1. Introduction to Selenium WebDriver  
     
   ii) WebDriver Environment Setup  
     
   iii) Create first Selenium Test Case  
   -------------------------------------------  
   **i) Introduction to Selenium WebDriver**  
   Selenium Tool’s Suite  
           Selenium IDE  
           Selenium RC  
           Selenium WebDriver  
           Selenium Grid  
     
   > In 2006 Selenium WebDriver was launched at Google.  
     
   > In 2008, the whole Selenium team decided to merge Selenium WebDriver with Selenium RC in order to form more powerful tool called Selenium 2.0  
     
   Selenium 1.0 + WebDriver = Selenium 2.0  
     
   Selenium 1.0  
     
   (Selenium IDE + Selenium RC + Selenium Grid)  
     
   Selenium 2.0  
     
   (Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)  
     
   Note: Now Selenium RC is only for maintenance projects.  
   ----------------------  
   > It is a most important tool in Selenium Suite.  
     
   > It has Programming interface only, no IDE.  
     
   > Selenium WebDriver supports various programming languages to write programs (Test scripts)  
     
   Java  
   Python  
   C#  
   Ruby  
   Perl  
   PHP  
     
   > Selenium WebDriver supports various Browsers to create and execute Test cases.  
     
   Mozilla Firefox  
     
   Google Chrome  
     
   IE  
     
   Safari  
     
   Opera etc...  
     
   > Selenium WebDriver supports various Operating environments.  
     
   MS Windows  
     
   Linux  
     
   Macintosh etc...  
     
   > Selenium WebDriver supports Data driven Testing and Cross browser testing.  
     
   > Selenium WebDriver is faster than other tools of Selenium suite.  
     
   > Selenium WebDriver supports Parallel test execution with the help of either JUnit or TestNG.  
   -------------------------------  
   Drawbacks of selenium Webdriver  
   > Selenium WebDriver doesn't have IDE (some difficult to create test cases)  
     
   > No Built-in Result reporting facility.  
     
   > No other tool Integration for Test management.  
     
   > No centralized maintenance of Elements/objects.  
   --------------------------------------------------  
   How we create Test Cases /Test Scripts / Tests in Selenium WebDriver:  
   In UFT:  
     
   Using Objects information and Test Methods we create Tests.  
     
   In Selenium WebDriver:  
     
   Using Element Locators and Webdriver Commands/methods we create Test Cases.  
     
   Selenium IDE:  
     
   Using Element locators and Selenese / Selenium IDE commands we create Test cases  
   -----------------------------------  
   Element Locators - to recognize elements/identify elements.  
     
   WebDriver commands/methods - to perform operations on elements.  
   --------------------------------------  
   Java Programming - for enhancing Test Cases  
     
   TestNG Framework - for grouping Test cases, batch Testing and generating Test Reports.  
   -----------------------------------------------------------  
   **ii) Selenium WebDriver Environment Setup.**  
   Steps:  
     
   1) Download and Install Java (JDK) software -to create programs (Test scripts)  
     
   2) Set Environment variable (path variable).  
     
   3) Download Eclipse IDE and extract - to write and execute Java programs.  
     
   4) Download Webdriver Java language binding (www.seleniumhq.org) and add WebDriver jar files to Java project in Eclipse IDE.  
   ---------------------------------------  
   5) Install Firebug and Firepath plug ins (Mozilla Firefox) for inspecting elements.  
     
   6) For Internet Explorer and Google Chrome, no need to install any plug in, they have  
   built in developer tools(F12) for inspecting elements.  
     
   7) Firefox driver is default driver in Selenium Webdriver, for IE and Chrome etc... Browsers then we need to download browser drivers.  
   --------------------------------------------  
   Download Selenium WebDriver Java language binding from www.seleniumhq.org website and extract.  
     
   Add WebDriver jar files to Java Project in Eclipse IDE  
   Navigation:  
     
   Create Java Project  
     
   > Select Java project and right click  
     
   > Build path  
     
   > Configure build path  
     
   > Select "Libraries" tab  
     
   > Click "Add external Jars"  
     
   > Browse path of the WebDriver jars.  
     
   > Add  
   -----------------------------------------  
   Create Selenium WebDriver Test Case  
   > Import Webdriver and Firefox/IE/Chrome Libraries. (In Selenium Test Case/Program)  
     
   > Using Element locators and Webdriver commands write test steps.  
     
   > Insert java programming statements to enhance Test cases.  
   -----------  
   > Using TestNG Annotations group test cases, execute test batches and generate detailed test reports.  
   --------------------------------------------------  
   **iii) Write first Selenium Test Case**  
   Manual Test Case  
   Test Case ID: gcrshop\_admin\_TC001  
     
   Test Case Name: Verify Admin Login in GCR shop Web Portal  
     
   Test Steps:  
     
   1) Launch the Browser and navigate to "www.gcrit.com/build3/admin"  
   2) Enter User name  
   3) Enter Password  
   4) Click Login Button  
     
   Input data:  
     
   Username = admin  
   Password =admin@123  
     
   Expected URL: "www.gcrit.com/build3/admin/index.php"  
     
   Actual: http://www.gcrit.com/build3/admin/index.php  
     
   Test Result: Pass  
   -------------------------------------------  
   \* Verification point: Capture the Browser URL after submission of Login details and compare with expected URL.  
   ----------------------------------------------------------------  
   Selenium WebDriver Test Case:  
   public class AdminLogin {  
     
   public static void main(String[] args) {  
   WebDriver driver = new FirefoxDriver(); //Launches Firefox Browser with blank url  
   driver.get("http://www.gcrit.com/build3/admin/login.php");  
   driver.findElement(By.name("username")).sendKeys("admin");  
   driver.findElement(By.name("password")).sendKeys("admin@123");  
   driver.findElement(By.id("tdb1")).click();  
     
   String url = driver.getCurrentUrl();  
   if (url.equals("http://www.gcrit.com/build3/admin/index.php")){  
   System.out.println("Login Successful -Passed");      
   }  
   else  
   {  
   System.out.println("Login Unsuccessful -Failed");          
   }  
   driver.close(); //Closes the Browser  
   }  
   }

**Web Elements and Element Locators**  
   
i) Web Elements  
  
ii) Element Locators  
---------------------------------------  
**i) Web Elements**  
Browser  
Page  
-----------------  
Edit Box  
Link  
Button  
Image, Image Link, Image Button  
Text box  
Text Area  
Check box  
Radio Button  
Drop down box  
List box  
Combo box  
Web table /HTML table  
Frame  
-----------------------------------  
1) Operations on Browser  
> Launch the browser,  
> Navigate to particular web page,  
> Close focused Browser  
> Close all Browsers that opened by WebDriver during execution  
---------------  
> Navigate from one URL to another  
> Navigate back to previous URL  
> Navigate forward  
> Refresh the Browser  
> Maximize the Browser  
Etc...  
-----------------------------------  
2) Operations on Web Page  
> Get Page Title  
> Get Page URL  
-----------------------------------  
3) Operations on Edit box  
  
> Enter a Value,  
> Clear the Value,  
> Check enabled status,  
> Check edit box existence,  
> Get the value etc...  
-----------------------------------  
4) Operations on Link  
  
> Click Link,  
> Check the link existence,  
> Check the link enabled status,  
> Return the Link Name  
Etc...  
-----------------------------------  
5) Operations on Button  
  
> Click  
> Check Enabled status  
> Display status  
Etc...  
-----------------------------------  
6) Operations Image  
  
Three types of Image elements in Web Environment  
  
a) General Image (No functionality)  
  
b) Image Button (Submits)  
  
c) Image Link (Redirects to another page/location)  
-----------------------------------  
7) Operations on Text Area  
  
> Return / Capture Text Area or Error message from a web page  
-----------------------------------  
8) Operations on Check box  
  
> Check if the check box is displayed or not?  
> Check if the check box is enabled or not?  
> Check if the check box is Selected or not?  
> Select the Check box  
> Unselect the Check box  
-----------------------------------  
9) Operations on Radio Button

> Select Radio Button  
> Verify if the Radio Button is Displayed or not?  
> Verify if the Radio Button is enabled or not?  
> Verify if the Radio Button is Selected or not?  
------------------------------------  
10) Operations on Drop down box  
  
> Check the Drop down box existence  
> Check if the Drop down is enabled or not?  
> Select an item  
> Items Count  
-----------------------------------  
11) Operations on List box  
  
12) Operations on Combo box  
-----------------------------------  
13) Operations on Web table /HTML Table  
  
> Get cell value  
> Rows Count  
> Cells Count Etc...  
-----------------------------------  
14) Operations on Frame  
-----------------------------------  
> Switch from Top window to a frame  
> Switch from a frame to Top window  
Etc...  
----------------------------------------------  
**ii) Element Locators**  
What is Locator?  
  
> Locator is an address that identifies a web element uniquely within the webpage. Locators are the HTML properties of a web element.  
  
Selenium WebDriver uses 8 element locators   
  
id,   
  
name,   
  
className,   
  
tagName,   
  
linkText,   
  
partialLinkText,   
  
cssSelector,   
  
xpath   
  
to find elements on Web pages.  
  
Why we need to use different locators?  
  
1) Developers may not provide all locators for all elements  
  
2) Some locators may be duplicated.  
  
So we have to choose any one unique locator to recognize the element.  
  
How to inspect elements?  
  
Download and install Firebug and Firepath plug ins/Add ons for Firefox Browser.   
  
If it Internet Explorer or Chrome, we no need to install any Add on, they provide built -in Developer Tools (F12) to inspect elements.  
  
Element Locators are common for all Browsers.  
-----------------------------------------  
1) id  
  
Syntax:  
  
By.id("id value")  
  
Examples:  
  
driver.findElement(By.id("Email"))  
  
driver- is Object  
  
findElement - WebDriver method  
  
By - pre-defined Class  
  
id - Element locater  
  
Email - id locator value  
-------------------------------------  
driver.findElement(By.id("Email")).sendKeys("gcrindia");  
--------------------------------------------  
Or  
  
WebElement Email = driver.findElement(By.id("Email"));  
Email.sendKeys("gcrindia");  
-------------------------------------------  
id locator for Button  
  
WebElement Email = driver.findElement(By.id("signIn"));  
Email.click();  
  
Or  
  
driver.findElement(By.id("signIn")).click();  
-------------------------------------------------------  
2) name  
  
Synatx:  
  
By.name("name value/locator name")         
  
Examples:  
  
driver.findElement(By.name("Email")).sendKeys("gcrindia");  
         
Or  
  
WebElement e = driver.findElement(By.name("Email"));  
e.sendKeys("gcrindia");  
------------------------------------------  
WebElement e = driver.findElement(By.name("signIn"));  
e.click();  
------------------------------------------  
3) className  
  
Syntax: By.className("class name value")  
  
Example:  
  
driver.findElement(By.className("textboxcolor")).sendKeys("Hyderabad");  
----------------------------------------------  
4) tagName  
  
Syntax:  
By.tagName("tag name value")  
  
Example:  
driver.findElement(By.tagName("input")).sendKeys("Hyderabad");  
--------------------------------------------------------------         
5) linkText  
  
Syntax:  
By.linkText("Link Text Value")  
  
Example:  
  
driver.findElement(By.linkText("Gmail")).click();  
-----------------------------------------------------                 
6) paritialLinkText  
  
Syntax:  
By.partialLinkText("Partial Link Text Value")  
  
Example:  
  
driver.findElement(By.partialLinkText("Gma")).click();  
-----------------------------------------------  
7) cssSelector  
Syntax:  
By.cssSelector("value")  
  
Example:  
driver.findElement(By.cssSelector(".gb\_m")).click();  
---------------------------------------------------  
8) xpath   
Xpath in XML document shows the direction of software web application's element location.  
  
Syntax:  
  
By.xpath("xpath value")  
  
driver.findElement(By.xpath(".//\*[@id='Email']")).sendKeys("abcdef");

**Selenium WebDriver Commands and Operations**  
  
> Selenium WebDriver Methods are used to perform operations on Web Elements.  
  
> Using Element Locators and WebDriver Methods we create Test Cases.  
  
Element Locators - for recognizing Elements  
  
WebDriver Methods - for performing operations on Elements.  
-----------------------------------  
**WebDriver Methods**  
  
1) get()   
Description: Opens a specified URL in the Browser window.  
  
Syntax:  
  
driverObject.get("URL");  
  
Example:  
  
driver.get("https://www.google.co.in");  
------------------------------------  
2) getTitle()  
Returns Title of the Browser.  
  
Syntax:  
  
String variable = driver.getTitle();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String Title = driver.getTitle();  
System.out.println(Title);  
------------------------------  
3) getPageSource()  
Returns HTML page source.  
  
Syntax:  
  
String stringName = driver.getPageSource();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String pageSource = driver.getPageSource();  
System.out.println(pageSource);  
------------------------  
4) getCurrentUrl();  
Returns Current URL of the Browser.  
  
Syntax:  
  
String stringName = driver.getCurrentUrl();  
  
Example:  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
-------------------------------  
**Browser Navigation Methods**  
5) navigate().to();  
Loads a new web page in the current browser window.  
  
Syntax:  
driverObject.navigate().to("URL");  
  
Example:  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
-------------------------------------  
6) navigate().back()  
It moves a single item back in the Browser history.  
  
Syntax:  
  
driver.navigate().back();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().back();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
Or  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://www.google.co.in");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
-----------------------------------  
7) navigate().forward();  
It moves single item forward in the Browser history.  
  
Syntax:  
  
driver.navigate().forward();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().back();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().forward();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
Or  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://www.google.co.in");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
-----------------------------------  
8) navigate().refresh()  
Refresh the current web page  
  
Syntax:  
  
driver.navigate().refresh()  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().refresh();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
------------------------------------  
Method Syntax in Java  
  
Object.method();  
Object.property().method();  
Class.method();  
-----------------------------------  
9) close()  
It closes the focused Browser.  
  
Syntax:  
  
driverObject.close();  
  
Example:  
  
driver.get("https://www.google.co.in");  
driver.close();  
-----------------------------------  
10) quit()  
It closes all browser that opened by WebDriver during execution.  
  
Syntax:  
  
driverObject.quit();  
  
Example:   
  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/LoginPage.html");  
driver.findElement(By.linkText("Sign In")).click();  
driver.quit();  
-----------------------------------  
11) findElement()  
It finds the first element within the current page using the give locator.  
  
driver.findElement(By.ElementLocator("Value"))  
  
Syntax:  
WebDriver driver = new FirefoxDriver();  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/LoginPage.html");  
driver.findElement(By.tagName("input")).sendKeys("abcd");  
  
Or  
  
WebElement Email = driver.findElement(By.id("Email"));  
Email.sendKeys("India");  
-----------------------------------  
12) sendkeys()  
Enters a value into Edit box/Text box  
  
Syntax:  
driver.findElement(By.ElementLocator("value").sendkeys("input data");  
  
Example:  
  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("Email")).sendKeys("India");  
}  
-----------------------------------  
13) clear()  
It clears the value   
  
Syntax:  
driver.findElement(By.ElementLocator("value").clear();  
  
Example:  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("Email")).sendKeys("India");  
Thread.sleep(5000);  
driver.findElement(By.id("Email")).clear();  
-----------------------------------  
14) click()  
Clicks an Element (Buttons, Links)  
  
Syntax:  
  
driver.findElement(By.ElementLocator("value").click;  
  
Example:  
  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("next")).click();  
-----------------------------------  
15) isEnabled()  
It checks weather the Element is in enabled state or not?  
  
Syntax:  
  
boolean variableName = driver.findElement(By.ElementLocator("value").isEnabled();  
  
Example:  
  
driver.get("https://www.gmail.com");  
boolean a = driver.findElement(By.id("next")).isEnabled();  
System.out.println(a);  
-----------------------------------  
16) isDisplayed()  
Checks if the Element is displayed or not? in the current web page.  
  
Syntax:  
  
boolean variableName = driver.findElement(By.ElementLocator("value").isDisplayed();  
  
driver.get("https://www.gmail.com");  
boolean a = driver.findElement(By.id("next")).isDisplayed();  
System.out.println(a);  
-----------------------------------  
17) isSelected()  
checks if the Element is Selected or not? in the current web page.  
  
Syntax:  
  
boolean variableName = driver.findElement(By.ElementLocator("value").isSelected();  
  
Example:  
  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/MultipleCheckbox.html");  
boolean a = driver.findElement(By.xpath("html/body/input[2]")).isSelected();  
System.out.println(a);//false  
driver.findElement(By.xpath("html/body/input[2]")).click();  
a = driver.findElement(By.xpath("html/body/input[2]")).isSelected();  
System.out.println(a);//true  
-----------------------------------  
18) manage().window().maximize()  
Syntax:  
  
driverObject.manage().window().maximize()  
  
Example:  
  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/MultipleCheckbox.html");  
Thread.sleep(5000);  
driver.manage().window().maximize();

**Handling Elements in Selenium WebDriver**  
  
Pre-requisites to create Test cases in Selenium  
  
i) Element Locators (To recognize/identify Elements)  
  
ii) WebDriver Methods (To perform operations on Elements)  
  
iii) Programming features (To enhance Test cases)  
----------------------------------------------  
iv) TestNG Annotations (Grouping Test Cases, Test Batch execution and generating reports.)  
----------------------------------------------  
**i) Element Locators**  
   
1) id  
  
2) name  
  
3) className  
  
4) tagName  
  
5) linkText  
  
6) parialLinkText  
  
7) cssSelector  
  
8) xpath  
------------------------------  
Web Elements  
   
Browser -driver object  
Page-----------  
Link  
Button  
Image, Image Button, Image Link  
Edit box  
Text Area  
Check box  
Radio Button  
Drop down box  
List box  
Combo box  
Web Table / HTML Table  
Frame etc...  
-----------------------------------------  
**ii) WebDriver Methods**  
   
**a) Methods on Browser**  
1) get()  
  
2) getTitle()  
  
3) getPageSource()  
  
4) getCurrentUrl()  
  
5) getwindowHandle()  
  
6) close()  
  
7) quit()  
--------------------  
**b) Browser navigation methods**  
1) navigate().to()  
  
2) navigate().back()  
  
3) navigate().forward()  
  
4) navigate().refresh()  
------------------------------  
**c) Methods on Elements**  
1) findElement()  
  
2) findElements()  
  
3) sendkeys()  
  
4) clear()  
  
5) click()  
  
6) isEnabled()  
  
7) isDisplayed()  
  
8) isSelected()  
  
9) getText()  
  
10) getAttribute()  
-----------------------------------  
**d) Others**  
1) manage.window.maximize()  
  
2) .explicitlyWait()  
-------------------------------------  
**iii) Java Programming features**  
  
**A) Java Fundamentals**  
   
1) Comments  
  
2) Data Types  
  
3) Modifiers  
  
4) Variables  
  
5) Operators  
  
6) Conditional Statements  
  
7) Loop Statements  
  
8) String handling  
  
9) Arrays in Java  
  
10) Built in Methods  
  
11) User defined Methods  
  
12) Input and Output Operations, File Handling  
  
13) Exception Handling  
  
**B) Java OOPS**  
   
1) Inheritance  
  
2) Polymorphism  
  
3) Abstraction   
  
4) Encapsulation  
------------------------------------  
**Handling Elements in Selenium**  
  
i) Handling Browser  
  
**Operations on Browser:**  
   
> Launch the Browser  
> Navigate to specified URL  
> Return Current URL  
> Get the Page Title  
> Return Page source  
> Return Window handle  
> Close focused Browser  
> Close all browsers that opened by selenium WebDriver during execution  
-----------  
> Navigate to another URL  
> Navigate back to previous URL  
> Navigate forward  
> Refresh the Browser  
> Maximize the Browser window.  
-------------------------------------  
**Examples:**  
   
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.co.in");  
  
String PageTitle = driver.getTitle();  
System.out.println(PageTitle);  
  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
String PageSource = driver.getPageSource();  
System.out.println(PageSource);  
  
String WindowHandle = driver.getWindowHandle();  
System.out.println(WindowHandle);  
  
driver.close();  
driver.quit();  
----------------------------------------  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.co.in");  
driver.navigate().to("https://login.yahoo.com/");  
System.out.println(driver.getCurrentUrl());  
  
driver.navigate().back();  
System.out.println(driver.getCurrentUrl());  
  
driver.navigate().forward();  
System.out.println(driver.getCurrentUrl());  
  
driver.manage().window().maximize();  
--------------------------------------------  
ii) Handling Edit box  
  
**Operations on Edit box**  
   
> Enter a value  
> Clear the Value  
> Return the Value  
> Check Displayed status  
> Check Enabled statues  
  
**Example:**  
   
//Finding Edit box using id locator  
driver.findElement(By.id("Email")).sendKeys("gcrindia");  
  
//Finding Edit box using name locator  
driver.findElement(By.name("Email")).sendKeys("gcrindia");  
  
//Finding Edit box using CSS Selector locator  
driver.findElement(By.cssSelector("#Email")).sendKeys("gcrindia");  
  
//Finding Edit box using xpath locator  
driver.findElement(By.xpath(".//\*[@id='Email']")).sendKeys("gcrindia");  
  
//Clear the Value  
driver.findElement(By.xpath(".//\*[@id='Email']")).clear();  
--------------------------------  
WebElement Email = driver.findElement(By.xpath(".//\*[@id='Email']"));  
  
Email.sendKeys("gcrindia");  
  
//Return Type of the Object  
Email.getAttribute("type");  
  
//Return the Value  
System.out.println(Email.getText());//gcrindia  
  
//Return Displayed status  
System.out.println(Email.isDisplayed());//true  
  
//Return Enabled status  
System.out.println(Email.isEnabled());//true  
  
//Clear the value  
driver.findElement(By.id("Email")).clear();  
---------------------------------------------  
iii) Handle Text Area  
  
**Capture Text Area/Capture Error Message**  
  
**Capture Text Area:**  
   
driver.get("https://www.gmail.com");  
String s = driver.findElement(By.xpath("html/body/div[1]/div[2]/div[1]/h1")).getText();  
System.out.println(s);  
  
**Capture Error Message:**  
   
WebDriver driver = new FirefoxDriver();  
driver.get("https://login.yahoo.com/");  
driver.manage().window().maximize();  
driver.findElement(By.xpath(".//\*[@id='login-signin']")).click();  
String ErrorMessage = driver.findElement(By.id("mbr-login-error")).getText();   
System.out.println(ErrorMessage);  
  
**Handle Window Popup**  
   
WebDriver driver = new FirefoxDriver();  
driver.get("https://mail.rediff.com/cgi-bin/login.cgi");  
driver.findElement(By.name("proceed")).click();  
  
Alert alert = driver.switchTo().alert();  
String Error\_Message =alert.getText();//Returns Error message  
System.out.println(Error\_Message);  
  
alert.accept();//Closes OK Button  
driver.findElement(By.id("login1")).sendKeys("Inda123");  
---------------------------------  
iv) Handle Button  
   
> Click  
> Check the Displayed status  
> Check the Enabled status  
> Return name of the Object  
> Return type of the Object  
  
**Example:**  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.gmail.co.in");  
  
System.out.println(driver.findElement(By.id("next")).isEnabled());//true  
  
driver.findElement(By.id("next")).click();  
  
System.out.println(driver.findElement(By.id("next")).isDisplayed());//true  
  
System.out.println(driver.findElement(By.id("next")).isEnabled());//false  
  
System.out.println(driver.findElement(By.id("next")).getAttribute("type"));  
  
System.out.println(driver.findElement(By.id("next")).getAttribute("name"));  
  
System.out.println(driver.findElement(By.id("next")).getAttribute("value"));

**Handling Elements in Selenium Part-2**  
  
i) Handle Browser  
ii) Handle Edit box  
iii) Handle Text Area, Error Message, Window Dialog  
iv) handle Button  
-----------------------  
**Return/Capture value from Edit box**  
  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("Email")).sendKeys("India123");  
System.out.println(driver.findElement(By.id("Email")).getAttribute("value"));  
}  
}  
-----------------------  
v) Handle Image  
  
Three types of Image elements in Web Environment.  
  
1) General Image (No functionality)  
  
2) Image Button (Submits)  
  
3) Image Link (Directs to another page/location)  
  
Example:  
  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
System.out.println(driver.findElement(By.id("hplogo")).isDisplayed());//true  
System.out.println(driver.findElement(By.id("hplogo")).getAttribute("title"));  
  
driver.navigate().to("http://newtours.demoaut.com/");  
driver.findElement(By.name("login")).click();  
  
driver.navigate().to("http://www.seleniumhq.org/");  
driver.findElement(By.xpath(".//\*[@id='choice']/tbody/tr/td  
  
[2]/center/a/img")).click();  
  
}  
}  
-----------------------  
vi) Handle Link  
  
Operations on Link  
  
> Click  
> Check the Link Existence  
> Check Enabled status  
> Return Link Name etc...  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
//driver.findElement(By.className("gb\_P")).click();  
//driver.findElement(By.linkText("Gmail")).click();  
//driver.findElement(By.partialLinkText("mail")).click();  
//driver.findElement(By.cssSelector(".gb\_P")).click();  
//driver.findElement(By.xpath(".//\*[@id='gbw']/div/div/div[1]/div  
  
[1]/a")).click();  
WebElement Gmail\_Link = driver.findElement(By.xpath(".//\*  
  
[@id='gbw']/div/div/div[1]/div[1]/a"));  
boolean linkStatus = Gmail\_Link.isDisplayed();  
System.out.println(linkStatus);//true  
  
linkStatus = Gmail\_Link.isEnabled();  
System.out.println(linkStatus);//true  
  
String LinkName = Gmail\_Link.getText();  
System.out.println(LinkName);  
  
Gmail\_Link.click();  
}  
}  
-----------------------  
vii) Handle Radio Button  
  
Operations on Radio Button  
  
> Select  
> Check Enabled status  
> Check Displayed status  
> Check Selected status  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("http://www.gcrit.com/build3/create\_account.php?  
  
osCsid=47gtsrhe41613u5r3eqhgdbas7");  
WebElement maleRadioButton = driver.findElement(By.xpath(".//\*  
  
[@id='bodyContent']/form/div/div[2]/table/tbody/tr[1]/td[2]/input[1]"));  
boolean elementStatus = maleRadioButton.isDisplayed();  
System.out.println(elementStatus);//true  
  
System.out.println(maleRadioButton.isEnabled());//true  
  
System.out.println(maleRadioButton.isSelected());//false  
  
maleRadioButton.click();  
  
System.out.println(maleRadioButton.isSelected());//true  
}  
}  
-----------------------  
viii) Handle Drop Down box  
  
Operations on Drop Down Box  
  
> Select an Item  
> Check Displayed status  
> Check Enabled status  
> Items count  
  
Example:  
  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("http://www.gcrit.com/build3/create\_account.php?  
  
osCsid=47gtsrhe41613u5r3eqhgdbas7");  
Select dropDown = new Select (driver.findElement(By.name("country")));  
//dropDown.selectByIndex(6);//Select an item by index  
//dropDown.selectByVisibleText("India");  
  
List<WebElement> e = dropDown.getOptions();  
int itemsCount = e.size();  
System.out.println(itemsCount);  
}  
}  
-----------------------  
ix) Handle Check box  
  
Operations on Check box  
  
> Select  
> Unselect  
> Check Displayed status  
> Check Enabled status  
> Check selected status  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("file:///E:/HTMLExamples/MultipleCheckbox.html");  
System.out.println(driver.findElement(By.xpath("html/body/input  
  
[2]")).isDisplayed());//true  
  
System.out.println(driver.findElement(By.xpath("html/body/input  
  
[2]")).isEnabled());//true  
  
System.out.println(driver.findElement(By.xpath("html/body/input  
  
[2]")).isSelected());//false  
  
driver.findElement(By.xpath("html/body/input[2]")).click();  
  
System.out.println(driver.findElement(By.xpath("html/body/input  
  
[2]")).isSelected());//true  
  
driver.findElement(By.xpath("html/body/input[2]")).click();  
  
System.out.println(driver.findElement(By.xpath("html/body/input  
  
[2]")).isSelected());//false  
-----------------------  
x) Handle Web Table / HTML Table  
  
Operations on Web Table:  
  
> Get cell value  
> Rows Count  
> Cells Count  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("file:///E:/HTMLExamples/htmlTable.html");  
String s = driver.findElement(By.xpath(".//\*[@id='students']/tbody/tr[2]/td  
  
[2]")).getText();  
System.out.println(s);  
WebElement htmlTable = driver.findElement(By.id("students"));  
  
List <WebElement> rows = htmlTable.findElements(By.tagName("tr"));  
int r = rows.size();  
System.out.println(r);  
  
List <WebElement> cells = htmlTable.findElements(By.tagName("td"));  
int c = cells.size();  
System.out.println(c);  
-----------------------  
xi) Handle inline Elements  
  
The span tag is used to group inline Elements in a Document.  
  
Example 1:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
driver.findElement(By.xpath(".//\*[@id='gbwa']/div[1]/a")).click();  
driver.findElement(By.xpath(".//\*[@id='gb36']/span[1]")).click();  
driver.navigate().back();  
-----------------------  
Example 2:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
driver.manage().window().maximize();  
driver.findElement(By.xpath(".//\*[@id='gbwa']/div[1]/a")).click();  
driver.findElement(By.xpath(".//\*[@id='gbwa']/div[2]/a[1]")).click();  
driver.findElement(By.xpath(".//\*[@id='gb300']/span[1]")).click();  
-----------------------  
Handle Frames  
Handle Mouse Over  
Working with Multiple browser windows.

**Handling Elements in Selenium Part-3**  
  
**In Handling Elements in Selenium Part-1**  
  
i) Handle Browser  
  
ii) Handle Edit box  
  
iii) Handle Text Area, Error Message, Window Dialog  
  
iv) Handle Button  
-----------------------  
**In Handling Elements in Selenium Part-2**  
  
v) Handle Images (General Image, Image Button and Image Link)  
  
vi) Handle Link  
  
vii) Handle Radio Button  
  
viii) Handle Drop down box  
  
ix) Handle Check box  
  
x) Handle Web Table / HTML Table  
  
xi) Handling inline Elements  
-----------------------  
**Handling Elements in Selenium Part-3**  
  
xii) Handle Frames  
  
> HTML frames are used to divide the Browser window into multiple sections, where each section can load a separate HTML document.  
  
> Frames are sections of Web page displayed on top window.  
  
> Whenever we access the page then focus on the top window.  
  
**Switch to a frame is done in two ways**  
  
**1) Using frame index**  
  
Syntax:  
  
driver.swithchTo().frame(int index);  
  
Example:  
  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
driver.switchTo().frame(2);  
driver.findElement(By.xpath("html/body/div[3]/table/tbody[2]/tr[1]/td[1]/a")).click();  
  
**2) Using frame name**  
  
Syntax:  
  
driver.switchTo().frame(String frame Name);  
  
Example:  
  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
driver.switchTo().frame("classFrame");  
driver.findElement(By.xpath("html/body/div[3]/table/tbody[2]/tr[1]/td[1]/a")).click();  
-----------------------  
> Top window to a Frame (frame index/frame name)  
  
> Switch from a frame to Top window  
  
Syntax:  
  
driver.switchTo().defaultContent();  
---------------------  
Navigation:  
  
> Launch the page   
> Switch to 3rd frame  
> Operate an element  
> Back to Top window  
> Switch to 1st frame  
> Operate an element  
--------------------------  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
  
//Switch to 3rd frame  
driver.switchTo().frame(2);  
driver.findElement(By.linkText("com.thoughtworks.selenium")).click();  
Thread.sleep(3000);  
//Switch from 3rd frame to Top window  
driver.switchTo().defaultContent();  
Thread.sleep(3000);  
//Switch to 1st frame  
driver.switchTo().frame(0);  
driver.findElement(By.linkText("org.openqa.selenium")).click();  
-----------------------  
WebDriver driver = new FirefoxDriver();  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
Thread.sleep(3000);  
//Switch to 3rd frame  
driver.switchTo().frame("classFrame");  
driver.findElement(By.linkText("com.thoughtworks.selenium")).click();  
Thread.sleep(3000);  
//Switch from 3rd frame to Top window  
driver.switchTo().defaultContent();  
Thread.sleep(3000);  
//Switch to 1st frame  
driver.switchTo().frame("packageListFrame");  
driver.findElement(By.linkText("org.openqa.selenium")).click();  
}  
}  
-----------------------  
xiii) Handle Mouse hover  
   
WebDriver driver = new FirefoxDriver();  
driver.get("http://www.carmax.com/");  
//create Action builder instance by passing WebDriver instance  
Actions builder = new Actions(driver);  
WebElement menuElement = driver.findElement(By.linkText("Sell Us Your Car"));  
builder.moveToElement(menuElement).build().perform();  
driver.findElement(By.linkText("FAQ")).click();  
-----------------------  
xiv) Handle Multiple Browsers  
  
String parent = driver.getWindowHandle();  
//System.out.println(parent);  
  
Set <String> Handles = driver.getWindowHandles();  
int BrowserCount = Handles.size();  
System.out.println(BrowserCount);  
  
for (String s1:Handles){  
if (! s1.equals(parent)){  
driver.switchTo().window(s1);  
System.out.println(driver.getCurrentUrl());  
}  
}  
driver.switchTo().window(parent);  
System.out.println(driver.getCurrentUrl());  
-----------------------  
xv) Handle Duplicate objects  
  
Redirecting to GCR Shop User Interface from Admin Interface  
(After Login to Admin Interface then Redirect to User Interface)  
  
Example:  
  
driver.get("http://www.gcrit.com/build3/admin/");  
driver.findElement(By.name("username")).sendKeys("admina");  
driver.findElement(By.name("password")).sendKeys("admin@123");  
driver.findElement(By.id("tdb1")).click();  
String url = driver.getCurrentUrl();  
if (url.equals("http://www.gcrit.com/build3/admin/index.php")){  
driver.findElement(By.linkText("Online Catalog")).click();  
}  
System.out.println(driver.getCurrentUrl());  
}  
}  
-----------------------  
**Assignment:**  
  
Enter Password in to 2nd Password Edit box in www.infibeam.com Registration page.

**Cross Browser Testing with Selenium**  
  
1) What is Cross Browser Testing?  
  
Cross Browser Testing is a type of Test to check that our Web Application works as expected in different Browsers.  
  
2) Why Cross Browser Testing?  
A web application can be opened in any web browser by the end user, So we need to ensure that the web application will work as expected in all popular browsers.  
  
3) Popular Web Browsers  
a) Google Chrome: It was released in 2008, its market share approximately 68%  
  
b) Mozilla Firefox: It was released in 2004, its market share approximately 19%  
  
c) Internet Explorer: It was released in 1995, its market share approximately 6.5%  
  
4) Working with different Browsers  
  
> Selenium WebDriver supports Browser compatibility tests on almost every popular browser, including Chrome, Firefox, IE, Opera and Safari.  
  
> The WebDriver API drives the web browser as the real user would drive it.  
  
> By default, Firefox driver comes with selenium-serverstanalone.jar library added.  
  
> For Chrome, IE, Safari, Opera, there are libraries that need to be instantiated externally.  
  
5) How to conduct Cross Browser Testing using Selenium WebDriver?  
  
a) Element Locators - Same for all Browsers.  
  
b) WebDriver Methods/Commands -Same for all Browsers.  
  
c) Programming features (Java/C#/Python/Perl/Ruby/PHP) - Same for all Browsers.  
  
d) JUnit / TestNG Annotations - Same for all Browsers.  
  
e) Browser Driver - various from one browser to another.  
  
Note: For Mozilla Firefox, just create the driver, For other browsers, libraries that need to be instantiated externally.  
------------------------  
f) Inspect Elements -  
  
For Mozilla Firefox -Built in feature Page Inspector,  
(Install Firebug and Firepath)  
  
For Chrome and IE - Built in Developer tools  
  
6) Create Browser Drivers  
   
(For Google Chrome, IE and Other Browsers, download Browser drivers and set   
  
path in Selenium Test Scripts)  
  
a) Mozilla Firefox Browser:  
  
WebDriver driverName = new FirefoxDriver();  
  
b) Google Chrome  
  
//Instantiate Chrome Browser driver  
  
System.setproperty("webdriver.chrome.driver", "driver .exe file path");  
WebDriver driverName = new ChromeDriver();  
  
c) IE Browser driver  
  
System.setproperty("webdriver.ie.driver", "driver .exe file path");  
WebDriver driverName = new InternetExplorerDriver();  
  
7) Create a Test Case and Execute using Mozilla Firefox, Chrome and IE Browsers.  
  
**Test Case:** Verify Launch Application (Google) functionality in Firefox, Chrome and IE Browsers.  
  
**Test Steps:**  
   
i) Launch the Browser  
ii) Navigate to https://www.google.com url  
  
**Verification point:**  
Capture the page Title (Actual) and Compare with Expected.  
  
**Expected Page Title:** Google  
-------------------------------  
**a) Test Case for Mozilla Firefox Browser**  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println("Google Application Launched - Passed");  
}  
else {  
System.out.println("Google Application Not Launched -Failed");      
}  
driver.close();  
------------------------------  
**b) Test Case for Google Chrome Browser**  
   
System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe");  
WebDriver driver = new ChromeDriver();  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println("Google Application Launched - Passed");  
}  
else {  
System.out.println("Google Application Not Launched -Failed");      
}  
driver.close();  
------------------------------  
**c) Test Case for internet Explorer Browser**  
  
System.setProperty("webdriver.ie.driver", "E:\\IEDriverServer.exe");  
WebDriver driver = new InternetExplorerDriver();  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println("Google Application Launched - Passed");  
}  
else {  
System.out.println("Google Application Not Launched -Failed");      
}  
driver.close();  
------------------------------  
8) Create a Test Case and Execute using Mozilla Firefox, Chrome and IE Browsers Continuously.  
  
public class TestCase1 {  
public static WebDriver driver;  
public static int browser;  
public static String BrowserName;  
  
public static void main(String[] args) {  
  
for (browser = 1; browser <= 3; browser++){  
if (browser == 1) {  
driver = new FirefoxDriver();  
BrowserName = "Mozilla Firefox Browser: ";  
}  
else if (browser == 2) {  
System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe");  
driver = new ChromeDriver();  
BrowserName = "Google Chrome Browser: ";  
}  
else if (browser == 3){  
System.setProperty("webdriver.ie.driver", "E:\\IEDriverServer.exe");  
driver = new InternetExplorerDriver();  
BrowserName = "Internet Explorer Browser: ";  
}  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println(BrowserName + " - Google Application Launched - Passed");  
}  
else {  
System.out.println(BrowserName + " - Google Application Not Launched -  
  
Failed");      
}  
driver.close();  
}  
}  
}

**Writing Selenium WebDriver Test Cases**  
  
**Pre-requisites  to create Selenium Test Cases**  
  
i) Test Scenario  
  
ii) Element Locators - To Locate / identify/recognize Elements.  
  
iii) Selenium WebDriver Commands/Methods - To perform Operations on Elements.  
  
iv) Programming Features - To enhance Test cases  
------------------  
v) JUnit/TestNG Annotations - To group Test cases, Batch Testing and generate Test Reports.  
---------------------------------------------------------------  
1) Test Case: Verify Internal and External Links in Wikipedia.org  
  
Internal Link: It redirects to another page or location in the same application.  
  
External Link: It redirects to another page or location in other application  
-----------  
**Test Steps:**  
  
i) Launch the Browser  
ii) Navigate to Selenium page in Wikipedia.org  
iii) Click "Create Account" Link  
iv) Capture Current Url  
v) Navigate back to Selenium Page  
vi) Click "selenium.org" Link  
vii) Capture Current Url  
viii) Close Browser  
---------------------  
**Verification Points:**  
   
i) Check if the First URL is an Internal Link or not?  
ii) Check if the second URL is an External Link or not?  
---------------------------------------  
**Selenium Webdriver Test Case:**  
WebDriver driver = new FirefoxDriver();  
driver.get("https://en.wikipedia.org/wiki/Selenium\_%28software%29");  
driver.findElement(By.linkText("Create account")).click();  
String url = driver.getCurrentUrl();  
//System.out.println(url);  
if (url.contains("wikipedia.org")){  
System.out.println("It is an Internal Link - Redirected to another page in the Same Application - Passed");  
}  
else{  
System.out.println("It is an External Link - Redirected to another page in the Other Application -Failed");  
}  
driver.navigate().back();  
driver.findElement(By.partialLinkText("seleniumhq.org")).click();  
url = driver.getCurrentUrl();  
  
if (! url.contains("wikipedia.org")){  
System.out.println("It is an External Link - Redirected to another page in the Other Application - Passed");  
}  
else{  
System.out.println("It is an Internal Link - Redirected to another page in the same Application - Failed");  
}  
driver.close();  
}  
}  
------------------------------------------------------  
2) Test Case: Verify Element Existence (Gmail link existence in Google home page)  
  
**Test Steps:**  
  
i) Launch the browser  
ii) Navigate to Google.com (Google Home page)  
------------------  
**Verification point:**  
  
Check the existence of Gmail link.  
  
Selenium Test Case:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
boolean existence = driver.findElement(By.linkText("Gmail")).isDisplayed();  
  
if (existence == true){  
System.out.println("Gmail Link Exists - Passed");  
}  
else {  
System.out.println("Gmail Link Not Exists - Failed");  
}  
}  
}  
------------------------------------------  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
  
try  
{  
if (driver.findElement(By.linkText("Gmailabc")).isDisplayed()){  
System.out.println("Gmail Link Exists - Passed");  
}  
}  
  
catch (NoSuchElementException e)  
{  
System.out.println("Gmail Link Not Exists - Failed");  
}  
driver.close();  
}  
}  
--------------------------------------------------------------  
3) Test Case: Login to Indian Railways Online web portal  
  
**Test Steps:**  
   
i) Launch the Browser  
ii) Navigate to https://www.irctc.co.in (Indian Railways Online web portal)  
iii) Enter User Id  
iv) Enter Password  
v) Enter Captcha (Verification Code)  
vi) Click Login Button  
-------------------------------  
**Verification Point:**  
   
Capture the URL and Compare with https://www.irctc.co.in/eticketing/home   
  
**Test Data:**  
   
User Id: gcreddy7 (Static Input)  
Password: gld938 (Static Input)  
  
Captcha: (Dynamic Input)  
---------------------------------------------------  
Selenium WebDriver Test Case:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.irctc.co.in/");  
driver.findElement(By.id("usernameId")).sendKeys("gcreddy7");  
driver.findElement(By.className("loginPassword")).sendKeys("gld938");  
  
Scanner scan = new Scanner(System.in);//System.in is an Input stream  
System.out.println("Enter Captcha");  
String captcha = scan.nextLine();  
  
driver.findElement(By.className("loginCaptcha")).sendKeys(captcha);  
driver.findElement(By.id("loginbutton")).click();  
String url = driver.getCurrentUrl();  
  
if (url.equals("https://www.irctc.co.in/eticketing/home")){  
System.out.println("Login Successful - Passed");  
}  
else{  
System.out.println("Login Unsuccessful - Failed");      
}  
driver.close();  
}  
}  
--------------------------------------------------------  
4) Test Case: Verify Customer Registration in gcrShop Web portal  
**Test Steps:**  
   
i) Launch the Browser  
ii) Navigate to http://gcrit.com/build3/  
iii) Enter all Mandatory fields  
iv) Click "Continue" Button  
-----------------------  
**Verification point:**  
   
Capture conformation message and compare with expected.  
  
WebDriver driver = new FirefoxDriver();  
driver.get("http://gcrit.com/build3/");  
driver.findElement(By.linkText("create an account")).click();  
driver.findElement(By.xpath(".//\*[@id='bodyContent']/form/div/div[2]/table/tbody/tr[1]/td[2]/input  
  
[1]")).click();  
driver.findElement(By.name("firstname")).sendKeys("Rahman");  
driver.findElement(By.name("lastname")).sendKeys("Mohommed");  
driver.findElement(By.name("dob")).sendKeys("10/20/1990");  
driver.findElement(By.name("email\_address")).sendKeys("rahman1237@gmail.com");  
driver.findElement(By.name("street\_address")).sendKeys("abcd xyz");  
driver.findElement(By.name("postcode")).sendKeys("12345");  
driver.findElement(By.name("city")).sendKeys("Hyderabad");  
driver.findElement(By.name("state")).sendKeys("Telangana");  
  
Select Dropdown = new Select (driver.findElement(By.name("country")));  
Dropdown.selectByVisibleText("India");  
  
driver.findElement(By.name("telephone")).sendKeys("9234565453");  
driver.findElement(By.name("password")).sendKeys("abcd123");  
driver.findElement(By.name("confirmation")).sendKeys("abcd123");  
driver.findElement(By.id("tdb4")).click();  
  
String ConformationMessage = driver.findElement(By.xpath(".//\*[@id='bodyContent']/h1")).getText();  
  
if (ConformationMessage.equals("Your Account Has Been Created!")){  
System.out.println("Customer Registration Successful - Passed");      
}  
else{  
System.out.println("Customer Registration Unsuccessful - Failed");      
}  
driver.close();  
--------------------------------------  
5) Test Case: Verify Customer Login in gcrShop Web portal  
**Test Steps:**  
  
i) Launch the Browser  
ii) Navigate to http://www.gcrit.com/build3/  
iii) Click "login" Link  
iv) Enter Email Address  
v) Enter Password  
vi) Click "Sign In" Button  
-----------------------  
**Verification Point:**  
   
Capture current url and compare with http://www.gcrit.com/build3/index.php  
  
**Selenium Test Case:**  
WebDriver driver = new FirefoxDriver();  
driver.get("http://gcrit.com/build3/");  
driver.findElement(By.linkText("login")).click();  
driver.findElement(By.name("email\_address")).sendKeys("rahman1237@gmail.com");  
driver.findElement(By.name("password")).sendKeys("abcd123");  
driver.findElement(By.id("tdb5")).click();  
String url = driver.getCurrentUrl();  
//System.out.println(url);  
  
if (url.contains("http://www.gcrit.com/build3/index.php")){  
System.out.println("Login Successful - Passed");  
}  
else{  
System.out.println("Login Unsuccessful - Failed");  
}  
driver.close();  
-------------------------------------------------

**Writing Selenium WebDriver Test Cases Part -2**  
  
**Writing Selenium WebDriver Test Cases**  
  
**6) Test Case:** Check Admin Login Functionality in gcrShop web portal (Positive Test Case)  
**7) Test Case:** Check Error Message/s in Admin Login Functionality(Negative Test Case)  
**8) Test Case:** Check Admin Functionality with valid and invalid inputs(Positive and Negative Testing)  
-------------------------------------  
6) Test Case: Check Admin Login Functionality in gcrShop web portal (Positive Test Case)  
  
**Test Steps:**  
i) Launch the Browser  
ii) Navigate to gcrShop Admin Interface (http://www.gcrit.com/build3/admin/)  
iii) Enter Valid User name  
iv) Enter Valid Password  
v) Click "Login" Button  
-------------------------  
**Verification Point:**  
Capture the url and compare with expected.  
  
**Expected url:**http://www.gcrit.com/build3/admin/index.php  
  
**Actual:**  
**Test Data:**  
User name = admin  
Password = admin@123  
-----------------------------------  
**Selenium Test Case:**  
   
WebDriver driver = new FirefoxDriver();  
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();  
  
String url = driver.getCurrentUrl();  
  
if (url.equals("http://www.gcrit.com/build3/admin/index.php")){  
System.out.println("Admin Login Successful - Passed");  
}  
else {  
System.out.println("Admin Login Unsuccessful - Failed");  
}  
driver.close();  
--------------------------------------  
7) Test Case: Check Error Message/s in Admin Login Functionality (Negative Test Case)  
   
**Test Steps:**  
i) Launch the Browser  
ii) Navigate to gcrShop Admin Interface (http://www.gcrit.com/build3/admin/)  
iii) Enter Invalid User name and / or Password  
iv) Click "Login" Button  
----------------------------  
**Verification point:**  
Capture the Error Message and compare with expected.  
  
**Expected:** Error: Invalid administrator login attempt.  
  
**Test Data:**  
User Name: admina  
Password: admin@123  
----------------------------------------------------  
**Selenium Test Case:**  
   
WebDriver driver = new FirefoxDriver();  
driver.get("http://www.gcrit.com/build3/admin/");  
driver.findElement(By.name("username")).sendKeys("admina");  
driver.findElement(By.name("password")).sendKeys("admin@123");  
driver.findElement(By.id("tdb1")).click();  
  
String url = driver.getCurrentUrl();  
if (url.contains("http://www.gcrit.com/build3/admin/login.php")){  
String ErrorMessage = driver.findElement(By.className("messageStackError")).getText();  
  
if (ErrorMessage.contains("Invalid administrator login attempt.")){  
System.out.println("Handling Invalid Inputs - Passed");  
}  
}  
else {  
System.out.println("Not Handling Invalid Inputs - Failed");  
}  
driver.close();  
----------------------------------  
Assignment:  
  
Verify the maximum Login attempts (For invalid inputs only)  
Verification: After 3 attempts it blocks the Login Functionality for 5 minutes.  
----------------------------------------------  
8) Test Case: Check Admin Functionality with valid and invalid inputs (Positive and Negative Testing)  
Test steps:  
i) Launch the Browser  
ii) Navigate to gcrShop Admin Interface (http://www.gcrit.com/build3/admin/)  
iii) Enter valid "User name"  
iv) Enter Valid "Password"  
v) Click "Login" Button  
---------------  
\* Repeat the navigation with Invalid User Name and / or Password  
  
Verification points:  
i) Capture the url and compare with expected.  
Expected: http://www.gcrit.com/build3/admin/index.php  
  
Test Data:  
User name = admin  
Password = admin@123  
--------------  
ii) Capture the Error message and compare with expected:  
Expected =Error: Invalid administrator login attempt.  
  
Test Data:   
User name = admina  
Password = admin@123a  
(Invalid User name and Invalid Password)  
  
Other Negative Scenarios:  
  
1) Valid User name and Invalid Password  
2) Invalid user Name and Valid Password  
3) Blank User name and Valid Password/Invalid Password  
4) Valid / Invalid User name and Blank password  
5) Blank User name and Blank password  
-----------------------------------  
**Selenium Test Case:**  
  
public class VerifyLogin {  
static int i;  
static String username, password, iteration;  
public static void main(String[] args) throws InterruptedException {  
WebDriver driver = new FirefoxDriver();  
  
for (i = 1; i <= 2; i++){  
if (i == 1)    {  
username = "admin";  
password = "admin@123";  
iteration = "Iteration 1:";  
}  
else if(i == 2){  
username = "admina";  
password = "admin@123a";  
iteration = "Iteration 2:";      
}  
  
driver.get("http://www.gcrit.com/build3/admin/");  
driver.findElement(By.name("username")).sendKeys(username);  
driver.findElement(By.name("password")).sendKeys(password);  
Thread.sleep(3000);  
driver.findElement(By.id("tdb1")).click();  
  
String url = driver.getCurrentUrl();  
  
if (url.contains("http://www.gcrit.com/build3/admin/index.php")){  
System.out.println(iteration + ("Input Data: ") + username + ", "+ password + " Admin Login   
  
Successful -Passed");  
driver.findElement(By.linkText("Logoff")).click();  
Thread.sleep(2000);  
}  
else if (! url.contains("http://www.gcrit.com/build3/admin/index.php")){  
String ErrorMessage = driver.findElement(By.className("messageStackError")).getText();  
if (ErrorMessage.contains("Invalid administrator login attempt.")){  
System.out.println(iteration + ("Input Data: ") + username + ", "+ password + " Handling Invalid   
  
Inputs -Passed");  
}  
else {  
System.out.println(iteration +  "- Failed");  
}  
}  
}  
driver.close();  
}  
}  
---------------------------------------------  
9) Test Case: Check communication between different browsers.  
   
**Test Steps:**  
i) Create Mozilla Firefox driver, Google chrome driver and IE driver.  
ii) Launch three different applications  
iii) Interact from one application to another  
iv) Close all browsers one by one.  
-------------------------------------------  
**Selenium Test Case:**  
  
WebDriver firefoxDriver = new FirefoxDriver();  
firefoxDriver.get("https://www.google.com");  
firefoxDriver.findElement(By.linkText("Gmail")).click();  
String text = firefoxDriver.findElement(By.xpath("html/body/div[1]/div[2]/div[1]/h2")).getText();  
  
System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe");  
WebDriver chromeDriver = new ChromeDriver();  
chromeDriver.get("http://www.gcrit.com/build3/create\_account.php?  
  
osCsid=1vbg1oj32ole3qrcv4b6mr7m24");  
chromeDriver.findElement(By.name("firstname")).sendKeys(text);  
Thread.sleep(3000);  
  
System.setProperty("webdriver.ie.driver", "E:\\IEDriverServer.exe");  
WebDriver IEDriver = new InternetExplorerDriver();  
IEDriver.get("https://in.mail.yahoo.com/");  
  
firefoxDriver.close();  
chromeDriver.close();  
IEDriver.close();

**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);  
  
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();

**Introduction to TestNG Framework**  
  
I) Overview  
  
II) Install TestNG and write First TestNG Test Case.  
  
III) Create multiple Test Cases and Run  
  
IV) Execute multiple programs/classes using XML  
-----------------------------------------------  
**I) Overview**  
  
> In Selenium using Java there are two Testing frameworks available,  
  
1) JUnit  
  
2) TestNG  
------------------------------  
**TestNG Testing Framework**  
> TestNG is a testing framework designed to simplify a broad range of Testing needs, from Unit Testing to System Testing.  
  
> Initially developed for Unit Testing, now used for all kinds of Testing.  
  
> TestNG is an open source framework, where NG stands for next generation.  
  
> TestNG inspired from Junit(Java platform) and NUnit (.NET platform), but introducing some new functionalities that make it more powerful and easier to use.  
--------------------------------------------  
Advantages of TestNG  
  
1) TestNG Annotations are easy to create Test Cases  
  
2) Test Cases can be grouped and prioritized more easily.  
  
3) Supports Parameterization.  
  
4) Supports Data driven Testing using Dataproviders.  
  
5) Generates HTML reports  
  
6) Parallel test execution is possible.  
  
7) Readily supports integration with other tools and plug ins like Eclipse IDE, build tools Ant, Maven etc...  
---------------------------------  
Note: Using TestNG we can create Test Cases, group Test Cases, prioritize Test Cases, execute Test Cases and generate Test Reports.  
---------------------------------------------  
**II) Install TestNG and write first Test Case**  
   
In Eclipse  
  
Help menu -> Install New Software -> Click Add  
-> Enter Name as "TestNG"  
-> Enter URL as "http://beust.com/eclipse/"  
-> Select "TestNG"  
-> Next -> Next -> Accept the Agreement -> Finish  
------------------------------------------------------  
Write TestNG Test Case  
  
**Manual Test Case**  
  
Test Case Name: Verify title of the Page  
  
Test Steps:  
  
1) Launch Browser  
  
2) Navigate to gmail.com  
------------------------------  
Verification point  
Capture the Page title and compare with expected  
  
Expected = Gmail  
  
Actual =   
  
Status =  
--------------------------------------------  
TestNG Test Case:  
  
public class Sample {  
@Test  
public void verifyTitle(){  
WebDriver driver = new FirefoxDriver();      
driver.get("https://www.gmail.com");  
String pageTitle = driver.getTitle();  
Assert.assertEquals(pageTitle, "Gmail");  
}  
}  
-------------------------------------------  
Note:  
1) main method is not used for TestNG programs.  
  
2) TestNG programs contains only methods that contain @Test Annotations  
  
3) if we don't write @Test Annotations then the methods are not going to be executed.  
----------------------------------------------------------  
**III) Write Multiple Test Cases**  
   
public class Sample {  
@Test   
public void testA(){  
Assert.assertEquals("Gmail", "Gmail");  
}  
@Test   
public void testC(){  
Assert.assertEquals("Gmail", "Google");  
}  
@Test   
public void testB(){  
Assert.assertEquals("Yahoo", "Yahoo");  
}  
}  
  
Note: TestNG Test cases are executed in Alphabetical order,  
  
If You want to control the Test execution process then use priority attribute.  
---------------------------------------------------------------  
public class Sample {  
@Test (priority = 3)  
public void abcd(){  
Assert.assertEquals("Gmail", "Gmail");  
}  
@Test (priority = 2)  
public void xyz(){  
Assert.assertEquals("Gmail", "Google");  
}  
@Test (priority = 1)  
public void pqr(){  
Assert.assertEquals("Yahoo", "Yahoo");  
}  
}  
-------------------  
General Test Execution Flow:  
abcd  
pqr  
xyz  
-----------------  
pqr  
xyz  
abcd  
--------------------  
public class Sample {  
@Test (priority = 3)  
public void abcd(){  
Assert.assertEquals("Gmail", "Gmail");  
}  
@Test (priority = 1, enabled = false)  
public void xyz(){  
Assert.assertEquals("Google", "Google");  
}  
@Test (priority = 2)  
public void pqr(){  
Assert.assertEquals("Yahoo", "Yahoo");  
}  
}  
--------------------------------  
public class Sample {  
@Test   
public void login(){  
System.out.println("Login Successful");  
}  
@Test (dependsOnMethods = {"login"})  
public void search(){  
System.out.println("Search Successful");  
}  
@Test (dependsOnMethods = {"search"})  
public void advancedSearch(){  
System.out.println("Advanced Search Successful");  
}  
@Test (dependsOnMethods = {"advancedSearch"})  
public void logout(){  
System.out.println("Logout Successful");  
}  
}  
-------------------------------------  
Hard dependency  
@Test (dependsOnMethods ={"methodName"})  
  
Soft Dependency  
@Test (dependsOnMethods ={"methodName"}, alwaysRun=true)  
--------------------------------------------------  
public class Sample {  
public WebDriver driver;  
@Test (priority=1)  
public void launchBrowser(){  
driver = new FirefoxDriver();      
}  
  
@Test (priority=2)  
public void verifyPageTitle1(){  
driver.get("https://www.gmail.com");  
Assert.assertEquals("Gmail", driver.getTitle());  
}  
  
@Test(priority=3)  
public void verifyPageTitle2(){  
driver.get("https://in.yahoo.com/");  
Assert.assertEquals("Yahoo", driver.getTitle());  
}  
@Test (priority=4)  
public void closeBrowser(){  
driver.close();  
}  
}  
--------------------------------------  
Test Execution Flow  
1) closebrowser  
2) launchBrowser  
3) verifyPageTitle1  
4) verifyPageTitle2  
  
Test Execution Flow (As per priorities) :  
  
1) launchBrowser  
2) verifyPageTitle1  
3) verifyPageTitle2  
4) closeBrowser  
--------------------------------------------------  
BeforeMethod and AfterMethod Annotations  
  
@BeforeMethod - Pre-condition for every Test case in a Class/Program  
@AfterMethod Post-condition for every Test case in a Class/Program  
  
**Example:**  
  
@BeforeMethod  
public void launchBrowser(){  
driver = new FirefoxDriver();      
}  
  
@Test   
public void verifyPageTitle1(){  
driver.get("https://www.gmail.com");  
Assert.assertEquals("Gmail", driver.getTitle());  
}  
  
@Test  
public void verifyPageTitle2(){  
driver.get("https://in.yahoo.com/");  
Assert.assertEquals("Yahoo", driver.getTitle());  
}  
@AfterMethod  
public void closeBrowser(){  
driver.close();  
}  
}  
--------------------------------------  
Test Execution Flow:  
launchBrowser -pre-condition for every test case.  
closeBrowser -post-condition for every test case  
  
verifyPageTitle1  
verifyPageTitle2  
---------------------------  
launchBrowser  
verifyPageTitle1  
closeBrowser  
  
launchBrowser  
verifyPageTitle2  
closeBrowser  
---------------------------------------------  
BeforeClass and AfterClass Annotations  
  
@BeforeClass -Pre-condition for All Test cases in a Class/Program  
@AfterClasee -Post-condition for All Test cases in a Class/Program  
  
**Example:**  
   
@BeforeClass  
public void launchBrowser(){  
driver = new FirefoxDriver();      
}  
  
@Test   
public void verifyPageTitle1(){  
driver.get("https://www.gmail.com");  
Assert.assertEquals("Gmail", driver.getTitle());  
}  
  
@Test  
public void verifyPageTitle2(){  
driver.get("https://in.yahoo.com/");  
Assert.assertEquals("Yahoo", driver.getTitle());  
}  
@AfterClass  
public void closeBrowser(){  
driver.close();  
}  
}

**TestNG Framework in Selenium Part-2**  
  
**Introduction to TestNG Testing Framework**  
I) Overview  
  
II) Install TestNG and write First TestNG Test Case  
  
III) Create multiple Test Cases and execute  
---------------------------------------------  
**TestNG Framework in Selenium Part-2**  
  
IV) Executing multiple programs / classes using XML file  
  
V) Grouping Test Cases  
  
VI) Parallel Test Execution  
-------------------------------------------------  
**IV) Executing multiple programs / classes using XML file**  
  
**Tags in XML**  
  
<suite name = "Suite Name">  
  <test name ="Test Name">  
    <classes>  
      <class name = "package.Class1Name"/>  
      <class name = "package.Class2Name"/>      
</classes>  
</test>  
</suite>  
----------------------------------------------  
**Create XML file**  
  
Select Java project/Package > Right click > New > Other...  
> Enter TestNG and Select TestNG Class  
> Enter source and package names  
> Enter XML file Name  
----------------------------------  
**XML File**  
<suite name="Ecommerce">  
  <test name="SanityTests">  
    <classes>  
      <class name="abcd.NewTest1"/>  
      <class name="abcd.NewTest2"/>  
    </classes>  
  </test>   
</suite>   
---------------------------  
**Class 1**  
  
public class NewTest1 {  
@BeforeClass  
public void login(){  
System.out.println("Login Successful");  
}  
@AfterClass  
public void logout(){  
System.out.println("Logout Successful");  
}  
@Test (priority = 1)  
public void addVendor(){  
System.out.println("Add Vendor Successful");      
}  
@Test(priority = 2)  
public void addProduct(){  
System.out.println("Add Product Successful");      
}  
@Test(priority = 3)  
public void addCurrency(){  
System.out.println("Add Currency Successful");      
}  
}  
----------------------------------  
**Class 2**  
  
@BeforeClass  
public void login(){  
System.out.println("Login Successful");  
}  
@AfterClass  
public void logout(){  
System.out.println("Logout Successful");  
}  
@Test (priority = 1)  
public void deleteVendor(){  
System.out.println("Delete Vendor Successful");      
}  
@Test(priority = 2)  
public void deleteProduct(){  
System.out.println("Delete Product Successful");      
}  
@Test(priority = 3)  
public void deleteCurrency(){  
System.out.println("Delete Currency Successful");      
}  
}  
-----------------------------------  
TestNG Annotations  
  
@Test - The annotated method is a part of a Test Case  
  
@BeforeMethod - The annotated method will be run before each Test method  
@AfterMethod - The annotated method will be run after each Test Method  
  
@BeforeClass - The annotated method will be run before the first test method in the current class is   
  
invoked.  
@AfterClass - The annotated method will be run after all the Test methods in the current class have   
  
been run.  
  
@BeforeTest - The annotated method will be run before any Test method belonging to classes inside   
  
the tag is run  
@AfterTest - The annotated method will be run after all the Test methods belonging to the classes   
  
inside the tage have run.  
-----------------------------------------------------  
**V) Grouping Test Cases**  
  
**XML File**  
  
<suite name="Suite" >  
  <test name="Test">  
  <groups>  
  <run>  
  <include name = "regression"/>  
  </run>  
  </groups>  
    <classes>  
      <class name="abcd.NewTest3"/>  
      </classes>  
  </test><!-- Test -->  
</suite><!-- Suite -->  
---------------------------------------------  
**Class File**  
  
public class NewTest3 {  
@Test(groups = {"sanity", "regression"}, priority =1)  
public void login(){  
System.out.println("Login Successful");  
}  
@Test (groups = {"sanity"}, priority =3)  
public void fundTransfer(){  
System.out.println("Fund Transfer Successful");  
}  
@Test(groups = {"sanity"}, priority =2)  
public void search(){  
System.out.println("Search Successful");  
}  
@Test (groups = {"regression"}, priority =2)  
public void advancedSearch(){  
System.out.println("Advanced Search Successful");  
}  
@Test(groups = {"regression"}, priority =3)  
public void prePaidRecharge(){  
System.out.println("PrePaid Recharge Successful");  
}  
@Test(groups = {"regression"}, priority =4)  
public void billPayments(){  
System.out.println("Bill Payments Successful");  
}  
@Test(groups = {"sanity", "regression"}, priority =10)  
public void logout(){  
System.out.println("Logout Successful");  
}  
}  
-------------------------------------  
**VI) Parallel Test Execution**  
  
1) Parallel Test Execution (Methods)  
**XML File**  
  
<suite name="Suite" parallel="methods" thread-count ="3">  
  <test name="Test">  
    <classes>  
      <class name="abcd.NewTest5"/>  
    </classes>  
  </test><!-- Test -->  
</suite><!-- Suite -->  
  
**Class File**  
  
public class NewTest5 {  
@Test  
public void testCase1(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 1 is Successful - Thread id is: "+ id);        
}  
@Test  
public void testCase2(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 2 is Successful - Thread id is: "+ id);        
}  
@Test  
public void testCase3(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 3 is Successful - Thread id is: "+ id);        
}  
}  
----------------------------------------------------   
Note: A Thread is a concurrent unit of execution.   
------------------------------------   
2) Parallel Test Execution (Classes)  
  
**XML File**  
  
<suite name="Suite" parallel="classes" thread-count ="2">  
  <test name="Test">  
    <classes>  
      <class name="abcd.NewTest5"/>  
      <class name="abcd.NewTest6"/>  
      </classes>  
  </test><!-- Test -->  
</suite><!-- Suite -->  
---------------------------------------------  
**Class 1 File**  
  
public class NewTest5 {  
@Test  
public void testCase1(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 1 is Successful - Thread id is: "+ id);        
}  
@Test  
public void testCase2(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 2 is Successful - Thread id is: "+ id);        
}  
@Test  
public void testCase3(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 3 is Successful - Thread id is: "+ id);        
}  
}  
---------------------------------------  
**Class 2 File**  
  
public class NewTest6 {  
@Test  
public void testCase4(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 4 is Successful - Thread id is: "+ id);        
}  
@Test  
public void testCase5(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 5 is Successful - Thread id is: "+ id);        
}  
@Test  
public void testCase6(){  
long id = Thread.currentThread().getId();  
System.out.println("Test Case 6 is Successful - Thread id is: "+ id);        
}  
}

**Data Driven Testing in Selenium**  
  
**Data Driven Testing using Data Provider**  
  
Examples:  
  
1) Read Test data (String type) from excel file and perform Data driven Testing for Admin Login Functionality.  
  
2) Read Test data (Integer type) from excel file and perform Data driven Testing for Addition Functionality.  
-------------------------------------------------------  
1) Read Test data (String type) from excel file and perform Data driven Testing for Login Functionality.  
  
**Steps for Admin Login**  
i) Launch the Browser  
ii) Navigate to http://www.gcrit.com/build3/admin/  
iii) Enter User name  
iv) Enter Password  
v) Click "Login" Button  
-----------------  
**Verification Point**  
Capture URL, and compare with expected.  
  
Expected = http://www.gcrit.com/build3/admin/index.php  
--------------------------------------  
Prepare Test data file.  
--------------------------------  
> Download Excel jar and extract  
> Add Excel jar to Java Project in Eclipse  
-------------------------------------  
Selenium Test Case:  
  
public class DatadrivenTest {  
public WebDriver driver;  
@Test(dataProvider ="testdata")  
public void login(String username, String password){  
driver = new FirefoxDriver();  
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();  
Assert.assertEquals("http://www.gcrit.com/build3/admin/index.php", driver.getCurrentUrl());  
driver.close();  
}  
      
@DataProvider(name = "testdata")  
public Object [] [] readExcel() throws BiffException, IOException {  
File f = new File("C:/Users/gcreddy/Desktop/Input.xls");  
Workbook w = Workbook.getWorkbook(f);  
Sheet s = w.getSheet("Sheet1");  
int rows = s.getRows();  
int columns = s.getColumns();  
//System.out.println(rows);  
//System.out.println(columns);  
  
String inputData [] [] = new String [rows] [columns];  
for (int i=0; i<rows; i++){  
    for (int j=0; j<columns; j++){  
    Cell c = s.getCell(j, i);  
    inputData [i][j] = c.getContents();  
    System.out.println(inputData[i][j]);  
}  
}  
return inputData;  
}  
}  
-----------------------------------------  
2) Read Test data (Integer type) from excel file and perform Data driven Testing for Addition Functionality.  
  
public class Datadriven2 {  
@Test(dataProvider="testdata")  
public void add(String val1, String val2, String val3){  
int a = Integer.parseInt(val1);  
int b = Integer.parseInt(val2);  
int c = Integer.parseInt(val3);  
int result = a + b + c;  
System.out.println(result);  
}  
@DataProvider(name="testdata")      
public Object [] [] readExcel() throws BiffException, IOException {  
File f = new File("C:/Users/gcreddy/Desktop/Input.xls");  
Workbook w = Workbook.getWorkbook(f);  
Sheet s = w.getSheet("Sheet1");  
int rows = s.getRows();  
int columns = s.getColumns();  
//System.out.println(rows);  
//System.out.println(columns);  
  
String inputData [] [] = new String [rows][columns];  
for (int i=0; i<rows; i++){  
    for (int j=0; j<columns; j++){  
        Cell c = s.getCell(j, i);  
        inputData[i][j] = c.getContents();  
        //System.out.println(inputData[i][j]);  
        }  
}  
return inputData;      
}  
}  
------------------------------------------------------------  
Difference between Reading String type data and Integer type data.  
  
If it is String type data then you can use the data as it is.  
  
If it is Integer type data, convert the data (from String type to Integer type) then use the data.  
-------------------------  
Why we need to convert the data?  
  
If you read data then Computer program considers the data as String type data,  
In order to perform mathematical operations then we need to convert the data.  
-----------------------------------------------------------  
Assignments  
  
i) After Assert fails then close the Browser.   
  
ii) Read data from excel (range of records)   
----------------------------------------  
Assignments  
  
i) Verify "Telephone Number" field in Customer registration form.  
  
Enter valid inputs for all fields except "Telephone Number" and click  
  
Capture error message and compare with expected.  
  
Write exception.  
------------------------------------------  
Data has 3 factors  
  
Type (Alfa bytes, numeric, Alfa-numeric etc...)  
Size (10 digits)  
Range (25 to 30 Years, 30 to 35 Years ....)  
--------------------------------------------------------  
ii) Verify Email field in http://niittrainercentral.com/UserRegistration.aspx form  
  
Check Email format  
something@something.something  
Or  
something@something.something.something  
------------------------------------------------------  
iii) Verify the Date field(Date of Birth:) in customer Registration form  
----------------------------------------------------------------  
Project Automation  
  
Selenium WebDriver + Programming (Java) + Testing Framework(JUnit/TestNG)  
Or  
Selenium WebDriver + Programming (C#) + Testing Framework(NUnit)  
Or  
Selenium IDE  
Or  
Selenium WebDriver + Programming (Java)