COMP 200 Final Project Ideas Fall 2004

1. *Early history of computing machines:* pick an early computing machine and explain how it was built and how it worked. Suggestions for machines: Babbage's differential engine, the World War II Enigma machine, the IBM Stretch, the IBM 650, the Burroughs 5000. See also http://www.computerhistory.org
2. *Data compression:* how are text files compressed (to create .zip or .gz files)? Are there limits on how much a file can be compressed? Why is image compression different than text compression?
3. *Privacy in the Internet age:* A broad range of privacy issues confront Internet users, including vulnerability of e-mail to outside readers, viruses, worms, spy-ware, spam mail-address harvesters, retail databases (how does Amazon make its recommendations), peer-to-peer file sharing, web site phishing.
    a. Write a broad overview of the threats and suggest techniques, behaviors, or tools that can help an informed user avoid them.
    b. Investigate one or more threats in greater depth.
4. *Spam:* What is it? How do spammers find e-mail addresses? How do they make money? Do people respond? What can you do to avoid being inundated with Spam?
5. *Spam:* How do mail filters work? How do spammers get around them?
6. *Computational Models:* Investigate some of the models that are used to reason about complexity classes. Examples include the Turing machine, Pebble machines, Random Access machines, and the lambda calculus (or recursive functions). Discuss the relative power of the various models. Show how you could encode one of the classic algorithms from class, such as bubblesort, in one or more of these models.
7. *Electronic voting:* Investigate the issues surrounding electronic voting — either digital voting machines or web-based balloting, or both. What are the policy issues that confront society as it moves toward electronic voting? How should we handle them? How do the security measures trade-off against the requirements for an election? (For example, does online, web-based voting work for SA elections or faculty voting on resolutions? What properties of an election make web-based voting simple or complex? What security properties are easy to enforce? Hard to enforce?)
8. *Alternate Electronic Voting Idea:* What problems arose in the national elections? or local elections? How did the procedures put in place by counties impact (positively or negatively) the potential for problems? What level of inaccuracy arises with paper ballots? How large an error (deliberate or unintentional) would be needed to change the outcome of the presidential election? or the senatorial election in Alaska?
9. *Public-key cryptography:* What is it? How does it work? What issues might arise if it were adopted for widespread use? How could you advertise a public-key with reasonable assurance of not being phished?
10. *Complexity theory:* Pick a problem that is known to be NP complete and sketch the details of a proof that it is, in fact, NP-complete. Your paper should target a COMP 200 student as its audience, rather than a COMP 482 student, so it should explain issues such as reducibility between problems that underlie the proof.

Revised List — Check Back Periodically for Additional Suggestions

11. *Machine learning:* Explain, compare, and contrast two techniques in machine learning, such as genetic algorithms and simulated annealing. Pick a problem and show how you might apply the two techniques to it.
12. *Viruses and worms:* Research how these programs work. You might pick a well-known virus, such as Blaster, and explain its workings (& its effectiveness) in detail. Alternatively, you might look at the techniques that a computer user can use to protect against these viruses. What habits, procedures, etc. should the user adopt? How do firewalls work and do they help prevent such attacks? How does virus scanning software work? What are the strengths and weaknesses of virus scanning software?
13. *Home Network Connections:* If you were a homeowner in Houston, what options are available to you for Internet service? What are the similarities and the differences between the available options? How does the price-performance curve look? What, realistically, are the tradeoffs? (How much bandwidth do you need or use?)
14. *Programming languages:* Pick two styles of programming language from Chapter 3 and implement QuickSort in both of them. Write up your experience, including how the approach and implementation differed between the two languages. Your QuickSort implementations must work.
15. *Internet Addressing:* Investigate the system used to assign internet addresses. Discuss issues such as: the number of available IP addresses; the domain-based naming system and its use to translate a textual name, such as diana.cs.rice.edu into an IP address, such as 128.42.2.16 5. What strategies do network managers use to "stretch" the number of IP addresses? (For example, the network at my house shares one IP address among at least 5 devices.) What are the social and commercial implications of domain-based naming?
16. *Complexity Classes:* Consider Figure 7.15 in the textbook. Discuss the various complexity classes mentioned in the figure and the underlying computational models that correspond to those classes. (For example, the control apparatus of a Turing machine models the distinction between the classes P and NP.)
17. *Complexity Classes:* Consider Figure 7.15 in the textbook. Relate the various complexity classes to the design of games (e.g., chess, monkey puzzles, roadblock). How do game designers use complexity to improve (?) their products.