# How to Set Priority of Test Cases in TestNG

## What is Priority in TestNG?

TestNG provides one more feature priority which is used to set the order of test cases as ours need.

Basically, Priority is an attribute that tells TestNG which orders have to follow to run the test method.

When we have multiple test cases and want to execute them in order, in this case, TestNG priority attribute is useful to execute the test cases in order. It has the following general form.

@Test(priority = 1) // Valid syntax.

  @Test(PRIORITY = 1) // Invalid syntax. IDE will show it as a compilation error.

If you are running multiple test cases and have not mentioned test priority, TestNG will assign all @Test a priority as zero(0) and execute them. i.e. the default value is zero for priority. Lower priorities will be scheduled first for execution.

If we define priority for multiple test cases, these test cases will get executed only after execution of all the test cases which don’t have any priority as the default priority will set to “priority = 0”.

Let’s understand a program where a class has no prioritized method.

## TestNG Tests without Priority

Let’s create a TestNG program where we will not define any priority with the test method. Look at the following source code.

**Program source code 1:**

package testNGPriority;

import org.testng.annotations.Test;

public class NoPriorityEx

{

@Test

public void one()

{

System.out.println("First");

}

@Test

public void two()

{

System.out.println("Second");

}

@Test

public void three()

{

System.out.println("Third");

}

}

Output:

First

Third

Second

In the preceding example program, we have declared a class that contains three test methods without any priority such as one(), two(), and three().

As you can see from the above output, TestNG executed test methods based on the alphabetical order of their method names irrespective of their place of implementation in the code.

So, to execute test methods as our needs, we will have to set priority with @Test annotation. So, let’s understand how to set priority of test cases in TestNG.

## How to Set Priority of Test Cases in TestNG?

Let’s take an example program where we will define priority with test methods to execute in sequential order.

**Program source code 2:**

package testNGPriority;

import org.testng.annotations.Test;

public class PriorityOrder

{

@Test(priority = 1)

public void one()

{

System.out.println("First");

}

@Test(priority = 2)

public void two()

{

System.out.println("Second");

}

@Test(priority = 3)

public void three()

{

System.out.println("Third");

}

}

Output:

First

Second

Third

After assigning priority to each test method, and after running the above code, the output has changed.

As you can see in the above output, the test method having lower priority is executed first i.e. execution happened in sequential order according to the priority assigned for the test method.

Thus, alphabetical order method name has not been considered as we provided the priorities and all test cases have been executed as our needs.

## Test Methods with Same Priority

Let’s make a program where we will define test methods with the same priority and check that TestNG executes them in which order.

**Program source code 3:**

package testNGPriority;

import org.testng.annotations.Test;

public class SamePriorityMethodEx

{

@Test(priority = 1)

public void firstTest()

{

System.out.println("First test method");

}

@Test(priority = 1)

public void secondTest()

{

System.out.println("Second test method");

}

@Test(priority = 0)

public void thirdTest()

{

System.out.println("Third test method");

}

@Test(priority = -1)

public void fourthTest()

{

System.out.println("Fourth test method");

}

@Test(priority = -2)

public void fifthTest()

{

System.out.println("Fifth test method");

}

}

Output:

Fifth test method

Fourth test method

Third test method

First test method

Second test method

In the preceding example program, a class contains five test methods in which two methods have the same priority. In these cases, TestNG considers the alphabetical order of the method names whose priority is the same.

Out of five test methods, three methods are executed based on their priority values. But methods firstTest() and secondTest() contains the same priority value (1).

So, TestNG considers the alphabetical order of ‘f’ and ’s’ and executes them accordingly.

## Combination of Prioritized and Non-prioritized methods

Let’s create a program where we will cover two cases (methods having the same priority value) and (more than one non-prioritized method) in one TestNG class.

Look at the following program source code to understand better.

**Program source code 4:**

package testNGPriority;

import org.testng.annotations.Test;

public class TestNGPriorityTest

{

@Test

public void c\_m1()

{

System.out.println("c\_m1");

}

@Test

public void b\_m1()

{

System.out.println("b\_m1");

}

@Test(priority = 1)

public void a\_m1()

{

System.out.println("a\_m1");

}

@Test(priority = 1)

public void d\_m1()

{

System.out.println("d\_m1");

}

@Test(priority = 0)

public void e\_m1()

{

System.out.println("e\_m1");

}

@Test(priority = 2)

public void f\_m1()

{

System.out.println("f\_m1");

}

}

Output:

b\_m1

c\_m1

e\_m1

a\_m1

d\_m1

f\_m1

**First preference:** In the preceding class, non-prioritized test methods b\_m1() and c\_m1() have been executed based on alphabetical order ‘b’ and then ‘c’.

**Second preference:** Prioritized methods e\_m1() has been executed first because of having the highest priority (0). As the priority of test methods a\_m1() and d\_m1() have the same.

Therefore, TestNG considered the alphabetical order of their methods names. So, between them, “a\_m1” was executed first and then “d\_m1”.

**Third preference:** Last, f\_m1() is executed by TestNG because it has the lowest priority (2).

Let’s automate a scenario by using the priority feature where we will automate Google home page and get the title of home page.

**Program source code 5:**

package testNGPriority;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

public class FirefoxTest

{

WebDriver driver; // Creating reference of WebDriver.

@Test(priority = 1)

public void driverSetup()

{

System.out.println("Running Firefox ");

driver = new FirefoxDriver(); // Create an object of ChromeDriver class.

}

@Test(priority = 2)

public void getURL()

{

driver.get("https://www.google.com");

}

@Test(priority = 3)

public void getTitle()

{

String title = driver.getTitle();

System.out.println(title);

}

@Test(priority = 4)

public void closeBrowser()

{

driver.close();

System.out.println("Test successfully passed");

}

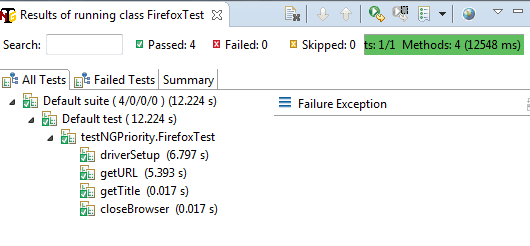
}

Output:

Running Firefox

Google

Test successfully passed

**Results of FirefoxTest:**[](https://www.scientecheasy.com/2019/06/testng-priority.html/)