algorithmic Paradigms, REVIEW EXAM OUT WED. |
| 2 J/12 | 3 | Recursion, Proofs, Induction, Recurrences, Divide and Conquer, REVIEW EXAM DUE WED., 2:00PM. |
| 3 J/19 (H) | 2, 3, 4 7.2–4 | <i>more</i> Recursion, Recurrences, Proofs, Induction, Searching Graphs: Depth First, Breadth First, Best First, |
| 4 J/26 | 7.4–5 | Topological Sort, Strongly Connected Components |
| 5 F/02 | 13.1–2 | review, MIDTERM 1 (WED.), NP-Completeness |
| 6 F/09 | 13.3.1–2, 10 | more NP-Completeness, Dynamic Programming |
| 7 F/16 (H) | 8 6 | Greedy Algorithms, Minimum Spanning Trees, Shortest Paths Priority Queue ADT, Amortized Analysis, Pairing Heaps |
| | 6,5 | Disjoint sets, Union-Find ADT |
| 8 F/23 | 13.3 | review, MIDTERM 2 (WED), more NP-Completeness |
| 9 M/02 | 9 | Transitive Closure, All Pairs Shortest Paths, |
| | 13.8 | more NP-Completeness (Traveling Salesperson Problem) |
| 10 M/09 | h.o. | Maximum Network Flow: the ultimate greedy algorithm, review. |
| 11 M/16 | | MIDTERM 3 (MON). |

H = holiday during the week.