**Junit**

**Whitebox Tesing** – Knowing the code/ functionality

**BlackBox Testing** – Without knowing the code and testing

**Unit Testing** – Testing small piece of code is Unit testing.

public int sum(int a, int b){ //src

If(a!=0 && b!=0)

return a+b;

}

public void testsum(){ //test

a=2, b=3;

if(5 == 5){

log(success);

}

else{

log(failed);

}

2, 3

-3+-6

9+0

null, 5

**TDD** – Test Driven Development – Based on the Test developer is enhancing his code.

Framework – is a semi finished application, which try to avoid redundancy of code

an essential supporting structure of a building, vehicle, or object

Boiler plate – test, utils, main, base

**Unit testing Frameworks** – Junit & Testng

**X-Unit** – Key Architecture for Unit Testing Framework

Java – Junit

Small Talk – Sunit

.Net – Nunit

**Junit Download**

1. Junit official website – download junit, hamcrest jars – add them to build path
2. Right click on project – properties- java build path – add libraries – junit 4
3. Adding Maven dependencies

**Junit 3**

Extend testcase

add test to all the test methods – or else it will not be treated as Junit test method

**Junit 4**

Everything is Annotation based

@BeforeClass – executes once before all classes

@AfterClass– executes once after all classes

@Before – executes before all the methods

@After– executes after all the methods

@Test – method / test case

@Ignore – If u don’t want to execute any particular method u can add @Ignore and execute

**timeout** – is an attribute for Test annotation.

**Expected** – is an attribute in Test Annotation, we are expecting for exception, if that

exception occurs, then the test method is passed or else it will be failed.

**Properties Class** – Utils Package – Used to read properties file. The data is stored in key value pair.

**Using Suiteclass.java**

@RunWith(Suite.Class)

@SuiteClasses(TestPrime.class, TestClac.class)

we can run the classes parallely.

**TestNG** – is also a unit testing framework, based on Junit n Nunit.

It is also derived from XUnit(Key Architecture)

Advantages of TestNg over Junit.

1. More annotation
2. All the configuration are done in XML file
3. Multithreaded environments and code is thread safe
4. No static methods declaration in Testng
5. It provides support for logging
6. Reporting is better than Junit
7. Group, add denpencies on test methods
8. Unit testing (small code till integration testing)

**Testng.xml**

Suite – we can have number of tests

test – we can have number of classes

classes – class

class – methods

**Annotations**-

@Test

@BeforeMethod

@AfterMethod

@Before class

@Afterclass

@BeforeSuite

@AfterSuite

@BeforeTest

@AfterTest

@DataProvider – we can pass test data to test method

**Attributes of Test**

**group –** we can group our test methods based on the their functionality.

**dependsOnMethods** – the dependent methods only excutes if the dependency is passed, setting some priority

**Harddependency** – if the dependency method is failed then the dependent methods will get skipped

**SoftDependency** - – if the dependency method is failed then the dependent methods will be executed,

we are only achiving some priority**(alwaysRun = true)**

**dependsOnGroups** – it will depend on group, only works when there is groups included in the test class

**innvocationCount –** It executes the test method for that particular number of times

**threadPoolSize** - we are giving some thread count to execute the test method parallely

**innvocationTimeOut** – we are giving some time and saying it has to execute in that particular amount of time for that invocationCount. if it exceeds then it throws threadTimeOutException.

**timeout** - we are giving some time and saying it has to execute in that particular amount of time. if it exceeds then it throws threadTimeOutException.

**expectedExceptions** – we can assert for exception

**Enabled** – if you don’t want to execute the test for particular run – then we can add enabled = false

**priority** – by default it is 0, all the priority test methods will be executing based on their priority.

@Parameters – is used to pass data from testing.xml,

<parameter name=*"input"* value=*"Hello World"*></parameter>

SoftAssert – is a class, and it acts as verify

IN TestNG.XML

Parallel – classes, tests, methods, instances, true, false

Classes – all the classes in the test will exceuting parallely

Junit – if you have any Junit test classes we can execute them in testing, using Junit = true

**Listeners** –

ITestListeners, IReporters interfaces, we can use for customized reports