```
package TestNGKeyWords;

import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

public class InvocationCountUse {
 @Test(invocationCount = 3)
 public void myTest()
 {
	Reporter.log("Hello",true);
 }

 @BeforeMethod
 public void test()
 {
	Reporter.log("Hi",true);
 }
}
```

```
package TestNGKeyWords;

import org.testng.Reporter;
import org.testng.annotations.Test;

public class PriorityUse {
 @Test(priority = -2)
 public void d()
 {
	Reporter.log("d is running",true);
 }
 @Test
 public void a()
 {
         Reporter.log("a is running",true);
 }
 @Test(priority = -1)
 public void v()
 {
         Reporter.log("v is running",true);
 }
}
```

```
package TestNGKeyWords;

import org.testng.Reporter;
import org.testng.annotations.Test;

public class EnableUse {
        @Test
         public void d()
         {
                 Reporter.log("d is running",true);
         }
         @Test(enabled = false )
         public void a()
         {
                 Reporter.log("a is running",true);
         }
         @Test
```

```
    public void v()
    {
        Reporter.log("v is running",true);
    }
}
```

```
package TestNGKeyWords;

import org.testng.Reporter;
import org.testng.annotations.Test;

public class TimeOutUse {
   @Test(timeOut = 1000)
    public void d() throws InterruptedException
    {
       Thread.sleep(2000);
        Reporter.log("d is running",true);
    }
    @Test
    public void a()
    {
        Reporter.log("a is running",true);
    }
    @Test
    public void v()
    {
        Reporter.log("v is running",true);
    }
}
```

```
package TestNGKeyWords;

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

public class DependtsOnMethodUSe {
	@Test
	 public void d()
	 {
		//Assert.fail();
		Reporter.log("d is running",true);
	 }
	 @Test(dependsOnMethods = {"d"}, priority =- 2)
	 public void a()
	 {
		 Reporter.log("a is running",true);
	 }
	 @Test
	 public void v()
	 {
		 Reporter.log("v is running",true);
	 }
}
```

```
package IncludeExclude;

import org.testng.Reporter;
import org.testng.annotations.Test;

public class MyTestClass {
	@Test
	 public void d()
	 {
		 Reporter.log("d is running",true);
		}
		@Test
		public void a()
		{
			Reporter.log("a is running",true);
		}
		@Test
		public void v()
		{
			Reporter.log("v is running",true);
		}
}
```

Include from single 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="IncludeExclude.MyTestClass">
      <methods>
      <include name="a"/>
      </methods>
      </class>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

Exclude from single 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="IncludeExclude.MyTestClass">
      <methods>
      <exclude name="a"/>
      </methods>
      </class>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

```java
package IncludeExclude;


import org.testng.Reporter;
import org.testng.annotations.Test;


public class MyTestClass2 {
	@Test
	 public void x()
	 {
		 Reporter.log("x is running",true);
	 }
	 @Test
	 public void y()
	 {
		 Reporter.log("y is running",true);
	 }
	 @Test
	 public void z()
	 {
		 Reporter.log("z is running",true);
	 }
}
```

Include or exclude from multiple classes

```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="IncludeExclude.MyTestClass">
      <methods>

      <include name="a"/>
      </methods>

      </class>
      <class name="IncludeExclude.MyTestClass2">
```

```
   <methods>
   <exclude name="x"/>

   </methods>
   </class>
  </classes>
 </test> <!-- Test -->
</suite> <!-- Suite -->
```