Course Information

You are responsible for everything on this handout — better read it!

Course homepage. www.cs.ucdavis.edu/∼rogaway/classes/120/winter06. (It's two clicks from Prof. Rogaway's homepage of www.cs.ucdavis.edu/∼rogaway/.) Please visit the page regularly.

Lectures. TR 1:40–3:00 in 107 Cruess.

Discussion Sections. F 4:10–5:00 in 107 Cruess.

Instructors. The course will have an unusual tag-team-teacher format; Earl Barr and Phillip Rogaway will be jointly teaching it. We'll speak in alternating sentences, or some such thing.

Earl Barr, Kemper, #3060, email: barr@cs.ucdavis.edu. Office hours on the course home page.

Phillip Rogaway, Kemper, #3063, rogaway@cs.ucdavis.edu, http://www.cs.ucdavis.edu/∼rogaway/. Office hours on the course home page.

Please don't use email for routine homework questions; come to office hours or use the newsgroup, instead.

Teaching Assistant. Our TA is Raza Saqib, sraza@ucdavis.edu. Office hours and location on the course home page.

Grading. Problem sets (20%), Quizzes (15%), Midterm (25%), Final (40%). Caveats: (1) you cannot get a passing grade in the class without getting a passing grade on the final; (2) In assigning grades we may deviate from the stated numerical percentages if we see a compelling reason to do so.

There will probably be 10 problem sets. The week with your midterm will probably have a problem set, as usual.

It is possible that the grader will grade a proper subset of the problems turned in.

Misgrading concerns must be submitted to the TA, in writing, within one week of when the problem is returned.

Prerequisites. ECS 20 is the prerequisite for this class (Math 108 is recommended and is interchangeable as far as we're concerned). This is a serious prerequisite in the sense that you will not do well in this course if you do not have mathematical maturity consistent with having taken, and understood, ECS 20. In particular, you need to be able to understand and create proofs. While we expect very little in the way of *particular* mathematical background, we do expect that sort of mathematical maturity. If you are a CS major who has trouble with math, consider taking some or all of your math electives before taking this class.

Text. Michael Sipser, *Introduction to the Theory of Computation*, 2nd edition. Course Technology, 2005. We will cover most of the material of Chapters 0–5, 7. You may use the prior edition of the book if you wish. There is also, apparently, an on-line version.

Course newsgroups There are two of these.

`ucd.class.ecs120` — This is for us and the TA to communicate things of general concern to the class (most typically, corrections to any homework problems found to be in error). You are responsible for anything we post to this newsgroup. Please don't post anything to this newsgroup.

`ucd.class.ecs120.d` — This is the "discussion" newsgroup that you may use to communicate among yourselves, and also to ask questions directed to other students at large, me, and the TAs. Read this newsgroup or not, the choice is yours. Please do not post what amounts to a solution to any homework problems on this newsgroup.

Problem sets. Homeworks will due on Tuesdays at 1:15 pm. Turn in your homeworks in the appropriately-labeled box on the second floor of Kemper. No late homeworks will be accepted.

Much of what one learns in this course comes from trying to solve the homework problems, so please work hard on them. We intend for you to find some of the problems challenging. If you're keeping up with the course and are reasonably creative, you should be able to solve most of the problems within a few hours of effort. But a few of the problems you might not be able to get. Don't let this discourage you.

Doing a conscientious job on the homeworks is the best preparation for the exams, and it is essential for mastery of the material.

Oddly, many students are more willing to spend long hours hacking in front of a machine than to spend them peacefully thinking beneath an old oak tree. We wonder why. Try not to be that way.

Your writeups should be clear, terse, and neat. Aim for elegance. Obsess a bit. *We encourage you to typeset your solutions.* Drawing associated pictures by hand is fine. We encourage top students—including anyone bound for graduate school—to use LaTeX (a typesetting program especially good for mathematical material). The elegance you should strive for includes (but goes far beyond) pretty typesetting.

For me, clarity of a solution is as necessary as correctness. Don't be surprised to lose points if you provide a correct solution with a poor writeup; we always encourage my graders to grade that way. As with an English paper, please don't turn in a first draft: you need to refine your writeup a time or two, making it shorter, simpler, and cleaner.

If you can't solve a problem, briefly indicate what you've tried and where the difficulty lies. Don't try to "fake" a solution. Know what you know and be clear about it. In grading exams you will find that we are not big on partial credit—get all that you can fully right.

Collaboration. For this particular class, *we discourage collaboration on homeworks.* When Prof. Rogaway asks people who have are good at this subject if they ever worked with anyone on their equivalent of this course, the answer is always *no.* This is an individualistic struggle.

That said, we don't *prohibit* collaboration, and some students sincerely believe that they learn better with it. If you do collaborate, the manner in which you collaborate will have a profound impact on how much you get out of the homeworks. (which will, in turn, have a big impact on

how you do on the exams). First, think about the problems and try to solve them on your own. If, after giving a problem some real though, you just can't get it, then you might wish to discuss it with other students, with me, or with the TA.

Academic misconduct. If you discuss problems with anyone, acknowledge him/her/them (*I discussed this problem set with so-and-so*). Also acknowledge any books which you consulted other than your own. Write up problems entirely on your own (even if you discuss a problem with someone else).

Some homework questions will have been used in prior years, either by Prof. Rogaway or other professors. *You absolutely may **not** consult old problem-set solutions for this class, or those of related classes.* Not "official" solutions posted to the web, and not those from another student in a prior term. (We are warning you right now that we have on many occasions sent students to Judicial Affairs for providing solutions that looked to us to influenced by an old problem-set solution.)

If you are having personal or academic problems which are motivating you towards academic misconduct, please come and talk to one of us, instead.

Some hints. Most students find this class very abstract and challenging. Some students tell us it is the hardest class that they have ever taken. So let us give you a few hints explaining what we are after and how to do well.

First, we really want you to *think*. Don't try to solve the problems by doing some sort of "pattern matching." It just won't work.

This course is about learning a certain sort of problem-solving skill more than it is about learning any specific material. Keeping this in mind may help put things into better perspective.

Even more than with other courses, you must not get behind. What we will do will keep building on what we have already done. Don't lose the thread. If you get seriously behind you will probably find it impossible to get back on track.

Be selective in note-taking. Actually, we would suggest that you don't take notes at all: the book is good and we will follow its development pretty closely. You can just sit and listen and think and follow. If you feel you must have notes, you might team up with others and take turns.

If you get involved in a study group, don't let the emphasis degenerate into an attempt to get as many homework points as possible. They homework points don't matter all that much and you'll learn more struggling on your own.

Parting thoughts. This is my favorite course in the CS/CSE curricula. We get at the question of what *is* computation. What could be more interesting or more fun?