

CS 1110 review session for prelim 2
Some overall notes
April 20, 2017

Announcements:

1. Some staff members are planning to hold other review sessions, sometime on the weekend or on Monday; announcements will be made.
2. A4 solutions are expected out this weekend; A4 grades are expected out Thursday April 27; Prelim grades are expected out May 1st.

Most important topics for the exam (not an exclusive list):

- Classes, methods, objects (as in A3 and A4 and various labs), sub-classes (lab 11)
- Iteration:
 - fluency with for-loops iterating through dictionaries, list items and through list indices (as in A3)
 - ability to write and analyze simple while-loops;
 - ability to understand a *loop invariant* and see whether a simple invariant is being correctly initialized and maintained by code.¹
- Recursion on sequences and object structures (as in A4, lab 9)

Questions from prior prelim 2's that are *not* “fair game” or require some modification:²

- In general, Fall prelim 2 class and sub-class questions have included sub-problems involving implementing getters and setters and asserting preconditions. We will not have such sub-problems, but other parts of the class and sub-class questions *are* fair game.
- In general, Spring 2015 and Spring 2016 use different variable naming conventions from what we use: we would reserve capital letters for class names, and use more evocative variable names.
- Fall prelim 2 solutions for which one-frame-drawn-per-line notation is used would need to be converted to our one-frame-per-function notation.
- 2013 Fall 6(b)
- 2014 Spring 6 partition2 is a bit too difficult for the level of practice we've had so far in the semester. The partition function is also a little hard, but within scope for our prelim 2.
- 2015 Spring 4(d), 6(b), 6(c)

¹For prelim 2, you are not responsible for knowing how to create your own invariants, or for the box notation in the invariant reading. These concepts will be tested on the final.

²Be warned that some solutions have typos. If your solution differs from the provided solution, *try actually running it*; it can be hard for beginning students to distinguish valid solution differences from invalid ones.

- 2015 Fall 3(a), 3(d), 6(a)

Comments:

1. Sources for more solved recursion problems were given in March 28 lecture slides. Lab 09's optional file has various recursion problems you can practice with.
2. Reading and worked examples for loops and invariants were posted with the April 18th lecture slides.
3. (repost from Piazza) Strategies for answering exam questions that involve coding:
 - (a) As mentioned in class, 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? if you aren't sure you understand a specification, ask.
 - (b) Once 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 you return satisfies the specification.
 - (c) Comment your code if you're doing anything unexpected. But don't overly comment - you don't have that much time. Use variable names that make sense, so we have some idea of your intent.
 - (d) If there's a portion of the problem you can't do and a part you can, you can try for partial credit by having a comment “# I don't know how to do <whatever>, but assume that variable starting_part contains ... (whatever it is you needed)”. That way you can use variable starting_part in the part of the code you can do.
4. (repost from Piazza, with slight modifications) Suggestions for prelim study strategies:

“Dear all, I just wanted to stress that a recommended strategy to study for the prelim (and, in general, how to learn how to program) is to do lots and lots of practice questions, such as those provided in the exam archive or in the demos we post on the lectures page or in the book or of course the assignments and labs, and to do some programming daily.

Analogy: If you were learning a new human language or a new martial art, your optimal strategy would NOT be to emphasize reading grammar texts or books, or to only try out speaking or some moves the night before the exam.

Instead, you would practice continuously, and in the case of a martial art, you would need to get hit in the face a number of times in order to develop the right blocking instincts.

Remember, each error message only makes you stronger!”