jUnit 4 Summary

Exerpt from http://courses.**cs.washington.edu**/courses/cse331/11au/sections/section4-cheat-sheet.pdf

Method annotations:

tag	description

`@Test`
`@Test(timeout = time)`
`@Test(expected = exception.class)`
	Turns a public method into a JUnit test case. Adding a timeout will cause the test case to fail after **time** milliseconds. Adding an expected exception will cause the test case to fail if **exception** is not thrown.

`@Before`	Method to run before every test case
`@After`	Method to run after every test case
`@BeforeClass`	Method to run once, before any test cases have run
`@AfterClass`	Method to run once, after all test cases have run

Assertion methods:

method	description
`assertTrue(`**test**`)`	fails if the Boolean test is `false`
`assertFalse(`**test**`)`	fails if the Boolean test is `true`
`assertEquals(`**expected, actual**`)`	fails if the values are not equal
`assertSame(`**expected, actual**`)`	fails if the values are *not* the same (by ==)
`assertNotSame(`**expected, actual**`)`	fails if the values are the same (by ==)
`assertNull(`**value**`)`	fails if the given value is *not* `null`
`assertNotNull(`**value**`)`	fails if the given value is `null`
`fail()`	causes current test to immediately fail

Unit testing tips

- The entire goal is FAILURE ATOMICITY – the ability to know exactly what failed when a test case did not pass
- Tests should be self-contained and not care about each other
- You cannot test everything! Instead, think about:
    - boundary cases,
    - empty cases,
    - error cases
    - behavior in combination (but not to excess)
- Each test case should test ONE THING
    - 10 small tests are better than 1 test 10x as large
    - Rule of thumb: 1 assert statement per test case
    - Try to avoid complicated logic
- Torture tests are ok, but only in addition to simple tests

JUnit best practices:

- Use descriptive test names
- Add a default timeout to every test
- Use private methods to get rid of redundant test code
- Create test suites using @RunWith and @Suite.SuiteClasses to run tests for several classes at once
- Build quick arrays and collections using array literals
    - int[] quick = new int[] {1, 2, 3, 4};
    - List list = Arrays.asList(7, 4, -3, 18);
    - Set set = new HashSet(Arrays.asList(5, 6, 10));