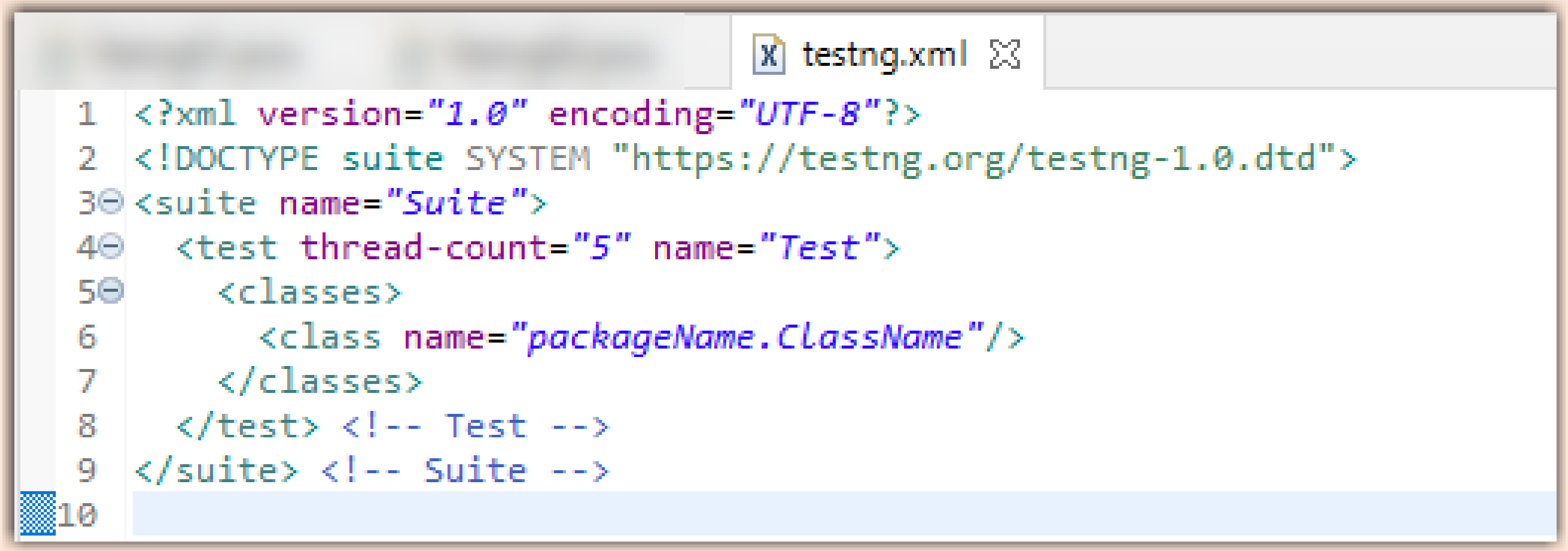
TestNG Slides

Following slides will be about “testng.xml” file.

Testng.xml file

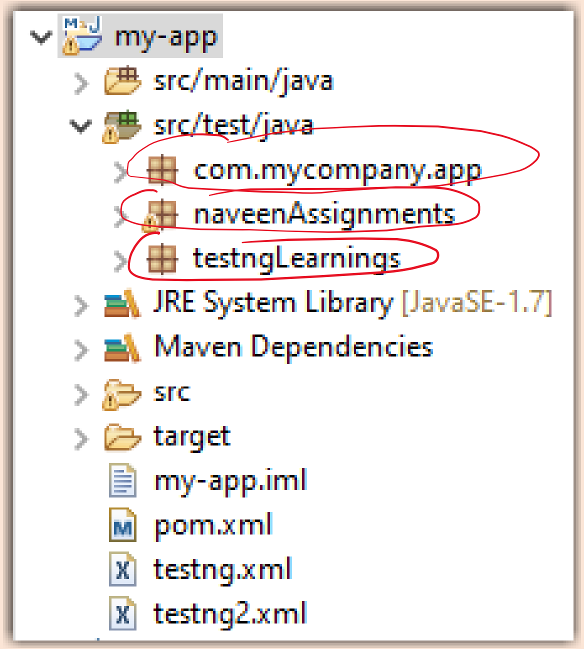


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>
      <class name="packageName.ClassName"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

① Suite
② Tests
↳ Classes
OR
↳ Packages

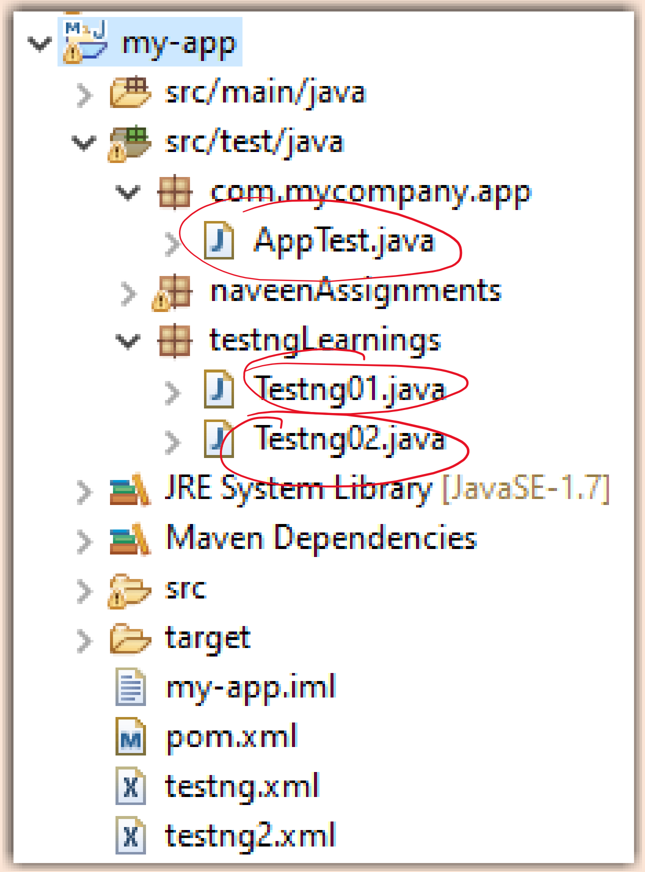
Inside <test> we can either have <classes> OR <packages>

Packages


my-app
src/main/java
src/test/java
com.mycompany.app
naveenAssignments
testngLearnings
JRE System Library [JavaSE-1.7]
Maven Dependencies
src
target
my-app.iml
pom.xml
testng.xml
testng2.xml

```
<packages>
    <package name="com.mycompany"/>
    <package name="testngLearnings"/>
</packages>
```

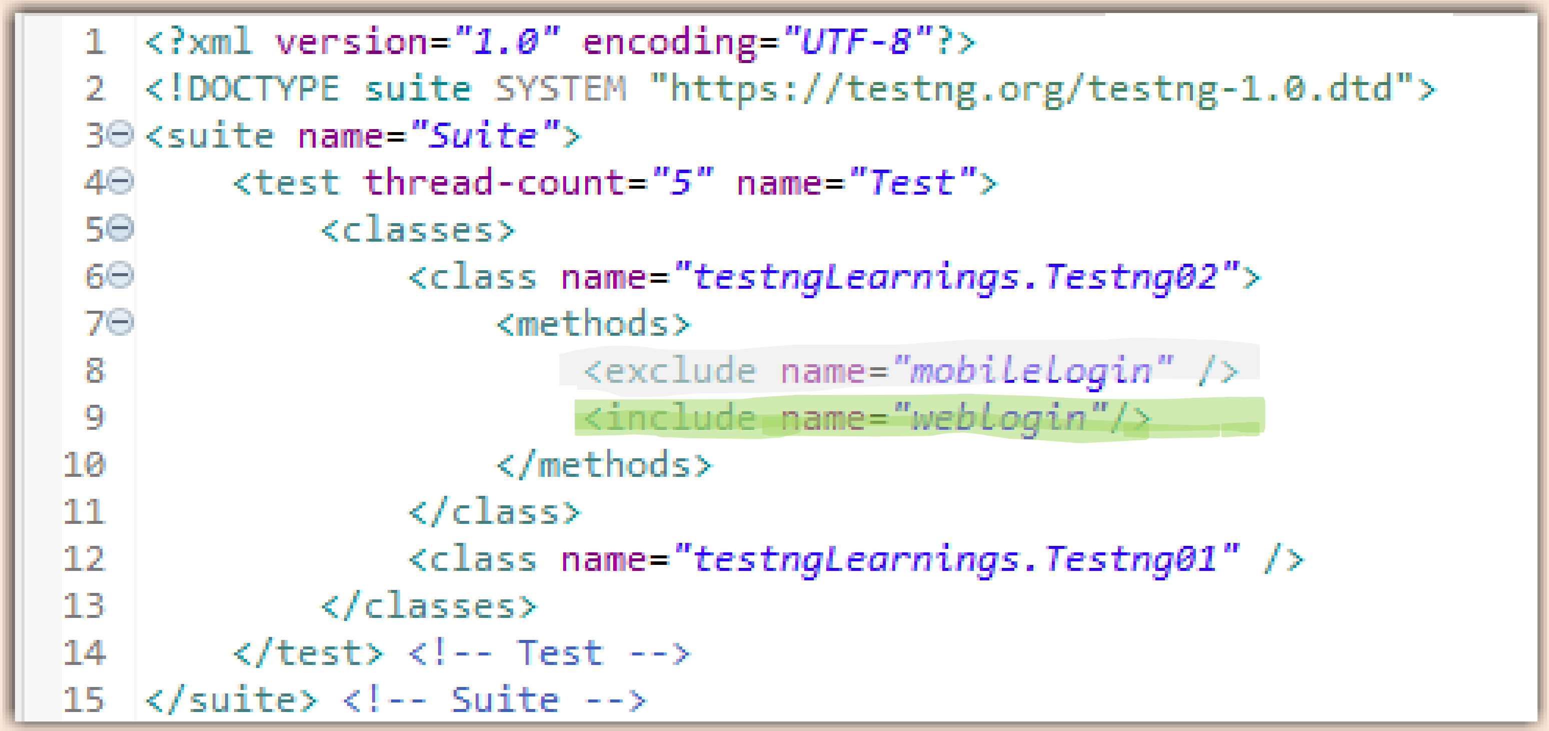
Classes



my-app
src/main/java
src/test/java
com.mycompany.app
AppTest.java
naveenAssignments
testngLearnings
Testng01.java
Testng02.java
JRE System Library [JavaSE-1.7]
Maven Dependencies
src
target
my-app.iml
pom.xml
testng.xml
testng2.xml

```
<classes>
        <class   name = "pack1.Test1" />
        <class   name = "pack2.Test2" />
        <class   name = "pack3.Test3" />

<classes>
```

Include/Exclude methods in testing.xml file:

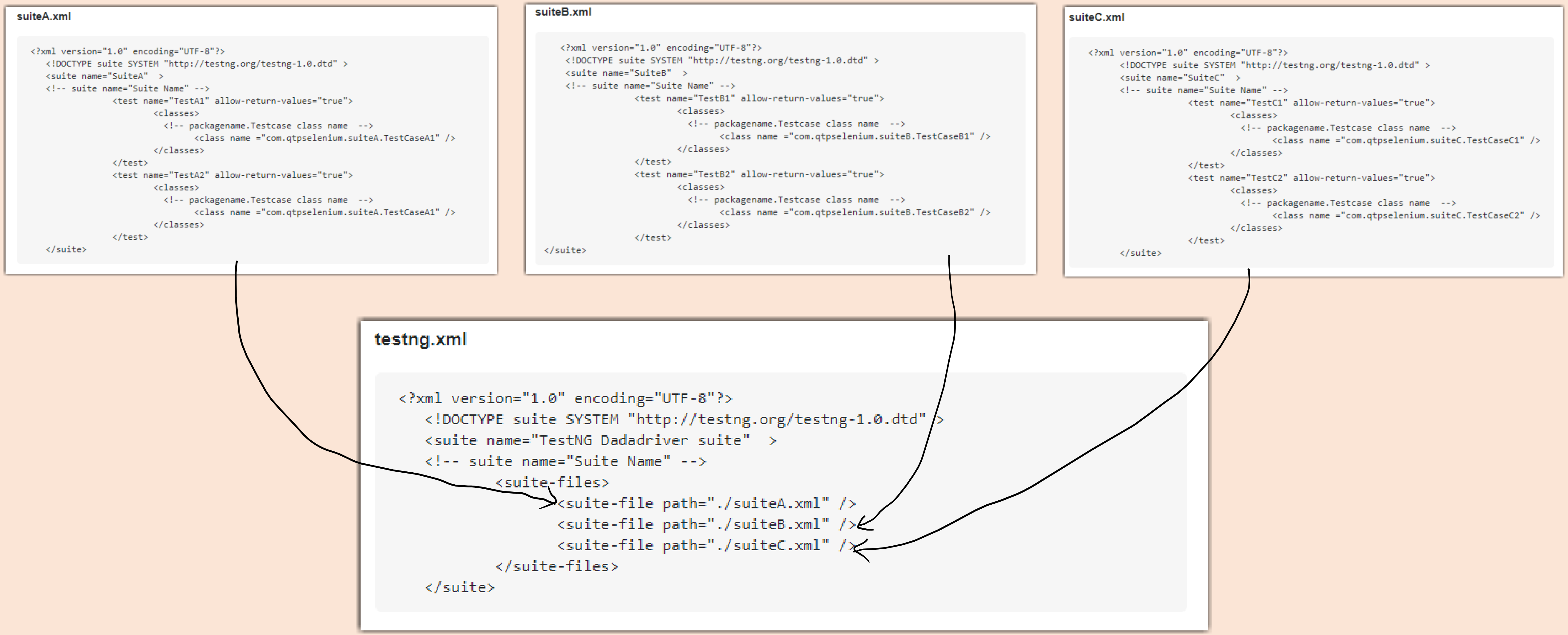


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <classes>
            <class name="testngLearnings.Testng02">
                <methods>
                    <exclude name="mobilelogin" />
                    <include name="weblogin"/>
                </methods>
            </class>
            <class name="testngLearnings.Testng01" />
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

- Suppose, from 100s of ‘tests(methods)’ we want to skip some ‘tests’, then we can use “exclude” function.
- Suppose, from 100s of ‘tests(methods)’ we want to run only some specific ‘tests’, then we can use “include” function.

Q: How to execute multiple suites?

A: Create multiple suiteX.xml file and in testng.xml file give the suite-file path.



suiteA.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
   <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
   <suite name="SuiteA"  >
   <!-- suite name="Suite Name" -->
                <test name="TestA1" allow-return-values="true">
                        <classes>
                          <!-- packagename.Testcase class name  -->
                                <class name ="com.qtpselenium.suiteA.TestCaseA1" />
                        </classes>
                </test>
                <test name="TestA2" allow-return-values="true">
                        <classes>
                          <!-- packagename.Testcase class name  -->
                                <class name ="com.qtpselenium.suiteA.TestCaseA1" />
                        </classes>
                </test>
   </suite>
```

suiteB.xml

```xml
    <?xml version="1.0" encoding="UTF-8"?>
     <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
     <suite name="SuiteB"  >
     <!-- suite name="Suite Name" -->
                  <test name="TestB1" allow-return-values="true">
                          <classes>
                            <!-- packagename.Testcase class name  -->
                                  <class name ="com.qtpselenium.suiteB.TestCaseB1" />
                          </classes>
                  </test>
                  <test name="TestB2" allow-return-values="true">
                          <classes>
                            <!-- packagename.Testcase class name  -->
                                  <class name ="com.qtpselenium.suiteB.TestCaseB2" />
                          </classes>
                  </test>
</suite>
```

suiteC.xml

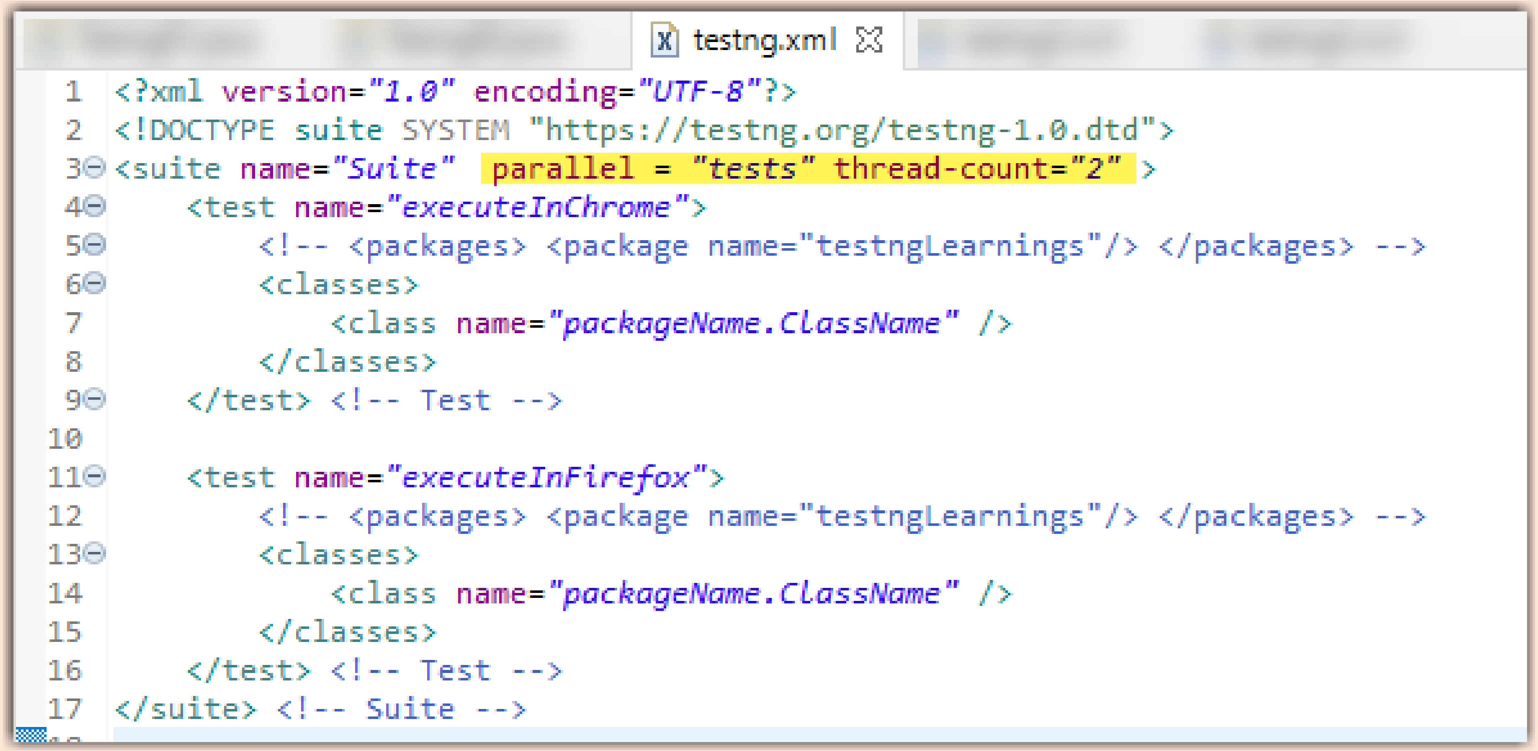
```xml
<?xml version="1.0" encoding="UTF-8"?>
      <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
      <suite name="SuiteC"  >
      <!-- suite name="Suite Name" -->
                   <test name="TestC1" allow-return-values="true">
                           <classes>
                             <!-- packagename.Testcase class name  -->
                                   <class name ="com.qtpselenium.suiteC.TestCaseC1" />
                           </classes>
                   </test>
                   <test name="TestC2" allow-return-values="true">
                           <classes>
                             <!-- packagename.Testcase class name  -->
                                   <class name ="com.qtpselenium.suiteC.TestCaseC2" />
                           </classes>
                   </test>
      </suite>
```

testng.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
   <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
   <suite name="TestNG Dadadriver suite"  >
   <!-- suite name="Suite Name" -->
           <suite-files>
                <suite-file path="./suiteA.xml" />
                <suite-file path="./suiteB.xml" />
                <suite-file path="./suiteC.xml" />
           </suite-files>
   </suite>
```

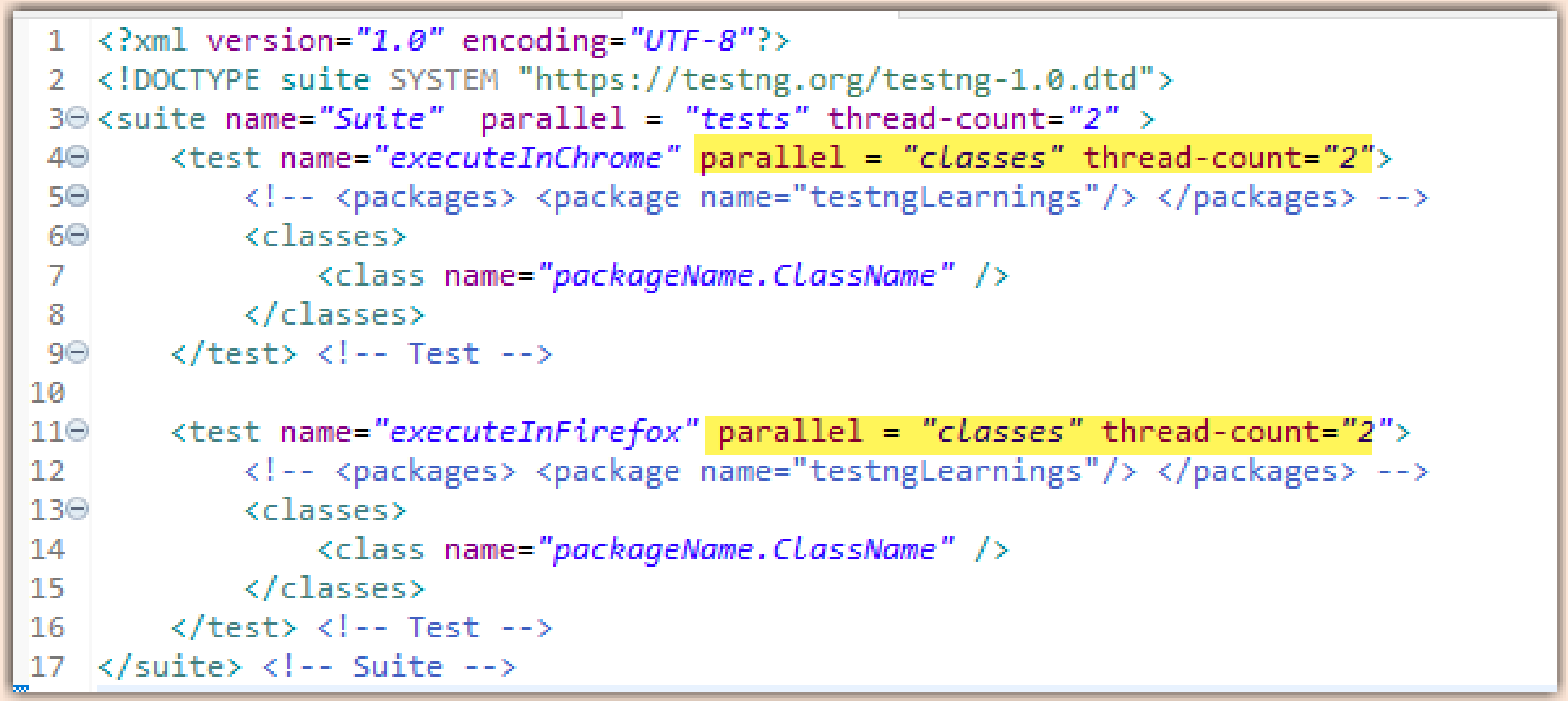
https://www.linkedin.com/in/imshaiknasir/ Source: https://stackoverflow.com/a/41755944

How to execute tests parallelly ?



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite"  parallel = "tests" thread-count="2" >
    <test name="executeInChrome">
        <!-- <packages> <package name="testngLearnings"/> </packages> -->
        <classes>
            <class name="packageName.ClassName" />
        </classes>
    </test> <!-- Test -->

    <test name="executeInFirefox">
        <!-- <packages> <package name="testngLearnings"/> </packages> -->
        <classes>
            <class name="packageName.ClassName" />
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

How to execute classes parallelly ?



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite"  parallel = "tests" thread-count="2" >
    <test name="executeInChrome" parallel = "classes" thread-count="2">
        <!-- <packages> <package name="testngLearnings"/> </packages> -->
        <classes>
            <class name="packageName.ClassName" />
        </classes>
    </test> <!-- Test -->

    <test name="executeInFirefox" parallel = "classes" thread-count="2">
        <!-- <packages> <package name="testngLearnings"/> </packages> -->
        <classes>
            <class name="packageName.ClassName" />
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

Following slides will be about “testng annotations”.

Hierarchy of Testng Annotations:

- *@BeforeSuite*
- *@BeforeTest*
- *@BeforeClass*
- *@BeforeMethod*

- *@Test*

- *@AfterMethod*
- *@AfterClass*
- *@AfterTest*
- *@AfterSuite*

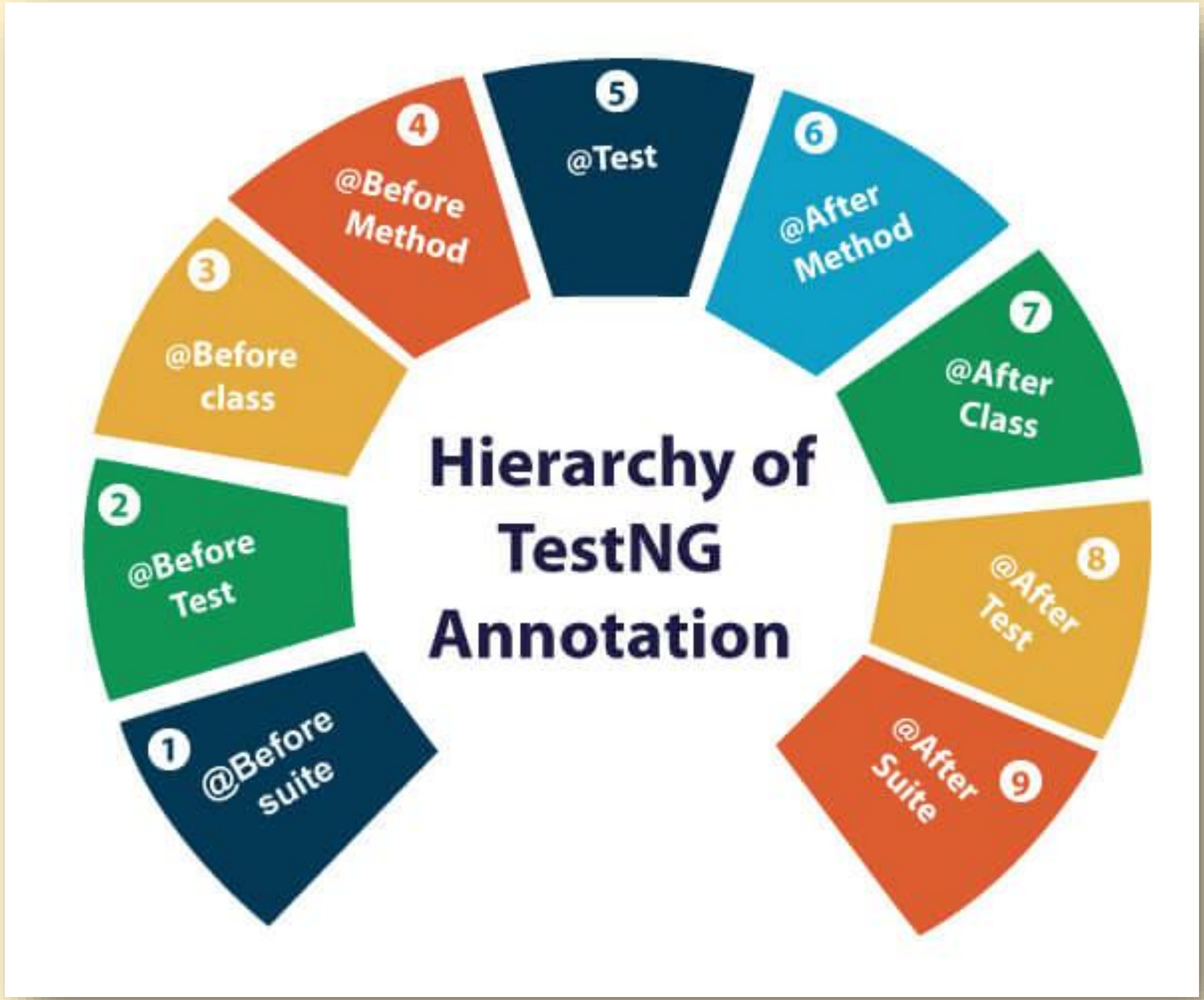


Image Source: https://static.javatpoint.com/tutorial/testng/images/hierarchy-of-testng-annotations.jpg

@BeforeSuite	The @BeforeSuite annotated method will run before the execution of all the test methods in the suite.

@BeforeTest	The @BeforeTest annotated method will be executed before the execution of all the test methods of available classes belonging to that folder.

@BeforeClass	The @BeforeClass annotated method will be executed before the first method of the current class is invoked.

@BeforeMethod	The @BeforeMethod annotated method will be executed before each test method will run.

@Test

@AfterMethod	The @AfterMethod annotated method will run after the execution of each test method.

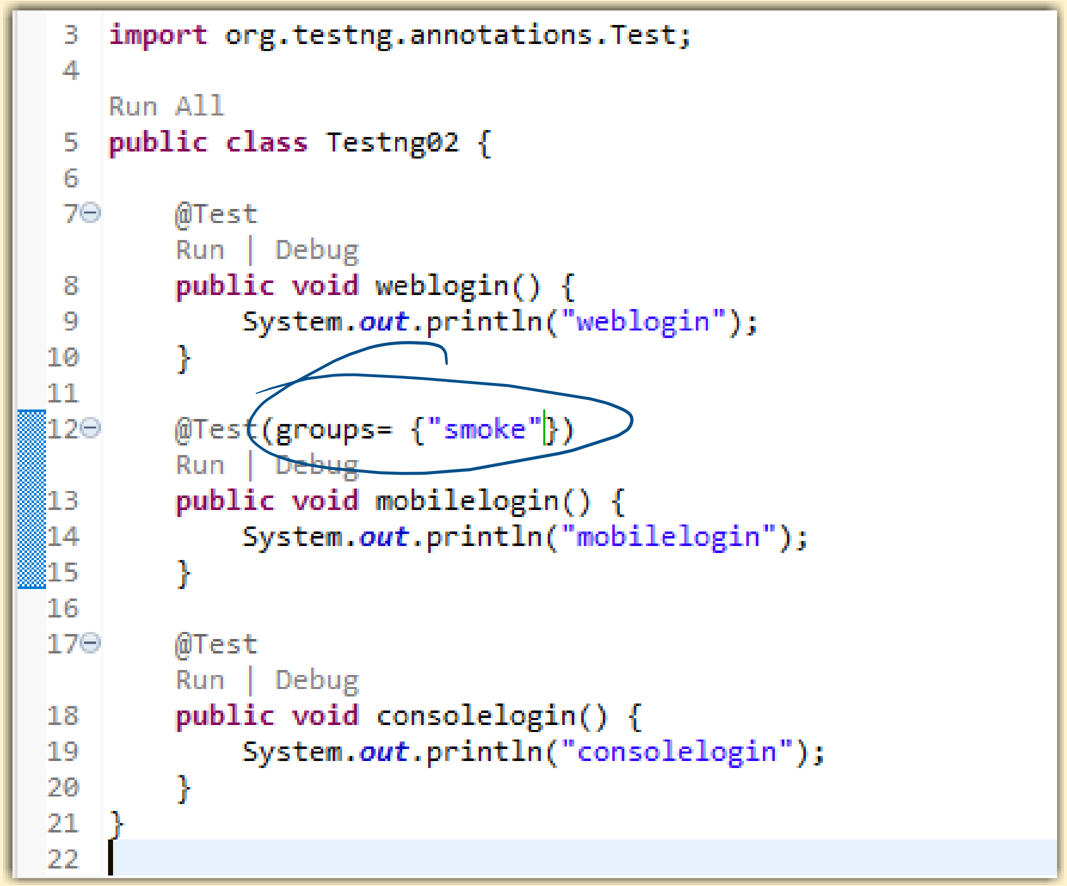
@AfterClass	The @AfterClass annotated method will be invoked after the execution of all the test methods of the current class.

@AfterTest	The @AfterTest annotated method will be executed after the execution of all the test methods of available classes belonging to that folder.

@AfterSuite	The @AfterSuite annotated method will run after the execution of all the test methods in the suite.

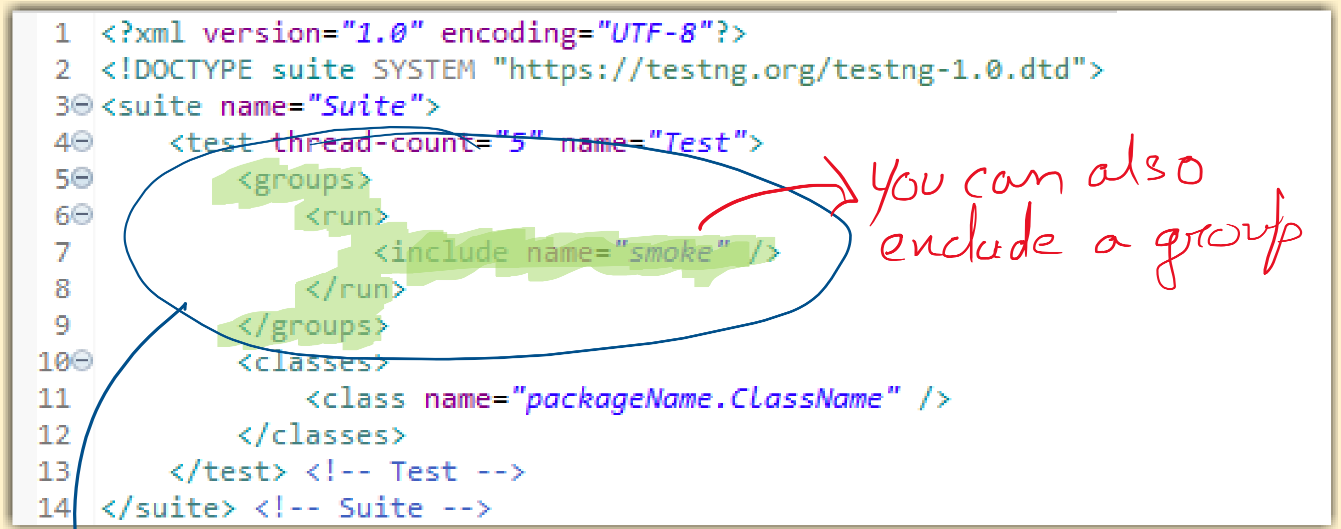
How to group test? And run those grouped tests at once ?

Test.java



```
import org.testng.annotations.Test;

Run All
public class Testng02 {

    @Test
    Run | Debug
    public void weblogin() {
        System.out.println("weblogin");
    }

    @Test(groups= {"smoke"})
    Run | Debug
    public void mobilelogin() {
        System.out.println("mobilelogin");
    }

    @Test
    Run | Debug
    public void consolelogin() {
        System.out.println("consolelogin");
    }
}
```

testng.xml



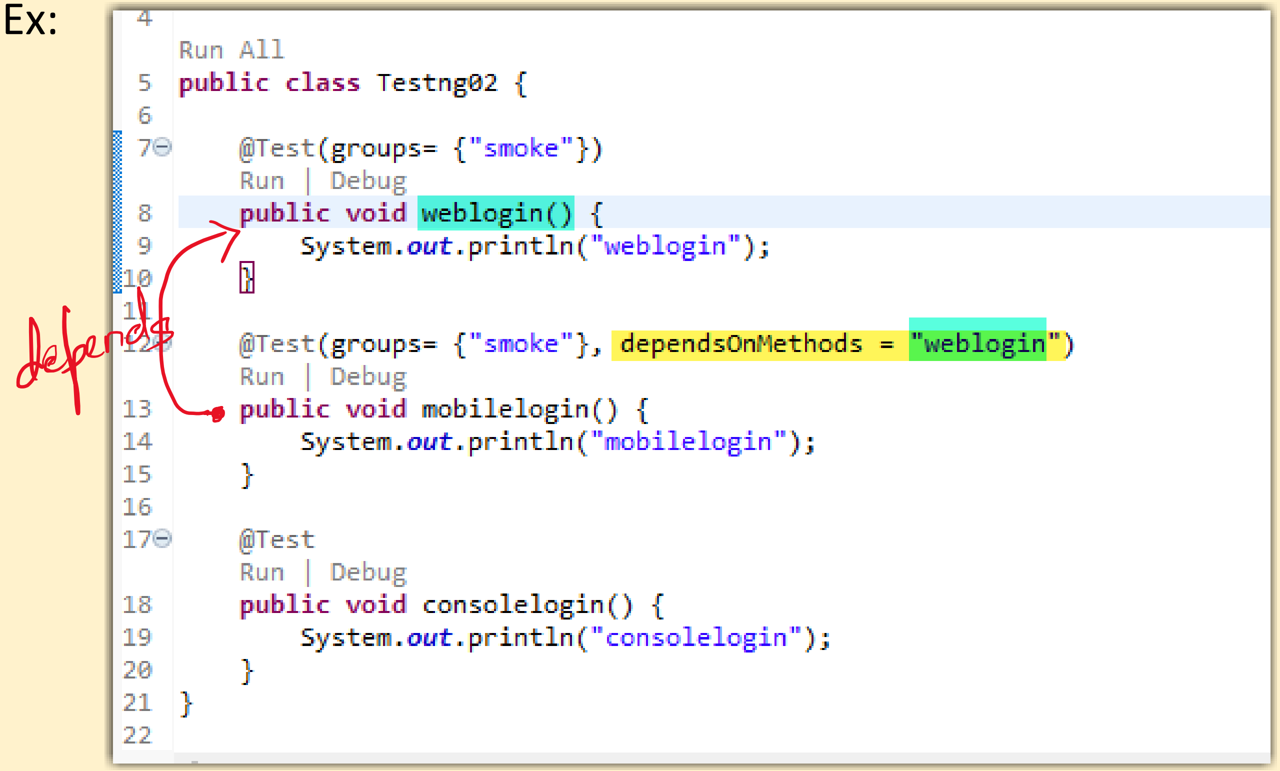
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <groups>
            <run>
                <include name="smoke" />
            </run>
        </groups>
        <classes>
            <class name="packageName.ClassName" />
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

You can also exclude a group

with the help of group, we can categorise the test to "smoke", "regression" or anything; & run those particular tests.

Dependency tests execution:

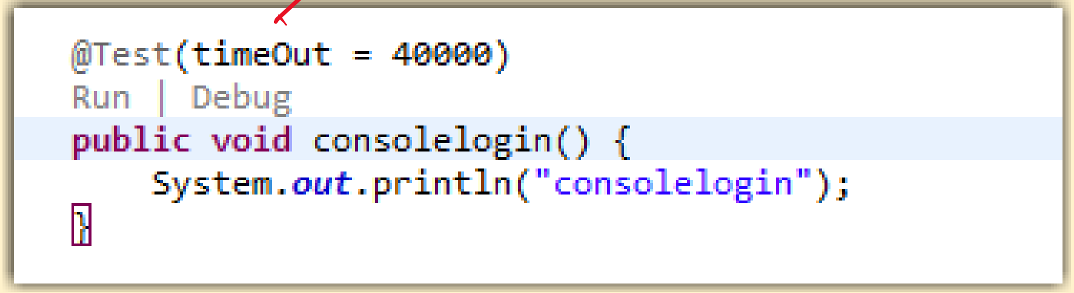
Example: 'TestB' depends on 'TestA'.
So unless TestA is executed, TestB can't be triggered.
In this case, we can use a dependency flag:
Ex:



```
Run All
public class Testng02 {

    @Test(groups= {"smoke"})
    Run | Debug
    public void weblogin() {
        System.out.println("weblogin");
    }

    @Test(groups= {"smoke"}, dependsOnMethods = "weblogin")
    Run | Debug
    public void mobilelogin() {
        System.out.println("mobilelogin");
    }

    @Test
    Run | Debug
    public void consolelogin() {
        System.out.println("consolelogin");
    }
}
```

“timeOut” Annotation helper:

Even though our code might have implicit waits as 10sec (for example), but this “timeOut” annotation helps in making sure that the test won’t fail until it reaches 40secs.



```
@Test(timeOut = 40000)
Run | Debug
public void consolelogin() {
    System.out.println("consolelogin");
}
```