

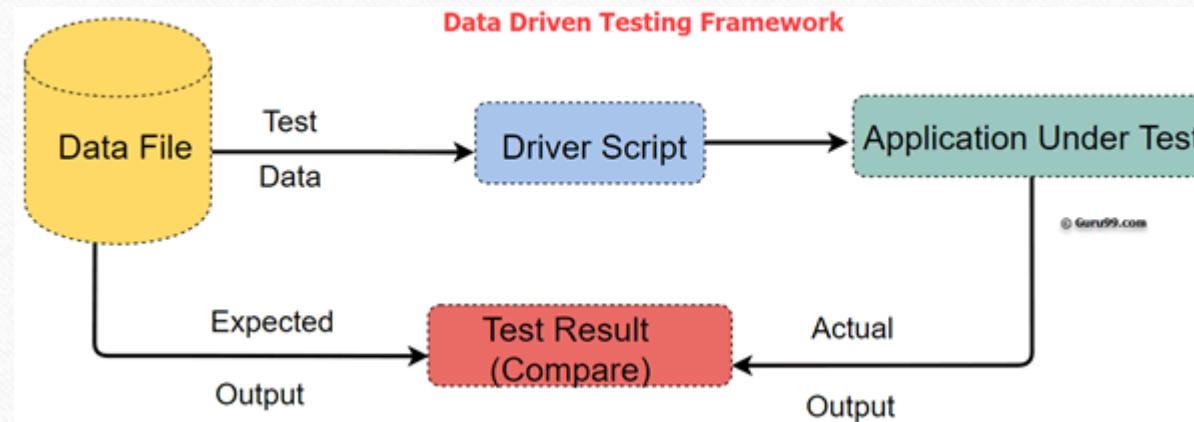
# SELENIUM FAQ'S

---

## Part-4

## Ques.76. What is a data driven framework?

- Ans. A data driven framework is one in which the test data is put in external files like csv, excel etc separated from test logic written in test script files. The test data drives the test cases, i.e. the test methods run for each set of test data values.





## Ques.77. What is a keyword driven framework?

---

- Ans. A keyword driven framework is one in which the actions are associated with keywords and kept in external files e.g. an action of launching a browser will be associated with keyword - launchBrowser(),.
- The code to perform the action based on a keyword specified in external file is implemented in the framework itself.

Testcase Name      Keywords      Object Name      Object Type can be xpath, name, css etc.      Value for textbox, area, url etc

Testcase	Keyword	Object	ObjectType	value
Reset Login In Application				
	GOTOURL			url
	SETTEXT	username	name	Demo
	SETTEXT	password	name	testPassword
	CLICK	resetButton	name	
Login In Application				
	GOTOURL			url
	SETTEXT	username	name	Demo
	SETTEXT	password	name	testPassword
	CLICK	loginButton	name	

function to execute keywords

Keyword File

TestCase	Keyword	Object
Reset Login In Application		
	GOTOURL	
	SETTEXT	username
	SETTEXT	password
	CLICK	resetButton
Login In Application		

```

public void perform(Properties p) {
    System.out.println("");
    switch (operation.toUpperCase()) {
        case "CLICK":
            driver.findElement(this.getObject(p, objectName, objectType)).click();
            break;
        case "SETTEXT":
            driver.findElement(this.getObject(p, objectName, objectType)).sendKeys(value);
            break;
        case "GOTOURL":
            driver.get(p.getProperty(value));
            break;
        case "GETTEXT":
            driver.findElement(this.getObject(p, objectName, objectType)).getText();
            break;
    }
}

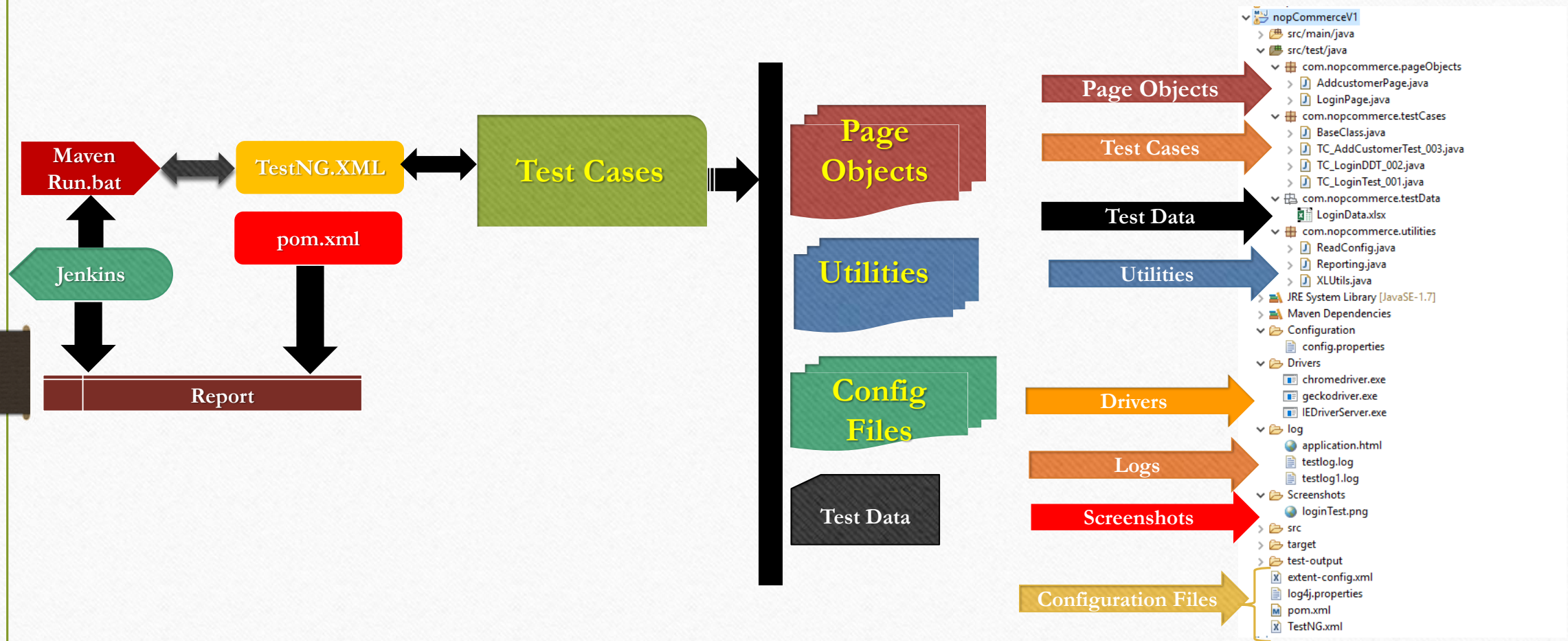
```



## Ques.78. What is a hybrid framework?

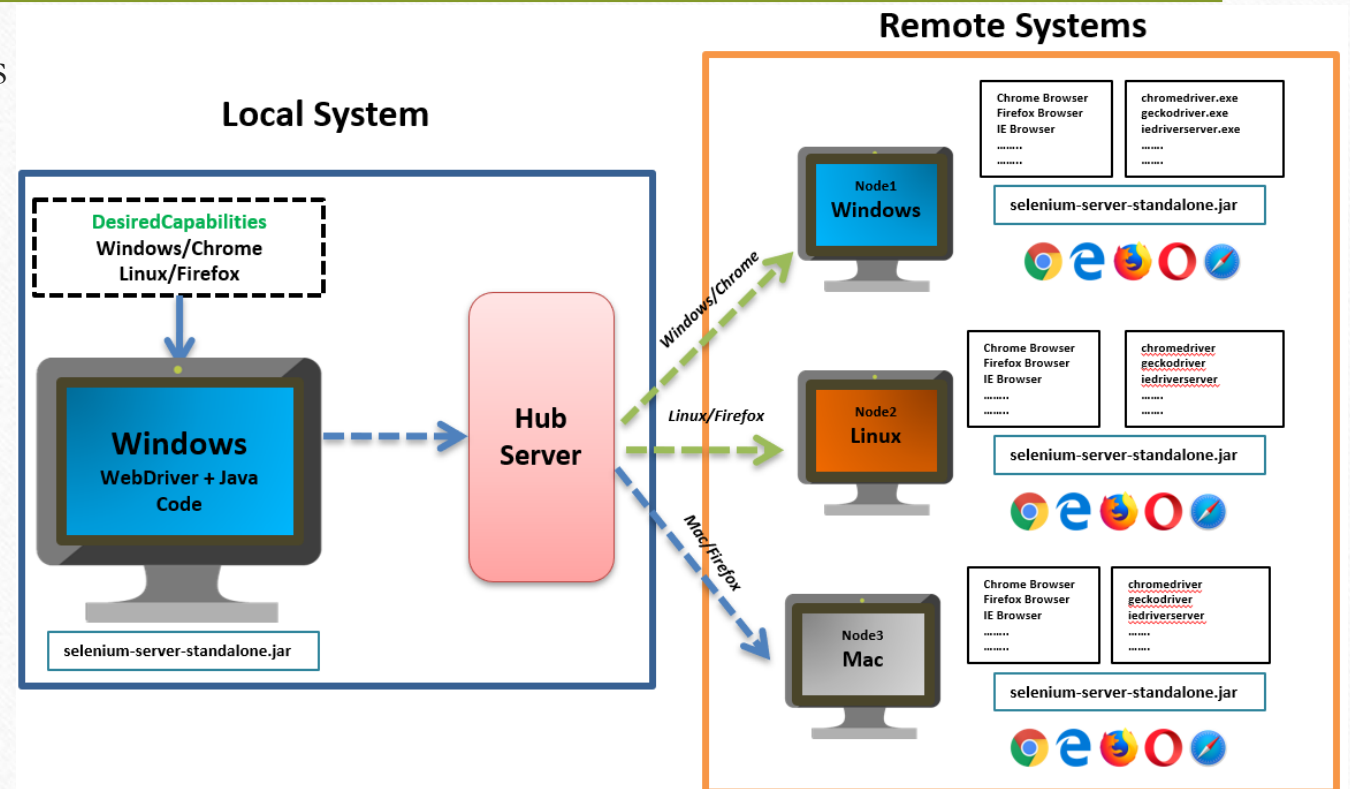
---

- Ans. A hybrid framework is a combination of one or more frameworks. Normally it is associated with combination of data driven and keyword driven frameworks where both the test data and test actions are kept in external files(in the form of table).



# Ques.79. What is selenium Grid?

- Ans. Selenium grid is a tool that helps in distributed running of test scripts across different machines having different browsers, browser version, platforms etc in parallel. In selenium grid there is hub that is a central server managing all the distributed machines known as nodes.





## Ques.80. What are some advantages of selenium grid?

---

- Ans.
- It allows running test cases in parallel thereby saving test execution time.
- Multi browser testing is possible using selenium grid by running the test on machines having different browsers.
- It is allows multi-platform testing by configuring nodes having different operating systems.



## Ques.81. What is a hub in selenium grid?

---

- Ans. A hub is server or a central point in selenium grid that controls the test executions on the different machines.

## Ques.82. What is a node in selenium grid?

---

- Ans. Nodes are the machines which are attached to the selenium grid hub and have selenium instances running the test scripts. Unlike hub there can be multiple nodes in selenium grid.



## Ques.83. Explain the line of code **Webdriver driver = new FirefoxDriver();**

---

- Ans. In the line of code *Webdriver driver = new FirefoxDriver();*
- 'WebDriver' is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.

**Ques.84 What is the purpose of creating a reference variable- 'driver' of type WebDriver instead of directly creating a FireFoxDriver object or any other driver's reference in the statement Webdriver driver = new FirefoxDriver();?**

---

- Ans. By creating a reference variable of type WebDriver we can use the same variable to work with multiple browsers like ChromeDriver, IEDriver etc.



## Ques.85. What is testNG?

---

- Ans. TestNG(NG for Next Generation) is a testing framework that can be integrated with selenium or any other automation tool to provide multiple capabilities like assertions, reporting, parallel test execution etc.

## Ques.86. What are some advantages of testNG?

---

- Ans. Following are the advantages of testNG:
- TestNG provides different assertions that helps in checking the expected and actual results.
- It provides parallel execution of test methods.
- We can define dependency of one test method over other in TestNG.
- We can assign priority to test methods in selenium.
- It allows grouping of test methods into test groups.
- It allows data driven testing using @DataProvider annotation.
- It has inherent support for reporting.
- It has support for parameterizing test cases using @Parameters annotation.



## Ques.87. What is the use of testng.xml file?

---

- Ans. testng.xml file is used for configuring the whole test suite.
- In testng.xml file we can create test suite, create test groups, mark tests for parallel execution, add listeners and pass parameters to test scripts.
- Later this testng.xml file can be used for triggering the test suite.

## Ques.88. How can we pass parameter to test script using testNG?

- Ans. Using @Parameter annotation and 'parameter' tag in testng.xml we can pass parameters to the test script.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="testsuite">
  <test name="sampletest">
    <parameter name="a" value="welcome" />
    <classes>
      <class name="parameterization.Test1" />
    </classes>
  </test>
</suite>

package parameterization;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class Test1 {
  @Parameters("a")
  @Test
  public void m1(String s)
  {
    System.out.println("the value from xml file is:" +s);
  }
}
```



## Ques.89. How can we create data driven framework using testNG?

- Ans. Using @DataProvider we can create a data driven framework in which data is passed to the associated test method and multiple iteration of the test runs for the different test data values passed from the @DataProvider method.
- The method annotated with @DataProvider annotation return a 2D array of object.

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

public class DataProviderExample {

    WebDriver driver;

    @BeforeClass
    void setup()
    {
        System.setProperty("webdriver.chrome.driver", "C://Drivers/chromedriver_win32/chromedriver.exe");
        driver=new ChromeDriver();
    }

    @Test(dataProvider="users")
    void loginTest(String uname,String pwd)
    {
        driver.get("http://newtours.demoaut.com/");
        driver.findElement(By.name("userName")).sendKeys(uname);
        driver.findElement(By.name("password")).sendKeys(pwd);
        driver.findElement(By.name("login")).click();
        Assert.assertEquals(driver.getTitle(), "Find a Flight: Mercury Tours:");
    }

    @DataProvider(name="users")
    String [][] logindata()
    {
        String data[][]={{"mercury","mercury"},{"asasa","mercury1"},{"mercury2","mercury2"} };
        return data;
    }

    @AfterClass
    void closebrowser()
    {
        driver.quit();
    }
}
```

Data  
Provider  
method

## Ques.90. What is the use of TestNG Listeners?

---

- Ans. TestNG provides us different kind of listeners using which we can perform some action in case an event has triggered.
- Usually testNG listeners are used for configuring reports and logging.
- One of the most widely used listener in testNG is ITestListener interface and TestListenerAdapter Class.
- It has methods like onTestSuccess, onTestFailure, onTestSkipped etc.
- We need to implement this interface creating a listener class of our own.



## Ques.91. What is the use of *@Listener* annotation in TestNG?

---

- We need to implement ITestListener interface by creating a listener class of our own.
- After that using the *@Listener* annotation, we can use specify that for a particular test class, our customized listener class should be used.

```

package testnglisteners;

import org.testng.ITestContext;
import org.testng.ITestListener;
import org.testng.ITestResult;

public class Mylisteners implements ITestListener {
    @Override
    public void onTestStart(ITestResult result) {
        System.out.println(" test is started");
    }

    @Override
    public void onTestSuccess(ITestResult result) {
        System.out.println(" test is passed");
    }

    @Override
    public void onTestFailure(ITestResult result) {
        System.out.println("test is failed");
    }

    @Override
    public void onTestSkipped(ITestResult result) {
        System.out.println("test is skipped");
    }

    @Override
    public void onTestFailedButWithinSuccessPercentage(ITestResult result) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStart(ITestContext context) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onFinish(ITestContext context) {
        // TODO Auto-generated method stub
    }
}

```

```

package testnglisteners;

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

@Listeners(testnglisteners.Mylisteners.class)
public class LoginTest {

    @Test
    void setup()
    {
        Assert.fail();
    }

    @Test
    void loginByEmail()
    {
        Assert.assertTrue(true);
    }

    @Test (dependsOnMethods={"setup"})
    void loginByFacebook()
    {
        Assert.assertTrue(true);
    }
}

```

## Ques.92. How can we make one test method dependent on other using TestNG?

---

- Ans. Using dependsOnMethods parameter inside @Test annotation in testNG we can make one test method run only after successful execution of dependent test method.

```
@Test(dependsOnMethods = { "preTests" })
```



## Ques.93. How can we set priority of test cases in TestNG?

---

- Ans. Using priority parameter in `@Test` annotation in TestNG we can define priority of test cases. The default priority of test when not specified is integer value 0. Example:

```
@Test(priority=1)
```

## Ques.94. What are commonly used TestNG annotations?

---

- Ans. The commonly used TestNG annotations are-
- @Test
- @BeforeMethod
- @AfterMethod
- @BeforeClass
- @AfterClass
- @BeforeTest
- @AfterTest
- @BeforeSuite
- @AfterSuite

## Ques.95. What are some common assertions provided by testNG?

---

- Ans. Some of the common assertions provided by testNG are-
- assertEquals(String actual, String expected, String message) - (and other overloaded data type in parameters)
- assertNotEquals(double data1, double data2, String message) - (and other overloaded data type in parameters)
- assertFalse(boolean condition, String message)
- assertTrue(boolean condition, String message)
- assertNotNull(Object object)
- fail(boolean condition, String message)
- true(String message)



## Ques.96. How can we run test cases in parallel using TestNG?

---

- Ans. In order to run the tests in parallel just add these two key value pairs in suite-
- `parallel="{methods/tests/classes}"`
- `thread-count="{number of thread you want to run simultaneously}"`.

```
<suite name="paralleltesting" parallel="tests" thread-count="5">
```

```

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

public class CrossBrowserTest {

    WebDriver driver = null;

    @Parameters("browser")
    @Test
    public void launchBrowser(String br) {
        if (br.equals("FF")) {
            // Firefox Browser
            System.setProperty("webdriver.gecko.driver", "C://Drivers/geckodriver-v0.19.1-win64/geckodriver.exe");
            driver = new FirefoxDriver();
        }

        else if (br.equals("IE")) {
            System.setProperty("webdriver.ie.driver", "C://Drivers/IEDriverServer_Win32_3.7.0/IEDriverServer.exe");
            driver = new InternetExplorerDriver();
        }

        else if (br.equals("CH")) {
            System.setProperty("webdriver.chrome.driver", "C://Drivers/chromedriver_win32/chromedriver.exe");
            driver = new ChromeDriver(); // opens the browser
        }

        driver.get("http://newtours.demoaut.com/");
    }

    @Test
    public void verifyTitle() {
        Assert.assertEquals(driver.getTitle(), "Welcome: Mercury Tours");
    }

    /**
     * @Test public void registration() { //selenium code for registration }
     */

    @AfterClass
    public void closeBrowser() {
        driver.quit();
    }
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="paralleltesting" parallel="tests" thread-count="5">

    <test name="FirefoxTest">
        <parameter name="browser" value="FF" />
        <classes>
            <class name="parameterization.CrossBrowserTest" />
        </classes>
    </test>

    <test name="ChromeTest">
        <parameter name="browser" value="CH" />
        <classes>
            <class name="parameterization.CrossBrowserTest" />
        </classes>
    </test>

    <test name="IETest">
        <parameter name="browser" value="IE" />
        <classes>
            <class name="parameterization.CrossBrowserTest" />
        </classes>
    </test>

</suite>

```

## **Ques.97. Name an API used for reading and writing data to excel files.**

---

- Ans. Apache POI API and JXL(Java Excel API) can be used for reading, writing and updating excel files.



## Ques.98. Name an API used for logging in Java.

---

- Ans. Log4j is an open source API widely used for logging in Java.
- It supports multiple levels of logging like - ALL, DEBUG, INFO, WARN, ERROR, TRACE and FATAL.

## Ques.99. What is the use of logging in automation?

---

- Ans. Logging helps in debugging the tests when required and also provides a storage of test's runtime behaviour.



# Ques.100. What is InvocationCount in TestNG?

- This is a TestNG attribute that defines number of times a test method should be invoked or executed before executing any other test method.

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class invocationCount {

    @Test(invocationCount = 5)
    public void getTitle() {
        System.setProperty("webdriver.chrome.driver", "C:/Drivers/chromedriver_win32/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://www.pavanonline Trainings.com/");
        driver.manage().window().maximize();
        System.out.println("Website Title: "+driver.getTitle());
        driver.quit();
    }

    @Test
    public void secondTest() {
        System.out.println("This will be executed at the end");
    }
}
```