Selenium WebDriver

Updated By: Ankit Mittal

techadda.ankit.mittal@gmail.com

DAY 1

IMPORTANT JAVA TOPICS FOR SELENIUM

1. Loops **for** and **for each, while**
2. Conditions **if** and **switch**
3. Exception Handling
4. Collection (List, Set, HashMap)
5. String Methods
6. OOPs concepts,Encapsulation, Polymorphism, Inheritance, Abstract class

WHAT IS SELENIUM?

Selenium is a **free** and **open source** web application automation tool. Also we can call it as **Functional Testing** web application automation tool.

1. Free: We can use Selenium for commercial purpose without purchasing any license.

Download Selenium

- **URL**: http://docs.seleniumhq.org/download/
- **Heading**: Selenium Stand Alone Server
- **Link**: Download Version 2.50.1
- **File:** selenium-server-standalone-2.50.1.jar

2. Open Source: We can view, download and customize the source code of selenium itself. We can see the source code in following website: https://github.com/SeleniumHQ/selenium

3. Web Application Automation Tool: Selenium is software which is used to test the web application automatically but we can't automate other type of applications such as Standalone (Desktop) and Client Server applications.

FLAVORS OF SELENIUM

1. Selenium core
2. Selenium RC(Selenium 1)
3. Selenium WebDriver (Selenium 2)
4. Selenium IDE
   a. Selendroid - Only for Android
   b. Appium – both Android and Apple

IQ1. What is the latest version of selenium is Selenium? **Selenium Webdriver 2.50.1**

IQ2. What are the languages supported by Selenium**? Java, C# (.net), Ruby, Python, Javascript, Pearl, PHP, Haskell, Objective C, R, Dart, and Tcl**

IQ3. What are the OS supported by selenium? **All OS's like Windows, Mac, Linux etc**

IQ4. Which OS is not supported by selenium? **Unix**

IQ5. Can we do Performance Testing using Selenium? **No. But we can integrate selenium with Jmeter.**

IQ6. What type of test cases we automate? **Regression Test Cases**

IQ7. Do we Automate Integration testing? **Yes. Different types of test cases which is part of regression.**

IQ8. Do we automate Negative test cases? **Yes.**

IQ9. Which test cases are automated first? **Smoke test cases. (Sanity, Dry Run(Automation), Build Verification Testing(BVT), Skim(UAT))**

IQ10. Is 100% Automation is possible? **No. Why? Because, we don't have technology to automate the features or it may be very costly or it may require manual intervention.**

Examples

- 3D games
- Verification of audio, video clips.
- Capturing the attendance using access cards and biometrics scanners, Entering product details using barcode scanner, Payment through credit card swiping. OTP.
- Captcha (Completely Automated Public Turing Test to tell computers and humans Apart)

REQUIRED SOFTWARE'S

1. JDK (Java Development Kit)
2. Eclipse, Browser (Firefox)
3. Selenium Jar file
4. Any a web application

STEPS TO CONFIGURE SELENIUM

1. Go to required location example **D:** & create a folder with the name **BCSM6.**

2. In Eclipse go to **File > Switch Workspace > Other**

3. Browse & select newly created folder and click on **OK**. It will restart the Eclipse.

4. Go to **File > New > Project** (Create a Java Project). Specify the name as **Automation** and click **Finish** and **Yes**.

5. Right click on Java Project (i.e **Automation**) and select **New**->**Folder**, give name as **jar file** and click **Finish**.

6. Copy the **selenium jar file**, right click on **jar file** folder and select **Paste**.

7. Expand **jar file** folder, right click on copied Selenium jar file, go to **Build Path** and select **Add to Build Path.**

8. **Right click on src, go to New > Package and give name as testaddda (Everything should be written in** small case and click Finish)

9. **Right click on testaddda and go to New >Class. Give class name as Demo. Select Public Static void** Main.

Write code as shown below and execute.

```
package testaddda
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo
  {
    public static void main(String[] args)
    {
     FirefoxDriver f=new FirefoxDriver();
    }
  }
```

Note:

 If browser is not opening it could be because of following reasons.

- **Incompatible version:** Please ensure that both selenium **jar file** and Firefox browser are up to date.

- **Browser installed in different location.**

    Normally, Firefox will be installed in C:/Program files.

    If it is installed in different location then we get following error.

    Ex: Error will be, **Cannot find Firefox binary in PATH**. To solve this issue specify the path of Firefox browser in environment variable path.

Script: Demo1

```java
package testaddda;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Point;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo1
{
  public static void main(String[] args) throws InterruptedException
   {
      //Open the browser
      FirefoxDriver driver=new FirefoxDriver();
      driver.get("file://D:/Demo1.html");
      Thread.sleep(2000);

      //Resize the browser
      Dimension d=new Dimension(200, 200);
      f.manage().window().setSize(d);
      Thread.sleep(2000);

      //Move the browser
      Point p=new Point(300, 200);
      f.manage().window().setPosition(p);
      Thread.sleep(2000);

      //Maximize the browser
      f.manage().window().maximize();
      }
   }
```

DAY 2

IQ11. How do you close the browser without using **close**() method?

Ans: Using **quit()** method

IQ12. How do you open the page without using **get()** method?

Ans: Using **navigate().to(url)**

IQ13. How do you click on back button?

Ans: Using **navigate().back()**

IQ14. How do you refresh the page?

Ans: Using **navigate().refresh()**

IQ15. What is the difference between **get()** and **navigate()** method?

Ans: Using **get()** method we can only open the web page, where as using **navigate()** method   we can open the page, click back and forward and we can refresh the web page.

```java
package testaddda;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo2
 {
   public static void main(String[] args) throws InterruptedException
        {
                FirefoxDriver driver=new FirefoxDriver();
                Thread.sleep(2000);
                driver.get("http://www.google.com");
                Thread.sleep(2000);
                driver.navigate().to("http://www.gmail.com");
                Thread.sleep(2000);
                driver.navigate().back();
                Thread.sleep(2000);
                driver.navigate().forward();
                Thread.sleep(2000);
                driver.navigate().refresh();
                Thread.sleep(2000);
                driver.quit();
        }
    }
```

IQ16. Write a script to open google.com and verify that title is Google and also verify that it is redirected to google.co.in

```java
package qspider;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo1
{
  public static void main(String[] args)
  {
      FirefoxDriver driver=new FirefoxDriver();
      driver.get("http://www.google.com");
      String title=driver.getTitle();
      if(title.equals("Google"))
      {
          System.out.println("Pass:Title is Google");
      }
      else
      {
       System.out.println("Fail:Title is not Google: actual title is"+title);
      }
      String url=driver.getCurrentUrl();
      if(url.contains("google.co.in"))
      {
        System.out.println("Pass: url has co.in");
      }
      else
      {
        System.out.println("Fail:url dont have co.in"+url);
      }
    }
  }
```

17. How selenium performs action on the browser?

Ans: By calling the **native methods** of the browser.

Q18. Which protocol is used by Selenium to interact/communicate with the browser?

Ans: **JSON Wire Protocol (Java Script Object Notation)**

DAY 3

HANDLING CHROME BROWSER

- Selenium performs the action on the browser by calling its **native method**.
- Firefox browser is open source hence Selenium can directly call its **native methods.** But for other browsers we need **driver executable file** (API).

For Google Chrome we can download the Driver Executable file from following URL:

http://chromedriver.storage.googleapis.com/index.html

File name for windows **is "chromedriver_win32.zip"**. After downloading the zip file extract it and copy paste the file to the required location. Ex: C:/

Write the code as shown below in main method and execute

```java
package qspider;
import org.openqa.selenium.chrome.ChromeDriver;
public class Demo6
{
 public static void main(String[] args) throws InterruptedException
 {
System.setProperty("webdriver.chrome.driver","D:/selenium_software/chromedriver.exe");
        ChromeDriver driver=new ChromeDriver();
        driver.get("file:///D:/Demo1.html");
        driver.close();
    }
}
```

Assignment: Write a script to open google.com in internet explorer (InternetExplorerDriver)

```java
package qspider;
import org.openqa.selenium.ie.InternetExplorerDriver;
public class Demo6a
{
  public static void main(String[] args)
  {
System.setProperty("webdriver.ie.driver","D:/selenium_software/IEDriverServer.exe");
    InternetExplorerDriver driver=new InternetExplorerDriver();
    driver.get("file:///D:/Demo1.html");
    driver.close();
  }
}
```

INHERITANCE AND METHOD OVERRIDING

```
class A
{
  void testA()
  {
    System.out.println("A1");
  }
  void testB()
  {
    System.out.println("B1");
  }
}
```

```
class B extends A
{
  void testB()
  {
    System.out.println("B2");
  }
  void testC()
  {
    System.out.println("C2");
  }
}
```

B b1=new B();

b1.testA(); -→A1

b1.testB();-→B2

b1.testC();-→C2

B b1=new B();

A a1=b1;

A1.testA();-→A1

A1.testB();-→B2

A1.testC();

DAY 4

RUNTIME POLYMORPHISM

We use runtime polymorphism in selenium so that it can execute the script on any browser. In order to do this we create the object of required browser and store it in the reference variable of parent interface called '**WebDriver".**

Script to open Chrome and IE browsers using user input:

```java
package qspider;
import java.util.Scanner;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;

public class Demo7
 {
   public static void main(String[] args)
   {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter browser?GC/FF/IE");
    String browser=sc.next();
    WebDriver driver;

    if(browser.equals("GC"))
     {
    System.setProperty("webdriver.chrome.driver","D:/selenium_software
      /chromedriver.exe");
      driver=new ChromeDriver();
     }
    else if(browser.equals("IE"))
     {
      System.setProperty("webdriver.ie.driver","D:/selenium_software
      /IEDriverServer.exe");
      driver=new InternetExplorerDriver();
     }
    else
     {
      driver=new FirefoxDriver();
     }

    driver.get("http://localhost/login.do");
    System.out.println(driver.getTitle());
    System.out.println(driver.getCurrentUrl());
    driver.close();
   }
 }
```
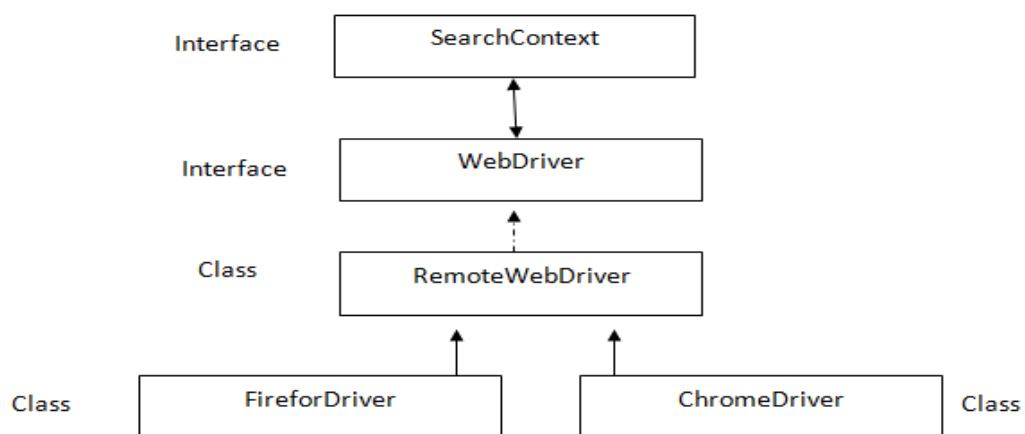
ARCHITECTURE OF SELENIUM WEBDRIVER:
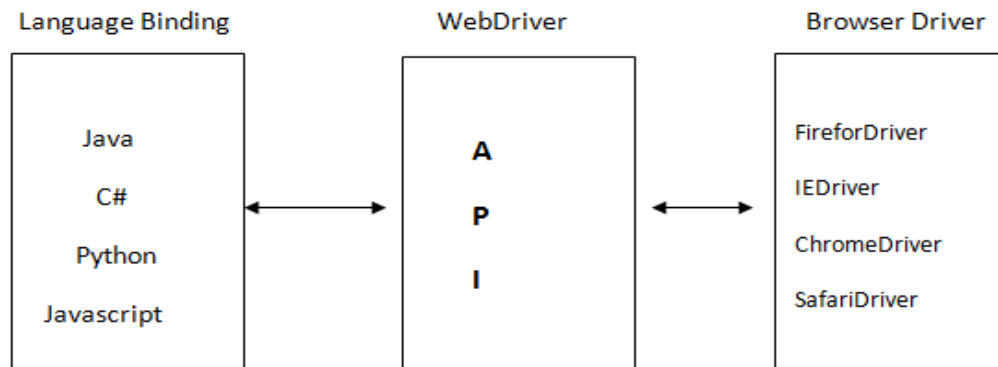
Selenium supports multiple coding languages. Each language has its own binding which communicates with WebDriver API. The WebDriver API performs the action on the browser by using browser specific drivers (Uses **JSON Wire Protocol**)

ARCHITECTURE OF WEBDRIVER API

SearchContext is super most interfaces which are extended by **WebDriver** interface. Abstract methods of these two interfaces is implemented in **Remote WebDriver** class and overridden in respective browser classes such as FirefoxDriver, ChromeDriver, InternetExplorerDriver, SafariDriver etc.

Q19. Explain the following statement: WebDriver driver=new FirefoxDriver();

    i.   **WebDriver** is a interface
    ii.   **driver** is reference variable
    iii.   = is assignment operator
    iv.   **new** is keyword
    v.   **FirefoxDriver** is constructor
    vi.   **;** is statement delimiter

WHAT IS WEB ELEMENT?

- Anything present on the web page is called as **WebElement**. Such as textbox, button, link etc.

- Elements are created using HTML. It stands for 'Hyper Text Markup Language'.

- In HTML pre-defined key words within angle brace. It is called as '**tag'**. We can use notepad to create the web page. After writing the code we should save the file with extension **.html**

  Example: Open the 'Notepad' and write following code:

```
<html>

    <body>

        <a href=http://localhose id="al" name ="nl"
class="cl">actitime</a>

    </body>

</html>
```

- Go to File and select Save. Navigate to required location. Specify the file name ex: Demo.html and click Save.

- Double click on newly created file which opens the file on default browser.

Selenium code to open the above web page:

```
WebDriver driver=new FirefoxDriver();
driver.get("file:///D:/Demo.html");
```

THE HTML ELEMENT CONTAINS FOLLOWING 3 COMPONENTS

1. Tag
2. Attribute
3. Text

1. **Tag:** Anything present after the less than (<) symbol.
     Ex: html, body, a
2. **Attribute**: Anything present after the tag till the greater than (>) symbol.
     Ex: href=http://localhose id="al" name ="nl" class="cl">
3. **Text**: Anything present after the greater than (>) symbol till the end of the tag.
     Ex: actitime

HOW TO SEE HTML ELEMENT?

- To see the source code of the element which is present on the web page, we right click on the element and select 'Inspect Element'.
- In Mozilla Firefox we also use an add-on called 'Fire Bug'.

STEPS TO INSTALL FIREBUG

1. **With internet connection**:
   o Go to tools>add-on, search for 'Firebug'.
   o Click install button on firebug.
2. **Without internet connection:**
   o Go to tools>add-on, click on 'Tools for all add-ons' button.
   o Select 'Install add-on from file' option.
   o Select 'Firebug' software. File name is 'firebug-2.0.12'
   o Click open and install

If right clicking (Context Click) is disabled then press F12. It will open Firebug window. Click on inspect button and then click on required element.

DAY 5

WHAT IS LOCATORS?

Locators are used to identify the element.

In Selenium before performing any action (click, type etc) we should find the element using **locators**. In Selenium there are 8 types are **locators**. All of them are **static methods** in **By** class (it is an abstract class).

- o All the methods takes string as argument and it returns an object of type **By.**
- o The **By** object is used as input argument for **findElement()** method.
- o Return type of **findElement()** method is **WebElement** (it is an Interface).

THE LIST OF SELENIUM LOCATORS:

1. By.tagName
2. By.id
3. By.name
4. By.className
5. By.linkText
6. By.partialLinkText
7. By.cssSelector
8. By.xpath

Code: Selenium code to click on a link using 'tagName':

```java
package testaddda;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo7
  {
      public static void main(String[] args)
      {
              WebDriver driver=new FirefoxDriver();
              driver.get("file:///D:///Demo.html");
              By b=By.tagName("a");
              WebElement e = driver.findElement(b);
              e.click();
      }
  }
```

Optimized code:

- using **tagName**

```
driver.findElement(By.tagName("a")).click();
```
In the browser find the element by tag name 'a' and click on it.

- using **id**

```
driver.findElement(By.id("a1")).click();
```

- using **name**

```
driver.findElement(By.name("n1")).click();
```

- using **className**

```
driver.findElement(By.className("c1")).click();
```

- using **linkText**

```
driver.findElement(By.linkText("actitime")).click();
```
Note: the locator 'linkText' can be used only if the element is a link
(tag of the element should be a).

- using **partialLinkText**

```
driver.findElement(By.partialLinkText("acti")).click();
```

Note: this locator is used to handle dynamic links.

```
driver.findElement(By.partialLinkText("Inbox")).click();
```

```
driver.findElement(By.partialLinkText("BuildNo")).click();
```

Important Note:

- ✓ If specified locator is matching with more than one element then **findElement**() method **returns the address of first matching element.**
- ✓ If the specified locator is not matching with any of the element then **findElement**() method will throw '**NoSuchElementException**'.

Script: Write a script to login to actitime application.

```java
package testaddda;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo8
 {
        public static void main(String[] args)
        {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://localhost/login.do");
        driver.findElement(By.id("username")).sendKeys("admin");
        driver.findElement(By.name("pwd")).sendKeys("manager");
        driver.findElement(By.id("loginButton")).click();
        }
    }
```

DAY 6

- using **cssSelector**

```
<html>
    <body>
            UN<input type="text">
            PW<input type="password">
    </body>
</html>
```

In the above sample page to identify the password field we can't use id, name, className, linkText, partialLinkText because they are not present. We can use 'tagName' but it has duplicate user field. In this situation we can use **cssSelector**. CSS stands for Cascading Style Sheets.

cssSelector has following syntax:

Tag[AttributeName='AttributeValue']

Ex: input[type='password']

To check whether CSS expression is correct or not, we can use **FirePath' in Mozilla Firefox**.

1. To install it go to **Tools>Add-Ons,** search for **FirePath**.
2. Click **Install** button of FirePath and restart the browser.
3. Open the required web page press **F12**(Firebug) click on **FirePath** tab.
4. Select **CSS**, type the expression and press **Enter**.
5. It will highlight matching elements.

Checking CSS in Google Chrome:

1. Press **F12** in chrome browser, then press **Ctr+F.**
2. Type the expression; it will highlight the source code of the matching element. If we place the mouse pointer on the source code, it will highlight the element on the page.

Ex: findElement by using cssSelector

```
package testaddda;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
```

```java
public class Demo6
  {
  public static void main(String[] args)
  {
   WebDriver driver=new FirefoxDriver();
   driver.get("file:///D:/Demo2.html");

driver.findElement(By.cssSelector("input[type='text']")).sendKeys("admin");

driver.findElement(By.cssSelector("input[type='password']")).sendKeys("manager");
  }
 }
```

- using **xpath**

xpath is the path of the element in HTML tree.

```html
<html>
    <body>
            FN<input type="text">
            LN<input type="text">
    </body>
</html>
```
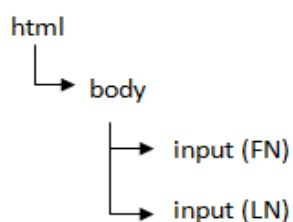
In the above sample web page we can't use cssSelector because it is same as first name field. In this case we can use 'xpath'.

We write the xpath expression using /(forward slash). The first forward slash represents beginning of the tree (root).

After every forward slash we should specify tag of immediate child element. We can also use index which starts from 1.

HTML Tree:

Xpath for First Name: /html/body/input[1]

Xpath for Last Name: /html/body/input[2]

Checking xpath using Firefox:

- o   Open the required page in Mozilla Firefox
- o   Press F12. Select 'xpath'.
- o   Type 'xpath' expression /html/body/input [1]. It will highlight matching element.

```java
package testaddda;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Day6Demo2
 {
   public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D/Demo2.html");
        driver.findElement(By.xpath("/html/body/input[1]")).sendKeys("a");
        driver.findElement(By.xpath("/html/body/input")).sendKeys("c");
        driver.findElement(By.xpath("/html/body/input[2]")).sendKeys("b");
    }
 }
```

TYPES OF XPATH

1. Absolute xpath
2. Relative xpath
3. xpath by Attribute
4. xpath by text() function
5. xpath by contains() function
6. Traversing in xpath
7. Independent-Dependent
8. xpath by group index

Let's consider the following html tree to deriver xpath expression
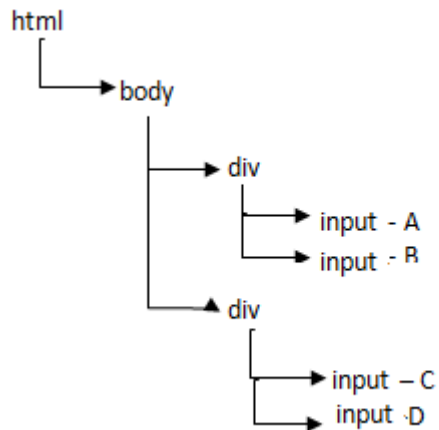
1. Absolute xpath

Specifying complete path of the element form the root till the element is called as absolute xpath. Ex:

Xpath	Matching Element
/html/body/div[1]/input[1]	A
/html/body/div[1]/input[2]	B
/html/body/div[2]/input[1]	C
/html/body/div[2]/input[2]	D
/html/body/div[1]/input	AB
/html/body/div[2]/input	CD
/html/body/div/input[1]	AC
/html/body/div/input[2]	BD
/html/body/div/input	ABCD
/html/body/div[1]/input[1]| /html/body/div[2]/input[2]	AD
/html/body/div[1]/input[2]| /html/body/div[2]/input[1]	BC
/html/body/div[1]/input[1]| /html/body/div[1]/input[2]| /html/body/div[2]/input[1]	ABC

DAY 7

Q20. WRITE A SCRIPT TO TAKE SCREENSHOT OF THE APPLICATION?

```java
package qspider;

import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.events.EventFiringWebDriver;
public class Copyscreenshot
{
  public static void main(String[] args) throws IOException
  {
    WebDriver driver=new FirefoxDriver();
    driver.get("http://localhost/login.do");
    EventFiringWebDriver e=new EventFiringWebDriver(driver);
    File srcfile=e.getScreenshotAs(OutputType.FILE);
    File destfile=new File("D:/actitime1.png");
    FileUtils.copyFile(srcfile,destfile);
  }
}
```

Limitations:

- We can only take screenshot in PNG(Portable Network Graphics) format.
- We can't take screenshot of the popup.
- We can't take screenshot of multiple browser.
- We can't take screenshot of required area on the Web page.

CONTI…. TYPES OF XPATH

2. Relative xpath

Absolute 'xpath' is very lengthy. In order to reduce the length of expression we can use relative 'xpath'.

In relative 'xpath' we use double forward slash(//) which represents any child, also called as 'descendent'.

Xpath	Matching element
//div[1]/input[1]	A
//div[1]/input[2]	B
//div[2]/input[1]	C
//div[2]/input[2]	D
//div[1]/input	AB
//div[2]/input	CD
//input[1]	AC
//input[2]	BD
//input	ABCD
//div[1]/input[1]| //div[2]/input[2]	AD
//div[1]/input[2]| //div[2]/input[1]	BC
//div[1]/input[1]| //div[1]/input[2]| //div[2]/input[1]	ABC

Q21. What is the different between single forward slash and double forward slash?

Ans: Single forward slash represent immediate child where as double forward slash represents any child (descendent).

Q22. What is the difference between '//a' and '//table//a'?

Ans: '//a' matches with all the links present which are in the entire page.

Whereas '//table//a' matches with all the links which are present inside the table.

Q23. Derive an 'xpath' which matches with all the images present on the web page?

Ans: '//img'

Q24. Write an 'xpath' which matches with all the links and all the images present on the web page?
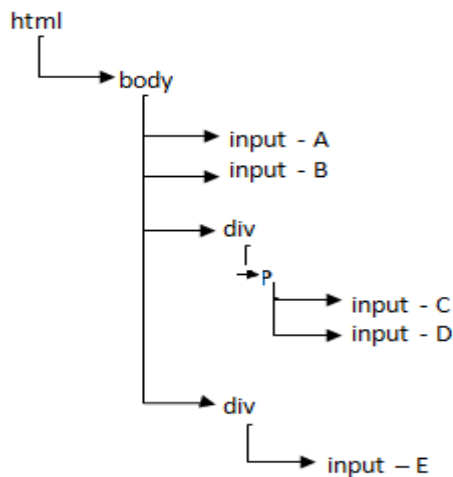
Ans: '//a|//img'

Important Note: **xpath** matches with hidden elements also.

Q25. What is the difference between //input and //div//input?

//Input matches with all the inputs present in the entire web page.

//div//input matches with all the inputs present inside the 'div'.

//div/input matches with all the immediate child input of 'div'

```
html
  └──► body
          ──► input - A
          ──► input - B
          ──► div
                └──► P
                        ──► input - C
                        ──► input - D
          ──► div
                └──► input – E
```

//div//input   ->   CDE

//div/input -> E

//input   -> ABCDE

//p/input -> CD

3.   xpath by Attribute

To identify the element uniquely, we can include attribute in the 'xpath' expression using below syntax

//tag[@AttributeName='AttributeValue']

//input[@id='username']

Note: We can use more than one attribute in an 'xpath' expression:

//input[@id='username'] [@type ='text']

//input[@id='username'] AND [@type ='text']

//input[@id='username'] OR [@type ='text']

4.   xpath by text() function

If Attribute is matching with more than one element or if the attribute is not present then we can identify the element using its text. It has following syntax:

 //tag[text()='textValue']

Examples:

//div[text()='Login ']

//div[text()='Users']

//td[text()='Java']

//div[@class='label'][text()='Users']

Important Note: If there is a 'Non Breakable Space' in attribute value or in text value then 'xpath' will not identify the element. Ex: HTML code present in source file.

5.   xpath by contains() function

When we inspect the element, we cannot make out whether the space is given using the space bar or by using the key work or using the ' '.

<html>
    <body>
      <botton type=" submit "> Sign in </button>
    </body>
</html>
HTML code displayed in firebug:

<button type=" submit "> Sign in </button>

Even though we write the 'xpath' by copy pasting the value from the source code displayed in the firebug, it will not match with any element.

Ex: //button[@type=' submit ']→ No Match

   //button[text()=' Sign in ']→No Match

Contains: We can use contains function when there is a 'Non Breakable Space' to identify the element. It has following syntax

1: //tag [contains(@AttributeName,'AttributeValue')]

2: //tag [contains(text(), 'textValue')].

Example:

 //button[contains(@type, 'submit')]

 //button[contains(@text, 'Sign in')]

 //input[contains(@value, 'Create Type of Work')]

Q26. How do you handle if there is 'Non Breakable Space' between the strings?

Ex: <button> Sign   in </button>

Xpath: //button [contains(text(),'Sign')][contains(text(),'in')]

Note: We can use contains function to handle dynamic element also. Ex: (build 27261). In this example build number 27261 will be changing. //span[contains(text(),'build')]

Q27. When do we use contains function?

Ans:We use 'contains' function if there is a 'Non Breakable Space' in attribute value or text value.

We use 'contains' function when the element is dynamic (some part of its value keeps changing)

DAY 9

6. Traversing in xpath

We can write 'xpath' expression which navigates from one element to another element which is called as traversing. In 'xpath' there are 2 types of traversing.

 i. Forward Traversing

ii. Backward Traversing.
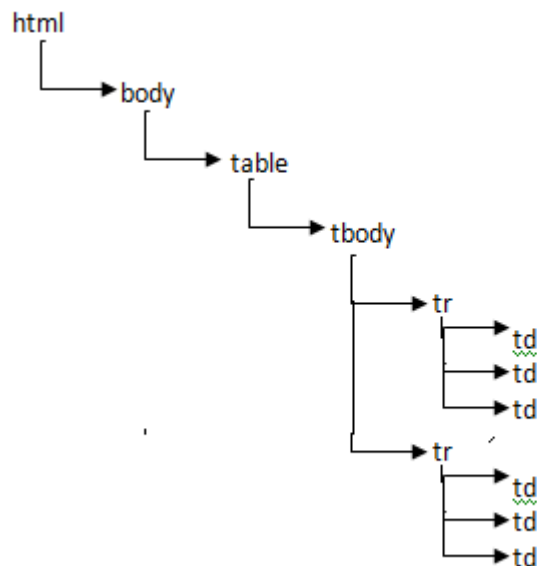
i.   Forward Traversing

Navigating from parent element to its any of the child element is called as forward traversing. Ex 1: Navigating from table node to java cell (//table/tbody/tr[1]/td[2]). Ex 2: Navigating from table to Unix cell (//table/tbody/tr[2]/td[2]).

ii.   Backward Traversing

 Navigating from child element to any of its parent element is called as backward traversing. Ex 1: Navigating from java cell to table node (//td[text()='Java']/../../..) Ex 2: Navigating from Unix cell to the table (//td[text()='Unix']/../../..)

7.   Independent Dependent xpath

If the element is completely dynamic or if the element is duplicate, then we can identify that element using some other element by applying a technique called independent, dependent xpath.

Ex:

1	Java	200
2	Unix	300

'xpath' to identify cost of Java:

//td[text()='300'] - In the first 'xpath' expression we are identifying the cost directly. If the cost of the Java book changes then this 'xpath' will not identify the element.

//td[text()='Java']/../td[3] - In the 2nd 'xpath' expression we are identifying the cost using the name of the subject. In this example Java is called as independent element and cost is called as dependent element. This will identify the cost even if it completely changes.

Steps to derive independent and dependent 'xpath'

1. Inspect the independent element and note the source code.
2. Place the mouse pointer on source code of independent element and move the mouse pointer on upward direction step by step till it highlights both independent and dependent element. It will be the common parent. Add it to HTML tree.
3. Navigates from common parent to dependent element using arrow key and add it to HTML tree as shown below.

common parent

tr

<td>Java</td>
Independent

<td>[3]
dependent
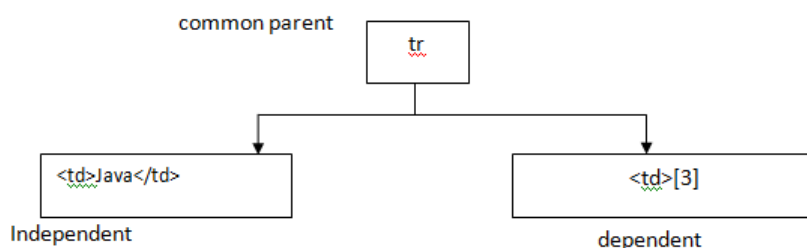
Derive the 'xpath' which navigates from independent element to common parent and then to dependent element. Ex: //td[text()='Java']/../td[3]

Q28: Deriver an 'xpath' which identifies download link of ruby which is present in Selenium download page?
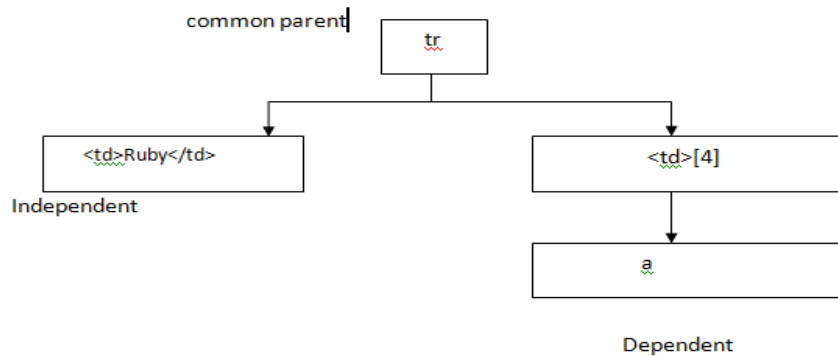
xpath: //td[text()='Ruby']/../td[4]/a

The above 'xpath' expression identifies the download link only if it is present in 4th column. If the column keeps changing we can use the below 'xpath'. Ex: //td[text()='Ruby']/..//a[text()='Download']

Q29: Deriver an 'xpath' to identify 'set by default' link of 'Billable'?
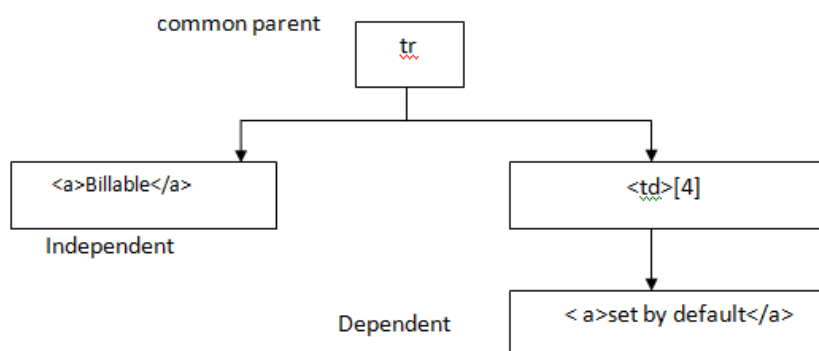
Assignment:

Q: Derive an 'xpath' to identify 'Phone Number' of Mumbai present in the IRCTC website?

Ans: //lable[text()='Mumbai']/../label[2]

Q: Derive an 'xpath' to identify 'Price' of 'MI 4I/Grey 16GB' present in Flipkart.com

Ans: //a[@title='Mi 4i (Grey , 16 GB)']/../../div[4]/div/div/span

8. xpath by Group Index

(//input)[1] – A

(//input)[2] – B

(//input)[3] - C

Ex: 'xpath' to identify help icon present in the 'ActiTime' application.

(//div[@class='popup_menu_arrow'])[3]

Q30: What is the difference between //a, //a[1] and (//a)[1]?

//a – Matches with all the links present in the entire web page

//a[1]- Matches with all the first link

(//a)[1] – Matches with only the first link

//input[@type='checkbox'] – Matches with all the checkbox in the entire web page

(//input[@type='checkbox'])[1]- Matches with the first checkbox

(//input[@type='checkbox'])[5] - Matches with the fifth checkbox

(//input[@type='checkbox'])[last()] - Matches with the last checkbox

Q31: Derive the 'xpath' which matches with first and last image?

(//img)[1]|(//img)[last()]

DAY 10

Important Note: Out of 8 locators we use following 4 important locators.

1. ID
2. Name
3. linkText
4. Xpath

Sometimes *xpath* written using one browser may not work in another browser. In such cases we can convert **xpath** into **cssSelector** as shown below.

//input[@name='UN'] → input[name='UN']

//input[@id='UN'] → input #UN or #UN

//class[@id='UN'] → input.UN

//a → a

//table/tbody → table>tbody

//table//td → table td

//*[@id='UN'] → #UN

//*[@class='UN'] → .UN

Note: * indicates any tag

In 'cssSelector' we can't identify the element using its text and cssSelector do not support backward traversing. We can't use independent dependent concept.

//table/.. → Not possible

//a[text()='abc'] → Not possible

Q32: How do you open Firefox browser with 'Add-ons'?

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Day10
{
  public static void main(String[] args)
   {
```

```
    System.setProperty("webdriver.firefox.profile", "default");
    WebDriver driver=new FirefoxDriver();
    driver.get("www.google.com");
    }
}
```

When Selenium opens the Firefox browser it opens with factory settings i.e. No add-ons, No History, No Auto Complete, No cookies. This is done to run the script faster.

SYNCHRONIZATION

Process of matching Selenium speed with application is called as Synchronization. On real time applications when Selenium try to find the element it may through 'NoSuchElementException' even though specified locator is correct. To handle this we can use 'Sleep' method of thread class as shown below.

```
package testaddda;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class DemoLogout
{
    public static void main(String[] args)
    {
    WebDriver driver=new FirefoxDriver();
    driver.get("http://localhost/login.do");
    driver.findElement(By.id("username")).sendKeys("admin");
    driver.findElement(By.name("pwd")).sendKeys("manager");
    driver.findElement(By.id("loginButton")).click();
    try
    {
        Thread.sleep(20000);
    }
    catch (InterruptedException e)
    {
        e.printStackTrace();
    }
    driver.findElement(By.id("logoutLink")).click();
    driver.close();
    }
}
```

DAY 11

USING IMPLICITLYWAIT

If we use **sleep()** method we should specify it in all the locations where application is slow. This will increase the time taken to write script, it consumes lot of space and increases maintenance of the script and it always waits for specified duration. Ex: if the duration is 20 sec, it will always waits for 20 sec even though element is displayed in the 5 sec.

To overcome all these limitations, we should use the synchronization option given by Selenium called **implicitlyWait** as shown below.

```
WebDriver driver=new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(20, TimeUnit, SECONDS);
```

The duration specified in **implicitlyWait** statement is used only by **findElement()** and **findElements().** But do not by any other methods.

It takes two arguments first one is a **duration** and the 2nd argument is the **TimeUnit** such as

- DAYS
- HOURS
- MINUTES
- SECONDS
- MILLISECONDS
- MICROSECONDS
- NANOSECONDS

If we use **implicitlyWait** then if the element is not located the **findElement()** method will keep searching for the element after every 500 MILLISECONDS. This is duration is called as "**Poling Period**". This is specified in a class called **FluentWait**.

If the element is not located even after the duration then we get **NoSuchElementException**.

Q33. Can we specify **ImplicitlyWait** statement multiple times in the Selenium script? **Yes.**

Q34. Is it necessary to write **ImplicitlyWait** statement multiple times? **No.**

USING EXPLICIT WAIT

WebDriverWaits itself is called Explicit Wait.

Wherever we can't use **implicitlyWait** (Other than findElement) we should use **Explicit Wait**. Since we specify the waiting condition explicitly it is called as **Explicit Wait**.

When the control comes to **wait.until** statement it will keep checking the condition after every 500 Mili Seconds. If the condition is satisfied it will go to next statement. If the condition is not satisfied even after the duration we get **TimeoutException**.

All the conditions are present in the class called **ExpectedConditions**. These conditions are also called as **Predicate**.

Q35: Print the title of the home page after login to the application.

```java
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class Demo
{
        public static void main(String[] args)
        {
        WebDriver driver=new  FirefoxDriver();
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
        driver.get("http://demo.actitime.com");
        driver.findElement(By.id("username")).sendKeys("admin");
        driver.findElement(By.name("pwd")).sendKeys("manager");
        driver.findElement(By.id("loginButton")).click();
        WebDriverWait wait=new WebDriverWait(driver,20);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("logoutLink")));
        String title=driver.getTitle();
        System.out.println(title);
        }
    }
```

Q36. What are the differences between 'Implicit' and 'Explicit' wait?

Implicit	Explicit
We do not specify the condition	We should specify the condition
We can handle 'Findelement' and 'Findelements'	We can handle any method
After the duration we get 'NoSuchElement' exception	After the duration we get 'Timeout' exception
Duration can be DAYS, HOURS, MINUTES, SECONDS etc.	Duration will be only Seconds

Script: Handling findElement using explicitWait

```java
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class Day11Demo3
{
  public static void main(String[] args)
   {
     WebDriver driver=new FirefoxDriver();
     driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
     driver.get("http://demo.actitime.com");
     driver.findElement(By.id("username")).sendKeys("admin");
     driver.findElement(By.name("pwd")).sendKeys("manager");
     driver.findElement(By.id("loginButton")).click();
     WebDriverWait wait=new WebDriverWait(driver,20);
     wait.until(ExpectedConditions.titleIs("actiTIME - Enter Time-Track"));
     driver.findElement(By.id("logoutLink")).click();
   }
 }
```

Q37. Write a script to login and logout from the application without specifying the waiting period or without using any of the Synchronization methods.

```java
package testaddda;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.NoSuchElementException;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo4
 {
   public static void main(String[] args)
    {
      WebDriver driver=new FirefoxDriver();
       driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
       driver.get("http://demo.actitime.com");
       driver.findElement(By.id("username")).sendKeys("admin");
       driver.findElement(By.name("pwd")).sendKeys("manager");
       driver.findElement(By.id("loginButton")).click();
      while(true)
        {
          try
            {
              driver.findElement(By.id("logoutLink")).click();
              break;
            }

          catch(NoSuchElementException e)
            {
              System.out.println("Bye");
            }
        }
    }
  }
```

DAY 12

Q38. Write a code to print the value present in the text box?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign1
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo2.html");
        WebElement tv=driver.findElement(By.id("t3"));
        String printText=tv.getAttribute("value");
        System.out.println(printText);
    }
}
```

Q39. Write a code to change the value present in the text box?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign2
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo2.html");
        WebElement tv=driver.findElement(By.id("t3"));
        tv.clear();
        tv.sendKeys("Webdriver");
    }
}
```

Q40. Write a script to remove text present in the text box without using clear method?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign3
{
    public static void main(String[] args)
    {
      WebDriver driver=new FirefoxDriver();
      driver.get("file:///D:/Demo2.html");
      WebElement tv=driver.findElement(By.id("t4"));
      tv.sendKeys(Keys.CONTROL+"a");
      tv.sendKeys(Keys.DELETE);
    }
}
```

Q41. Write a script to clear the text present in the text box by pressing back space?

```java
package qspider;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Assign4
{
    public static void main(String[] args)
    {
      WebDriver driver=new FirefoxDriver();
     driver.get("file:///D:/Demo2.html");
        WebElement tv=driver.findElement(By.id("t4"));
        String st=tv.getAttribute("value");
        int count=st.length();
        for(int i=1;i<=count;i++)
        tv.sendKeys(Keys.BACK_SPACE);
    }
}
```

Q42. Write a script to copy & paste the value present in one text box into another text box?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign5
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo2.html");
        WebElement v1=driver.findElement(By.id("t1"));
        v1.sendKeys(Keys.CONTROL+"a");
        v1.sendKeys(Keys.CONTROL+"c");
        WebElement v2=driver.findElement(By.id("t4"));
        v2.clear();
        v2.sendKeys(Keys.CONTROL+"v");
    }
}
```

Q43. Write a script to print text of the link?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign6
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://www.facebook.com");
WebElement
v1=driver.findElement(By.xpath("//input[@id='persist_box']/../../../../td[2]/a"
));
        String text = v1.getText();
        System.out.println(text);
    }
}
```

Q44. Write a script to print x and y coordinates of an element?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign7
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://www.facebook.com");
        WebElement em=driver.findElement(By.id("email"));
        Point p=em.getLocation();
        System.out.println("X coordinate (in pixels): "+p.getX());
        System.out.println("Y coordinate (in pixels): "+p.getY());
    }
}
```

Q45. Write a script to verify that email text box and Next button present in Gmail login page are aligned horizontally? (x value should be same)

```java
package qspider;

import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Assign9
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://www.gmail.com");
        WebElement em=driver.findElement(By.id("Email"));
        Point p1=em.getLocation();
```

```java
    int x1=p1.getX();

    System.out.println("X value of email field: "+x1);

    WebElement nxt=driver.findElement(By.id("next"));

    Point p2=nxt.getLocation();

    int x2=p2.getX();

    System.out.println("X value of next button: "+x2);

    if(x2-x1<=0)
     {
      System.out.println("Email textbox and next button aligned  horizontally");
     }
    else
     {
      System.out.println("Not alligned Horizontally");
     }
  }
}
```

Q46. Write a script to print width and height of a text box?

```java
        package qspider;


        import org.openqa.selenium.By;

        import org.openqa.selenium.Dimension;

        import org.openqa.selenium.WebDriver;

        import org.openqa.selenium.WebElement;

        import org.openqa.selenium.firefox.FirefoxDriver;


        public class Assign9
        {
          public static void main(String[] args)
          {
                WebDriver driver=new FirefoxDriver();
                driver.get("http://www.gmail.com");
                WebElement em=driver.findElement(By.id("Email"));
                Dimension s = em.getSize();
                System.out.println("Height of the textbox: "+s.getHeight());
                System.out.println("Width of the textbox: "+s.getWidth());
          }
        }
```

Q47. Write a script to verify that width of email textbox and next button is same which are present in Gmail login page?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign9
  {
   public static void main(String[] args)
    {
     WebDriver driver=new FirefoxDriver();
      driver.get("http://www.gmail.com");
    WebElement em=driver.findElement(By.id("Email"));
    Dimension s1 = em.getSize();
    int w1 = s1.getWidth();
    WebElement nxt=driver.findElement(By.id("next"));
    Dimension s2=nxt.getSize();
    int w2=s2.getWidth();

    System.out.println("Width of Email textbox: "+w1);
    System.out.println("Width of next button: "+w2);
    if(w1==w2)
    {
    System.out.println("Width of email textbox and next button is same");
    }
    else
    {
    System.out.println("Width of email textbox & next button is not same");
    }
   }
  }
```

Q48. Write a script to verify that height of email password and login button which are present in FB login page are same?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign8
{
	public static void main(String[] args)
	{
	  WebDriver driver=new FirefoxDriver();
	  driver.get("http://www.facebook.com");
	  WebElement em=driver.findElement(By.id("email"));
	  Dimension s1=em.getSize();
	  int h1=s1.getHeight();
	  System.out.println("Height of Email textbox: "+h1);
	  WebElement pwd=driver.findElement(By.id("pass"));
	  Dimension s2=pwd.getSize();
	  int h2=s2.getHeight();
	  System.out.println("Height of passowd textbox: "+h2);

	   WebElement button=driver.findElement(By.id("u_0_v"));
	   Dimension s3=button.getSize();
	   int h3=s3.getHeight();
	   System.out.println("Height of login button: "+h3);
	  if(h1-h2==0&&h2-h3==0&&h3-h1==0)
	  {
	   System.out.println("Height of email, password and login button is same");
	  }
	  else
	  {
	   System.out.println("Height of email, password and login button is not the same");
	  }
	}
}
```

Q49. Write a script to verify that email text box present in Facebook login page is empty?

Hint: get the value and check the length of it. Length should be 0.

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign8
 {
   public static void main(String[] args)
    {
       WebDriver driver=new FirefoxDriver();
       driver.get("http://www.facebook.com");
       WebElement em=driver.findElement(By.id("email"));
       String t = em.getAttribute("value");
       if(t.length()==0)
        {
          System.out.println("Textbox present in Facebook login page is empty");
        }
       else
        {
          System.out.println("Textbox present in Facebook login page is not empty");
        }
    }
}
```

Q50. Write a script to verify the status of the check box which is present in FaceBook login page?   Note: IsSelected method is used to verify the checkbox or radio button is selected.

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign
{
        public static void main(String[] args)
        {
           WebDriver driver=new FirefoxDriver();
           driver.get("http://www.facebook.com");
           WebElement chkbox=driver.findElement(By.id("persist_box"));
           WebElement radbtn=driver.findElement(By.id("u_0_e"));
```

```java
        if(chkbox.isSelected())
        {
                System.out.println("Checkbox is selected");
        }
        else
        {
                System.out.println("Checkbox is deselected");
        }
        if(radbtn.isSelected())
        {
                System.out.println("Radio button is selected");
        }
        else
        {
           System.out.println("Radio button is deselected");
        }
    }
}
```

Q51. Write a script to verify whether login button is enabled or not which is present in the FB page?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class assign10
{
        public static void main(String[] args)
        {
          WebDriver driver=new FirefoxDriver();
          driver.get("http://www.facebook.com");
          WebElement log=driver.findElement(By.id("u_0_v"));
          if(log.isEnabled())
           {
                System.out.println("Login button is enabled");
           }
          else
           {
                System.out.println("Login button is disabled");
           }
        }
}
```

Q52. Write a script to verify that logo of actitime is displayed on the login page?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Assign11
{
    public static void main(String[] args)
     {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://localhost/login.do");
        WebElement logo=driver.findElement(By.id("logoContainer"));
         if(logo.isDisplayed())
        {
            System.out.println("Logo is displayed");
             }
         else
            {
          System.out.println("Log is not displayed");
            }
    }
}
```

DAY 13

Q53. How do you execute an exe file in Selenium?

In Selenium there is No option to run exe file. We can use runtime class to execute the exe files.

```java
package testaddda;
import java.io.IOException;
public class Day13Demo1
{
   public static void main(String[] args) throws IOException
       {
         Runtime.getRuntime().exec("C:/Windows/system32/calc.exe");
         }
}
```

Q54. Write a script to delete all the cookies present in the browser?

```java
package testaddda;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo
{
       public static void main(String[] args)
       {
               WebDriver driver=new FirefoxDriver();
               driver.manage().deleteAllCookies();
       }
}
```

Q55. Write a script verify whether page is loaded within 3 seconds?

```java
package qspider;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.TimeoutException;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo1
{
   public static void main(String[] args)
    {
       WebDriver driver=new FirefoxDriver();
       driver.manage().timeouts().pageLoadTimeout(1,TimeUnit.SECONDS);
       try
```

```
        {
          driver.get("http://localhost/login.do");
          System.out.println("Page is loaded within 3sec");
        }
      catch(TimeoutException e)
        {
          System.out.println("Page is not loaded within 3sec");
        }
    }
}
```

Q56. What are the different ways of clicking on a button?

1. click()
2. sendKeys()
3. submit()//this works only if button code is submit
4. javascript
5. AutoIt
6. Robot Class

```
package testaddda;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo
{
    public static void main(String[] args)
     {
    WebDriver driver=new FirefoxDriver();
    driver.get("http://demo.vtiger.com");
    String xp="//button[text()='Sign in']";
    WebElement btn = driver.findElement(By.xpath(xp));
    btn.sendKeys(Keys.ENTER);
     }
}
```

Q57. How do you get the font size of the text box? Or how do you get style property of an element? Ans:

Using **getCssValue()**

```java
package testaddda;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Day13Demo5
{
        public static void main(String[] args)
        {
                WebDriver driver=new FirefoxDriver();
                driver.get("http://demo.vtiger.com");
                WebElement un = driver.findElement(By.id("username"));
                String s = un.getCssValue("font-size");
                System.out.println(s);
                driver.close();
        }
}
```

Q58. Write a script to print background color of a textbox?

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Demo
{
        public static void main(String[] args)
        {
                WebDriver driver=new FirefoxDriver();
                driver.get("http://localhost/login.do");
                WebElement un=driver.findElement(By.id("username"));
                System.out.println(un.getCssValue("color"));
        }
}
```

IMPORTANT METHODS OF WEBELEMENT INTERFACE

i. clear()

ii. click()

iii. getAttribute()

iv. getCssValue()

v. getLocation()

vi. getSize()

vii. getText()

viii. isDisplayed()

ix. isEnabled()

x. isSelected()

xi. sendkeys()

xii. submit()

JAVASCRIPT

If the method of Selenium is not working such as sendkeys(), click() etc. then we can use javascript as alternative option. Ex: if the textbox is disabled we can use sendkeys() method. In such cases we can use javascript.

Steps to Run Java Script Manually

1. Open the required page in Firefox. Press F12 to open 'Firebug'.
2. Click on 'Console' tab.
3. Type the java script in the textbox available at the bottom of 'Firebug' window and press enter.

   |>alert ('Hi') ← (press Enter)

Executing Java Script in Selenium

- To run the Java Script in Selenium we should use **executeScript()** method.
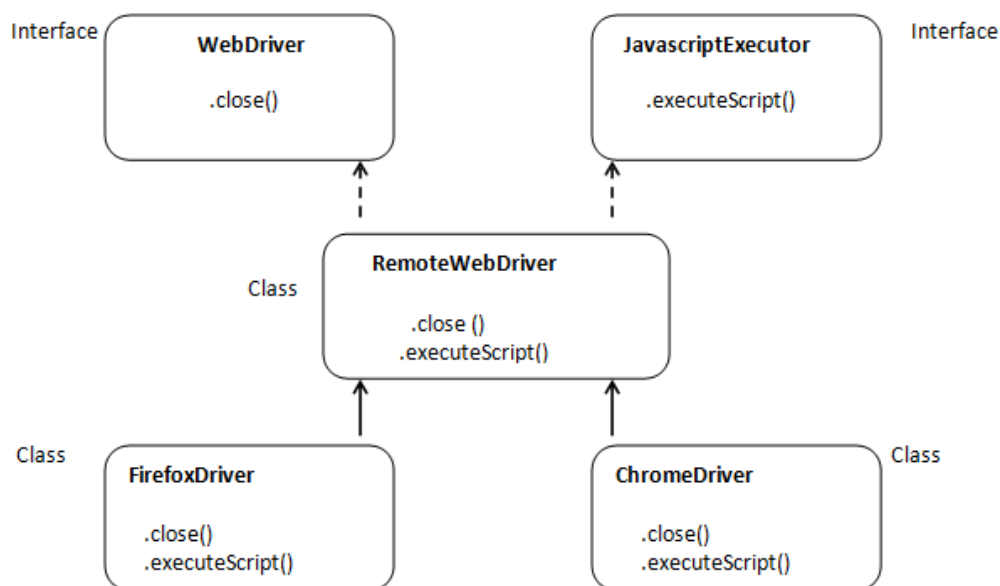
The **executeScript()** method declared is declared in **JavascriptExecutor** interface which is implemented in **RemoteWebDriver,** but after creating the object of the browser we will 'upcast' it to **WebDriver** interface which will hide all the methods of **JavascriptExecutor**. In order to access those methods we should 'downcast' it.

```java
package qspider;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
public class Javascript1
{
  public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://localhost/login.do");
        RemoteWebDriver r=(RemoteWebDriver) driver;
        r.executeScript("alert('Hi')");
    }
}
```

Q59. How do you click on the button using Java Script?

```java
package testaddda;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
public class Demo
{
    public static void main(String[] args)
    {
            WebDriver driver=new FirefoxDriver();
            driver.get("http://localhost/login.do");
            RemoteWebDriver r= (RemoteWebDriver) driver;
            String c="document.getElementById('loginButton').click()";
            r.executeScript(c);
    }
}
```

Q60. How do you enter the text into the textbox without using 'sendkeys'?

Using Javascript.

```java
package testaddda;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
public class Demo9
{
 public static void main(String[] args)
  {
     WebDriver driver=new FirefoxDriver();
     driver.get("http://demo.vtiger.com/");
     RemoteWebDriver r= (RemoteWebDriver) driver;
     String c="document.getElementById('username').value='admin'";
     r.executeScript(c);
  }
}
```

Q49. Write a code to remove the value present in the textbox using Java Script?

```java
 package testaddda;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
public class Demo10
{
  public static void main(String[] args)
  {
      WebDriver driver=new FirefoxDriver();
      driver.get("http://demo.vtiger.com/");
      RemoteWebDriver r= (RemoteWebDriver) driver;
      String c="document.getElementById('username').value=''";
      r.executeScript(c);
  }
}
```

Q61. How do you enter the value if textbox is disabled? **Ans:** Using **Javascript**.

```java
package qspider;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
public class Javascript2
{
  public static void main(String[] args)
  {
      WebDriver driver=new FirefoxDriver();
      driver.get("file:///D:/Demo2.html");
      RemoteWebDriver r= (RemoteWebDriver) driver;
      String c="document.getElementById('t1').value='Wonderful'";
      r.executeScript(c);
  }
}
```

Q62. How do you scroll the web page? By using java **script(scrollTo)**

Q63. Write a script to scroll to the bottom of the web page?

```java
package testaddda;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
public class Demo11
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("https://ww.google.com/");
        RemoteWebDriver r= (RemoteWebDriver) driver;
        String c="window.scrollTo(0,document.body.scrollHeight)";
        r.executeScript(c);
    }
}
```

Q64. Write a script to scroll to the specific element? Hint: get the Y coordinate of the element using get location method and pass it as argument for 'scrollTo' method.

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
public class Javascript3
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/mydownloads/Google%20News.htm");
        driver.manage().window().maximize();
        RemoteWebDriver r= (RemoteWebDriver) driver;
        WebElement el=driver.findElement(By.linkText("Business »"));
        System.out.println(el.getText());
        Point l = el.getLocation();
        int y=l.getY();
        String s="window.scrollTo(0,"+y+")";
        r.executeScript(s);
    }
}
```

DAY 14

HANDLING MULTIPLE ELEMENTS

In order to handle multiple elements we use **findElements**() method which returns **List of WebElement** [**List<WebElement>**]. List should be imported from java.util package.

Under the List we frequently use the following two methods.

1. list**.size()** - It returns element present in the List (return type int)
2. list**.get()** – It returns element present in the specified index (return type WebElement)

Note: For findElements() method we can use any of the 8 locators, but frequently use is xpath.

Sample page

```
<html>
      <body>
              <a href="http://www.testaddda.com">testaddda</a>
              <a href="http://www.techaddda.com">techaddda</a>
      </body>
</html>
```
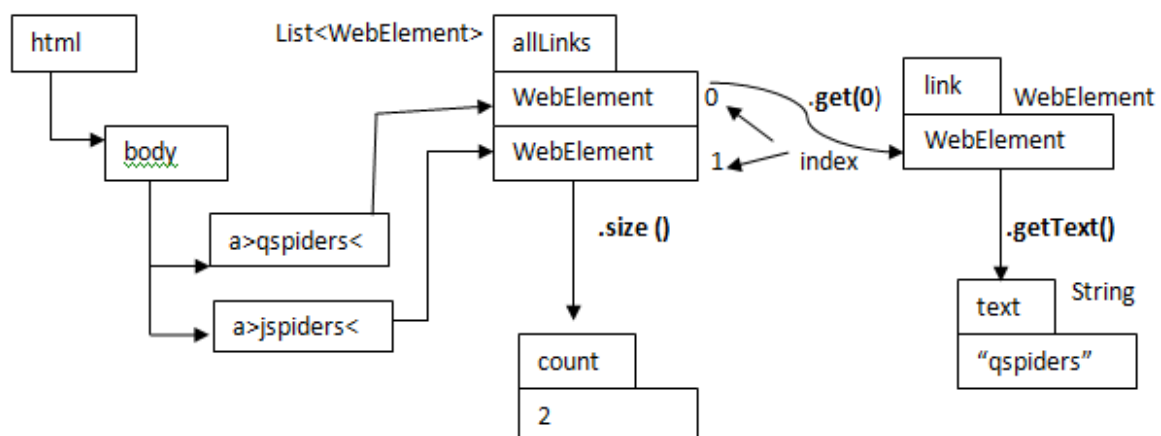

Return types:
List<WebElement>, WebElement, String

Sample html page > Demo4.html

```html
<html>
    <body>
        <div>
            <a href="http://www.testaddda.com">testaddda</a>
            <a href="http://www.techaddda.com">techaddda</a>
            <br><br>
            <input type="checkbox"><br><br>
            <input type="checkbox"><br><br>
            <input type="checkbox"><br><br>
            <input type="checkbox"><br><br>
            <input type="checkbox"><br><br>
            <input type="checkbox"><br><br>
        </div>
        <br><br>
        <div>
            <table border="1">
              <tbody>
                <tr>
                    <td>1</td>
                    <td>Java</td>
                    <td>300</td>
                </tr>
                <tr>
                    <td>2</td>
                    <td>Selenium</td>
                    <td>400</td>
                </tr>
                <tr>
                    <td>3</td>
                    <td>CRM</td>
                    <td>500</td>
                </tr>
              </tbody>
            </table>
        </div>
    </body>
</html>
```

```java
package qspider;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Elements1
{
  public static void main(String[] args)
  {
    WebDriver driver=new FirefoxDriver();
    driver.get("file:///D:/Demo4.html");
    List<WebElement> allLinks = driver.findElements(By.xpath("//a"));
    int count = allLinks.size();
    System.out.println("Total Number of links: "+count);
    WebElement link = allLinks.get(0);
    String text = link.getText();
    System.out.println(text);
  }
}
```

Q65. Write a script to print text of links present on the page.

```java
package qspider;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Elements1
{
  public static void main(String[] args)
  {
    WebDriver driver=new FirefoxDriver();
    driver.get("file:///D:/Demo4.html");
    List<WebElement> allLinks = driver.findElements(By.xpath("//a"));
    for(int i=0;i<allLinks.size();i++)
    {
      String text = allLinks.get(i).getText();
      System.out.println(text);
    }
  }
}
```

Note: alternate for loop

```
for(WebElement link:allLinks)
  {
    System.out.println(link.getText()
  }
```

Q66. Write a script to count the no.of checkboxes present on the page.

Q67. Write a script to select all the check boxes present on the page from top to bottom

```java
package qspider;

import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Elements1
{
  public static void main(String[] args)
  {
    WebDriver driver=new FirefoxDriver();
    driver.get("file:///D:/Demo4.html");
    List<WebElement> allchkbox = driver.findElements(By.xpath("//input"));
    int count = allchkbox.size();
    System.out.println("Total no. of checkboxes: "+count);

    //select all check boxes from top to bottom
    for(int i=0;i<count;i++)
    {
      WebElement chkbox = allchkbox.get(i);
      chkbox.click();
    }
    //deselect all checkboxes from bottom to top
    for(int i=count-1;i>=0;i--)
    {
      allchkbox.get(i).click();
    }
  }
}
```

Q68. Write a script to print all the url present on the page.

```java
package qspider;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Elements1
{
  public static void main(String[] args)
  {
    WebDriver driver=new FirefoxDriver();
    driver.get("file:///D:/Demo4.html");
    List<WebElement> allLinks = driver.findElements(By.xpath("//a"));
    for(WebElement link:allLinks)
    {
      String url = link.getAttribute("href");
      System.out.println(url);
    }
  }
}
```

Q69. Write a script to print the contents of the table.

```java
package qspider;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Elements1
{
  public static void main(String[] args)
  {
    WebDriver driver=new FirefoxDriver();
    driver.get("file:///D:/Demo4.html");
    List<WebElement> allCells = driver.findElements(By.xpath("//td"));
```

```
      int count=allCells.size();
      for(int i=0;i<count;i++)
      {
         WebElement cell = allCells.get(i);
         String text = cell.getText();
         System.out.println(text);
      }
   }
}
```

Q70. Write a script to print all numbers present in the table.

Q71. Write a script to print sum of all the numbers present in the table.

```java
package qspider;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Elements1
{
   public static void main(String[] args)
   {
      WebDriver driver=new FirefoxDriver();
      int sum=0;
      driver.get("file:///D:/Demo4.html");
      List<WebElement> allCells = driver.findElements(By.xpath("//td"));

      int count=allCells.size();
      for(int i=0;i<count;i++)
      {
         String text = allCells.get(i).getText();
         try
         {
              int x = Integer.parseInt(text);
              System.out.println(x); //prints only numbers
              sum=sum+x;
         }
```

```
    catch(NumberFormatException e)
    {
        //System.out.println(text); //Prints only strings
    }
  }
  System.out.println("Sum is: "+sum);
 }
}
```

Q72. Write a script to print on numbers present in the table without using Try-Catch

```
package qspider;
import java.util.List;
import org.apache.commons.lang3.StringUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Elements1
{
  public static void main(String[] args)
  {
    WebDriver driver=new FirefoxDriver();
    driver.get("file:///D:/Demo4.html");
    List<WebElement> allCells = driver.findElements(By.xpath("//td"));
    int count=allCells.size();
    for(int i=0;i<count;i++)
    {
      String text = allCells.get(i).getText();
      if(StringUtils.isNumeric(text))
      {
        System.out.println(text);
      }
    }
  }
}
```

Q73. What are the difference between findElement() and findElements()

findElement()	findElements()
Return type is WebElement	Return Type is List<WebElement>
If the specified locator is matching with Multiple elements, it returns first matching element.	If the specified locator is matching with multiple elements it returns all the matching elements.
If the specified locator is not matching with any of the element then it will throw NoSuchElementException.	If the specifies locator is not matching with any of the element it will not throw any exception instead of this it returns empty list.

DAY 15

HANDLING LISTBOX

Sample html page: Demo5.html   (Single selcte listbox

```html
<html>

<body>

    <select name="country">

    <option value="USD">United States</option>

    <option value="GBU">United Kingdom</option>

    <option value="CAD">Canada</option>

    <option value="BZL">Brazil</option>

    <option value="AUD">Australia</option>

    option value="CAD">Canada</option>

 </select>

</body>
```

Sample webpage: Demo6.html (Multiselect Listbox)

```html
<html>
<body>

  <select name="country" multiple size="10">
   <option value="USD">United States</option>
   <option value="GBU">United Kingdom</option>
   <option value="CAD">Canada</option>
   <option value="BZL">Brazil</option>
   <option value="AUD">Australia</option>
  <option value="CAD">Canada</option>


 </select>
</body>
</html>
```
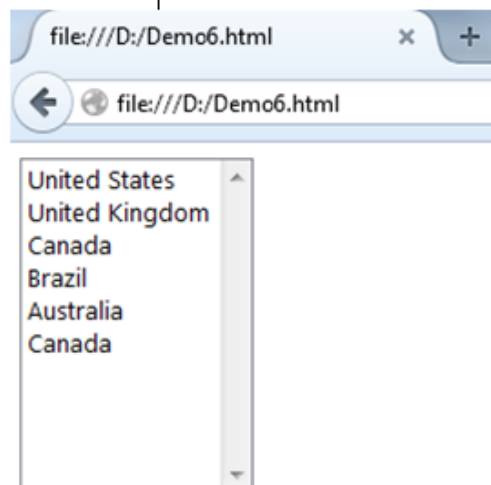
To handle the listbox, we use Select class of selenium. It should be imported from the following packages: **import org.openqa.selenium.suport.ui.Select**

Select class has parameterized constructor (single arg constructor) it takes an argument type **WebElement** (address of the listbox). In order to select the required option present in the listbox we can use any one the following method **of Select** class.

1. **selectByVisibleText(str**) > takes string argument
2. **selectByIndex(int)** > takes integer argument
3. **selectByValue(str)** > takes string argument

If the specified option is duplicate in will select first matching option(in dropdown list) and if the specified option is not present(text, value or index), we get ***NoSuchElementException.***

Select class can also be used to handle mulitselect listbox. If the specified option is duplicate in mutliselect listbox, it selects all the matching option.

In *Select* class we also have the following 4 methods. This can be used on Multiselect listbox

1. **deselectByVisibleText(str)**
2. **deselectByIndex(int)**
3. **deselectByValue(str)**
4. **deselectAll()**

Q74. Script to verify whether the listbox is single select or mutliselect

Note: We use **isMultiple()** methods present in Select class.

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class Listbox1
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo6.html");
        WebElement listBox = driver.findElement(By.name("country"));
        Select select=new Select(listBox);
        boolean v = select.isMultiple();
```

```
        //returns true if it is multiselect listbox
        //returns false if it is single select listbox
        System.out.println(v);
        //selects specific option
        select.selectByVisibleText("United Kingdom");
        select.selectByIndex(3);
        select.selectByValue("AUD");
        //deselect specific option
        select.deselectByVisibleText("United Kingdom");
        select.deselectByIndex(3);
        select.deselectByValue("AUD");
    }
}
```

Q75. Write a script to count no.of options present in the listbox

Q76. Write a scrtipt to select all the options present in the listbox and deselect all the option

```
package qspider;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class Listbox1
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo6.html");
        WebElement listBox = driver.findElement(By.name("country"));
        Select select=new Select(listBox);
        List<WebElement> allOptions = select.getOptions();
        int count=allOptions.size();
        System.out.println(count);
        for(int i=0;i<count;i++)
        {
                select.selectByIndex(i);
        }
        select.deselectAll();
    }
}
```

Q77. Write a script to print all the contents of the listbox

```java
package qspider;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class Listbox1
{
    public static void main(String[] args
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo6.html");
        WebElement listBox = driver.findElement(By.name("country"));
        Select select=new Select(listBox);
        List<WebElement> allOptions = select.getOptions();
        for(int i=0;i<allOptions.size();i++)
        {
            WebElement option = allOptions.get(i);
            String text = option.getText();
            System.out.println(text);
        }
    }
}
```

Q78. Write a script to search for specified option in the listbox

```java
package qspider;
import java.util.List;
import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class Listbox1
{
    public static void main(String[] args)
    {
        System.out.println("Enter option to search:");
        Scanner sc=new Scanner(System.in);
        String eText=sc.next();
        int found=0;
        System.out.println("Searching....");
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo6.html");
        WebElement listBox = driver.findElement(By.name("country"));
        Select select=new Select(listBox);
        List<WebElement> allOptions = select.getOptions();
        for(int i=0;i<allOptions.size();i++)
        {
            String atext = allOptions.get(i).getText();
                if(atext.equals(eText))
                {
                        found++;
                }
        }
        if(found==0)
        {
                System.out.println(eText+" is not found"); //No matching
        }
        else if(found==1) // if found >1 then duplicates
        {
            System.out.println(eText+" is found"); //matching found
        }
        else
        {
```
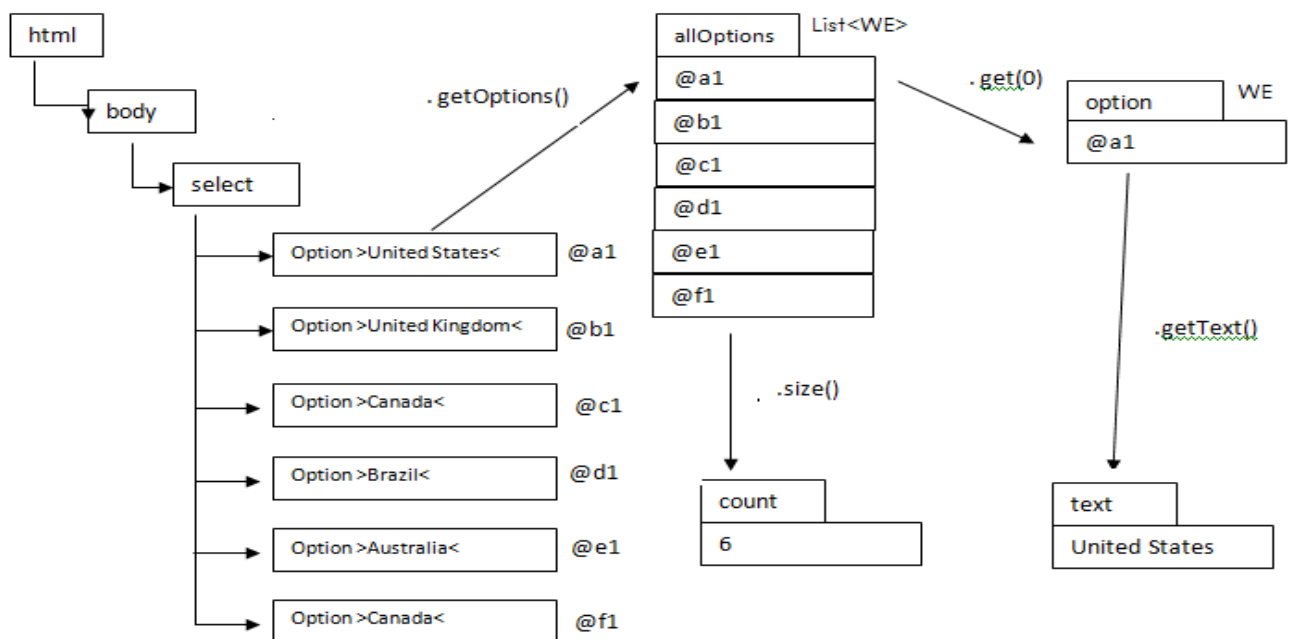
```java
                System.out.println(eText+ "is duplicate");
        }
        driver.close();
    }
}
```

Q79. Write a script to print the content of list in sorted order.

```java
package qspider;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class Listbox1
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo6.html");
        WebElement listBox = driver.findElement(By.name("country"));
        Select select=new Select(listBox);
        List<WebElement> allOptions = select.getOptions();

        ArrayList<String> allText=new ArrayList<String>();

        for(int i=0;i<allOptions.size();i++)
        {
            String text = allOptions.get(i).getText();
            allText.add(text);
        }
        Collections.sort(allText);
        for(String s:allText)
        {
            System.out.println(s);
        }
    }
}
```

Q80.

- Australia

- Brazil

- Canada

- Canada

- United Kingdom

- United States

a.) //Write a script to print only duplicates.

b.) //Write a script to print all the options except duplicates

```java
package qspider;
import java.util.HashSet;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class Listbox1
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo6.html");
        WebElement listBox = driver.findElement(By.name("country"));
        Select select=new Select(listBox);
        List<WebElement> allOptions = select.getOptions();
        HashSet<String> allText=new HashSet<String>();
        for(int i=0;i<allOptions.size();i++)
          {
                String text = allOptions.get(i).getText();
                if(allText.add(text)==false)
                  {
                        System.out.println(text); //print duplicate
                  }
          }
    }
}
```

//Write a script to print all the options except duplicates

```java
package qspider;
import java.util.HashSet;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class Listbox1
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/Demo6.html");
        WebElement listBox = driver.findElement(By.name("country"));
        Select select=new Select(listBox);
        List<WebElement> allOptions = select.getOptions();

        HashSet<String> allText=new HashSet<String>();

        for(int i=0;i<allOptions.size();i++)
        {
            String text = allOptions.get(i).getText();
            allText.add(text);
        }

        for(Object o:allText)
        {
            System.out.println(o);
        }

    }
}
```

DAY 16

Q81. Write a script to sort the content of the listbox without using sort() method of collection (Hint: using comparator() method of String class, it it returns any positive then swap the values.)

HANDLING AUTOSUGGESTION

In some of the textbox when we type a letter or set of characters it will automatically display set of options which is called as Autosuggestion.

Example: Google

Steps:

1. Open the browser
2. Go to www.google.com
3. Type "selenium" in the search textbox which display autosuggestions
4. Count the no.of autosuggestions
5. Print all the autosuggestions
6. Select one of them.

Script:

```java
package qspider;
import java.util.List;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Autosug1
{
 public static void main(String[] args)
 {
    WebDriver driver=new FirefoxDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.get("http://www.google.com");
    driver.findElement(By.id("lst-ib")).sendKeys("selenium");
```

```java
        String xp="//div[contains(text(),'selenium')]";
        List<WebElement> allAut = driver.findElements(By.xpath(xp));
        int count=allAut.size();

        System.out.println("No of autosuggesions:"+count);
        System.out.println("List of Autosuggestions..");
        int position=0;
        for(int i=0;i<count;i++)
          {
              WebElement aut = allAut.get(i);
              String text=aut.getText();
              System.out.println(text);
              if(text.equals("selenium"))
              {
                  position=i;
              }
          }
        allAut.get(position).click();
      }
    }
```

Q82. Write a script to perform the following steps

- Go to www.makemytrip.com
- specify Bang in the "from" field
- count the no of autosuggestions displayed
- print all the autosuggestions
- print the second option without using index

  Hint: press key down using sendKeys() method.

```java
        package qspider;
        import java.util.List;
        import org.openqa.selenium.By;
        import org.openqa.selenium.Keys;
        import org.openqa.selenium.WebDriver;
        import org.openqa.selenium.WebElement;
        import org.openqa.selenium.firefox.FirefoxDriver;
        public class Autosug1
        {
         public static void main(String[] args) throws InterruptedException
```

```
        {
            WebDriver driver=new FirefoxDriver();
            driver.manage().window().maximize();
            driver.get("http://www.makemytrip.com/flights");
            WebElement textbox=driver.findElement(By.id("from_typeahead1"));
            textbox.clear();
            textbox.sendKeys("Bang");
            Thread.sleep(2000);
            List<WebElement> allOptions =
        driver.findElements(By.xpath("//div[@class='tt-suggestion']"));
            int count=allOptions.size();
            System.out.println("No. of autosuggestions"+count);
            System.out.println("List of autosuggstions");
            for(int i=0;i<count;i++)
            {
                String text=allOptions.get(i).getText();
                System.out.println(text);

            }
        textbox.sendKeys(Keys.ARROW_DOWN);
        textbox.sendKeys(Keys.ARROW_DOWN);
        textbox.sendKeys(Keys.ENTER);
        }
    }
```

If listbox is developed using select tag then only we can use Select class.

If the listbox developed using some other tag such as input, div, li, ul(unorderedlist),ol etc then if we try to use Select class we get *UnexceptedTagNameException.*

To handle this situation we can use **click()** or **sendKeys**() method.

Q83. How do you handle listbox without using Select class

Ans: using sendKeys() or click() method

Q84. How do you handle dropdown menu.

Ans: Dropdown menu is a element on which if we move the mouse pointer it will display list of options to handle dropdown menu. And we use *moveToElement()* method of *Actions* class. t has parameterized constructor it takes an argument of type **WebDriver**.Whenever we call any methods of **Actions** class we should always call *perform()* method.

Eample

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;


public class Autosug2
{
  public static void main(String[] args)
  {
      WebDriver driver=new FirefoxDriver();
      driver.get("http://www.actimind.com");
      WebElement menu =    driver.findElement(By.xpath("//span[text()='About
Company']"));
      Actions actions=new Actions(driver);
      actions.moveToElement(menu).perform();
      driver.findElement(By.linkText("Basic Facts")).click();
  }
 }
```

Q85. Write a script to perform following steps: 1. go to istqb.in, 2. go to Foundation > Registration > Corporate Registration > select Online Registration

```java
package qspider;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;
public class Autosug2
{
  public static void main(String[] args)
  {
      WebDriver driver=new FirefoxDriver();
      driver.manage().window().maximize();
      driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
      driver.get("http://www.istqb.in");
       Actions actions=new Actions(driver);
      WebElement menu = driver.findElement(By.linkText("FOUNDATION"));
      actions.moveToElement(menu).perform();
```

```
        WebElement submenu1=driver.findElement(By.linkText("REGISTRATION"));
        actions.moveToElement(submenu1).perform();


        WebElement submenu2 = driver.findElement(By.linkText("Corporate
Registration"));
        actions.moveToElement(submenu2).perform();


        driver.findElement(By.linkText("Online Registration")).click();
   }
  }
```

Q86. How do you handle context menu?

 Ans: Right clicking is also called **as Contextclick**.

When we right click on any element we get list of options which is called as context menu.

To right click on the element we use *contextClick()* method of *Actions* class. and to select required option in the contextmenu, we press the shortcut such as "t" for new tab, "w" for newwindow etc using *sendKey()* method of *Actions* class.

 Example

```
    package qspider;
    import org.openqa.selenium.By;
    import org.openqa.selenium.WebDriver;
    import org.openqa.selenium.WebElement;
    import org.openqa.selenium.firefox.FirefoxDriver;
    import org.openqa.selenium.interactions.Actions;
    public class Autosug3
    {
     public static void main(String[] args)
     {
       WebDriver driver = new FirefoxDriver();
       driver.get("http://localhost/login.do");
       WebElement link=driver.findElement(By.linkText("Actimind Inc."));
       Actions actions=new Actions(driver);
       actions.contextClick(link).perform();
       actions.sendKeys("t").perform();
     }
    }
```

Q87. How do you perform drag and drop action in selenium.

Ans: using **dragAndDrop()** method of **Actions** class

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;

public class Autosug4
{
 public static void main(String[] args)
 {
    WebDriver driver=new FirefoxDriver();
    String url="http://www.dhtmlgoodies.com/submitted-scripts/i-google-like-drag-drop/index.html";
    driver.get(url);
    Actions actions=new Actions(driver);
    WebElement source=driver.findElement(By.xpath("//h1[text()='Block 1']"));
    WebElement target=driver.findElement(By.xpath("//h1[text()='Block 3']"));
    actions.dragAndDrop(source, target).perform();
  }
}
```

Note: We can use Actions class to double click on the element

Ex: actions.doubleClick(fileElement).perform();

DAY 17

ENCAPSULATION

Hiding the data and binding with methods is called as Encapsulation.Data will be stored in a variable in java for any given variable we should perform following steps:

1. **Declaration**
2. **Initialization**
3. **Utilization**

```java
public class A
{
    private int i;
     public  void A(int j)
      {
         i=j;
      }
public int getValue()
      {
       return i;
      }
}

public class B
  {
     public static void main(String[] args)
      {
        A a=new A(5);
        int x=a.getValue()
       System.out.println(x);
      }
    }
```

There are two classes in the given example; the purpose of class A is only to manage the variable i. whereas the purpose of class B is only to execute the code.

USING ENCAPSULATION IN SELENIUM

- Selenium code to enter "admin" in the username textbox.

```
driver.findElement(By.id("username")).sendKeys("admin");
```

- Above code can we written as shown below

```
WebElement unTextBox= driver.findElement(By.id("username"));
unTextBox.sendKeys("admin");
WebElement unTextBox;
unTextBox= driver.findElement(By.id("username"));
unTextBox.sendKeys("admin");
```

Example:// class LoginPage

```
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
public class LoginPage
  {
    private WebElement unTextBox;
    public LoginPage(WebDriver driver)
     {
       unTextBox=driver.findElement(By.id("username"));
     }
    public void setUserName()
     {
        unTextBox.sendKeys("admin");
     }
  }
```

//class MainMethod

```
package qspider;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class MainMethod
{
  public static void main(String[] args)
   {
      WebDriver driver=new FirefoxDriver();
      driver.get("http://localhost/login.do");
      LoginPage l=new LoginPage(driver);
      l.setUserName();
   }
}
```

Enhanced programe:

//class LoginPage

```java
package qspider;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
public class LoginPage
{
    private WebElement unTextBox;
    private WebElement pwTextBox;
    private WebElement loginButton;
    public LoginPage(WebDriver driver)
     {
            unTextBox=driver.findElement(By.id("username"));
            pwTextBox=driver.findElement(By.name("pwd"));
            loginButton=driver.findElement(By.id("loginButton"));
     }
    public void login(String un, String pw)
     {
            unTextBox.sendKeys(un);
            pwTextBox.sendKeys(pw);
            loginButton.click();
        }
    }
```

//class MainMethod

```java
package qspider;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class MainMethod
{
    public static void main(String[] args)
    {
         WebDriver driver=new FirefoxDriver();
        driver.get("http://localhost/login.do");
        LoginPage l=new LoginPage(driver);
        l.login("admin","manager");
    }
  }
```

Note: The login() method present in LoginPage class can be used to enter valid and invalid inputs:

```java
public class MainMethod
  {
    public static void main(String[] args)
      {
         WebDriver driver=new FirefoxDriver();
         driver.get("http://localhost/login.do");
         LoginPage l=new LoginPage(driver);
         l.login("abc","xyz");


         Thread.sleep(2000);
         l.login("admin","manager");
      }
  }
```

But when we execute the above code we get ***StaleElementReferenceException*** because when it clicks on loginbutton after entering invalid username and password, page will be reloaded and address of the element will be changed. But the reference variables such as unTextBox will be holding old address. It will try to enter valid username using old address which is invalid. Hence we get the Exception.

SCRIPT TO EXPLAIN **StaleElementReferenceException**

```java
public class MainMethod
{
  public static void main(String[] args)
  {
      WebDriver driver=new FirefoxDriver();
      driver.get("http://localhost/login.do");
      //stores username address in @a1
      WebDriver unTexBox=driver.findElement(By.id("username"));
      //refresh and username get new address x1
      driver.navigate().refresh();
      //try to enter admin using old address a1
      unTextBox.sendKeys("admin");
  }
}
```

DAY 18

POM CONCEPTS

To avoid StaleElementReferenceException we use Page Object Model POM class. POM is one of the Java design pattern. POM concept is used by both developers and testengineers (automation) to develop and test webpages.

In POM class we declare the element using **FindBy Annotation** and we write it as **@FindBy**. It should be imported from the following package

Import org.openqa.selenium.support.FindBy;

Syntax 1: single element

> @FindBy(locator="locator value")

> private WebElement elementname;

Syntax2: multiple element

> @FindBy(locator="locator value")

> private List<WebElement> elementname;

To initialize the element we use **initElements()** method of PageFactory class. It takes two arguments

- **WebDriver**
- **Object of POM class**

initElement() method will only loads the element (reference variable),but it will not initialize actually. Element are actually initialized during runtime when we try to perform any action on the element. This process is called as **Lazy Initialization**. This will avoid **StaleElementReferenceException.** Since only one element is loaded at a time it will improve the performance of the script execution.

Example: //POM class

```java
package qspider;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
public class LoginPage
{
    @FindBy(id="username")
    private WebElement unTextBox;
    @FindBy(name="pwd")
```

```java
        private WebElement pwTextBox;
        @FindBy(id="loginButton")
        private WebElement loginButton;
        public LoginPage(WebDriver driver)
          {
              PageFactory.initElements(driver, this);
          }
        public void login(String un, String pw)
        {
                unTextBox.sendKeys(un);
                pwTextBox.sendKeys(pw);
                loginButton.click();
          }
      }


        //class MainMethod


        package qspider;
        import org.openqa.selenium.WebDriver;
        import org.openqa.selenium.firefox.FirefoxDriver;
        public class MainMethod
          {
            public static void main(String[] args) throws InterruptedException
            {
                WebDriver driver=new FirefoxDriver();
                driver.get("http://localhost/login.do");
                LoginPage l=new LoginPage(driver);
                l.login("abc","xyz");
                Thread.sleep(3000);
                l.login("admin","manager");
            }
          }
```

Q88. what is pom?

Ans: it is one of the java designing pattern to develop and test webpage.

Q89. How do you declare the element in the pom class?

Ans:Usng FindBy Annotation (@FindBy)

- @FindBy(id="username")

- private WebElement unTextBox;

Q90. How do you handle multiple elements in pom class?

Ans: we handle it using @FindBy itself and we change the datatype to List<WebElement>

```java
package qspider;
import java.util.List;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
public class LoginPage
{
        @FindBy(xpath="//input[@type='checkbox']")
        private List<WebElement> allChkBox;
        public LoginPage(WebDriver driver)
        {
                PageFactory.initElements(driver, this);
        }
        public void selectAllChkBox()
        {
                int count=allChkBox.size();
                for(int i=0;i<count;i++)
                {
                        allChkBox.get(i).click();
                }
        }
 }
```

// Main method

```java
package qspider;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class MainMethod
{
  public static void main(String[] args)
    {
       WebDriver driver=new FirefoxDriver();
      driver.get("file:///D:/Demo4.html");
      LoginPage l=new LoginPage(driver);
      l.selectAllChkBox();
      //deselects all
      l.selectAllChkBox();
    }
}
```

Q91. What happens if we do not use intiElements() method in POM class?

Ans: we get NullPointerException

Q92. Can we develop POM class without the constructor?

Ans: Yes. We should explicitly call initElements() method.

//class LoginPage

```java
package qspider;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
public class LoginPage
{
  @FindBy(id="username")
  private WebElement unTextBox;


    public void setUserNAme(String un)
  {
        unTextBox.sendKeys(un);
  }
}
```

//Main Method
```java
package qspider;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.PageFactory;
public class MainMethod
{
  public static void main(String[] args)
    {
       WebDriver driver=new FirefoxDriver();
      driver.get("http://localhost/login.do");
      LoginPage l=new LoginPage();
      PageFactory.initElements(driver, l);
      l.setUserNAme("admin");
    }
  }
```

Q93. What is the difference between Page Object Model and PageFactory

Ans: POM is java design concept. PageFactory is class which implements cocept.

Q94. What is the advantage of using POM class?

Ans: it will avoid StaleElementReferenceException and it improve the performance

Q95. What is Object Repository?

It is the location where we store the objects(elements)

Q96. What is Page Object Repository?

Ans: It is the location where we store the elements present on the page.

Page ObjectModel is also called as Page Object Repository

LIMITATIONS OF @FindBy

-    We cannot use variables in place of locatorvalue in the FindBy Annotation

Below code is invalid.
int i=1;
@FindBy(id="username"+i)
private WebElement unTextBox;

DAY 19

TESTNG – TEST NEXT GENERATION

It is unit testing framework. Basically TestNG is used by developers to perform unit testing and it is also used in selenium to perform BlackBox testing.

- **Run multiple test classes**
- **Generate Reports**
- **Rerun only failed classes etc**

TestNG is available as plug-in for Eclipse IDE

STEPS TO INSTALL TestNG

1. Open the eclipseIDE > click on Help > select Eclipse Market Place
2. Search for TestNG
3. Click Install Button of TestNG
4. Select "Accept" and click finish.
5. Click "ok" on the popup
6. Click "yes" which restarts the Eclipse
7. Right click JavaProject and go to Properties
8. Click on Java Build Path
9. Click on "Libraries" tab
10. Click on "Add Library"
11. Select TestNG > click Next > click finish and click on OK, this will associate TestNG with current Java Project.

FOLLOWING 4 ARE THE REQUIRED JAR FILES:

1. testing.jar
2. jcommander.jar
3. bash-2.0b4.jar
4. snakeyaml.jar

TestNG class   While creating TestNG class we should not use

   i.   Default package
   ii.   No main() method
   iii. No System.out.println

Example:

```
package testNg;
import org.testng.Reporter;
import org.testng.annotations.Test;
public class TestNGdemo1
 {
   @Test
   public void testDemo()
   {
        Reporter.log("Welcome to testNG", true);
   }
}
```

When we execute the above code it will automatically generate execution result in html format, in order to see it

1. Refresh the Javaproject which will displays "test-output" folder
2. Expand the folder & right click on "emailable-report.html"
3. Goto Open With and select WebBrowser

TestNG Suit

It is an xml file , which contains list of all TestNG classes which are to be executed.

Suite file is used for Batch execution. To create it:

1. Right click on Java Project
2. Goto TestNG
3. Select Convert to TestNG
4. Click finish
5. It creates TestNG.xml file inside the JavaProject

To Execute it

1. Right click on xml file.
2. Goto Run as & select TestNG suite.

Content of TestNG Suite

```xml
<suite name="Suite" parallel="none">
  <test name="Test">
    <classes>
      <class name="testNg.TestNGdemo1"/>
    </classes>
  </test>
</suite>
```

DAY 20

Q97. How do you execute only failed test classes?

Ans: using *TestNG-Failed.xml* file present in *test-output* folder

Q98. If class contains multiple test methods in which order they are executed?

Ans: Alphabetical order (ascending)

Q99. How to execute the test method in required order?

Ans: using priority.

Note: Default priority is 0.

If the priority is duplicate then those methods will be executed in alphabetical order.

We can specify –ve value for priority and it will execute them in ascending order.

Variable and decimal numbers are not allowed.

Q100. How do you run a test method multiple-time?

Ans: using invocationCount.

Note: Defualt invocationCount is 1.

If we specify 0 or –ve number it will not execute the test method.

Fraction numbers and variables are not allowed.

Q101. What are the import annotations used in TestNG with respect to selenium.

Ans:

 @Test – this indicates test method.

@BeforeMethod – this method will be executed before the execution of every test methods.

@AfterMethod – this method is executed after execution of every method

@BeforeClass – this method is executed only once at the beginning of the class

@AfterClass – this method is executed only once at the end of the class.

Example:

```java
package testNg;
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
public class TestngDemo2
{
   @BeforeClass
   public void openApp()
      {
         Reporter.log("OpenApp", true);
      }
   @AfterClass
   public void closeApp()
      {
       Reporter.log("CloseApp",true);
      }
   @BeforeMethod
   public void login()
      {
       Reporter.log("Login",true);
      }
   @AfterMethod
   public void logout()
      {
       Reporter.log("Logout",true);
      }
   @Test (priority=2, invocationCount=2)
   public void editUser()
      {
         Reporter.log("Edit User",true);
      }
   @Test
   public void register()
      {
       Reporter.log("Register",true);
      }
   @Test
```

```
    public void deleteUser()
    {
        Reporter.log("Delete User",true);
    }
}
```

Output>>

OpenApp

Login

Delete User

Logout

Login

Register

Logout

Login

Edit User

Logout

Login

Edit User

Logout

CloseApp

Q102. How do make a test depend on other test?

Ans: By using dependsonMethods option.

Q103. If both primary and dependency are specified which one will be used?

Ans: dependency

Example

```java
package testNg;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;

public class TestngDemo3
{
  @Test (priority=2)
  public void createUser()
  {
        Reporter.log("Create User",true);
      Assert.fail();
  }
  @Test (priority=1,dependsOnMethods={"createUser"})
  public void deleteUser()
  {
        Reporter.log("Delete User",true);
  }
}
```

In the above example deleteUser() method depends on createUser() method i.e If createUser() method is executed successfully (passed), then only it will execute deleteUser() method.

If createUser() method is failed, then it will skip the execution of deleteUser() method.

Q104. What if 2 methods are dependent on each other?

Ans: we get TestNGException

 Error is Cyclic dependencies

Q105. How do you intentionally fail the test?

Ans: Using Assert.fail()

Q106. How do you compare actual and expected results without using if-else statement?

Ans: Assert.assertEquals() statement

```java
package testNg;

import java.util.Scanner;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

public class TestngDemo4
{
    public WebDriver driver;
    public String eTitle;
    @BeforeMethod
    public void preCondition()
     {
            System.out.println("Enter expected title:");
            Scanner sc=new Scanner(System.in);
            eTitle=sc.next();
            Reporter.log("eTitle: "+eTitle,true);
            driver=new FirefoxDriver();
     }
    @Test
    public void testGoogle()
     {
            driver.get("http://www.google.com");
            String aTitle=driver.getTitle();
            Reporter.log("aTitle: "+aTitle,true);
            Assert.assertEquals(aTitle, eTitle);
     }


    @AfterMethod
     public void postCondition()
      {
            driver.close();
      }
}
```

Note: If comparison fails then the statements which are present after the Assert statement of current test method will not be executed

Example

```
@Test
Public void testGoogle()
{
 Reporter.log("Step1",true);
Assert.assertEquals("abc","xyz");
Reporter.log("step2",true)
}
```

During the execution of above test method it will print step1 that comparison fails, hence it will not print step2.

Q107. What are the important methods available under **Assert** Class?

Ans:

- **assertEquals()**
- **assertNotEquals()**
- **assertTrue()**
- **assertFalse()**
- **assertSame()**
- **assertNotSame()**
- **assertNull()**
- **assertNotNull()**
- **fail()**

All the above methods are static methods of Assert Class. In order to continue the execution even after failure of the comparison, we can use **SoftAssert** . But all methods are nonstatic

Example

```
package testNg;
import org.testng.Reporter;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;
public class TestngDemo3
{
    @Test
    public void testGoogle()
    {
            SoftAssert s= new SoftAssert();
            Reporter.log("Step1",true);
            s.assertEquals("xyz","abc");
            Reporter.log("Step2",true);
            Reporter.log("Step3",true);
            s.assertAll();
            Reporter.log("Step4",true);
    }
}
```

Output

Step1

Step2

Step3

 FAILED: testGoogle

 Note: To update status of the comparison into the result window we should use assertAll() method. Any statement after assertAll() method will not be executed if comparison fails.

Q109. What is the difference between Assert and SoftAssert ?

Assert	SoftAssert
If comparison fails remaining statement will not be executed in current class	Executes remaining statements even if comparison fails
All methods are static	All methods are non-static
We do not call assertAll() method	We should call assertAll() method

DAY 21

AUTOMATION FRAMEWORK

It is the standard rules, best practices and folder structure which should be followed while automating the application testing.

We should follow the Automation Framework to have constistancy.

In automation framework we have 3 stages.

I.   Automation Framework Design

II.   Automation Framework Implementation

III. Automation Framework Execution

I.   AUTOMATION FRAMEWORK DESIGN

This is the initial stage where automation lead or manager will specify files and folder structures, naming conventions and rules which should be followed to develop and execute the automation script.

Based on the design framework is catogarized into following types:

1. Method-driven Automation Framework

2. Data-driven Automation Framework

3. Module-driven Automation framework

4. Keyword-driven Automation framework

5. Hybrid Automation Framework

Note: The above type of Framework can be customized and implemented in a company with different names. Such as Cucumber, Robot, Protractor, Craft etc. Generally, folder structures are decided based on the file types

Example:

File Types	Location
.java	javaproject/src
.class	javaproject/bin
.xml	javaproject
.html	javaproject/test-output
.jar	javaproject/jarfiles
.exe	javaproject/exefiles
.xlsx	javaproject/testdata
.bat	javaproject
.war	javaproject

Steps to configure Automation Framework

1. Goto required location (ex: D:) drive and create a folder (ex: BCSM6).

2. Go to **File > Switch Work Space** other, browse and select above folder and click OK.

3. Create a javaproject with the name **"Automation".**

4. Create a folder with the name **"exefiles"**under "**Automation"** and copy paste "**chromedriver.exe"** and **"IEDriverServer.exe".**

5. Create a folder with the name **"jarfiles"** under **Automation** and copy paste "**selenium server standalone" jar file**.

6. Expand "**jarfile"** folder and right click on "**selenium server standalone"** jar file and select **Build Path** > **Add to Build Path.**

7. Associate **TestNG** to the javaproject.

8. Create a folder with the name **"testdata"** under **"automation"** which is used to store excel files.

9. Create two packages under **"src"** with the name **"pom"** and **"scripts"**.

   o pom packages is used to store POM class.

   o script package is used to store TestNG class

While writing scripts we develop two types of classes:

1. **POM class**
2. **TestNG class**

First we should always develop **pom** class. It is used to store elements and perform the action.

We use **TestNg** class for execution purpose.

Ex: //POM class

```java
package pom;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
public class LoginPage
{
  @FindBy(id="username")
  private WebElement unTextBox;
  public LoginPage(WebDriver driver)
   {
     PageFactory.initElements(driver, this);
   }
```

```java
        public void setName(String un)

        {

            unTextBox.sendKeys(un);

        }

    }

//TestNG

    package scripts;

    import java.util.concurrent.TimeUnit;

    import org.openqa.selenium.WebDriver;
    import org.openqa.selenium.firefox.FirefoxDriver;
    import org.testng.annotations.Test;

    import pom.LoginPage;
    public class Demo
    {
     @Test
     public void TestDemo()
        {
            WebDriver driver=new FirefoxDriver();
            driver.manage().timeouts().implicitlyWait(10,
            TimeUnit.SECONDS);
            driver.get("http://localhost");
            LoginPage l=new LoginPage(driver);
            l.setName("admin");
        }
    }
```

DAY 22

HANDLING FRAMES

1. Webpage present inside another webpage is called as embedded webpage.

   Developer creates embedded webpage using **iframe** html tage.

2. If right click on any element which inside the frame it will display **This frame** option in the context menu.

3. To transfer the control into the frame we should use following statemen:t

   driver.switchTo().frame(arg);

   where arg > can be index of the frame (int) or id of the frame (string) or element of the frame (WebElement).

   In order to transfer the control back to the main page we can use the following statements:

   driver.switchTo().defaultContent();

4. In case of nested frames, to switch from child frame to parent frame we can use following statements;

   driver.switchTo().parentFrame();

   Example:

   Content of DemoB.html


   ```
   <html>
           <body>
                   t2:<input id="t2" type="text"/>
           </body>
   </html>
   ```


   Content of DemoA.html


   ```
   <html>
           <body>
                   t1:<input id="t1" type="text"/><br><br>
                   <iframe id="f1" class="c1" src="DemoB.html"/>
           </body>
   </html>
   ```

Script:

```java
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

class Demo
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("file:///D:/DemoA.html");
        driver.findElement(By.id("t1")).sendKeys("abc");

        driver.switchTo().frame(0);
        //driver.switchTo().frame("f1");

        //WebElement f=driver.findElement(By.className("c1"));
        //driver.switchTo().frame(f);
        driver.findElement(By.id("t2")).sendKeys("xyz");

        driver.switchTo().defaultContent();
        driver.findElement(By.id("t1")).sendKeys("123c");
    }

}
```

Assignment:

Write a script to perform following steps:

1. open the browser and enter the following url http:// www.zoho.com/crm/lp/login.html
2. Enter invalid email address, invalid password and click on Sign in.

    Verify that error msg is displayed.

```java
package example;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

class Exam
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
        driver.get("https://www.zoho.com/crm/lp/login.html");

        driver.findElement(By.id("lid")).sendKeys("admin1");

        driver.findElement(By.id("pwd")).sendKeys("1234");
        driver.findElement(By.id("submit_but")).click();
        driver.switchTo().defaultContent();
        driver.switchTo().frame("zohoiam");

        WebElement errMsg=driver.findElement(By.id("msgpanel"));
        if(errMsg.isDisplayed());
        {
            System.out.println(errMsg.getText());
        }
    }
}
```

Note: If the page is refreshed or new page is loaded then control will be automatically transferred to main page (from the frame).

Embeded web page can also be created using **frameset** html tag.

Conti...Automation Framework

II.   AUTOMATION FRAMEWORK IMPLIMENTAION

This is the second stage of the framework where we convert Manual testcased (regression) into automation script byu developing two types of classes **POM** and **TestNG**.

Sample Test Cases (TC)

TC1: Valid Login

Pre-condition: Login page should be present

Steps:

1.   Enter valid user name.
2.   Enter valid password.
3.   Click on Login button.
4.   Click on Logout link.

Post-condition: Browser should be closed.

TC2: Invalid login
Steps:

1.   Enter invalid username
2.   Enter invalid password
3.   Click on Login button.
      Verify that errot message is displayed.

TC3:
 Verify Build Number

1.   Enter valid username
2.   Enter valid password
3.   Click on login button
4.   click on Help
5.   click on About actiTIME
6.   verify that build number is 272661
7.   click on logout
8.   click on close

Steps to develop POM class

1. Execute the test case manual which gives more clarify on the steps which is to be automated.

2. While executing the testcase notedown the page, its elements and the action

3. For each page present on the application create a POM class under "pom" package.

4. The name of the class should be same as Title of the page ending with the word Page.

5. In every POM class we should declare the element usig **FindBy annotation @FindBy,** initialize it using **initElemetns()** and uitilize it using getters and setters methods.

Example

LoginPage

Elements	Actions
unTextBox	senKeys()
pwTextBox	sendKeys()
loginButton	click()
errorMessage	Verify

Enter Time-Track Page

Elements	Actions
logoutButton	click()
Help	click()
About actiTime	click()
BuildNumber	Verify
Close	click()

//POM for LoginPage

```java
package pom;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.testng.Assert;

public class LoginPage
 {
  @FindBy(id="username")
  private WebElement unTextBox;
  @FindBy(name="pwd")
  private WebElement pwTextBox;
  @FindBy(id="loginButton")
  private WebElement loginButton;
  @FindBy(xpath="//span[contains(text()'is invalid')]")
  private WebElement errMsg;

  public LoginPage(WebDriver driver)
     {
      PageFactory.initElements(driver, this);
     }
  public void setUserName(String un)
     {
       unTextBox.sendKeys(un);
     }
  public void setPassword(String pw)
     {
       pwTextBox.sendKeys(pw);
     }
  public void clickLoginButton()
     {
       loginButton.click();
     }
  public void verifyErrMsg()
     {
        Assert.assertTrue(errMsg.isDisplayed());
     }

  }
```

//**EnterTimeTrackPage POM class**

```java
package pom;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.testng.Assert;
public class EnterTimeTrackPage
{
 @FindBy(id="logoutLink")
 private WebElement logoutLink;
 @FindBy(xpath="(//div[@class='popup_menu_arrow'])[3]")
 private WebElement help;
 @FindBy(linkText="About actiTIME")
 private WebElement aboutActiTIME;
 @FindBy(xpath="//span[contains(text(),'build')]")
 private WebElement buildNumber;
 @FindBy(xpath="//im[@title='Close']")
 private WebElement close;
    public EnterTimeTrackPage(WebDriver driver)
     {
        PageFactory.initElements(driver, this);
     }
    public void clickLogoutLink()
     {
        logoutLink.click();
     }
    public void clickAboutActiTime()
     {
        aboutActiTIME.click();
     }
   public void verifyBuildNumber(String eBuildNumber)
     {
        String aBuildNumber=buildNumber.getText();
        Assert.assertEquals(aBuildNumber, eBuildNumber);
     }
    public void clickClose()
     {
        close.click();
     }
  }
```

DAY 23

Summary:

Class #1: LoginPage

Methods:

1. setUserName

2. setPassword

3. clickLoginButton

4. verifyErrMsg

Class #2: EnterTimeTrackPage

Methods:

1. clickLogoutLink

2. clickHelp

3. clickAboutActiTIME

4. verifyBuildNumber

5. clickClose

Developing TestNG class

- For every manual test case we should develop a 'TestNG' class. In all the test cases some of the steps will be common such as pre-conditions and post-conditions. In order to handle this we use 'Inheritance' concept.

    Ex: Create a class with the name 'BaseTest' and scripts package and write the code as shown below.

```java
package script;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeMethod;
public class BaseTest
{
        public WebDriver driver;
        @BeforeMethod
        public void preCondition()
        {
         driver=new FirefoxDriver();
        driver.get("http://localhost");
```

```
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    }
            @AfterMethod
    public void postCondition()
    {
        driver.close();
    }
}
```

- We cannot directly execute 'BaseTest' class as there is No test method
- All the TestNG class should extend 'BaseTest' class

Important Note: Steps to convert manual test cases into automation script or steps to develop TestNG class

1. Create a class inside script package and name of the class should be same as respective test case ID
2. Extend the **TestNG** class from **BaseTest** class
3. Create a method using **@Test** annotation and the name of the method should start with test and end with the class name
4. Write the manual test case steps as inline comment
5. Below each inline comment call the required method of POM class
6. Execute the script and ensure that it is working fine

Hiding Methods of Object class

1. In eclipse IDE go to Window→select Preferences
2. Go to Java→Appearance→Type filters and click on Add 'java.lang.Object'. Click on 'OK' and again on 'Ok' button.

Automation Script for Test Case #1:

```
package scripts;
import org.testng.annotations.Test;
import pom.EnterTimeTrackPage;
import pom.LoginPage;
public class ValidLogin extends BaseTest
{
    @Test

    public void testValidLogin()
    {
```

```java
        //Enter valid user name
        LoginPage l=new LoginPage(driver);
        l.setUserName("admin");


        //Enter valid password
        l.setPassword("manager");


        //Click on login button
        l.clickLoginButton();


        //Click on logout link
        EnterTimeTrackPage e=new EnterTimeTrackPage(driver);
        e.clickLogoutLink();
    }
}
```

Automation Script for Test Case #2:

```java
package scripts;
import org.testng.annotations.Test;
import pom.LoginPage;
public class InvalidLogin extends BaseTest
{
    @Test
    public void testInvalidLogin()
    {
        //Enter invalid user name
        LoginPage l=new LoginPage(driver);
        l.setUserName("abc");
        //Enter invalid password
        l.setPassword("xyz");
        //Click on login button
        l.clickLoginButton();
        //Verify Error Message
        l.verifyErrMsg();
    }
}
```

Automation Script for Test Case #3:

```java
package scripts;
import org.testng.annotations.Test;
import pom.EnterTimeTrackPage;
import pom.LoginPage;
public class VerifyBuildNumber extends BaseTest
{
        @Test
        public void testVerifyBuildNumber()
        {
                //Enter valid user name
                                LoginPage l=new LoginPage(driver);
                                l.setUserName("admin");
                                //Enter valid password
                                l.setPassword("manager");
                                //Click on login button
                                l.clickLoginButton();

                                EnterTimeTrackPage e=new EnterTimeTrackPage(driver);

                                //click on Help
                                e.clickHelp();

                                //click on About ActiTIME
                                e.clickAboutActiTime();

                                //Verify Build Number
                                e.verifyBuildNumber("(build 29885)");

                                //close About ActiTIME popup
                                e.clickClose();

                                //Click on logout link
                                e.clickLogoutLink();
        }
}
```

III.   AUTOMATION FRAMEWORK EXECUTION

To run all the scripts present in the frame work we use TestNG suite file.

To Create It:

1. Right click on Java Project(Ex: **Automation**),

2. Select **TestNG→Convert to TestNG** and click on **Finish**.

3. It creates **testng.xml** file inside Javaproject folder as shown below.

```
<suite name="Suite" parallel="none">
 <test name="Test">
  <classes>
   <class name="scripts.VerifyBuildNumber"/>
   <class name="scripts.InvalidLogin"/>
   <class name="scripts.ValidLogin"/>
  </classes>
 </test>
</suite>
```

4. Right click on **testng.xml** file;

5. Go to "**Run As**" and select "**TestNG suite**".

6. It will execute all the scripts present in the suite file and generate the results in HTML format (emailable-report.html) inside the**"test-output"** foler (refresh javaproject folder).

Executing Framework from command prompt:

1. Go to the location where eclipse is present i.e. right click on the eclipse short cut and select open file location.

2. Go to plugins → org.testng.elcipse→lib folder

3. Copy bsh, jcommander, snake and testng jar files.

4. Paste then inside "jarfiles" folder of framework

5. Open the command prompt and navigate to location where TestNG.xml is present

   D:\BCSM6\Automation>

6. Type following command and execute:

   java -cp bin;jarfiles/ org.testng.TestNG testng.xml*

Using batch file:

Instead of opening command prompt navigating the required location and typing and executing the command, we can use Batch file so that all these steps can be done by just double clicking it.

Example:

1.   Open the Notepad and type the following command;

"*java -cp bin;jarfiles/* org.testng.TestNG testng.xml*"

2.   Go to File and select Save. Navigate to java project folder (Ex: *D:/BCSM6/Automation)* and specify the name file as **"RunMe.bat"** and click Save.

3.   Double click on the batch file "**RunMe.bat"**

DAY 24

METHOD-DRIVEN FRAMEWORK

Executing the script by calling the methods present in the frameworks is called as 'Method Driven Framework'. Methods will avoid repetition of the steps and they will increase re-usability of the code.

In order to test the features thoroughly that is with all possible inputs only methods are not sufficient we should use external files such as excel file. If this feature is available in the framework then such type of framework is called **'Data-Driven Framework'**.

To handle the excel file we use API provided by Apache called POI (Poor Obfuscation Implementation). We can download it from following URL: **http://mirror.fibergrid.in/apache/poi/release/bin/poi-bin-3.13-20150929.zip**. After downloading the file extract it which creates a folder with the name 'poi-3.13'

It has many jar files, only following 4 jar files are required.

1. poi-3.13-20150929
2. poi-ooxml-3.13-20150929
3. poi-ooxml-schemas-3.13-20150929
4. xmlbeans-2.6.0

Copy above jar files into 'jarfiles' folder of the Framework and then associate them to Java project.

Steps to read Data from Excel Sheet

1. Open the excel file, go to sheet 1.
2. Go to Row 0, Go to Cell 0
3. Get the value present in the call and print it

```java
package example;
import java.io.FileInputStream;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;
```

```java
public class ExcelData1
{
        public static void main(String[] args) throws
EncryptedDocumentException, InvalidFormatException, IOException
        {
                FileInputStream fis=new FileInputStream("D:/BCSM6/Book1.xlsx");
                Workbook wb=WorkbookFactory.create(fis);
                Cell c = wb.getSheet("Sheet1").getRow(0).getCell(1);
                System.out.println(c);

                //Sheet s = wb.getSheet("Sheet1");
                //Row r = s.getRow(0);
                //Cell c = r.getCell(0);
                //String v = c.toString();
                //System.out.println(v);
        }
}
```

IQ:

1. Write a script to print content of the excel sheet where it has data in 3X3 matrix

```java
package example;


import java.io.FileInputStream;
import java.io.IOException;


import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;
public class ExcelData2
{
        public static void main(String[] args) throws
EncryptedDocumentException, InvalidFormatException, IOException
        {
                FileInputStream fis=new FileInputStream("D:/BCSM6/Book1.xlsx");
                Workbook wb=WorkbookFactory.create(fis);
                for(int i=0;i<=2;i++)
```

```
            {
                for(int j=0;j<=2;j++)
                {
                    Cell c = wb.getSheet("Sheet1").getRow(i).getCell(j);
                    System.out.println(c+" ");
                }
            System.out.println();
            }
        }
    }
```

Note: Sheet name, row index or cell index is invalid we get 'NullPointerException'

In order to count number of rows present in the sheet we should use **getLastRowNum**() method which returns the index of the last row.

To count the number of cells in the specified row we should use **getLastCellNum**() method which returns the count and not the index.

```java
package example;
import java.io.FileInputStream;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class ExcelData3
{
    public static void main(String[] args) throws EncryptedDocumentException,
InvalidFormatException, IOException
    {
        FileInputStream fis=new FileInputStream("D:/BCSM6/Book2.xlsx");
        Workbook wb=WorkbookFactory.create(fis);

        int rc=wb.getSheet("Sheet1").getLastRowNum();
        System.out.println(rc);

        int cc=wb.getSheet("Sheet1").getRow(0).getLastCellNum();
        System.out.println(cc);
    }
}
```

	0	1	2	3	4	
0	**A**	**A**	**A**			→3
1	**B**				**B**	→5
2		**C**	**C**			→3
3				**D**		→4
4						→0
5		**F**				→2

Integrating excels features in the framework

1. Ensure that **POI jar files** are associated with framework
2. Create a package with the name **generics** under **src**
3. Then create class with the name **Excel**
4. Write the code as shown below

```java
package generics;
import java.io.FileInputStream;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;


public class Excel
{
    public static String getCellData(String xlpath,String sheet,int row,int cell)
    {
        String v="";
        try
        {
            FileInputStream fis=new FileInputStream(xlpath);
            Workbook wb=WorkbookFactory.create(fis);
            v=wb.getSheet(sheet).getRow(row).getCell(cell).toString();
        }
        catch(Exception e)
        {
        }
        return v;
    }
```

```
public static int getRowCount(String xlpath,String sheet)
{
        int rc=0;
        try
        {
                FileInputStream fis=new FileInputStream(xlpath);
                Workbook wb=WorkbookFactory.create(fis);
                rc=wb.getSheet(sheet).getLastRowNum();
        }
        catch(Exception e)
        {
        }
        return rc;
}
}
```

Taking data from Excel sheet in framework

1. Go to **testdata** folder of the framework and create an excel file with the name **TDR.xlsx** (Test Data Repository)

2. Rename the Sheet1 as **ValidLogin** (respective class name)

3. Enter the data as shown below

UserName	Password
admin	manager

4. **Save** and close the excel file

5. Update the **TestNG class** as shown below

```java
package scripts;
import generics.Excel;
import org.testng.annotations.Test;
import pom.EnterTimeTrackPage;
import pom.LoginPage;

public class ValidLoginExcel extends BaseTest
{
    @Test
    public void testValidLogin1()
    {
        String xlpath="./testdata/TDR.xlsx";
        //DOT->current path of Java Project
        String sheet="ValidLogin";
        String un=Excel.getCellData(xlpath, sheet, 1, 0);
        String pw=Excel.getCellData(xlpath, sheet, 1, 1);
        LoginPage l=new LoginPage(driver);
        l.setUserName(un);
        l.setPassword(pw);
        l.clickLoginButton();
        EnterTimeTrackPage e=new EnterTimeTrackPage(driver);
        e.clickLogoutLink();
    }
}
```

DAY 25

Executing invalid login scripts with multiple inputs

```java
package scripts;

import org.testng.annotations.Test;

import pom.LoginPage;
import generics.Excel;
public class InvalidLoginExcel extends BaseTest
{
    @Test
    public void testInvalidLogin() throws InterruptedException
    {
    String xlpath="./testdata/TDR.xlsx";
    String sheet="InvalidLogin";
    int rc=Excel.getRowCount(xlpath,sheet);
    for(int i=1;i<=rc;i++)
    {
    String un=Excel.getCellData(xlpath, sheet, i, 0);
    String pw=Excel.getCellData(xlpath, sheet, i, 1);
    LoginPage l= new LoginPage(driver);
    l.setUserName(un);
    l.setPassword(pw);
    l.clickLoginButton();
    Thread.sleep(1000);
    l.verifyErrMsg();

    }
}
}
```

Q. How do you send a data from TestNG suite file or xml file into TestNG Methods?

Ans: Using Parameter.

Example:

//Sending part

```xml
<suite name="Suite" parallel="none">
  <test name="Test">
    <parameter name="state" value="Kararnatak"/>
    <parameter name="city" value="Bangalore"/>
    <parameter name="area" value="Basavanagudi"/>
      <classes>
      <class name="example.DemoA"/>
    </classes>
  </test>
</suite>
```

//Receiving Part

```java
package example;

import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import org.testng.xml.XmlTest;

public class DemoA
{
  @BeforeMethod
  public void precondition(XmlTest x)
   {
      String s=x.getParameter("state");
      Reporter.Log(s,true);
   }
  @Test
  public void testA(XmlTest x)
  {
     String c=x.getParameter("city");
     Reporter.Log(c,true);
  }
```

```java
      @AfterMethod
      public void postCondition(XmlTest x)
       {
          String a=x.getParameter("area");
          Reporter.log(a,true);
       }
      }
```

Q. How do you execute all the scripts simultaneously on multiple browsers?

Ans: using parallel option available in TestNg along with testNG parameter.

Ex: update precondition method of BaseTest class as shown below

```java
@BeforeMethod
      public void preCondition(XmlTest x)
      {
              String browser=x.getParameter("browser");
              Reporter.log("Browser is:"+browser,true);


              if(browser.equals("GC"))
              {

      System.setProperty("webdriver.chrome.driver","./exefiles/chromedriver.exe");
                              driver=new ChromeDriver();
              }
              else
              {
                      driver=new FirefoxDriver();
              }
      driver.manage().window().maximize();
      driver.get("http://localhost");
      driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
      }
```

Content of TestNG.xml

```xml
<<suite name="Suite" parallel="tests">
 <test name="TestGC">
<parameter name="browser" value="GC"/>
    <classes>
      <class name="scripts.VerifyBuildNumber"/>
      <class name="scripts.InvalidLogin"/>
      <class name="scripts.ValidLogin"/>
    </classes>
  </test>

  <test name="TestFF">
<parameter name="browser" value="FF"/>
    <classes>
      <class name="scripts.VerifyBuildNumber"/>
      <class name="scripts.InvalidLogin"/>
      <class name="scripts.ValidLogin"/>
    </classes>
  </test>
</suite>
```

When we execute TestNG suite file it will create two threads. Because we have specified **parallel** option as **tests a**nd we have 2 test blocks.

If first thread will execute all the scripts on Chrome browser, where as second thread will execute all the scripts on Firefox browser.

SELENIUM GRID

To run automation scripts on multiple browsers on multiple environments (remote computers) we can use Selenium Grid.

In Selenium Grid, we will have two types of systems:

1. **Hub** – it is the computer where framework is present and we start the execution from this computer, after the execution result is also stored in this system. Selenium grid has only one Hub.

2. **Node** – Node refers to remote computer on which actual execution happens. In Selenium grid there could be one or more node communicating with the hub . To implement this concept, we should perform following steps

    a. **Start the hub**

    b. **Start the node**

    c. **Execute Framework using RemoteWebDriver**.

a.   Start The Hub

1. Go to the system where framework is present.
2. Copy-Paste selenium jar file to the required location.

    (ex: D drive) rename it if it is required (s.jar).
3. Open the command prompt and execute following command

    java –jar d:\s.jar –role hub
4. It should display following message

    "Selenium Grid hub is up and running"

b.   Start the Node

1. Go to the remote system where the brower has to be opened and script should be executed.

    Node system should contain following software's:

    a. jdk

    b. Selenium jarfile

    c. firefox browser

    d. chrome browser with chromedriver exe file

2. Open the command prompt and execute following command

    java –jar e:\s.jar –role node –hub http://192.168.1.23:4444 –

    Dwebdriver.chrome.driver=e:/cd.exe

    (We can store it in bat file)

c.   Execute Framework using remotecontrol

1.   Go to precondition method of BaseTest class and update the code as shown below

Old code

driver=new ChromeDriver();

New code

driver=new RemoteWebDriver(DesiredCapabilities.chrome());

Old code

driver=new FirefoxDriver();

New code:

driver= new RemoteWebDriver(DesiredCapabilities.firefox());

Execute the framework using suitefile or .bat file

DAY 26

JENKINS

It is a Contineous Integraton Tool used by developers for managing build creation process, which includes following steps:

1. Download the latest source code.
2. Complile the code.
3. Compress the code.
4. Install the build.
5. Send a Email notification.

In order to execute the framework automatically after the installation of the build we should integrate framework with JenKins.

To implement the above concept we should perform following steps:

1. Install and configure JenKins.
2. Integrate Framework with JenKins.
3. Create the Build.

Install and Configure JenKins

1. Download **jenkins.war** file from http://jenkins-ci.org/ and copy it to required location (ex: D:)
2. Execute following command in the command prompt:

   java –jar d:\jenkins.war
3. It should display following messge:

   Jenkins is fully up and running
4. Open the browser and enter the url of the Jenkins ex: *http://localhost:8080/*
5. Click on **Create New Jobs**
6. Specify the Name
7. Select the first Radio button **Free Style Project**
8. Click **OK** and Click **Save**.

   Note: all the above steps done by Developers

Integrate Framework with JenKins. (Imp: by Automation Engineers)

1. Open the homepage of Jenkins and click on the **Job Name.**
2. Click on **Configure**
3. Click on **Advanced**
4. Select **Use Custom Workspace**
5. specify the path of Javaproject folder where batch file is present

Directory: *D:\BCSM6\Automation*

6. Click on Addd Build Step and select Execute windows batch command.

7. Specify the batch file name "RunMe.bat" and click Save.

Create the Build. by dev

1. Go to homepage of Jenkins

2. Click on **Job Name**

3. Click on **Build Now**

    Note 1: When developer clicks on Build Now Jenkins will create a build, installs the build and then it will start the framework execution.

    Note 2: Every time the build creation process started it will display link under Build History, which can be used to see the details.

    ex: click on the link present under build history and then click on Console output.

    We can view execution result of the Automation in Jenkins page itself.

FLOW DIAGRAM OF THE FRAMEWORK
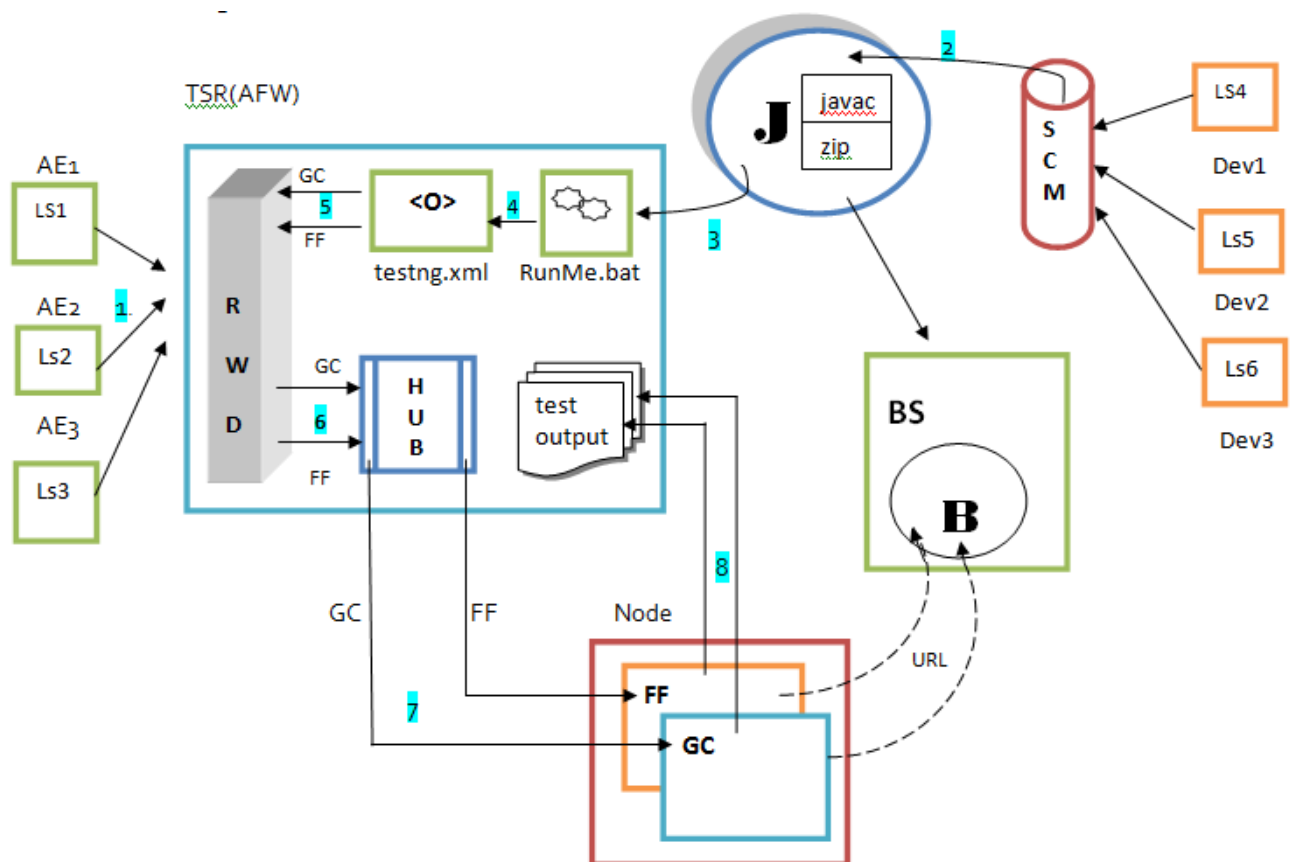
LS - Local System

AE -Automation Engineer

TSR > Test Script Repository

AFW - Automation Framework

Dev – Developer

SCM - Software Conf iguration Management

J- Jenkins,

B – Build

BS- Build Server

Steps:

1. Every day automation engineers will convert manual testcases into Automation scripts and store them in TSR.
2. Developer writes code in their local system and stores them in SCM and it is detected by Jenkins which will creates a build and installs it in Build Server.
3. After installing the build and it will execute the batch file.
4. Batch file will execute TestNg suite file (testng.xml).
5. Since the parallel option is specified as **tests** and we have 2 test blocks, it will create two threads: one for chrome and another for firefox browser.
6. Since we are using RemoteWebDriver class, all the scripts will be sent to hub.
7. Hub will forward it to suitable node.
8. Scripts will be executed in the node and results will be stored back in the **test-output** folder of framework.

ARCHITECTURE OF THE FRAMEWORK
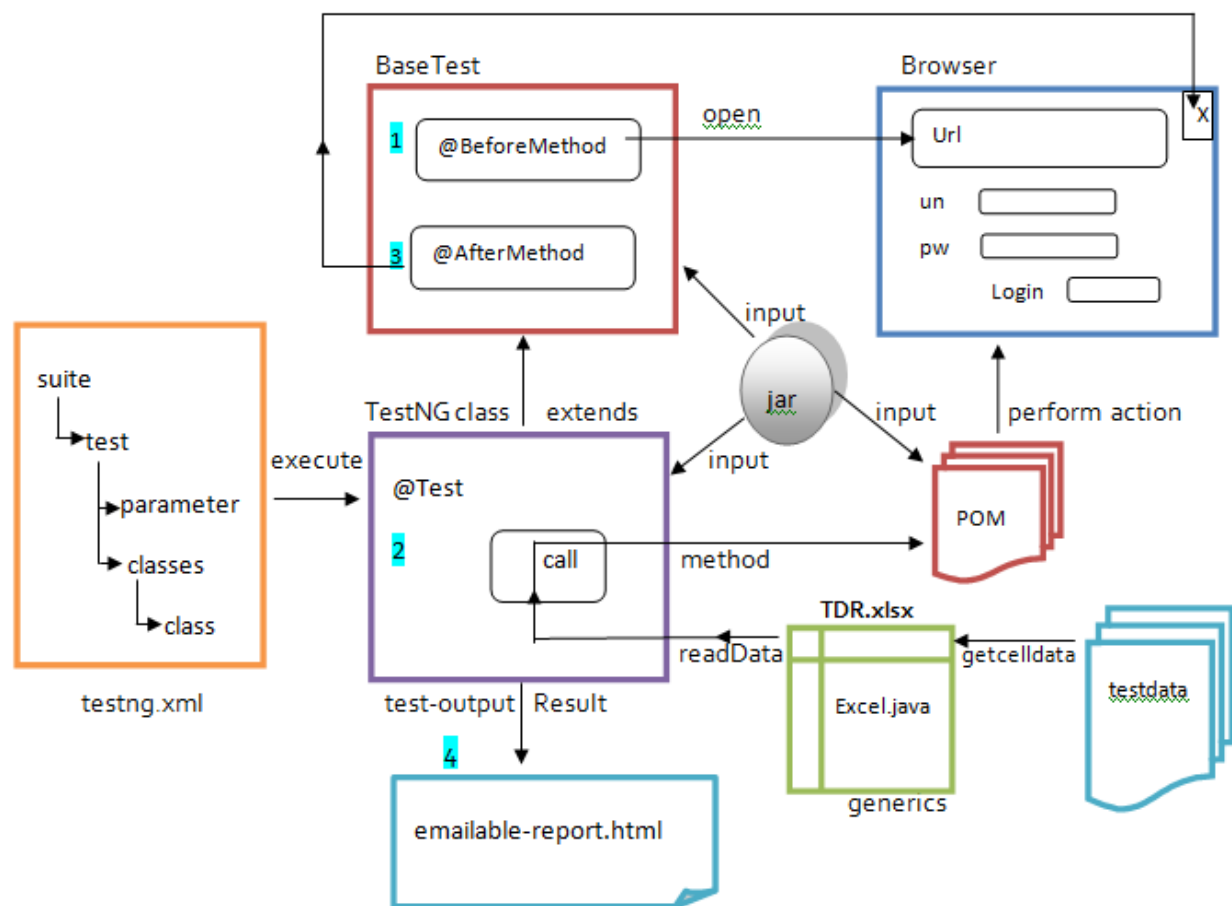
1. The framework is developed using TestNG, POM and Exel libraty. It is a combination of Data-Driven and Method-Driver framework, which we call it as Hybrid Framework.

2. The execution is controlled by TestNG suite file which has list of TestNG classes which are to be executed.

3. Each TestNG class has test method and also extends from BaseTest class which has @BeforeMethod and and @AfterMethod.

4. First @BeforeMethod is executed which opens the browser and enters the url.

    WebDriver driver=new FirefoxDriver();

    driver.get("");

5. After executing Before Method it will start the execution of testmethod. The testmethod takes the data from excel sheet and performs the action by calling the method present in POM class.

    ex:

    String un=Excel.getCellData(xlpath,sheet,1,0);

    LoginPage l=new LoginPage(driver);

    l.setUserName(un);

6. After executing testmethod, it will execute AfterMethod which closes the browser.

   driver.close();

7. After executing all the scripts it will generate the result it html format (test-ouput folder) of the framework.

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
TestGC	3	0	0	278,656		
TestFF	0	3	0	93,525		
Total	3	3	0	372,181		

DAY 27

HANDLING POPUP

With respect to Selenium we can categorize the popup into following types:

1. Hidden Division Popup
2. Alert and Confirmation Popup
3. File Upload Popup
4. File Download Popup
5. Child Browser Popup
6. Window Popup

1. HIDDEN DIVISION POPUP

Characteristics:
1. We cannot move the popup
2. We can Inspect the popup
3. It will be colorful

Solution:
To handle Hidden Division Popup we use **findElement()** method.
Ex: calendar popup, a type of hidden division popup

Write a script to select a date in Calendar Popup.

```java
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Demo
{
        public static void main(String[] args)
        {
                WebDriver driver=new FirefoxDriver();
                driver.manage().window().maximize();
                driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
                driver.get("http://www.yatra.com/");
                driver.findElement(By.id("BE_flight_depart_date")).click();
                driver.findElement(By.id("a_2016_3_19")).click();
        }
    }
```

Assg: Write a script to select today's date in the calendar:

```java
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Demo
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.get("http://www.yatra.com/");
        driver.findElement(By.id("BE_flight_depart_date")).click();
        driver.findElement(By.xpath("//td[contains(@class,'curent-date')]")).click();
    }
}
```

2. ALERT AND CONFIRMATION POPUP

Characteristics:

1. We can move the popup
2. We cannot inspect the popup
3. If the popup has Alert Symbol (!) it is called as Alert Popup
4. If the popup has Confirmation (?) Symbol, it is called as confirmation popup

    Note: Both are called as javascript popup.

Solution:

To Alert Popup we use **switchTo** and **alert** statement then we use following methods of Alert Interface.

1. getText() - get the msg present on the pop
2. accept() – to click on OK
3. dismiss() – to click on Cancel.

```java
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Demo
{
```

```java
        public static void main(String[] args)
        {
                WebDriver driver=new FirefoxDriver();
                driver.manage().window().maximize();
                driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
                driver.get("https://www.irctc.co.in/eticketing/loginHome.jsf");
                driver.findElement(By.id("loginbutton")).click();
                Alert alert=driver.switchTo().alert();
                String msg=alert.getText();
                System.out.println(msg);
                alert.accept();
                //after alert is closed control will be transferred back to page
                //alert.dismiss(); // we get NoAlertPresentException
        }
    }
```

3. FILE UPLOAD POPUP

Characteristics:

1. Clicking on **Browse** button will display a popup with the title file upload.

2. We can move the popup, but we cannot inspect it.

Solution

1. To handle file upload popup we specify Absolute path of the file as an argument for sendKeys() method.

```java
    import org.openqa.selenium.By;
    import org.openqa.selenium.WebDriver;
    import org.openqa.selenium.firefox.FirefoxDriver;
    class Demo
    {
            public static void main(String[] args)
            {
               WebDriver driver=new FirefoxDriver();
               driver.get("http://www.2shared.com/");
               driver.findElement(By.id("upField")).sendKeys("D:\\Book1.xlsx");
            }
     }
```

Note: Above code will not work if Attachment Icon is present instead of Browse button.

like in Gmail. In such cases we use **AutoIt** function.

Assg: Write a script to login to Naukri and upload CV.

```java
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Demo
  {
      public static void main(String[] args)
      {
            WebDriver driver=new FirefoxDriver();
            driver.manage().window().maximize();
            driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
            driver.get("http://www.naukri.com/");
            driver.findElement(By.id("login_Layer")).click();
            driver.findElement(By.id("eLogin")).sendKeys("abc@gmail.com");
            driver.findElement(By.id("pLogin")).sendKeys("xyz123");
            driver.findElement(By.xpath("//button[text()='Login']")).submit();
            driver.findElement(By.xpath("//a[text()='View Profile']")).click();
            driver.findElement(By.id("uploadLink")).click();
            driver.findElement(By.id("attachCV")).sendKeys("D:\\Resume.docx");
      }
  }
```

4.   FILE DOWNLOAD POPUP

Characteristics:

1.   We can move the popup
2.   We cannot inpsect the popup
3.   popup will have two radio button: **Open with** and **Save file**

Solution

To handle File Download Popup, we use **setPreference()** of **FirefoxProfile class.** Which will
 programmatically change the settings of the browser so that it will download the file automatically
 without displaying the popup.

We can refer following website to change any settings of Firefox:

http://kb.mozillazine.org/About.config_entries

```java
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxProfile;
class Demo
{
    public static void main(String[] args)
    {
        FirefoxProfile profile=new FirefoxProfile();
        //if file is .zip never display popup download it directly
        String key="browser.helperApps.neverAsk.saveToDisk";
        String value="application/zip";
        profile.setPreference(key,value);
        //open firefox with above setting

        WebDriver driver=new FirefoxDriver(profile);
        driver.get("http://docs.seleniumhq.org/download/");
        String xp="//td[text()='Java']/../td[4]/a";
        driver.findElement(By.xpath(xp)).click();
    }
}
```

Note: We cannot handle file download popup display in other browser. We should use **AutoIt.**

5. CHILD BROWSER POPUP

Characteristics:

1. We can move the popup
2. We can inspect the popup
3. It will have minimize and maximize button with address bar.

Q1: What is the difference between close() and quit() method

Ans: **close()** method closes current browser, whereas **quit()** method closes all the browser

Q2. Write a script to count no. of browsers open by selenium.

Q3. Write a script to print window handle of all the browsers.

Q4.. Write a script to print title of all the browsers

Q5. Write a script to close all the browsers without using quit() method.

```java
import java.util.Set;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Demo
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://www.naukri.com/");
        Set<String> allWH=driver.getWindowHandles();
        int count=allWH.size();
        System.out.println(count);
        for(String wh:allWH)
        {
            driver.switchTo().window(wh);
            String title=driver.getTitle();
            System.out.println(title);
            driver.close();
        }
    }
}
```

Assgn:

2. Write a script to close only child browser.

```java
import java.util.Set;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Demo
{
  public static void main(String[] args)
   {
      WebDriver driver=new FirefoxDriver();
      driver.get("http://www.naukri.com/");
      String mTitle=driver.getTitle();
      Set<String> allWH=driver.getWindowHandles();
      for(String wh:allWH)
      {
       driver.switchTo().window(wh);
       String title=driver.getTitle();
       if(mTitle.equals(title))
        {
         System.out.println("Main Browser is "+mTitle);
        }
      else
        {
         System.out.println(title+" - child browser is closed");
         driver.close();
        }

      }
    }
  }
```

2. Write a script to close specified browser.

```java
import java.util.Scanner;
import java.util.Set;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Demo
{
    public static void main(String[] args)
    {
        WebDriver driver=new FirefoxDriver();
        driver.get("http://www.naukri.com/");
        Set<String> allWH=driver.getWindowHandles();
        System.out.println("List Of Browsers:");
        for(String wh:allWH)
        {
            driver.switchTo().window(wh);
            String title=driver.getTitle();
            System.out.println(title);
        }
        System.out.println();
        //Close specific browser
        System.out.println("Enter Specific browser name:");
        Scanner sc=new Scanner(System.in);
        String browser=sc.next();
        for(String wh:allWH)
        {
            driver.switchTo().window(wh);
            String title=driver.getTitle();
            if(browser.equals(title))
            {
                driver.close();
            }
            else
            {
                driver.switchTo().window(wh);
            }
        }
    }
}
```

6. WINDOW POPUP

If the popup displayed in application is not Hidden Division, Alert and confirmation, File upload, File Download and child Browser popup. Then it is called as Window Popup. In Selenium we cannot handle window popup.And to handle the popup we should use **autoIt.**

autoIt is a free window based application automation tool, which can be downloaded from following website:

url: **https://www.autoitscript.com/site/autoit/downloads/**

File: autoit-v3-setup.exe

To install it: Double click on the downloaded setup file, follow the default instructions given in the setup wizard (next.. next.. till finish).

Steps to inspect element in Autoit:

1.   Go to Start > All Programs > Autoit > select Window Info
2.   Drag and drop "finder tool" on required element or popup.

Steps to write code in AutoIt and Executing it:

1.   Go to Start > All Programs > AutoIt > select Script Editor
2.   Write the code as below

     WinWaitActive("Calculator")

     WinClose("Calculalor")
3.   Go to file and select Save
4.   Navigate to required location ex: D specify the name ex: script1 and click Save extension will be .au3
5.   Go to tools and select Compile and it creates new .exe file (same location)
6.   Double click to execute it

To run the complete code programmatically in java we should write the code as shown below

```java
package example;
import java.io.IOException;
class Demo
{
        public static void main(String[] args) throws IOException
        {
                Runtime.getRuntime().exec("D:\\Window\\system32\\calc.exe");
                Runtime.getRuntime().exec("D:\\script1.exe");
        }
}
```

Q. Write a script to perform following steps:

1. Open firefox browser

2. Go to download page of selenium

3. Click on link 2.52.0

4. Activate the window popup

5. Press left arrow key so that control will be transferred to Save File button

6. Press Enter key.

Solution

AutoIt script:

```
WinActivate("Opening selenium-server-standalone-2.52.0.jar")
Sleep(1000)
Send("{LEFT}")
Sleep(1000)
Send("{ENTER}")
```

Selenium script:

```java
package example;
import java.io.IOException;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class Exam
{
      public static void main(String[] args) throws IOException
      {
            WebDriver driver=new FirefoxDriver();
            driver.get("http://docs.seleniumhq.org/download/");
            driver.findElement(By.linkText("2.52.0")).click();
            Runtime.getRuntime().exec("D:\\script1.exe");

      }
   }
```

EXCEPTIONS:

1. InterruptedException (Java checked)
2. IllegalStateException (Java Unchecked)
3. NoSuchElementException (Selenium Unchecked)
4. IOException (Java checked)
5. TimeoutException (Selenium Unchecked)
6. UnreachableBrowserException (Selenium Unchecked)
7. InvalidElementStateException (Selenium Unchecked) – when we try to enter data in a disabled textbox.
8. IndexOutOfBoundsException (Java Unchecked)
9. NumberFormatException (java Unchecked)
10. NoSuchFrameException (selenium unchecked)
11. UnexpectedTagNameException (selenium unchecked)
12. StaleElementReference Exception (selenium unchecked)
13. TestNGException (TestNG unchecked
14. EncryptedDocumentException (Java unchecked)
15. Invalid FormatException (Java unchecked)
16. NoAlertPresentException (selenium unchecked)
17. SessionNotFoundException (Selenium unchked)
18. NoWindowException (Selenium Unchecked)

LIMITATIONS OF SELENIUM WEB DRIVER:

1. We can automate only web applications
2. We can't minimize the browser
3. Using selenium we can't perform any action on the browser which is already opened. Every time it will open the new browser.
4. We can take screenshot only in png format. Other format is not supported.
5. We can't take screenshot of popup, multiple browsers and specific area.
6. We can't specify the password in encrypted format.
7. We cannot handle file upload popup if it has attachment icon instead of Browse button.
8. We cannot handle file download popup in the browsers other than Firefox
9. We cannot handle new tab in selenium.
10. We cannot handle window popup.