Contents

Junit 4

Unit Testing

- Tests the individual subprograms, subroutines, or procedures to compare the function of the module to its specifications
- helps developers find errors in code.
- helps you write better code.
- saves time later in the production/development cycle.
- provides immediate feedback on the code.

TDD (Test Driven (First) Development)

It is a technique in which you write unit tests before writing the application functionality.

JUnit

- free, open source, software testing framework for Java.
- It is a library put in a jar file.
- It is not an automated testing tool.
- JUnit tests are Java classes that contain one or more unit test methods.
- Package : **org.junit.Test**

Annotations

- @Test – used to signify a method is a test method
- @Before – can do initialization task before each test run
- @After – cleanup task after each test is executed
- @BeforeClass – execute task before start of tests
- @AfterClass – execute cleanup task after all tests have completed
- @Ignore – to ignore the test method

Assert Statements

- Fail(String)
- assertTrue(boolean)
- assertEquals([String message],expected,actual)
- assertNull([message],object)
- assertNotNull([message],object)
- assertSame([String],expected,actual)
- assertNotSame([String],expected,actual)
- assertThat(String,T actual, Matcher<T> matcher)

Testing Exceptions

```
@Test(expected = ArithmeticException.class)
public void divideByZeroTest() {
calobj.divide(15,0);
}
```

Testing timeout Example

@Test(timeout=100) fails if the test takes longer than 100 milliseconds for execution.