1. **@BeforeTest** 🡪 This method will execute only once before any @Test Method execute irrespective of how many @Test Methods are present in one Class

**Ex :** In a class, if two @Test methods are present and one @BeforeTest method is present then @BeforeTest will execute only once

1. **@BeforeMethod 🡪** This method will execute every time before the @Test method execute.

**Ex :** If 2 @Test methods are present in a class then **@BeforeMethod** will execute 2 times

1. Also whenever we **extend Parent Class** always **Parent class @Test method** will **execute first** before child class @Test method.
2. **@AfterTest** 🡪 This method will execute only once **After the last @Test Method execute** irrespective of how many @Test Methods are present in one Class

**Ex :** In a class, if two @Test methods are present and one @BeforeTest method is present then @AfterTest will execute only once after @Test method 2

1. **@AfterMethod 🡪** This method will execute every time after the @Test method execute.

**Ex :** If 2 @Test methods are present in a class then **@AfterMethod** will execute 2 times after every @Test method.

1. All @Test methods will execute in alphabetical order if there is no priority parameter passed to @Test method.

Note: All the 6 above mentioned points are covered in below program

**package** TestNGPractice;

**import** org.testng.annotations.Test;

**public** **class** BaseTest {

@Test

**public** **void** basetestmethod1(){

System.***out***.println(" Parent Class Test Method 1 ");

}

@Test

**public** **void** basetestmethod2(){

System.***out***.println(" Parent Class Test Method 2 ");

}

}

-----------------------------------------------------------------------------------------------------------------------

**package** TestNGPractice;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** BeforeMethod\_Inheritance **extends** BaseTest{

@BeforeTest

**public** **void** BeforeTestMethod(){

System.***out***.println(" @BeforeTest Method ");

}

@AfterTest

**public** **void** AfterTestMethod(){

System.***out***.println(" @AfterTest Method ");

}

@BeforeMethod

**public** **void** BeforeMethosMethod(){

System.***out***.println(" @BeforeMethod Method ");

}

@AfterMethod

**public** **void** AfterMethosMethod(){

System.***out***.println(" @AfterMethod Method ");

}

@Test

**public** **void** childClassTestMethod1(){

System.***out***.println("Child class Test Method 1");

}

@Test

**public** **void** childClassTestMethod2(){

System.***out***.println("Child class Test Method 2");

}

}

Output :

[RemoteTestNG] detected TestNG version 6.13.1

**@BeforeTest Method**

**@BeforeMethod Method**

**Parent Class Test Method 1**

**@AfterMethod Method**

**@BeforeMethod Method**

**Parent Class Test Method 2**

**@AfterMethod Method**

**@BeforeMethod Method**

**Child class Test Method 1**

**@AfterMethod Method**

**@BeforeMethod Method**

**Child class Test Method 2**

**@AfterMethod Method**

**@AfterTest Method**

PASSED: basetestmethod1

PASSED: basetestmethod2

PASSED: childClassTestMethod1

PASSED: childClassTestMethod2

===============================================

Default test

Tests run: 4, Failures: 0, Skips: 0

===============================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**LL2) If we want to execute only selected @Test methods out of many, we can use groups option in TestNG**

Ex : Suppose if we have 4 @Test methods in a class and we want to run only selected methods then we should write code like below

**@Test(groups=”somename”)**

Ex Code:

**package** TestNGPractice;

**import** org.testng.annotations.Test;

**public** **class** GroupsRun {

@Test

**public** **void** test(){

System.***out***.println(" Default test method.");

}

@Test(groups="Priority1")

**public** **void** test1(){

System.***out***.println(" Priority1 test method.");

}

@Test(groups="Priority2")

**public** **void** test2(){

System.***out***.println(" Priority2 test method.");

}

@Test(groups="Priority3")

**public** **void** test3(){

System.***out***.println(" Priority3 test method.");

}

}

**Testing.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name=*"Suite"*>

<test thread-count=*"5"* name=*"Test"*>

<groups>

<run>

<include name=*"Priority3"*></include>

</run>

</groups>

<classes>

<class name=*"TestNGPractice.GroupsRun"*></class>

</classes>

</test>

<!-- Test -->

</suite> <!-- Suite -->

**Ouput : [**RemoteTestNG] detected TestNG version 6.13.1

Priority3 test method.

===============================================

Suite

Total tests run: 1, Failures: 0, Skips: 0

===============================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**LL3)** **@DataProvider**

Note : I usually prefer to use Microsoft Excel as the format for storing my parameters

**package** TestNGPractice;

**import** org.testng.annotations.DataProvider;

**import** org.testng.annotations.Test;

**public** **class** DataProviderAnnotation {

@Test(dataProvider="getData",invocationCount=1)

**public** **void** test1(String un, String pwd){

System.***out***.println(un+" "+pwd);

}

@DataProvider

**public** Object[][] getData(){

// i stands for number of times test case should run

// j stands for number of parameters it should send for one go

//Object[][] data=new Object[i][j];

Object[][] data=**new** Object[2][2];

data[0][0]="One\_UN";

data[0][1]="One\_PWD";

data[1][0]="Two\_UN";

data[1][1]="Two\_PWD";

**return** data;

}

}

* **test1()** method will execute **1 time** even though we mentioned ‘i’ value as 2 in data provider class**, this is because we mentioned invocationCount=1 in test1() method. In the above program it is mentioned like**

Object[i][j] = Object[2][2] , so i=2.

So test1() method is suppose to execute twice but it will execute 1 time since **invocationCount=1.**

* Suppose if we **remove** **invocationCount** and run the test1() method, then it will run 2 times since the number of ‘i’ value passing is 2 in Object[2][2].

So first priority is for **invocationCount**

* If we change ‘i’ value as 3 , then test1() method will execute 3 times only if invocation count is not defined in @Test method or it is equal to ‘i’ value.
* The @Test method that wants to receive data from this DataProvider needs to use a dataProvider name equals to the name of this annotation.
* The name of this data provider. If it's not supplied, the name of this data be provider will automatically set to the name of the method.

Ex : In the below program , I have mentioned @Dataprovider name as “likhi”, so I have to pass the same in @Test mentod as value to dataProvider parameter.

@Test(dataProvider="likhi",invocationCount=1)

**public** **void** test1(String un, String pwd){

System.***out***.println(un+" "+pwd);

}

@DataProvider(name="likhi")

**public** Object[][] getData(){

// i stands for number of times test case should run

// j stands for number of parameters it should send for one go

//Object[][] data=new Object[i][j];

Object[][] data=**new** Object[2][2];

data[0][0]="One\_UN";

data[0][1]="One\_PWD";

data[1][0]="Two\_UN";

data[1][1]="Two\_PWD";

**return** data;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**LL4) PARALLEL Execution:**

* TestNG provides multiple ways to execute tests in separate threads. In testng.xml, if we set**'parallel'**attribute on the tag to 'tests', testNG will run all the Test blocks in testing.xml file in the same thread, but each block will be in a separate thread.
* If we want to run methods/classes in separate threads, we need to set 'parallel' attribute on the tag to 'methods' / 'classes'
* This helps us to run test methods / classes / tests in parallel. By using parallel execution, we can reduce the 'execution time' as tests are executed simultaneously in different threads.
* **To run scripts parallel, value can be tests/classes/methods/suites. Default value is none**
* The below is the simple testng.xml file, if you observe, we are defining two attributes 'parallel' and 'thread-count' at suite level. As we want test methods to be executed in parallel, we have provided 'methods'. And 'thread-count' attribute is used to pass the number of maximum threads to be created.

<!DOCTYPE suite SYSTEM "[http://testng.org/testng-1.0.dtd">](http://testng.org/testng-1.0.dtd)

[<](http://testng.org/testng-1.0.dtd)**[suite](http://testng.org/testng-1.0.dtd)** name="Parallel test suite" parallel="methods" thread-count="2">

  <**test** name="Regression 1">

    <**classes**>

      <**class** name="com.parallel.TestParallelOne"/>

    </**classes**>

  </**test**>

</**suite**>

parallel="methods"/”classes”/”tests”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# LL5) Parameterization in TestNG using testng.xml

**package** com.parameterization;

**import** org.testng.annotations.Parameters;

**import** org.testng.annotations.Test;

**public** **class** **TestParameters** {

@Parameters({ "browser" })

@Test **public** **void** **testCaseOne**(String browser) {

System.out.println("browser passed as :- " + browser);

}

}

The below is the testng.xml file, in which we need to pass the parameter values for the test method

<!DOCTYPE suite SYSTEM "[http://testng.org/testng-1.0.dtd">](http://testng.org/testng-1.0.dtd)

[<](http://testng.org/testng-1.0.dtd)**[suite](http://testng.org/testng-1.0.dtd)** name="Parameterization Test Suite">

<**test** name="Testing Parameterization">

<**parameter** name="browser" value="Firefox"/>

<**classes**>

<**class** name="com.parameterization.TestParameters" />

</**classes**>

</**test**>

</**suite**>

In testng.xml, parameter values can be set at both suite and test level

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LL6) If we comment “<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">” in testing.xml file we will get a **warning** saying

“[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >" at the top of your file, otherwise TestNG may fail or not work as expected.”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LL7)