

Course Information

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

Course homepage. www.cs.ucdavis.edu/~rogaway/classes/120/winter02. It's two clicks from my homepage of www.cs.ucdavis.edu/~rogaway/. Please visit the page regularly.

Lectures. Section 1: TR 10:30–11:50 in 3 Temp. Section 2: TR 1:40–3:00 in 212 Wellman. Exams must be taken with the section in which you are enrolled.

Discussion Sections. T 6:10–7:00 in 107 Cruess and T 7:10–8:00 in 107 Cruess. Go to either one.

Instructor. Phillip Rogaway, Eng. II, #3063, Phone: 752–7583, email: rogaway@cs.ucdavis.edu, URL: www.cs.ucdavis.edu/~rogaway/

Please don't use my phone or email for routine homework questions; come to office hours and/or use the newsgroup instead. Sorry, it's just not practical to respond to tens of students emailing me homework questions.

Teaching Assistants. We have two TAs: **Jason Coit** and **Fan Zhao**. Office hours in #3104 or #3106. Both TAs will use cs120t@cs.ucdavis.edu.

Everyone's office hours. I find that whatever I write down gets outdated after a short period of time, so please look at the course web page for current hours.

Grading. Problem sets (20%), Quiz (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 I deviate from the stated numerical percentages if I see a compelling reason to (eg, a homework score much higher than your exam scores may get ignored; a midterm much lower than your final may get de-emphasized).

The quiz will be Thursday of week 4: January 31. The midterm will be Thursday of week 7: February 21. Dates subject to change.

There will probably be 10 problem sets. The week of exams will have have problem sets, as usual.

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

Misgrading concerns must be submitted within one week, in writing.

Prerequisites. ECS 20 (formerly ECS 100) is the prerequisite for this class (with Math 108 recommended and interchangeable as far as I am concerned). This is a serious prerequisite, in the sense that you will not do well in this course (you might not even pass) 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 I expect very little in the way of

particular mathematical background, I 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*, PWS Publishing Company, 1997. We will cover most of the material of Chapters 0–5, 7.

Course newsgroups

`ucd.class.ecs120` — This is for me and the TAs to communicate things of general concern to the class (most typically, correction to any homework problems found to be in error). You are responsible for anything we post to this newsgroup. You yourselves should not post anything to this newsgroup.

`ucd.class.ecs120.d` — This is the “discussion” newsgroup which 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. Do not post what amounts to a solution to any homework problems on this newsgroup.

Problem sets. Homeworks will usually be due on Thursdays at 10:00 am in the turn-in box in Engineering II, #0086. No late homeworks will be accepted. Write your section number (1 or 2) prominently on your solutions.

Much of what one learns in this course comes from trying to solve the homework problems, so please work hard on them. I intend for you to find some of the problems challenging. If you’re working at it, most of you will be able to get most of the problems with a reasonable amount of effort. Some problems you may not be able to get. Don’t let it 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, to me, many students are happier to spend long hours hacking in front of a machine than peacefully thinking beneath an oak tree. Try not to be that way. If we are not willing to stretch and challenge our minds, what exactly are we? The skills you learn in this class are more applicable than you might imagine.

Your writeups should be clear, terse and neat. Aim for elegance. Obsess a bit. Doing your solutions in \LaTeX is always nice, I encourage all serious students to learn this tool. But keep clear that the elegance we are striving for 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; I always encourage our graders to do this. 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, 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 I am not big on partial credit; don’t expect it. Get all that you can fully right.

Collaboration. I strongly discourage collaboration on homeworks. When I ask 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 a very individualistic struggle.

That said, I 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 (and this, in turn, will 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 thought, you just cannot solve 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, make certain to acknowledge him/her/them. 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 me or other professors). Do not consult old problem set solutions for this class — neither solutions of another student from a prior term nor solutions handed out in class from a prior term.

If you are having personal or academic problems which are motivating you towards academic misconduct, come talk to me or a TA. We're not monsters.

Some hints. Most students find this class very abstract and challenging. Some students tell me it is the hardest undergraduate class they take. I've taught this class lots of times, and I want you to succeed. So you might like to listen to my hints. Or you might not!

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

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, I myself would never take notes in a course like this. The book is good and I follow it pretty closely. I think it's better to just sit there and listen and follow. If you feel you need notes, you might team up with others so that you only sometimes have to be scribbling stuff down.

If you get involved in a study group, don't let the emphasis degenerate into an attempt to jointly work out the problems and get as many of the homework points as possible. They don't matter 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?