Final Programming Project

Proposal due:	*Nov 20*
Implementation Plan due:	*Nov 30 - Dec 1*
Prototype due:	*Dec 7-8*
Project due:	*Dec 9-10*

For the last two weeks of lab this semester, you will work on a project of your own design. While exactly what you do is entirely up to you, it should ideally demonstrate, in a substantial way, what you have learned during the semester.

To prepare for your final project, please spend some time thinking about what you would like to do. You may select a project similar to those described on the "Final Project" page of the course web site, or suggest a project of your own. (We prefer the latter, since it will probably be of more direct interest to you.) In particular, you *may* design a project that helps you in another course or serves as a tool in a research problem.

Please submit a short discussion of your plans (about 1-2 pages typed?). **Submitting this proposal is an indication that, if the project concept is accepted, you will follow through with the proposed project.** In other words, think carefully about what you write! In your proposal please address the following questions:

1. Briefly, what do you want to do for your final project? Feel free to draw a picture of your running program.
2. What programming skills do you think you need to (*or* would like to) spend more time developing?
3. For the purposes of focusing our evaluation, what aspects of this project will address those skills discussed in question 2?

You may work with a partner if you prefer, but this is *not required*.

Project Evaluation

When evaluating your projects, we expect to use the following categories:

• Completeness	40-50%
• Correctness	10-15%
• Code formatting and documentation	10-15%
• Coding style	15%
• Educational value	15%

Completeness will be judged by determining to what degree you have written Java code that represents a reasonable attempt to implement the full functionality described for your program in your original proposal.

Correctness will measure how many (or few) flaws remain in your code.

Code formatting and documentation evaluates the degree to which you have added comments and used spacing and indentation to make your program as readable and clear as possible. Consult the style guide on the course web page.

Coding style refers to the clarity of your code, ignoring issues of spacing and documentation. Issues like using good variable names, using local variables rather than instance variables when possible, breaking large complex methods into smaller private methods, etc. fall into this category.

Educational value is our attempt to capture the degree to which you choose a project that promises to help you extend you own knowledge/skills. This is why questions 2 and 3 above are important. Educational value will not be judged strictly by how challenging a task you pick. It will be judged based on whether the task you picked appears to exercise and/or further develop the skills you learned in this course (or possibly in other courses).