CSEE W4119 - Computer Networks
(Call # 77951)
Course Information

Professor Dan Rubenstein — Spring 2007
(revision 1: 1/23/07)

Contact Information			
	Dan Rubenstein (Instructor)	Dhruv Chopra (TA)	Kyung-Wook Hwang (TA)
Office	CEPSR 816	CS TA Room	CEPSR 807 (revised 1/25/07)
Phone	(212) 854-0050	TBD	TBD
e-mail	danr@cs.columbia.edu	dc2378@columbia.edu	kh2210@columbia.edu
Mailbox	in Mudd 1312 (EE Office)	TBD	TBD
Office Hours	Tu 9:30-10:30, W 3:00-4:00 (revised 1/23/07) or by appt.	Mon 1-3pm	Th 1-2pm, (revised 1/26/07) F 11am-12pm

Course URL: http://www.cs.columbia.edu/~danr/4119.

Also see Courseworks (https://courseworks.columbia.edu/) for additional handouts, etc.

Course meeting time / location: 2:40pm - 3:55 pm on M,W in Mudd 1024

Pre-Requisites

C or Java programming, Course in algorithms, Course in probability

Description

Topics: Introduction to computer networks and the technical foundations of the Internet, including applications, protocols, local area networks, algorithms for routing and congestion control, security, elementary performance evaluation. Several programming assignments required.

This course is a joint EE/CS course. It requires both programming skills and mathematical training.

Grading

Your grade consists of:

10% **Homework:** Unless otherwise specified, homework will be due one week after it is assigned and should be turned in by beginning of class. At that time, a physical copy of the assignment must be received (CVN students have an additional 24 hours to send in their homework). If you will not attend class on that day, you should slide the homework under my office door (CEPSR 816) by **2:30 pm** on the day it is due. Before class but after 2:30, I will collect the homeworks from my office. On-campus students: E-mailed/faxed homework and late assignments **will not** be accepted unless approved in advance. Approval will only be given under extreme circumstances. You are expected to produce your work in a timely manner.

You may discuss and work on questions with other students in the class. However, you should write your solutions on your own. In other words, if I were to later ask you to re-derive one of your homework solutions or to solve a similar problem when you were without your friends, you should be able to do so or have a clear understanding of how to approach the problem. This can only be learned by doing, so you should do your homework.

25% **Programming Assignments**: same rules as homework. You should write your own code. Note our ability to check the compiled code for similarities.

25% **Mid-term:** March 6 in-class, closed book, no calculators.

40% **Final:** TBD by registrar, closed book, no calculators.

Exams: I try to test your understanding of a concept, and not just straightforward regurgitation of formulae, i.e., **why** certain rules, laws, and techniques hold and are used. Hence, I try to design the midterm and final questions to test your understanding of the concepts, not your memorization skills. I realize that some memorization will undoubtedly be required, but hopefully the memorized concepts will be those that can be re-derived via your intuition. I usually take a problem covered in class and put a small "twist" on it, so that blind application of the method won't work, but if you have the kind of understanding I am looking for (e.g., the kind that the inventor of the method had), you will know how to adapt the method.

A note on effort: Your grade will mainly be a reflection of how you perform on the midterm and final. Homework grades don't have much of an effect, as long as homework is turned in (i.e., most students typically get most of the problems right). **You should do the homework so that you learn the material.** If you find yourself copying or getting solutions from someone else without putting in the effort of solving them yourself, you'll probably find yourself doing poorly on the exams. You won't get much sympathy from me if you come crying to me at the end of the term that you did well on the homework yet poorly on the midterm and final.

If you are a bad test-taker, there is hope! Show me (i.e., in office hours and class) that you understand what is going on, and I take that into account when assigning the final grade.

How much I care about helping students is directly proportional to how much you seem to care about the class (i.e., via attendance, homework, coming to office hours). I have nothing personal against students who think the class is a waste of their time or think they have better things to do with their time. I also have lots to do besides teaching, and will only make the extra effort for those students who earn it by putting in the extra effort themselves (active in class, active at office hours).

Reading / Texts

Note that we will really only be following the textbook loosely, i.e., the course is not contained entirely within the required text.

- **Required:** James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3rd ed. Addison-Wesley, 2005, ISBN 0-321-22735-2
- **Optional:** Dimitri Bertsekas and Robert Gallager *Data Networks (2nd ed.)*, Prentice Hall, 1992. ISBN 0-13-200916-1. Significant mathematical treatment (graduate 6000 level - practical stuff a bit out of date).
- **Optional:** Andrew S. Tanenbaum, *Computer Networks* (4th ed.), Prentice Hall, 2003. ISBN 0-13-066102-3. A lot like Kurose/Ross
- **Optional:** Alberto Leon-Garcia and Indra Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures, 2nd ed.*, McGraw-Hill, 2004. ISBN 0-07-246352X. A bit more mathematical than Kurose-Ross, but less than B&G.

Computing Accounts

You need access to a computer with Berkeley Sockets or the Java equivalent

Cheating

In short: don't do it. Be warned now - I take cheating very seriously. If you are caught cheating on the midterm or final, you will fail the class and I will likely take additional action which can result in your suspension or expulsion from Columbia. It's not worth putting yourself in this position.

If a grade is that important to you then you should be putting in the extra effort, i.e., reading the book, coming to office hours, etc.

You must use common sense about when to collaborate / use notes / calculators, etc. If you are unsure of a policy, you should ask me or the TAs first *before* doing something you (and I) might consider unethical. Both I and the TAs will be putting a lot of time into teaching you this course. Our goal is to teach you the material. Grades on homeworks, midterms, and finals are not only a means to evaluate you, but also a means to force you to learn the course material.

If you do your own work but facilitate someone else's cheating, you run a risk of getting in trouble as well. This is because you run the risk of having me determine who copied from whom. If you feel that someone is pressuring you to help them in a way that makes you uncomfortable, come talk to me / send me e-mail. You should feel free (and actually I would encourage you) to

- Discuss homework problems / give hints / work together through a part of a problem that you are stuck on
- Study for the midterm / final together

Student Feedback

I'm always looking for ways to improve the course. If you have any comments or criticism about the course, or find any mistakes or misleading facts / comments in the lecture, please feel free to contact me. This includes comments on the material being covered, teaching style, pace of the class, workload, etc. I will try and accommodate, but I can't make any promises...

Things to know about Professor Rubenstein

- I teach what I believe is fundamental material. Often, this means I gravitate toward the theoretical side and de-emphasize practical details, which I assume Columbia students are more than capable of picking up on their own. I expect students to have decent mathematical sophistication (i.e., know probability and algorithms). The way I teach does **not** prepare you directly for a job as a network programmer, or give you the know-how to work at CUIT, or hack Skype, build your own P2P software, etc. It's not *what* is taught, so much as the thought process behind the evaluation and understanding.

  There are students who think this approach is inappropriate (read a nasty review of me on CULPA that compares my lecture style to a sushi chef who gives you a "dead fish"). Another student wrote in the class wrote in their (private) Courseworks review: "Professor Rubenstein taught us everything we didn't need to know, and didn't teach us anything we needed to know." I view these opinions as incredibly short-sighted. The classroom is probably both the last and best place to learn about theories, models, abstract thinking. Previous students frequently tell me that all that math we focused on actually turned out to be useful, but in unexpected ways, while alot of what seemed more relevant at the time is no longer relevant. In my opinion, it's the most useless place to spend time learning the details - most of which I don't even know myself. You learn them by experience (on the job, or in a lab, or on a project... not by someone lecturing them to you).

  Not everyone agrees with me. If you disagree with me, the smart thing to do is to drop the course.

- I write on the board (i.e., I don't use slides, I don't provide handouts, etc.) so if you want to know what is going on in class, come to class, or get a friend to take notes. Strangely enough, if you're paying attention, I believe you learn alot more when taking notes yourself. For some reason, when you

just read off of notes, your brain tricks itself into thinking it's seen everything when it hasn't. Also, if you have good pre-packaged notes, what do you need me for?

- I use the book as a rough guide, but I don't follow it verbatim. I will leave out lots of material that is covered in the book, and will interject material that I think is relevant (i.e., a more theoretical/mathematical treatment than what is provided in the book). **Some students hate this.** Sorry, one book is too hard and outdated for this course (Bertsekas and Gallagher), the others (including Kurose and Ross) are too lightweight

- I respond to e-mail in batches. I get between 50-100 e-mails a day that require a response. I read everything as it comes in (Unless traveling, I check e-mail several times during the day, before I go to bed, when I wake up, etc.) but if an e-mail takes more than a minute to craft a response, I usually wait to answer it. Roughly once or twice a week, I do a sweep of my inbox and respond to e-mails in a batch. So if you have questions on the homework and cannot come to office hours, my advice is to not wait until the last minute if you want me to answer questions via e-mail.

- Some Outside Thoughts (from Columbia Underground Listing of Professor Ability (CULPA) at www.culpa.info)

  For ELEN E4710: Intro to Networking Theory (from someone who isn't my biggest fan):

  > "My worst academic experience. Pay 200% attention during lectures if you are taking his class since nothing he teaches in lectures will be found in the textbooks. Never buy the textbook he assigned since I have never used it once. Professor Rubenstein seems to be a very smart person and he gets annoyed with questions. Don't take his class if you have problems getting up. He always has 9:00am classes, hws are due at the beginning of the class and LATE assignments are never accepted, even if you arrive 30 seconds after the lecture has started. Homeworks are very very difficult, so are exams. you are doing really well if you get above 70% in HWs. Midterms and final scores usually averaged around $\sim 40\%$.
  >
  > Workload: Around 7 - 8 assignments. Due dates usually get postponed since he never finishes the materials on time."

  Alot of this was true for 4710: I didn't follow a book at all (I didn't think any book was at the right math level) but expect to be closer to the book here. I don't agree that I don't like questions.

- I'm willing to put time into helping those students who truly put effort into the class. I judge this by them coming to office hours and showing me where they get stuck on a problem, clarifying something they didn't understand in office hours, answering relevant questions in class, and/or showing me in class that they're truly thinking about the material (i.e., asking an insightful question).

- I find it a waste of time to put effort into students who don't put effort into the class. Some of you will disappear after a few lectures, copy your homework, slack off on the programming assignments, and then get crushed on the midterm (and probably on the final later on as well). Don't come to me if all you want is sympathy, or to explain to me that you understand the material and don't know what happened. You can come to me if you want to get back on track, but my suggestion will be: do your homework, do the programming assignments, come to class, come to office hours, i.e., put in effort.