CS 1110 Exam Success

Recommendations for preparing

1. **Be able to do this semester's assignments and labs so far fluently and "from scratch"**.
2. Do relevant problems from previous exams.
    1. You will want to be comfortable answering coding questions on paper, since that is how you'll take the exams.
    2. **Warning:** it is often difficult for students to recognize whether their answers are actually similar to or are actually distant from solutions we would accept as correct. So, rather than saying "oh, my solution looks about the same", we suggest you try out your answers by coding them up in Python where possible, and see what happens on test instances that the exam problems typically provide. You can also check with course staff members.
3. **Buddy up:** at office hours, lab, or via Ed Discussions, try to find a study partner who would be well-matched with you, and try solving problems together.
4. Go through the lecture slides, making sure you understand each example, including the code examples that are posted along with the lecture slides. Go over iClicker questions and their answers.

Strategies for answering coding questions

1. When asked to write a function body, always first read the specifications carefully: what are you supposed to return? Are you supposed to alter any lists or objects? What are the preconditions? Do you understand the given examples/test cases? I*f you aren't sure you understand a specification, ask.*
2. Do NOT spend time writing code that checks or asserts preconditions, in the interest of time. Don't worry about input that doesn't satisfy the preconditions.
3. After you write your answer, double-check that it gives the right answers on the test cases --- any we give you, plus any you think of. Also, double check that what your code returns on those test cases satisfies the specification.
4. Comment your code if you're doing anything unexpected. But don't overly comment - you don't have that much time.
5. Use variable names that make sense, so we have some idea of your intent.
6. If there's a portion of the problem that you can't do and a part that you *can*, you can try for partial credit by having a comment like

    ```
    # I don't know how to do <x>, but assume that variable `start`
    # contains ... <whatever it is you needed>"
    ```

    That way you can use variable start in the part of the code you know how to do.