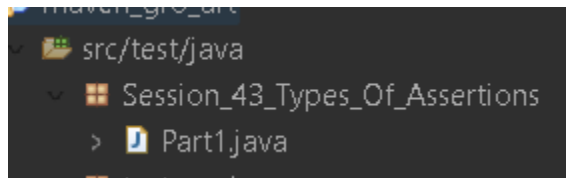


Create Package and Class:



Code 1: BeforeMethod and AfterMethod

```
package Session_43_Types_Of_Assertions;
```

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

```
public class Part1 {

    public static void p(Object...objs) {
        for (Object obj: objs) {
            System.out.print(obj);
        }
        System.out.println();
    }

    @BeforeMethod
    void login() {
        p("This is login...");
    }

    @AfterMethod
    void logout() {
        p("This is logout...");
    }

    @Test(priority = 1)
    void search() {
        p("This is search...");
    }

    @Test(priority = 2)
    void advancedsearch() {
        p("This is advancedsearch...");
    }

}
```

Notes:

Use [ctrl + shift + o] to import the module. Right click on the script -> Run As -> TestNG Test

Output:

```
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
This is login...
This is search...
This is logout...
This is login...
This is advancedsearch...
This is logout...
PASSED: Session_43_Types_Of_Assertions.Part1.advancedsearch
PASSED: Session_43_Types_Of_Assertions.Part1.search
```

```
=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====
```

```
=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====
```

Code 2: BeforeClass and AfterClass

```
package Session_43_Types_Of_Assertions;
```

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

```
public class Part2 {
    public static void p(Object...objs) {
        for (Object obj: objs) {
            System.out.print(obj);
        }
        System.out.println();
    }
}
```

```
@BeforeClass
```

```

void login() {
    p("This is login...");
}

@AfterClass
void logout() {
    p("This is logout...");
}

@Test(priority = 1)
void search() {
    p("This is search...");
}

@Test(priority = 2)
void advancedsearch() {
    p("This is advancedsearch...");
}
}

```

Output:

```

[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
This is login...
This is search...
This is advancedsearch...
This is logout...
PASSED: Session_43_Types_Of_Assertions.Part2.search
PASSED: Session_43_Types_Of_Assertions.Part2.advancedsearch

```

```

=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

```

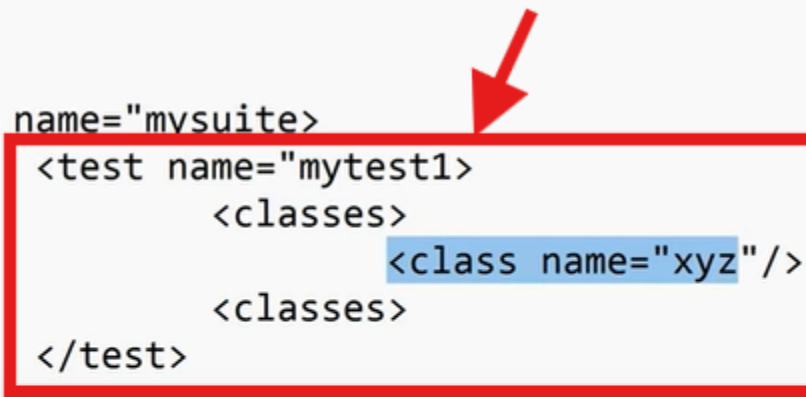
```

=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====

```

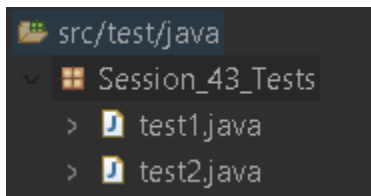
Single Test:

```
xml
-----
<suite name="mysuite">
  <test name="mytest1">
    <classes>
      <class name="xyz"/>
    </classes>
  </test>
  <test name="mytest2">
    <classes>
      <class name="abc"/>
    </classes>
  </test>
</suite>
```



In the xml context, @BeforeTest happens after each test and @AfterTest happens after each tests.

Create a package and classes



write the code and convert to TestNG

Code for test1.java:

```
package Session_43_Tests;
```

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

```
public class test1 {
    @Test
    void abc() {
        System.out.println("From abc...");
    }
}
```

```
        @BeforeTest
        void bt() {
            System.out.println("From bt...");
        }
    }
```

Code for test2.java:

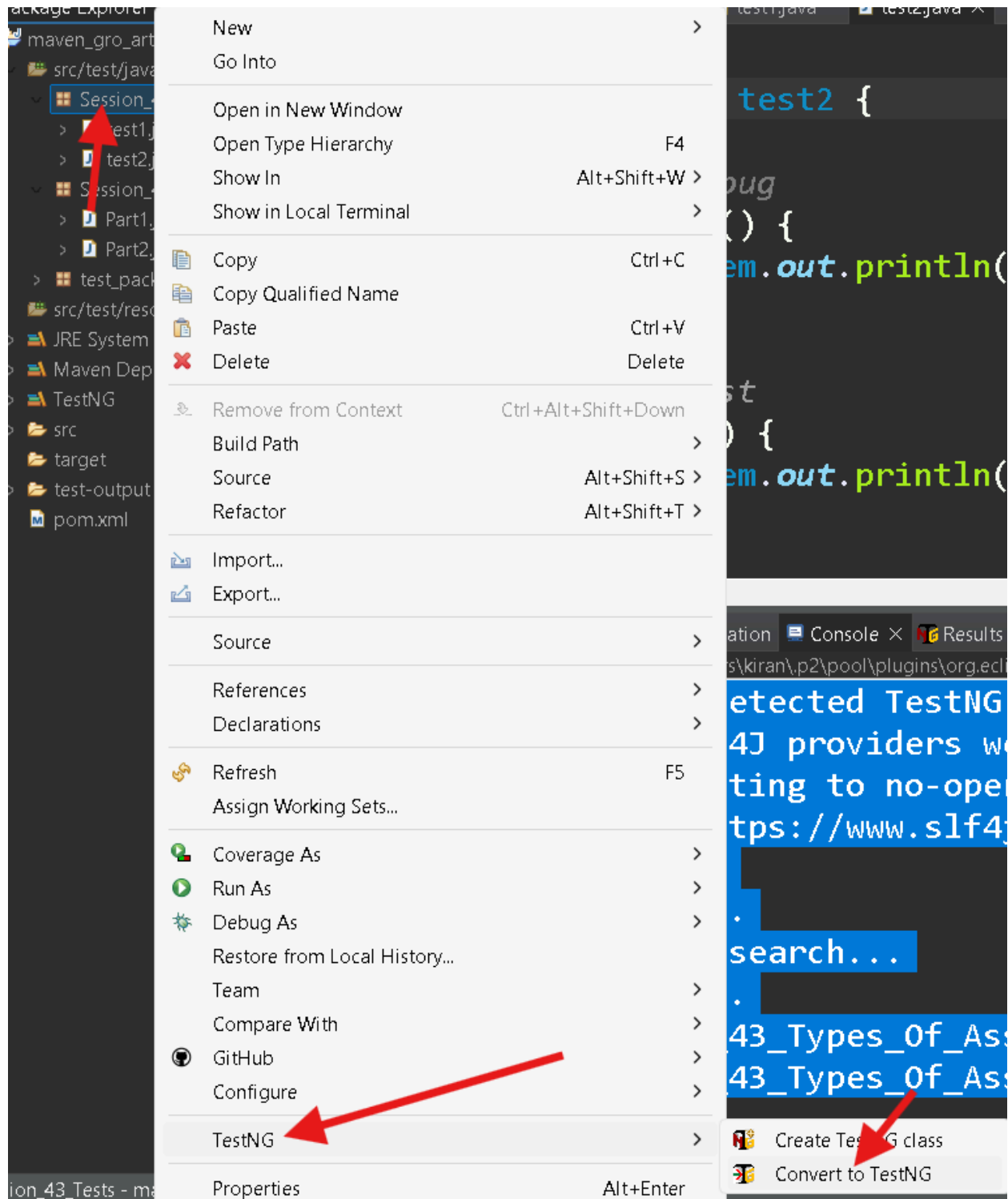
```
package Session_43_Tests;

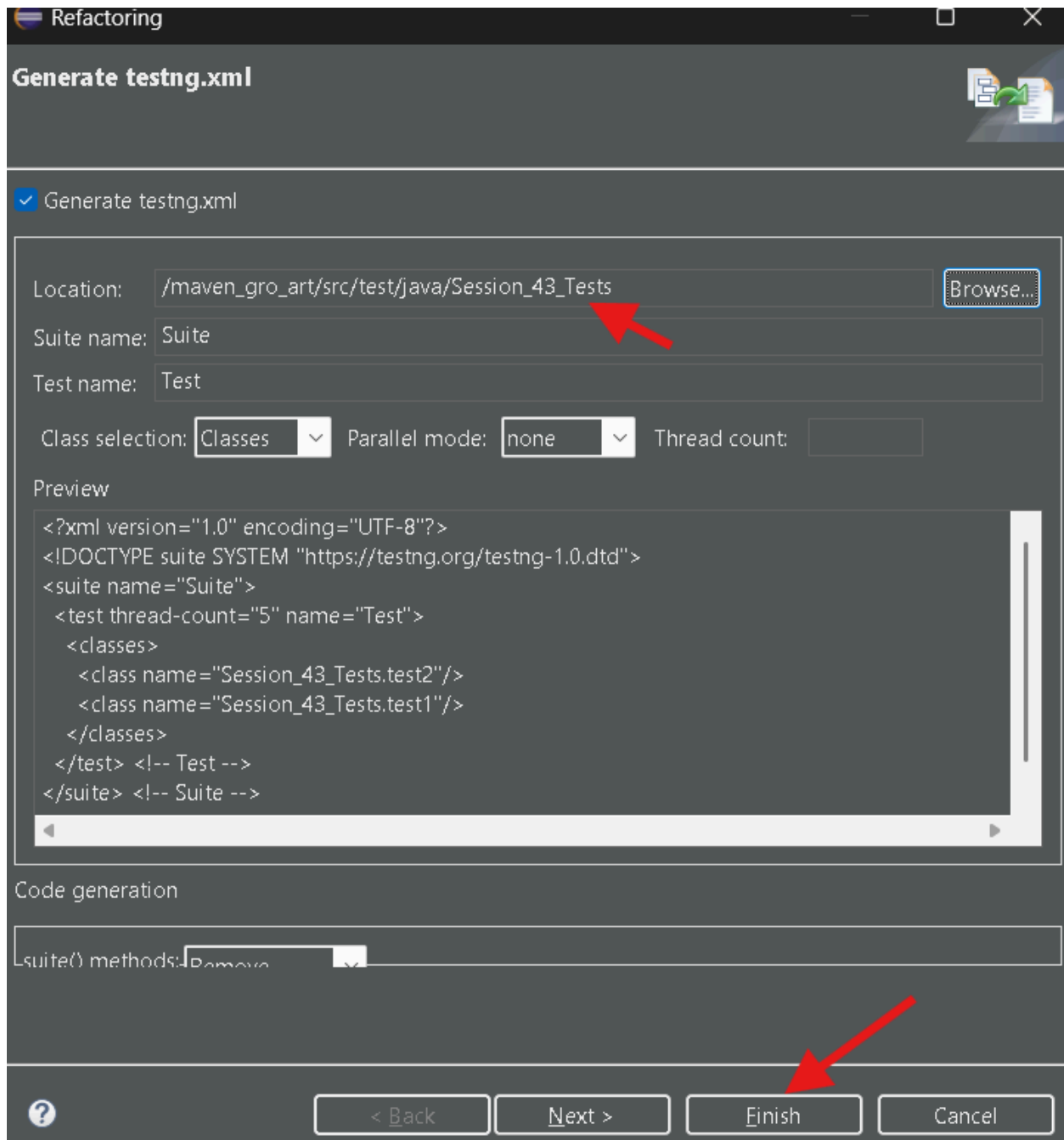
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

public class test1 {
    @Test
    void abc() {
        System.out.println("From abc...");
    }

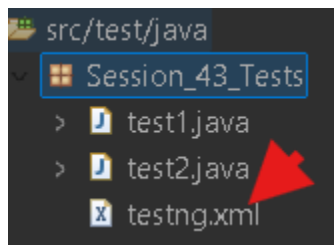
    @BeforeTest
    void bt() {
        System.out.println("From bt...");
    }
}
```

Convert to the TestNG:



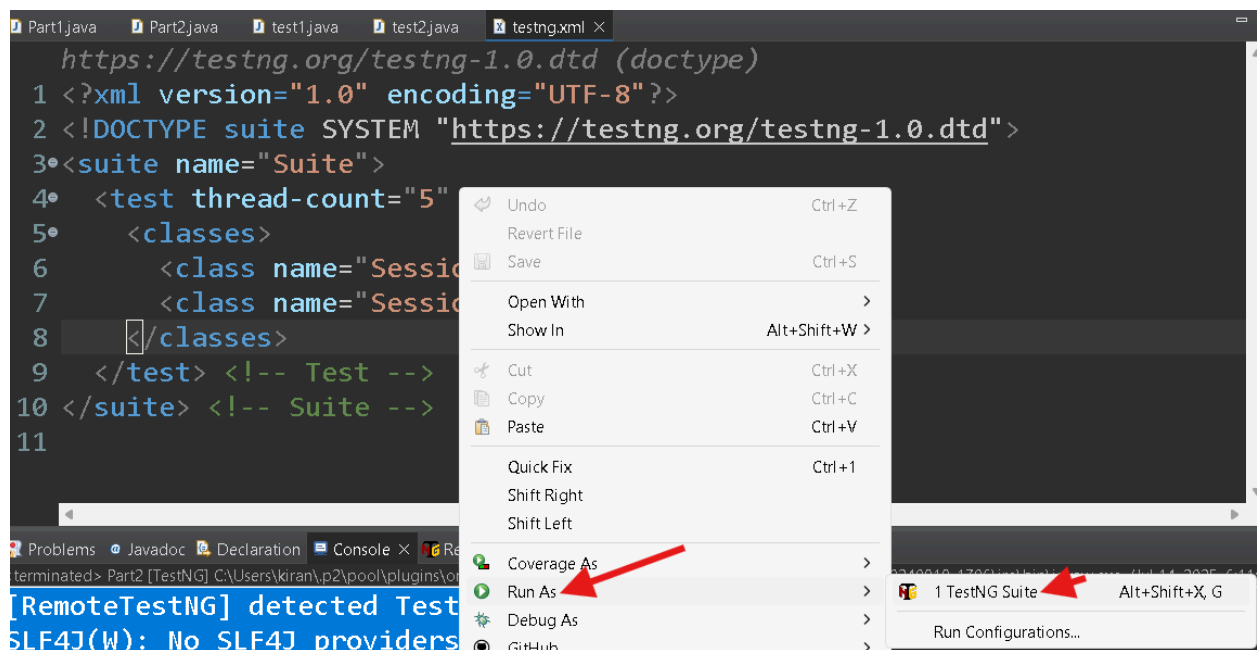


Testng.xml gets created:



Code:

```
<?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="Session_43_Tests.test2"/>
      <class name="Session_43_Tests.test1"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```



Output:

```
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
From bt...
From xyz...
From abc...
From at...
```

```
=====
Suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====
```


Code:

```
<?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_1">
    <classes>
      <class name="Session_43_Tests.test1"/>
      <class name="Session_43_Tests.test2"/>
    </classes>
  </test> <!-- Test -->
  <test thread-count="5" name="Test_2">
    <classes>
      <class name="Session_43_Tests.test1"/>
      <class name="Session_43_Tests.test2"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

Output:

```
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
From bt...
From abc...
From xyz...
From at...
From bt...
From abc...
From xyz...
From at...
```

```
=====
Suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
=====
```

Create class test3.java as:



Code in test3.java:

```
package Session_43_Tests;

import org.testng.annotations.AfterSuite;
import org.testng.annotations.BeforeSuite;

public class test3 {
    @BeforeSuite
    void bs() {
        System.out.println("From bs...");
    }

    @AfterSuite
    void as() {
        System.out.println("From as...");
    }
}
```

Code in 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_1">
    <classes>
      <class name="Session_43_Tests.test1"/>
      <class name="Session_43_Tests.test2"/>
      <class name="Session_43_Tests.test3"/>
    </classes>
  </test> <!-- Test -->
  <test thread-count="5" name="Test_2">
    <classes>
      <class name="Session_43_Tests.test1"/>
      <class name="Session_43_Tests.test2"/>
      <class name="Session_43_Tests.test3"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

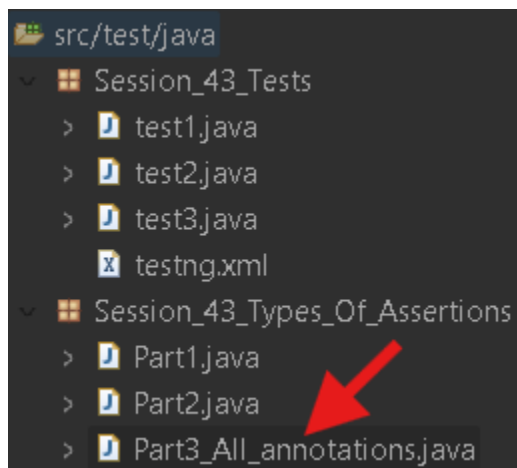
Output:

```
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
From bs...
```

From bt...
From abc...
From xyz...
From at...
From bt...
From abc...
From xyz...
From at...
From as...

```
=====
Suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
=====
Suite has test, test has classes, classes has methods.
```

Create All_Annotations class:



Code:

```
package Session_43_Types_Of_Assertions;

import org.testng.annotations.*;

public class Part3_All_annotations {
    public static void p(Object...objs) {
        for (Object obj:objs) {
            System.out.print(obj);
        }
        System.out.println();
    }
    @BeforeSuite
    void bs() {p("From BeforeSuite");}
```

```

    @AfterSuite
    void as() {p("From AfterSuite");}
    @BeforeTest
    void bt() {p("From BeforeTest");}
    @AfterTest
    void at() {p("From AfterTest");}
    @BeforeClass
    void bc() {p("From BeforeClass");}
    @AfterClass
    void ac() {p("From AfterClass");}
    @BeforeMethod
    void bm() {p("From BeforeMethod");}
    @AfterMethod
    void am() {p("From AfterMethod");}
    @Test(priority = 1)
    void t1() {p("first test");}
    @Test(priority = 2)
    void t2() {p("second test");}
}

```

After running the TestNG Test:

```

[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
From BeforeSuite
From BeforeTest
From BeforeClass
From BeforeMethod
first test
From AfterMethod
From BeforeMethod
second test
From AfterMethod
From AfterClass
From AfterTest
PASSED: Session_43_Types_Of_Assertions.Part3_All_annotations.t2
PASSED: Session_43_Types_Of_Assertions.Part3_All_annotations.t1

```

```

=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

```

From AfterSuite

=====

Default suite

Total tests run: 2, Passes: 2, Failures: 0, Skips: 0

=====

Create testng.xml file for all annotation java class:

```
<?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="Session_43_Types_Of_Assertions.Part3_All_annotations"/>
```

```
    </classes>
```

```
  </test> <!-- Test -->
```

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

Run the testng suite:

[RemoteTestNG] detected TestNG version 7.11.0

SLF4J(W): No SLF4J providers were found.

SLF4J(W): Defaulting to no-operation (NOP) logger implementation

SLF4J(W): See <https://www.slf4j.org/codes.html#noProviders> for further details.

From BeforeSuite

From BeforeTest

From BeforeClass

From BeforeMethod

first test

From AfterMethod

From BeforeMethod

second test

From AfterMethod

From AfterClass

From AfterTest

From AfterSuite

=====

Suite

Total tests run: 2, Passes: 2, Failures: 0, Skips: 0

=====

Suite > test > class > method

Naming conventions: where test is included in the method name like: testTitle()

Disadvantages of using if statement is it still shows PASSED even if title is not same.

The screenshot shows an IDE with a project named 'Session_43_Types_Of_Assertions'. The 'Part4_Assertions_demo.java' file is open, containing the following code:

```
1 package Session_43_Types_Of_Assertions;
2
3 import org.testng.annotations.Test;
4
5 Run All
6 public class Part4_Assertions_demo {
7     @Test
8     Run | Debug
9     void testTitle() {
10         String expected_title = "opencart";
11         String actual_title = "openshop";
12         if (expected_title.equals(actual_title)) System.out.println("Test passed!");
13         else System.out.println("Test failed!");
14     }
15 }
```

The console output shows the following messages:

```
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Test failed!
PASSED: Session_43_Types_Of_Assertions.Part4_Assertions_demo.testTitle
```

The screenshot shows the same IDE with the 'Part4_Assertions_demo.java' file open, containing the following code:

```
1 package Session_43_Types_Of_Assertions;
2
3 import org.testng.Assert;
4 import org.testng.annotations.Test;
5
6 Run All
7 public class Part4_Assertions_demo {
8     @Test
9     Run | Debug
10     void testTitle() {
11         String expected_title = "opencart";
12         String actual_title = "openshop";
13         Assert.assertEquals(expected_title, actual_title);
14     }
15 }
```

Run as TestNG Test:

[RemoteTestNG] detected TestNG version 7.11.0

SLF4J(W): No SLF4J providers were found.

SLF4J(W): Defaulting to no-operation (NOP) logger implementation

SLF4J(W): See <https://www.slf4j.org/codes.html#noProviders> for further details.

FAILED: Session_43_Types_Of_Assertions.Part4_Assertions_demo.testTitle

java.lang.AssertionError: **expected [openshop] but found [opencart]**

at org.testng.Assert.fail(Assert.java:111)

at org.testng.Assert.failNotEquals(Assert.java:1590)

at org.testng.Assert.assertEqualsImpl(Assert.java:150)

at org.testng.Assert.assertEquals(Assert.java:132)

at org.testng.Assert.assertEquals(Assert.java:656)

at org.testng.Assert.assertEquals(Assert.java:666)

at

Session_43_Types_Of_Assertions.Part4_Assertions_demo.testTitle(Part4_Assertions_demo.java:11)

```
        at
java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor
.java:103)
        at java.base/java.lang.reflect.Method.invoke(Method.java:580)
        at
org.testng.internal.invokers.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.jav
a:141)
        at org.testng.internal.invokers.TestInvoker.invokeMethod(TestInvoker.java:687)
        at org.testng.internal.invokers.TestInvoker.invokeTestMethod(TestInvoker.java:230)
        at org.testng.internal.invokers.MethodRunner.runInSequence(MethodRunner.java:63)
        at
org.testng.internal.invokers.TestInvoker$MethodInvocationAgent.invoke(TestInvoker.java:995)
        at org.testng.internal.invokers.TestInvoker.invokeTestMethods(TestInvoker.java:203)
        at
org.testng.internal.invokers.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:154)
        at org.testng.internal.invokers.TestMethodWorker.run(TestMethodWorker.java:134)
        at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
        at org.testng.TestRunner.privateRun(TestRunner.java:741)
        at org.testng.TestRunner.run(TestRunner.java:616)
        at org.testng.SuiteRunner.runTest(SuiteRunner.java:421)
        at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:413)
        at org.testng.SuiteRunner.privateRun(SuiteRunner.java:373)
        at org.testng.SuiteRunner.run(SuiteRunner.java:312)
        at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
        at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95)
        at org.testng.TestNG.runSuitesSequentially(TestNG.java:1274)
        at org.testng.TestNG.runSuitesLocally(TestNG.java:1208)
        at org.testng.TestNG.runSuites(TestNG.java:1112)
        at org.testng.TestNG.run(TestNG.java:1079)
        at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)
        at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:293)
        at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:91)
```

```
=====
Default test
Tests run: 1, Failures: 1, Skips: 0
=====
```

```
=====
Default suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====
```

Code:

```
package Session_43_Types_Of_Assertions;

import org.testng.Assert;
import org.testng.annotations.Test;

public class Part4_Assertions_demo {
    @Test
    void testTitle() {
        String expected_title = "opencart";
        String actual_title = "openshop";
        if (expected_title.equals(actual_title)) {
            System.out.println("Passed! message...");
            Assert.assertTrue(true);
        } else {
            System.out.println("Failed! message...");
            Assert.assertTrue(false);
        }
    }
}
```

Output:

```
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Failed! message...
FAILED: Session_43_Types_Of_Assertions.Part4_Assertions_demo.testTitle
java.lang.AssertionError: expected [true] but found [false]
    at org.testng.Assert.fail(Assert.java:111)
    at org.testng.Assert.failNotEquals(Assert.java:1590)
    at org.testng.Assert.assertTrue(Assert.java:57)
    at org.testng.Assert.assertTrue(Assert.java:67)
    at
Session_43_Types_Of_Assertions.Part4_Assertions_demo.testTitle(Part4_Assertions_demo.java:16)
    at
java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
    at java.base/java.lang.reflect.Method.invoke(Method.java:580)
    at
org.testng.internal.invokers.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:141)
    at org.testng.internal.invokers.TestInvoker.invokeMethod(TestInvoker.java:687)
```



```

        at org.testng.internal.invokers.TestInvoker.invokeTestMethod(TestInvoker.java:230)
        at org.testng.internal.invokers.MethodRunner.runInSequence(MethodRunner.java:63)
        at
org.testng.internal.invokers.TestInvoker$MethodInvocationAgent.invoke(TestInvoker.java:995)
        at org.testng.internal.invokers.TestInvoker.invokeTestMethods(TestInvoker.java:203)
        at
org.testng.internal.invokers.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:154)
        at org.testng.internal.invokers.TestMethodWorker.run(TestMethodWorker.java:134)
        at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
        at org.testng.TestRunner.privateRun(TestRunner.java:741)
        at org.testng.TestRunner.run(TestRunner.java:616)
        at org.testng.SuiteRunner.runTest(SuiteRunner.java:421)
        at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:413)
        at org.testng.SuiteRunner.privateRun(SuiteRunner.java:373)
        at org.testng.SuiteRunner.run(SuiteRunner.java:312)
        at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
        at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95)
        at org.testng.TestNG.runSuitesSequentially(TestNG.java:1274)
        at org.testng.TestNG.runSuitesLocally(TestNG.java:1208)
        at org.testng.TestNG.runSuites(TestNG.java:1112)
        at org.testng.TestNG.run(TestNG.java:1079)
        at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)
        at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:293)
        at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:91)

```

```

=====
Default test
Tests run: 1, Failures: 1, Skips: 0
=====

```

```

=====
Default suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====

```

There are two main types of assertions:

1. **Hard Assertions**
2. **Soft Assertions**

Hard Assertions:

```
package Session_43_Types_Of_Assertions;
```

```
import org.testng.Assert;
```

```

import org.testng.annotations.Test;

public class Part5_Hard_Assertions {
    @Test
    void test() {
//        For strings
        Assert.assertEquals("test1", "test1");
//        For integers
        Assert.assertEquals(123, 123);
        Assert.assertNotEquals(123, "123");
        Assert.assertTrue(1 == 1);
        Assert.assertFalse(1 == 2);
//        without any condition it gets failed due to
//        Assert.fail();
    }
}

```

Run as TestNG Test

Output:

```

[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
PASSED: Session_43_Types_Of_Assertions.Part5_Hard_Assertions.test

```

```

=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

```

```

=====
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====

```

Code:

```

package Session_43_Types_Of_Assertions;

import org.testng.Assert;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;

public class Part6_Hard_Vs_Soft_Assertions {

```

```

@Test(priority = 1)
void test_Hard_Assertions() {
    System.out.println("1First line...");
    System.out.println("1Second line...");
    Assert.assertEquals(true, false);
//    In hard assertions if about test fails it does not execute below code
//    So make sure assertion must be placed at the last line
    System.out.println("1third line...");
    System.out.println("1forth line...");
}
@Test(priority = 2)
void test_Soft_Assertions() {
    System.out.println("2First line...");
    System.out.println("2Second line...");
    SoftAssert sa = new SoftAssert();
    sa.assertEquals(true, false);
//    In Soft assertions if about test fails it does execute below code
//    So make sure assertion must be placed at the last line
//    PASSED:
Session_43_Types_Of_Assertions.Part6_Hard_Vs_Soft_Assertions.test_Soft_Assertions
    System.out.println("2third line...");
    System.out.println("2forth line...");
    // need to write below code otherwise assertions gets true even if it fails
    sa.assertAll();
}
}

```

Output:

```

[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
1First line...
1Second line...
2First line...
2Second line...
2third line...
2forth line...
FAILED:
Session_43_Types_Of_Assertions.Part6_Hard_Vs_Soft_Assertions.test_Soft_Assertions
java.lang.AssertionError: The following asserts failed:
    expected [false] but found [true]
    at org.testng.asserts.SoftAssert.assertAll(SoftAssert.java:46)
    at org.testng.asserts.SoftAssert.assertAll(SoftAssert.java:30)

```

```
    at
Session_43_Types_Of_Assertions.Part6_Hard_Vs_Soft_Assertions.test_Soft_Assertions(Part6
_Hard_Vs_Soft_Assertions.java:30)
    at
java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor
.java:103)
    at java.base/java.lang.reflect.Method.invoke(Method.java:580)
    at
org.testng.internal.invokers.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.jav
a:141)
    at org.testng.internal.invokers.TestInvoker.invokeMethod(TestInvoker.java:687)
    at org.testng.internal.invokers.TestInvoker.invokeTestMethod(TestInvoker.java:230)
    at org.testng.internal.invokers.MethodRunner.runInSequence(MethodRunner.java:63)
    at
org.testng.internal.invokers.TestInvoker$MethodInvocationAgent.invoke(TestInvoker.java:995)
    at org.testng.internal.invokers.TestInvoker.invokeTestMethods(TestInvoker.java:203)
    at
org.testng.internal.invokers.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:154)
    at org.testng.internal.invokers.TestMethodWorker.run(TestMethodWorker.java:134)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
    at org.testng.TestRunner.privateRun(TestRunner.java:741)
    at org.testng.TestRunner.run(TestRunner.java:616)
    at org.testng.SuiteRunner.runTest(SuiteRunner.java:421)
    at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:413)
    at org.testng.SuiteRunner.privateRun(SuiteRunner.java:373)
    at org.testng.SuiteRunner.run(SuiteRunner.java:312)
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95)
    at org.testng.TestNG.runSuitesSequentially(TestNG.java:1274)
    at org.testng.TestNG.runSuitesLocally(TestNG.java:1208)
    at org.testng.TestNG.runSuites(TestNG.java:1112)
    at org.testng.TestNG.run(TestNG.java:1079)
    at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)
    at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:293)
    at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:91)
```

FAILED:

```
Session_43_Types_Of_Assertions.Part6_Hard_Vs_Soft_Assertions.test_Hard_Assertions
java.lang.AssertionError: expected [false] but found [true]
    at org.testng.Assert.fail(Assert.java:111)
    at org.testng.Assert.failNotEquals(Assert.java:1590)
    at org.testng.Assert.assertEqualsImpl(Assert.java:150)
    at org.testng.Assert.assertEquals(Assert.java:132)
    at org.testng.Assert.assertEquals(Assert.java:1079)
```

```
    at org.testng.Assert.assertEquals(Assert.java:1043)
    at org.testng.Assert.assertEquals(Assert.java:1089)
    at
Session_43_Types_Of_Assertions.Part6_Hard_Vs_Soft_Assertions.test_Hard_Assertions(Part6
_Hard_Vs_Soft_Assertions.java:12)
    at
java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor
.java:103)
    at java.base/java.lang.reflect.Method.invoke(Method.java:580)
    at
org.testng.internal.invokers.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.jav
a:141)
    at org.testng.internal.invokers.TestInvoker.invokeMethod(TestInvoker.java:687)
    at org.testng.internal.invokers.TestInvoker.invokeTestMethod(TestInvoker.java:230)
    at org.testng.internal.invokers.MethodRunner.runInSequence(MethodRunner.java:63)
    at
org.testng.internal.invokers.TestInvoker$MethodInvocationAgent.invoke(TestInvoker.java:995)
    at org.testng.internal.invokers.TestInvoker.invokeTestMethods(TestInvoker.java:203)
    at
org.testng.internal.invokers.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:154)
    at org.testng.internal.invokers.TestMethodWorker.run(TestMethodWorker.java:134)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
    at org.testng.TestRunner.privateRun(TestRunner.java:741)
    at org.testng.TestRunner.run(TestRunner.java:616)
    at org.testng.SuiteRunner.runTest(SuiteRunner.java:421)
    at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:413)
    at org.testng.SuiteRunner.privateRun(SuiteRunner.java:373)
    at org.testng.SuiteRunner.run(SuiteRunner.java:312)
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95)
    at org.testng.TestNG.runSuitesSequentially(TestNG.java:1274)
    at org.testng.TestNG.runSuitesLocally(TestNG.java:1208)
    at org.testng.TestNG.runSuites(TestNG.java:1112)
    at org.testng.TestNG.run(TestNG.java:1079)
    at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)
    at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:293)
    at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:91)
```

```
=====
Default test
Tests run: 2, Failures: 2, Skips: 0
=====
```

```
=====
Default suite
Total tests run: 2, Passes: 0, Failures: 2, Skips: 0
=====
```