**Writing Selenium WebDriver Test Cases Part 3**  
  
**Writing Selenium WebDriver Test Cases using User defined Methods/Reusable components.**  
  
**I) Create User defined Methods**  
  
public class Methods {  
public static WebDriver driver;  
//Launch Browser  
public void launchBrowser(){  
driver = new FirefoxDriver();      
}  
//Admin Login without Parameters  
public void adminLogin(){  
driver.get("http://www.gcrit.com/build3/admin/");  
driver.findElement(By.name("username")).sendKeys("admin");  
driver.findElement(By.name("password")).sendKeys("admin@123");  
driver.findElement(By.id("tdb1")).click();  
}  
//Admin Login With Parameters  
public void adminLogin(String username, String password){  
driver.get("http://www.gcrit.com/build3/admin/");  
driver.findElement(By.name("username")).sendKeys(username);  
driver.findElement(By.name("password")).sendKeys(password);  
driver.findElement(By.id("tdb1")).click();  
}  
//Close Browser  
public void closeBrowser(){  
if (! driver.toString().contains("null")){  
driver.close();  
}  
}  
public static void main(String[] args) {  
Methods obj = new Methods();  
obj.launchBrowser();  
obj.adminLogin();  
obj.closeBrowser();  
obj.launchBrowser();  
obj.adminLogin("admin", "admin@123");  
obj.closeBrowser();  
}  
}  
--------------------------------------------------  
**II) Creating Test Cases using User defined Methods.**  
Test Case 1: Redirect to user Interface from Admin Interface  
  
//Create Object/Instance  
TestCase1 object = new TestCase1();  
object.launchBrowser();  
object.adminLogin("admin", "admin@123");  
driver.findElement(By.linkText("Online Catalog")).click();  
String url = driver.getCurrentUrl();  
  
if (url.equals("http://www.gcrit.com/build3/")){  
System.out.println("Redirected to User Interface -Passed");      
}  
else {  
System.out.println("Redirected to User Interface -Failed");      
}  
object.closeBrowser();  
--------------------------------------------------  
Test Case 2: Admin Login Functionality with valid inputs(Positive Testing)   
  
Test Case 2: Admin Login Functionality with valid inputs(Positive Test Case)  
  
//Create Object/Instance  
TestCase2 obj2 = new TestCase2();  
obj2.launchBrowser();  
obj2.adminLogin();  
String url = driver.getCurrentUrl();  
  
if (url.contains("http://www.gcrit.com/build3/admin/index.php")){  
System.out.println("Admin Login Successful - Passed");      
}  
else {  
System.out.println("Admin Login Unsuccessful - Failed");      
}  
obj2.closeBrowser();  
---------------------------------------------------  
Test Case 3: Admin Login Functionality with invalid inputs (Negative Testing)  
  
Test Case 3: Admin Login Functionality with invalid inputs(Negative Test Case)  
  
//Create Object/Instance  
TestCase3 obj3 = new TestCase3();  
obj3.launchBrowser();  
obj3.adminLogin("admina", "admin@123a");  
String ErrorMessage = driver.findElement(By.className("messageStackError")).getText();  
  
if (ErrorMessage.contains("Error: Invalid administrator login attempt.")){  
System.out.println("Handling Invalid Inputs - Passed");      
}  
else {  
System.out.println("Not Handling Invalid Inputs - Failed");  
}  
obj3.closeBrowser();  
-------------------------------------------------------  
Write multiple Test Cases in a Program/Class  
  
public class TestCases extends Methods{  
  
public static void main(String[] args) {  
//Create Object/Instance  
TestCases obj4 = new TestCases();  
//Test Case 1: Redirect to user Interface from Admin Interface  
obj4.launchBrowser();  
obj4.adminLogin("admin", "admin@123");  
driver.findElement(By.linkText("Online Catalog")).click();  
String url = driver.getCurrentUrl();  
  
if (url.equals("http://www.gcrit.com/build3/")){  
System.out.println("Test Case 1: -Redirected to User Interface -Passed");      
}  
else {  
System.out.println("Test Case 1: Redirected to User Interface -Failed");      
}  
obj4.closeBrowser();  
//---------------------------------------------  
//Test Case 2: Admin Login Functionality with valid inputs(Positive Test Case)  
obj4.launchBrowser();  
obj4.adminLogin();  
String url2 = driver.getCurrentUrl();  
  
if (url2.contains("http://www.gcrit.com/build3/admin/index.php")){  
System.out.println("Test Case 2: Admin Login Successful - Passed");      
}  
else {  
System.out.println("Test Case 2: Admin Login Unsuccessful - Failed");      
}  
obj4.closeBrowser();  
//------------------------------------------  
//Test Case 3: Admin Login Functionality with invalid inputs(Negative Test Case)  
obj4.launchBrowser();  
obj4.adminLogin("admina", "admin@123a");  
String ErrorMessage = driver.findElement(By.className("messageStackError")).getText();  
  
if (ErrorMessage.contains("Error: Invalid administrator login attempt.")){  
System.out.println("Test Case 3: -Handling Invalid Inputs - Passed");      
}  
else {  
System.out.println("Test Case 3: -Not Handling Invalid Inputs - Failed");  
}  
obj4.closeBrowser();  
}  
}  
--------------------------------------------------------------  
**Synchronization**  
  
1) What is Synchronization?  
  
General:  
  
Process of coordinating or matching two or more activities /devices/processes in time.  
  
Test Automation:  
  
Process of matching the speeds of AUT (application Under Test) and Test Tool in order to get proper execution.  
  
2) Why Synchronization is required?  
  
During Test execution Test tool gives instructions one by one with same speed, but AUT takes less time for some steps execution and more time for some steps execution, in order to keep them in Sync then Synchronization is required.  
  
3) Types of Synchronization  
  
a) Unconditional Synchronization  
  
In this we specify timeout value, we will make the tool to wait certain amount of time and then proceed.  
  
Syntax:  
  
Thread.sleep(time in mille seconds);  
  
Example:  
  
Thread.sleep(9000);  
  
b) Conditional Synchronization  
  
i) It will not work for all commands/statements in the application  
  
ii) It works only for findElement and findElements statements  
  
Syntax:  
  
driver.manage().timeouts().implicitlyWait(Time in Seconds, TimeUnit.SECONDS);

The implicit wait will tell to the web driver to wait for certain amount of time before it throws a "No Such Element Exception". The default setting is 0. Once we set the time, web driver will wait for that time before throwing an exception.

4) Examples  
  
//Unconditional Synchronization  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
Thread.sleep(10000);  
driver.findElement(By.linkText("Gmail")).click();  
  
//Conditional Synchronization  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
driver.findElement(By.linkText("Gmail")).click();  
------------------------------------------------------------------