6.005 Recitation 9: Quiz Review

Administrivia

- Quiz 1 will be held in class on Wednesday, March 14.
- One page (double-sided) of crib sheet will be allowed.

Quiz Review

Define terms:

**static typing**

**static checking**

Define: (in relation to Java when applicable)

public, private, protected, instance variables, constructor, 'this', private, final, static, rep-

**dynamic typing**

**overloading**

**Specification:**
precondition – what is the keyword for this? (requires)
postcondition – (effects)
frame condition – (modifies)

A specification A is at least as strong as a specification B:

- A's precondition is no stronger than B's
- A's postcondition is no weaker than B's, for the states that satisfy B's precondition

Checked vs. Unchecked Exceptions

If a method might throw a checked exception, the possibility must be declared in its
signature. For an unchecked exception, in contrast, the compiler will not check for try-
catch or a throws declaration.

State machine semantics:

Formally, a state machine consists of:

Input alphabet : A well defined set of possible inputs
Output alphabet: A well defined set of possible outputs
States: A (finite) set of states Transitions
Rules: that define how to move from state to state for each possible input and state
combination
Start state: The state that the machine will be in initially
There are four common kinds of coverage for state machines:
- all-actions includes every event in at least some test
- all-states visits every state (of the abstract state machine) in at least some test
- all-transitions makes every legal transition in the state machine in some test
- all-paths explores every possible path of transitions through the state machine

**Grammars:**

production has form

non-terminal ::= expression of terminals and non-terminals and operators

option: A::= B?     an A is a B or is empty

grouping: A ::= (B C)* parentheses for grouping     an A is zero or more B-C pairs

1+iteration: A ::= B+     is equivalent to A ::= BB*     an A is one or more B’s character

classes:     A::= [abc] is equivalent to     A ::= a | b | c

A::= [^b]     is equivalent to     A::= a | c | d | e | f | ... (all other characters)

**Operators:**

- Creators create new objects of the type. A constructor may take an object as an argument, but not an object of the type being constructed.
- Producers create new objects from old objects of the type. The concat method of String, for example, is a producer: it takes two strings and produces a new one representing their concatenation.
- Mutators change objects. The add method of List, for example, mutates a list by adding an element to the end.
- Observers take objects of the abstract type and return objects of a different type. The size method of List, for example, returns an integer

**Visitors**

Visitors vs. Interpreters (pros and cons)

**Testing**

Partitioning stats, white box vs. black box testing