CS1632, Lecture 9: Unit Testing, part 1

Wonsun Ahn

What is unit testing?

- Unit testing involves testing the smallest coherent "units" of code, such as functions, methods, or classes.
- It is white-box; you are looking at and testing the code directly.
- Ensures that the smallest pieces of the code work correctly (NOT that they work correctly with the rest of the system – very localized)

Examples

1. Testing that a .sort method actually sorts elements
2. Testing that a nil/null throws an exception
3. Testing that a formatNumber method formats a number properly
4. Checking that passing in a string to a function which expects an integer does not crash
5. Testing that a .send and .receive method exist on a class

Unit testing

his is usually done by the developer writing the code, another
eveloper (esp. in pair programming), or (very occasionally), a
hite-box tester.

What’s the point?

1. Problems found earlier
2. Faster turnaround time
3. Developer understands issues with his/her code
4. "Living documentation“
5. Able to tell if your changes caused issues elsewhere by running full test suite

What do unit tests consist of?

- (optional) Set up code
- Preconditions
- Execution Steps
- Postconditions - a/k/a Assertions (a/k/a asserts, shoulds, musts)
- (optional) Tear down code

Example (in natural language, not code)

reate two linked lists with the same number of nodes and same data i
ch node.

compare them with the equality operator, they SHOULD be equal.

r "they MUST be equal.")

r "I ASSERT that they will be equal")

Postconditions = assertions

When you think "should" or "must", that is the assertion. It's what you're testing for.

It's the EXPECTED BEHAVIOR of the unit test.

When you execute the test, that's when you'll find out the OBSERVED BEHAVIOR.

If the expected behavior matches the observed behavior, the test passes; otherwise it fails.

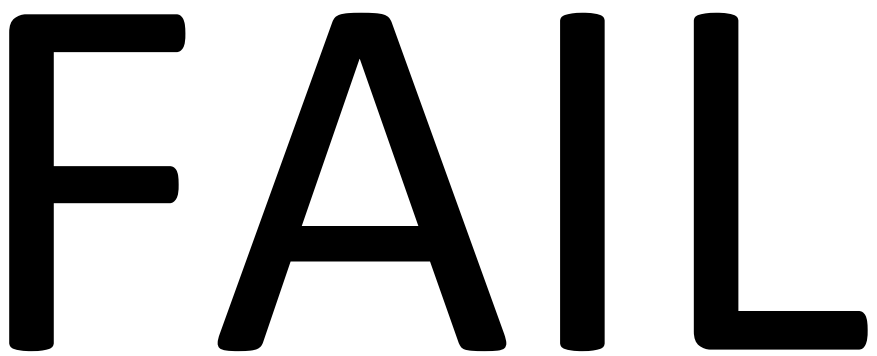
Junit assertions

Some possible assertions using JUnit:

assertEquals, assertArrayEquals, assertTrue, assertFalse, assertNull, assertNotNull, assertSame, assertThat(*something*), assertNotSame...

fail()

fail() makes a test automatically fail. Usually because you know it's going to fail anyways and you don't want to waste time running it, or check that a certain part of code is not reached.



Unit test example

```
//  0. A LL should always equal itself
@Test
public void testEqualsSelf() {
    LinkedList<Integer> ll = new
LinkedList<Integer>();
    assertEquals(ll, ll);
}
```

More linked list test examples

ample_code/LinkedListTest.java

Unit is not the only unit test framework out there

- Not even for Java!
- But the xUnit frameworks are common and easy to understand
- Ideas should apply to other testing frameworks easily

Now Please Read Textbook Chapter 13