CMPS 101-01 (60673) Syllabus: Algorithms and Abstract Data Types Apr. 3, 2017

Instructor:	Allen Van Gelder, 355 Engr. II, 459-4611, avg@cs.ucsc.edu Office Hours: Mon., Wed. 2:00–3:00, and appt./drop in
Teaching Assts.:	
Texts:	Baase and Van Gelder, *Computer Algorithms, 3rd Edition* (3rd printing preferred) Supplemental materials in the `Handouts` subdirectory mentioned on the next page, including errata and some sample Java code. *A Discipline of Data Abstraction using ANSI C* download `c-adt.pdf` or `c-adt.ps`, 22 pp. Also in the `Handouts` subdirectory.
Optional:	Gehani, *C: An Advanced Introduction* ANSI C Edition. Hodges, *Introduction to Berkeley Unix and ANSI C*.
Reference:	Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms* 2nd or 3rd ed. Sedgewick, *Algorithms in C*. Aho, Hopcroft, Ullman, *Data Structures and Algorithms*. Roberts, Eric S., *The Art and Science of C*.

Optional texts are on reserve at the library. Look before you leap.

Prerequisites: The prerequisites for this class are: CMPS 12B/M, CMPE 16(16H), Math 19B(20B), and *possibly* Math 23A or AMS 10 or . . ., although documents are inconsistent about this. Transfer students **must** have credit for these courses approved by the School of Engr office, 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 `unix.ic` CruzID account.
Visit `http://its.ucsc.edu/unix-timeshare/`.

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***.

Plagiarism: Plagiarism constitutes 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. **This is includes copying from the Internet or posting your code to a general-access site on the Internet.** 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.

Course Work: This course will have substantial programming assignments in **C** (to be done individually), several written homeworks, and three midterms. These will be weighted as follows.

Midterms 75% **Programs** 15% **Homeworks** 10%

Students may be requested to show student-ids at the exams. In order to pass the course you must **pass the exams component** and **pass the programs component by showing competence in C programming**, as well as passing all components combined.

Online Information: Bookmark the class URL: `http://classes.soe.ucsc.edu/cmps101/Spring17`. It has info to help you get started with your CruzId account. Its `Handouts` subdirectory has class-related documents, including this syllabus. **You are responsible to be aware of the class mailing list**, especially near deadlines, for announcements, questions and answers, mainly about assignments: `cmps101-spring17-02@soe.ucsc.edu`. With web browsers and other software, be sure to disable "threads," so you see all the official messages.

Reading: I recommend that you read through the appropriate sections before lecture, and note areas where you have questions; be sure they get answered in class. Later, go back and read carefully, to ensure that you fully understand the material. The main text is intended to be understandable on its own. Optional texts should be on reserve at the library.
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.
Programs will be written in ANSI C. ***Minimum proficiency*** in Unix and C are required to pass the class; passing the programming part is not just a matter of numerical grades. at least one of pa03 or pa04 must do all assigned functions correctly. Programming assignments ***must*** be done ***individually***. There is ***no pair programming***.

The programming assignments must work on `unix.ic.ucsc.edu` running from your (and our) CruzID accounts. We will have electronic program submission, and automated grading, and will also work with a "show and tell" grading system, where needed. Additional technical requirements are stated in the assignments.

CruzId (Class Account): Your `unix.ic` account is assigned automatically, referred to as your CruzID or "email account." **This is not your student number**.

Tentative Schedule

Topics marked *Review* were at least partially covered in CMPS 12B 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	A/03	1.3, 1.5.1	*Review*: Math, Asymptotic Notation
		1.1, 1.2, 1.3, 1.4, 1.5	*Review*: Java for Algorithms, Math
		1.4, 1.5, 2, adt-c.pdf	Analyzing Algorithms, Asymptotic Notation, Abstract Data Types, Encapsulation, Dynamic Sets,
		7.1–2	Graphs.
2	A/10	3	Recursion, Proofs, Induction, Recurrences,
		4.1–6, 4.8,	Insertion, Quick, Merge, and Heap Sorts, Priority Queue ADT, **hw01 DUE**.
3	A/17	3	*more* Recurrences, and Sorting Analysis,
		4.7, 4.9	Sorting lower bounds, Comparison of Sorts, **pa01 DUE**.
4	A/24	7.1–4, 7.6	Depth first search, Connected Components, Undirected DFS, Topological Sort, **hw02 DUE**.
5	M/01	7.5–6	Strongly Connected Components, Breadth-First Search, review, **MIDTERM 1** (WED 05/03) **pa02 DUE**.
6	M/08	7.5, 6.7, 8.1	*more* SCCs, Greedy Algorithms, Priority Queues, **pa03 DUE**.
7	M/15	8.2	Minimum Spanning Trees, review, **MIDTERM 2** (WED 05/17)
8	M/22	8.3	Prim's Algorithm, Dijkstra Shortest Path Algorithm, **hw03 DUE**.
9	M/29(H)	6.6, 8.4	Union-Find ADT, Kruskal's Algorithm, **pa04 DUE**.
10	J/05	9.1–4	All-Pairs Shortest Paths, review, **MIDTERM 3** (FRI 06/09).
11	J/12		Finals Week. Tu730PM