**TestNG Practise**

**Selenium Automation sctipt**

**---------------------**

**-> We are write the selenium automation script in two models**

**a) JUnit**

**b) TestNg**

**-> compare the junit testng is powerfull.**

**-> TestNg (Test Next Generation ).**

**-> We are the test case in testng.**

**-> the testng containg no main method.**

**-> Test case are running is very easly**

**-> At a time we are running paralley two testcase.**

**-> Generating reports very easy.**

**How to write the TestNg Programs**

**----------------------------**

**-> we are write the test case in functions.**

**-> We are runn the test cases (Functions) using testng Annatations**

**TestNg Annatations**

**----------------**

**1) @Test : It is TestAnnatation**

**2) @Test(priority=1) : using this annatation we are running function based on priority number.**

**3) @Test(enabled=false) : using this annatation we are stop the running of the testcase.**

**4) BeforeMethod : This annatation is used to run this test case before every test case running.**

**5) AfterMethod : This annnation is used to run this test case after every test case before.**

**6) @DataProvider(Name="provider")**

**Example 1:**

**---------**

**package examplePack;**

**import org.testng.annotations.Test;**

**public class Sample**

**{**

**@Test(priority=1)**

**public void Login()**

**{**

**System.out.println("Login Sucessfully....");**

**}**

**@Test(priority=3)**

**public void WithDraw()**

**{**

**System.out.println("WithDraw Sucessfully.....");**

**}**

**@Test(priority=2)**

**public void Diposit()**

**{**

**System.out.println("Diposit Sucessfullly....");**

**}**

**@Test(priority=4)**

**public void Logout()**

**{**

**System.out.println("Logout Sucessfully.....");**

**}**

**}**

**Example 2:**

**---------**

**package examplePack;**

**import org.testng.annotations.Test;**

**public class Sample**

**{**

**@Test(priority=1)**

**public void Login()**

**{**

**System.out.println("Login Sucessfully....");**

**}**

**@Test(priority=2)**

**public void WithDraw()**

**{**

**System.out.println("WithDraw Sucessfully.....");**

**}**

**@Test(enabled=false)**

**public void Diposit()**

**{**

**System.out.println("Diposit Sucessfullly....");**

**}**

**@Test(priority=4)**

**public void Logout()**

**{**

**System.out.println("Logout Sucessfully.....");**

**}**

**}**

**Example 3 :**

**----------**

**package examplePack;**

**import org.testng.annotations.AfterMethod;**

**import org.testng.annotations.BeforeMethod;**

**import org.testng.annotations.Test;**

**public class Sample**

**{**

**@BeforeMethod**

**public void Login()**

**{**

**System.out.println("Login Sucessfully....");**

**}**

**@Test(priority=1)**

**public void WithDraw()**

**{**

**System.out.println("WithDraw Sucessfully.....");**

**}**

**@Test(priority=2)**

**public void Diposit()**

**{**

**System.out.println("Diposit Sucessfullly....");**

**}**

**@AfterMethod**

**public void Logout()**

**{**

**System.out.println("Logout Sucessfully.....");**

**}**

**}**

**Example 4**

**--------**

**package examplePack;**

**import org.testng.annotations.DataProvider;**

**import org.testng.annotations.Test;**

**public class Sample**

**{**

**@Test(dataProvider="mydata")**

**public void Display(String n1,String n2)**

**{**

**System.out.println(n1+" , "+n2);**

**}**

**@DataProvider(name="mydata")**

**public Object[][] GetDetails()**

**{**

**String name1="Mahesh";**

**String name2="Raju";**

**String[][] data=new String[1][2];**

**data[0][0]=name1;**

**data[0][1]=name2;**

**return data;**

**}**

**}**

**Example 5:**

**---------**

**package examplePack;**

**import org.testng.annotations.DataProvider;**

**import org.testng.annotations.Test;**

**public class Sample**

**{**

**@Test(dataProvider="mydata")**

**public void Display(String eno,String ename,String salary)**

**{**

**System.out.println(eno+" , "+ename+" , "+salary);**

**System.out.println("");**

**}**

**@DataProvider(name="mydata")**

**public Object[][] GetDetails()**

**{**

**String[][] data=new String[3][3];**

**data[0][0]="100";**

**data[0][1]="Raju";**

**data[0][2]="25000";**

**data[1][0]="200";**

**data[1][1]="Prakash";**

**data[1][2]="35000";**

**data[2][0]="300";**

**data[2][1]="Krishna";**

**data[2][2]="45000";**

**return data;**

**}**

**}**

**Example 5:**

**---------**

**package examplePack;**

**import java.io.File;**

**import java.io.IOException;**

**import org.testng.annotations.DataProvider;**

**import org.testng.annotations.Test;**

**import jxl.Cell;**

**import jxl.Sheet;**

**import jxl.Workbook;**

**import jxl.read.biff.BiffException;**

**public class Sample**

**{**

**@Test(dataProvider="mydata")**

**public void Display(String eno,String ename,String salary)**

**{**

**System.out.println(eno+" , "+ename+" , "+salary);**

**System.out.println("");**

**}**

**@DataProvider(name="mydata")**

**public Object[][] GetDetails() throws BiffException, IOException**

**{**

**// Open files**

**File f=new File("D:\\MyData.xls");**

**Workbook wb=Workbook.getWorkbook(f);**

**Sheet sh=wb.getSheet(0);**

**// get the used rows and columns count**

**int rcount=sh.getRows();**

**int ccount=sh.getColumns();**

**// Row Loop**

**String[][] data=new String[rcount][ccount]; // Array String**

**for(int r=1;r<rcount;r++)**

**{**

**for(int c=0;c<ccount;c++)**

**{**

**Cell myvalue=sh.getCell(c, r);**

**String value=myvalue.getContents();**

**data[r][c]=value;**

**}**

**}**

**return data;**

**}**

**}**

**SoftAssertDemo:**

**package** samplePack;

**import** org.testng.Assert;

**import** org.testng.annotations.Test;

**import** org.testng.asserts.SoftAssert;

**public** **class** SoftAssertDemo {

@Test

**public** **void** softAssert(){

SoftAssert assertion = **new** SoftAssert();

System.***out***.println("Soft assert started");

assertion.assertEquals(12, 13);

System.***out***.println("Soft assert ended");

assertion.assertAll();

}

@Test

**public** **void** hardAssert(){

System.***out***.println("hard assert started");

Assert.*assertEquals*(12, 13);

System.***out***.println("hard assert ended");

}

}