JUnit Tests

Clicker Question:

Be honest; I can't see your individual answers. How did you test your code in CSCI 150 (or whatever was your last programming class)?

A. I typed in the test cases I was given. If my program ran correctly with those it must have been okay.
B. I also tested extreme cases -- what happens if you get unexpected input.
C. I added print statements to my code to make sure that I got to certain parts of the code.
D. I was supposed to test my code?

JUnit is an Eclipse extension that helps with unit testing. The old way to test was to insert a bunch of print statements into your code to check the values of variables, then to run your program with specific inputs to see if it printed what you expected. This suffers from several problems:

- It is slow -- you have to insert the print statements then remove them when you are done.
- You have to re-enter the input each time you run a test.
- It is often not clear if your program passed or failed a test.
- You have to have a running program before you can start testing.

JUnit is designed to address these problems. Rather than modifying your code it creates a new class for testing. You enter your test cases once and run the test suite until your code passes all of the tests. The JUnit tests include *assertions* about the state of your program -- whether two things are equal, whether some condition is true, and so forth. If any of these assertions is incorrect when your program runs, it fails the test.

For example, to test your MyArrayList class you might create two variables

```
        MyArrayList<Integer> test = new MyArrayList<Integer>();
        ArrayList<Integer> real = new ArrayList<Integer>();
```

The *test* variable and the *real* variable should behave identically so you can go through a series of steps and assert that their behavior is the same:

```
        test.add(5);
        real.add(5);
        assertEquals("Size after addition", test.size(), real.size());
        assertEquals("First addition", test.get(0), real.get(0));
```

and so forth. Best of all, you don't have to modify your code to do the testing.