Important Java topics required for Selenium.

1. Class and Objects
2. Static and non-Static methods
3. Conditional Statements like if and if-else
4. Looping Statements like while, do while, for, for each
5. OOPS Concept like Inheritance, polymorphism, encapsulation, abstraction
6. Collection Framework - List, Set and Maps
7. Exception Handling
8. Methods of String Class
9. File Handling.

Required Softwares.

1. JDK (Java Development Kit)
2. Eclipse, Browser (Chrome)
3. Selenium Jar File
4. Driver Executable (chromedriver. exe)
5. Any web application.

Selenium Introduction :

Selenium is a free and open Source web application automation tool (Functional testing web application automation tool.

1. From where we can download the Selenium for free?

https: // www. selenium. dev / downloads /

2. What is the latest stable version of Selenium?

→ 3.141.59

3. What are the flavors of Selenium?

→ Selenium Core, Selenium IDE (Record & play), Selenium RC (Remote Control), Selenium WebDriver, Selendroid, Appium, Watium.

4. What are the browsers supported by Selenium?

→ Google Chrome, FireFox, Internet Explorer, Safari, Edge, Opera, HTML unit, phantom JS (Headless browser)

5. What are the ~~langu~~ languages supported by Selenium?

→ Java, C#, Ruby, perl, python, PHP, JavaScript, Objective C, NodeJS, Haskell, R, TCL, Elixit, DART.

6. Using Selenium what type of application can we automate?

→ Only web applications.

7. Is Selenium is a open source? how?

→ Yes, Selenium is an open source. It means we can access source code of Selenium software itself. (we can download & customize as per our requirement)

8. What are the ~~browsers~~ supported OS supported by Selenium

→ windows, Linux, Mac, other unix flavours like Android, IOS, ubuntu .... etc.

9. Which OS is not supported by Selenium?

→ unix.

10. Can we do performance testing using Selenium?

→ No, But we can integrate Selenium with Jmeter.

11. What type of test Cases we automate ?

→ Regression Test Cases.

12. Do we Automate Integration testing ?

→ Yes. Different types of test Cases which is part of regression.

13. Do we automate Negative Test Cases ?

→ Yes.

14. Which test Cases are automated first ?

→ Smoke test Cases. (Sanity, Dry Run (Automation), Build verification Testing (BVT), Skim (UAT)).

15. Is 100% Automation is possible ? If No why ?

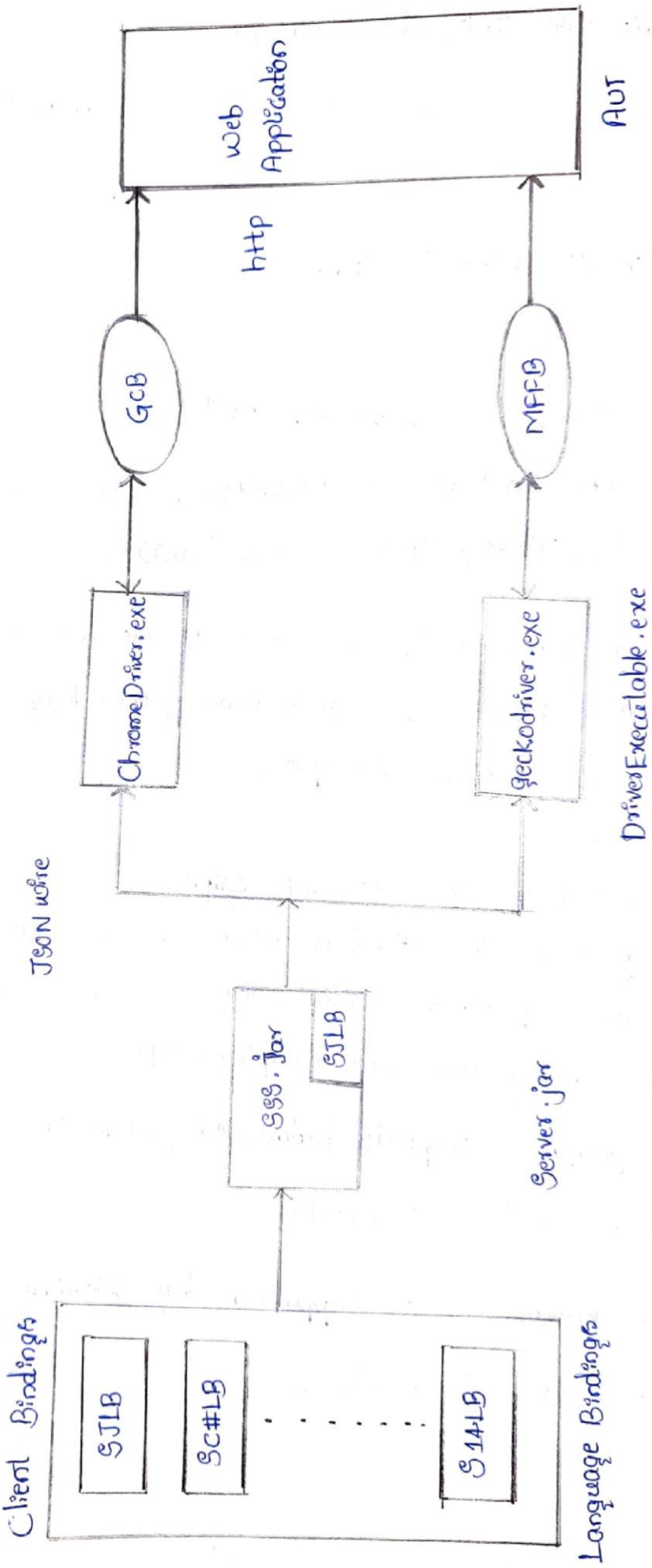
→ No, Because we don't have technology to automate the features or it may. Examples.

- 3D Games
- Verification of audio, video Clips.
- Capturing the attendance using access Cards & biometric Scanners, Entering product details using barcode scanner, payment through Credit Card Swiping, OTP.
- Captcha (Completly Automated public Turing test to tell Computers & humans Apart)

16. Which is browser & OS Supported by Selenium IDE ?

→ FireFox & windows.

Architecture of Selenium :



Selenium supports 14 different coding languages like Java, python, Ruby etc. these are called as language Binding or Client Binding.

These Client Binding communicates with Selenium Stand alone. Then Server performs the action on the browser with the help of driver executables. In order to perform action it uses JSON wire protocol (Java Script object Notation).

Selenium Server internally contains Selenium Java Client binding also, hence while installing Selenium we use only Server.jar file & driver executables.

Steps to install Selenium :

1. Download Selenium Server jar file & browser specific driver executable file in the download page of Selenium website.

2. Extract the driver executable file (unzip the file).

3. In eclipse create a new java project & create 2 new folders for driver & jar inside the java project.

4. Copy & paste driver executable file into driver folder.

5. Similarly copy & paste Selenium jar file into jar folder.

6. Add jar file to build path.

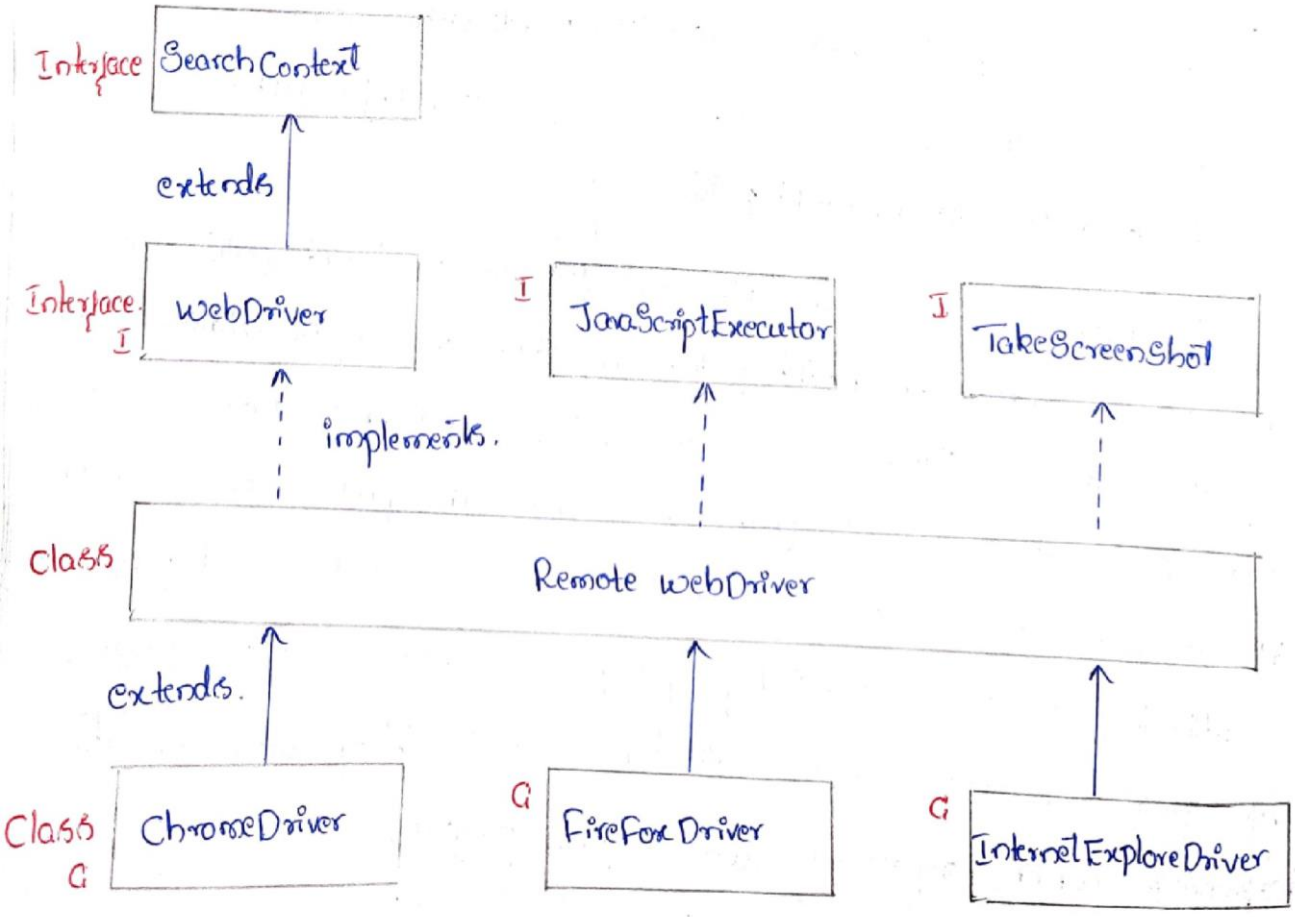
7. Note : Don't add driver.exe file to the build path since jar file is added to build path. we can use import statement to import required class of Selenium. Eclipse will not allow us to add exe file to the build path. To specify the path either we add it manually to environment variable or we should add it programmatically using System.Set property statements. while specifying the path we use relative path by specifying '.' operator at the beginning which represent current java project.

8. We should set the path before opening the browser or else we will get illegal state exception.

9. For easy maintenance we set the path of all the driver executables in static block as below.

```
→ public class Demo {
    Static {
    System.setproperty ("webdriver.Chrome.driver", "./driver/
                                                ChromeDriver.exe");
    Public Static Void main (String [] args) {
    ChromeDriver driver = new ChromeDriver ();
    driver.get ("www.google.com");
    driver.Close ();
    }
  }
}
```

Architecture of Selenium WEBDriver.



The Supermost interface is Search Context. which is extended by webdriver interface.

webdriver interface is implemented in Remote webDriver Class.

Remote webDriver Class implements other interfaces also such as Java Script executor, Take Screen Shot etc.

All the browser specific classes such as Chrome Driver, FireFox Driver, Internet Explorer Driver etc extends RemotewebDriver Class.

Note : Apply filter : Eclipse will display methods of object class also when we press Ctrl button after any reference Variable. To hide it go to window tab in eclipse → preference → java → appearences → type filters, click on add button & type java.lang.object then click ok & click apply & Close.

17. How Selenium performs action on the browser ?

By Calling the native methods of the browser.

18. which protocol is used by Selenium to interact / communicate with the browser ?

JSON wire protocol (Java Script object Notation)

Methods of Search Context :

1. find Element (By arg[]) : WebElement
2. find Elements (By arg[]) : List <WebElement>

Take Screen Shot Methods :

1. getScreenShotAs ()

WebDriver Methods :

1. Close ()
2. get ()
3. get CurrentUrl ()
4. get pageSource ()
5. getTitle ()
6. getwindowHandle ()
7. getwindowHandles ()
8. manage ()
9. navigate ()
10. quit ()
11. SwitchTo ()

Java Script Executer Method :

1. ExecuteAsyncScript ()
2. executeScript ()

Web Element Methods :

1. Clear ()
2. Click ()
3. getAtribute ()
4. getCSSValue ()
5. get Location ()
6. getRect ()
7. get Size ()
8. getTagName ()
9. getText ()
10. isDisplayed ()
11. isEnabled ()
12. isSelected ()
13. SendKeys ()
14. Submit ()

What is upcasting ? is it used in Selenium ? why ?

Converting Subclass object into Supertype is called upcasting.

Example : webDriver driver = new ChromeDriver ();

In Selenium we use upcasting so that we can execute same script on any browser.

Explain webDriver driver = new ChromeDriver ();

webDriver is an interface,

driver is an reference variable.

= is an assignment operator.

New is a keyword to create an object

ChromeDriver () is a constructor to initialize the object.

; is the statement delimiter or used to end the statement.

```
→ public class DemoA {
  public static void test (webDriver driver) {
    driver.get ("https://www.google.com/");
  System.out.println (driver.getTitle());
  driver.close ();
  }
}

public class DemoB {
static {
System.setproperty ("webdriver.chrome.driver", "./driver/
                                    Chromedriver.exe");
System.setproperty ("webdriver.gecko.driver", "./driver/
}                                   geckodriver.exe");
```

```
public static void main (String [] args) {
    WebDriver driver = new ChromeDriver ();
    DemoA. test (driver);
    WebDriver driver1 = new FirefoxDriver ();
    DemoA. test (driver 1);
  }
}
```

How do you enter url without using get method ?

using navigate().toUmethod.

driver.navigate().to ("http://www.google.com");

what is the difference between navigate() & get() method ?

By using get method we can enter the url only, whereas by using navigate we can enter the url, click on back, forward & refresh functions.

what is the difference between get and to method ?

There is no difference between get and to method. 'to' method internally calls 'get' method, hence there is no difference between to & get method. Both are used to enter the url.

⟹

```
Class NavigateClass {
Static {
System. Setproperty ("webdriver. ChromeDriver", "./driver/
                                    ChromeDriver. exe");
}

public static void main (String [] args) {
```

```
webDriver driver = new ChromeDriver();
driver.navigate().to("http // www.google.com");
thread.Sleep(3000);
driver.get("https // www.gmail.com");
thread.Sleep(3000);
driver.navigate().refresh();
thread.Sleep(3000);
driver.Close();
}
}
```

How to close browser without using Close method?

By using quit method.

Ex : driver.quit();

What is the difference between Close() & quit() methods?

Close() method Closes Current Browser where as quit() Closes all the Browsers.

Write a Script to open google.com & Verify that title is Google & also verify that it is redirected to google.co.in.

```
Public Class Demo1
{
Static {
System.Setproperty("webdriver.ChromeDriver", "./driver/
                                ChromeDriver.exe");
}
```

```
public Static void main (String [] args) {
 webDriver driver = new ChromeDriver ();
 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);
 }
}
}
```

Write a script to delete all the cookies present in the browser & maximise the browser.

```
Package qsp;
public class CookiesAndMaximise {
 Static {
System.Setproperty ("webdriver.Chrome.driver", "./driver/Chromedriver
                                                .exe");
 }
 public Static void main (String [] args) {
    WebDriver driver = new ChromeDriver ();
    driver.manage().deleteAllCookies();
    driver.manage().window().maximize();
    driver.Close();
 }
}
```

write a program to print Html Source Code (Script) of the webpage.

```
Package qsp;
Public class HtmlCode {
Static {
System.Setproperty ("webdriver.chrome.driver", "./driver/
                                        Chromedriver.exe")
}
Public Static void main (String [] args) {
    WebDriver driver = new ChromeDriver ();
    driver.get ("https://www.google.com/");
    String htmlCode = driver.getpageSource();
    System.out.println (htmlcode);
    driver.Close();
 }
}
```

Web Element : Anything present on a webpage is called web element. web element are created using HTML language (Hyper Text Markup language)

Each HTML elements contains three things

"Tag", "Attribute" and "Text"

Example : Html of a login button is

 login </ div>

In the above HTML code, div is the tag, id is attribute name d1 is attribute value & login is visible text.

In order to see the HTML code of the required element right click on that element & select the option 'inspect', if right click is disabled on the page then,

Cntl + shift + i, or press f12, it will display developers tool bar.

In order to inspect another element, click on inspect button & then click on required element.

In Selenium, before performing any action such as clicking, typing, selecting etc... we should find the elements using locators.

Locators : Locators are used to find the elements. In Selenium there are 8 types of locators, & all of them are static method present in 'By' class

By is an abstract class

Ex : By.tagName ();

1. tagName ()
2. id ()
3. name ()
4. ClassName ()
5. linkText ()
6. partialLinkText ()
7. Css Selector ()
8. Xpath ()

1. TagName :

Write a program to click on the google link present in static web page by using tag name locator.

Write what is the return type of find element ?

→ web element.

What is the argument accepted by find element method ?

→ By type

If specified locator is not matching with any of the elements, then what find element does ?

→ It throws no such element exception.

If the specified locator is matching with multiple elements what find element does ?

→ It takes the first ab available element & it returns the address of first matching element.

Write a program to click google link elements by using tagname, id, name, Class name.

```
public Class TagName {
Static {
    System.setproperty ("webdriver.chrome.driver",
                        "./driver/Chromedriver.exe");
}
```

```
Public Static void main (String [] args) throws InterruptedException {
  webDriver driver = new ChromeDriver();
driver.get ("file:c:\\users\\new user\\Desktop\\Selenium");
  // webElement e = driver.findElement (By.tagName ("a"));
  // System.out.println(e); // gives the address of the element
  // e.click();
  driver.navigate().back();
  driver.findElement (By.tagName ("a")).click(); // Code optimi-
                                                  -zation using tagname.
  Thread.Sleep (2000);
  driver.navigate().back();

  driver.findElement (By.id ("d1")).click(); // using id
  Thread.Sleep (2000);
  driver.navigate().back();

  driver.findElement (By.name ("n1")).click(); // using name
  Thread.Sleep (2000);
  driver.navigate().back();

  driver.findElement (By.ClassName ("c1")).click(); // class
                                                       Name
  Thread.Sleep (2000);
  driver.navigate().back();

  driver.close();
  }
}
```

Link text and partial Link text :

Both link text & partial Linktext locators can be used to find the link. If we try on any other type of element, we get no such element exception.

Ex : driver. findElement (By. LinkText ("Google")). Click ();

If the text of the link is Changing, partially link text.

Example : HTML Code of a link is,

<a ....> Inbox (25) </a>

Selenium Code :

driver. findElement (By. partialLinkText ("Inbox")). Click ();

Limitations of partial LinkText :

1. Elements should be a link.

   Ex : <Span> Inbox (25) </ Span> // Cannot be used

2. Text should be partially Changing.

   Ex : <Span> 25 </ Span> // we cannot use partial link.

CssSelector : (Cascading Style Sheet)

CssSelector is one of the locator in Selenium.

Syntax : tag [Attribute name = 'Attribute value']

Example : a [id = 'd1']

a [name = 'n1']

a [class = 'c1']

a [href = 'http : // jspiders. com / ']

These are all called as Css Expression.

Limitations of CSS Selector:

we can check CSS expression in the browser by using following steps.

1. Inspect any element.
2. press Cntrl + f
3. type the above expression.

1 of 1 ⟶ One matching element.

1 of 3 ⟶ multiple matching element.

0 of 0 ⟶ no matching element.

```
public class LinkTest {
    Static {
System.Setproperty ("webdriver.Chrome.driver", "./driver/
                                    Chromedriver.exe);
}

public Static void main (String[] args) {
  webDriver driver = new ChromeDriver ();
driver.get ("file:C:\/users/Samskruthi/OneDrive/Desktop/
                                    Demo.html");
driver.findElement (By.CssSelector ("a [id = 'd1']")).Click ();
driver.navigate ().back ();
driver.findElement (By.CssSelector ("a [name = 'n1']")).Click ();
driver.navigate ().back ();
driver.findElement (By.CssSelector ("a [Class = 'c1']")).Click ();
driver.navigate ().back ();
```

```
driver.FindElement(By.CssSelector("a[href='http://www.qspiders.com/']")).click();
driver.Close();
}
}
```

Note: If we do any syntax mistake while writing CSS expression or Xpath expression, then we get invalid Selector ~~expression~~ Exception.

Xpath :

It is the path of the element in a HTML tree is called as Xpath. while writing Xpath it starts with "." which also represents Current webpage or HTML document. using "." is not mandatory.

To navigate from parent element to Child element we use single "/".

Xpath Expression : ./html/body/a or /html/body/a

Xpath in Selenium : driver.findElement(By.Xpath("/html/body/a").Click();

The above Xpath are called as absolute Xpaths.

Types of Xpath :

1. Absolute Xpath
2. Relative Xpath
   a. Xpath by Attribute
   b. Xpath by text() function.

c. Xpath by Contains () function.

d. Traversing in Xpath.

e. Independent - Dependent Xpath

f. Xpath by group index.

Absolute Xpath :

Starting from the html tag to the required or desired element is called as absolute Xpath.

```
html
└→ body
    1 ├→ div
      │   1 ├→ a - link 1
      │   2 └→ a - link 2
    2 └→ div
        1 ├→ a - link 3
        2 └→ a - link 4
```

Xpath by index :

/ html / body / div [1] / a [1] ⟶ link 1

/ html / body / div [1] / a [2] ⟶ link 2 (All are absolute Xpath)

/ html / body / div [2] / a [1] ⟶ link 3

/ html / body / div [2] / a [2] ⟶ link 4

/ html / body / div [2] / img [1] ⟶ image

/ html / body / div / a ⟶ link 1, 2, 3, 4

/ html / body / div / a [1] ⟶ link for 1 & 3

/ html / body / div / a [2] ⟶ link for 2 & 4

/ html / body / div [1] / a ⟶ link for 1 & 2

/ html / body / div [2] / a ⟶ link 3 & 4.

Xpath Index :

In Xpath, we can use index which starts from 1, if there is another element under the same parent with the same tag then index become 2 & so on.

Relative Xpath :

/ ⟶ Child

// ⟶ Descendent

// div [1] / a [1] ⟶ link 1

// div [1] / a [2] ⟶ link 2

// div [2] / a [1] ⟶ link 3 (all are relative Xpath)

// div [2] / a [2] ⟶ link 4

// img ⟶ image // div [2] / img // div / img

// a [2] ⟶ link 2 & 4

// div [1] / a ⟶ link 1 & 2

// div [2] / a ⟶ link 3 & 4

// a ⟶ link 1, 2, 3, 4

In relative Xpath, it starts with "//" which represents descendent (Child, grand Child, great grand child ... etc)

What is the difference between / & //

/ represent ⟶ Child

// represent ⟶ Descendent.

what is the difference between //a & //table//a

// a matches with all the links which are present anywhere on the webpage.

//table//a matches all the links which are inside the table.

write the Xpath for all the links and all the images

//a|//img ⟶ all the links and all the images

link 1 & 4 ⟶ //div[1]/a[1]|//div[2]/a[2]

link 2 & 3 ⟶ //div[1]/a[2]| //div[2]/a[1]

link 1, 2 & 3 ⟶ //div[1]/a | //div[2]/a[1]

1. Xpath with attribute :

we can also use attribute of the element while writing the Xpath.

The Syntax is //tag[@AN='AV'] ⟶ path with attribute

AN is Attribute Name & AV is Attribute Value.

//tag[@AN1 = 'AV1' and @AN2 = 'AV2']

//tag[@AN1 = 'AV1' or @AN2 = 'AV2']

//tag[not(@AN = 'AV')]

write an Xpath to identify following elements

1. all the text box ⟶ //input[@type = 'text'] both A1 & B1 is selected.

2. all the buttons ⟶ //input[@type = 'button']

3. all the check box ⟶ //input[@type = 'Checkbox']

4. all the text box & button → // input [@type = 'text' or @type = 'button'] → o/p A1, B1, A2, B

5. only first text box → // input [@type = 'text' @value = 'A']

6. only Second text box → // input [@type = 'text' @value = 'B']

7. First text box & first button:

// input [@value = 'A'] — A1 & A2

8. only Selected check box:

// input [@type = 'checkbox' and @checked]

or

// input [@checked]

9. unselected Check box:

// input [@type = 'Checkbox' and not (@checked)]

or

// input [not (@checked) and @type = 'checkbox']

2. Xpath with Text function:

In Xpath we can also use text (visible text).

Syntax is: // tag [text() = 'text value']

html Code of login button is:

< div > Login </ div >

Xpath is:

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

hlm

3. Xpath with Contains : If text value is partially Changing we Can use Contains () function on the Xpath. because in this type, Version or value Changes as an when new Version releases.

usually if you see any number element in visible text it is usually dynamic in nature, it is likely to Change in future.

Syntax : // tag [contains (text() , 'value')]

html code : <nobr> actiTIME 2020.3 online </nobr>

Xpath : // <nobr> [Contains (text (), 'actiTIME')]

How do you identify element without using partial link text ?

By using Xpath with Contains function.

Ex : html Code : <a> Inbox (25) </a>

Xpath : // a [Contains (text (), 'Inbox')]

Note : we can use Contains function for attribute also the Syntax is,

// tag [Contains (@AttributeName , 'AttributeValue')]

Ex : html Code :

< img src = " /img/ default / login/ logo .... hash = 12365456 4">

Xpath : // img [Contains (@src, 'logo')]

4. Xpath by traversing:

Navigating from one element to another element is called as traversing.

There are two types of traversing.

1. Forward traversing.

2. Backward traversing.

Navigating from parent element to child element is called as forward traversing.

Navigating from child element to parent element is called as backward traversing.

Sample html table code web Table.

```
<table border = "1">

<tr>
<td> java </td>
<td > 100 </td>
</tr>

<tr>
<td > Selenium </td>
<td > 300 </td>
</tr>

</table>
```

<tr> : table row

<td> : table data.

Example for forward traversing [From html to Selenium)

/ html / body / table / tbody / tr[2] / td [1] → absolute Xpath.

// td [. = 'Selenium'] → relative Xpath.

Example for backward traversing [From Selenium to html]

// td [. = 'Selenium'] / .. / .. / .. / .. / ..

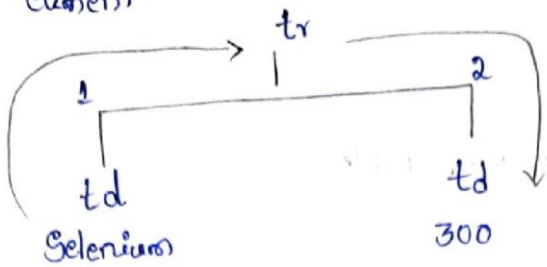
(/ .. will move cursor upto one level same as in Cmd, Cd. :)

5. Independent Dependent Xpath:

If the value is changing completely then we handle it using independent Dependent Xpath.

Here we use the Xpath using backward & forward traversing

we always starts from Independent element and ends with Dependent element.



// td [. = 'Selenium'] / .. / td [2]

Steps to Construct html tree and write the Xpath:

Identify the independent & Dependent element.

Inspect independent element & place the mouse pointer on the Source of independent element.

Note down the Html Code of independent element.

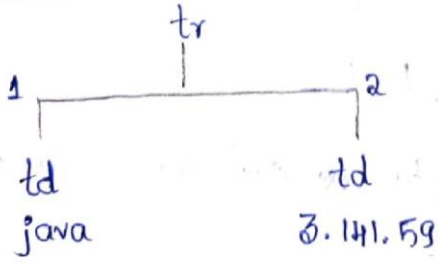
Move the mouse pointer in the upward direction in step by step till it highlights both independent & Dependent element, it will Common parent.

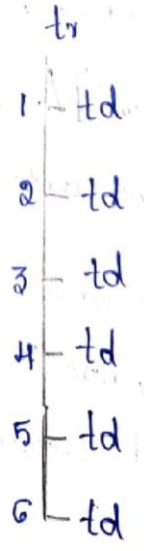
Add the path Common parent above the Code independent element

use down arrow key to navigate from Common Parent to dependent element.

Add its path below the Common parent.

write an Xpath to identify Stable version of java present in the web table.





Xpath: // td [text () = 'java'] / .. / td[2]

Xpath for Ruby download link :

~~//td [text () = 'ruby'] ../td [6] /../ a [1]~~

//td [text () = 'Ruby'] ../td [5] / a [1]

write an Xpath to identify the status of testing present in type of work web table in actitime application under settings.

//a [. = 'testing'] /.. /../ td [2]

write an Xpath to identify set by default link for manufacturing type of work.

//a [text () = 'manufacturing'] /../../ td [5]/a

6: Xpath by Group Index :

html

└ body

1 → div

1 → a → A

2 → a → B

2 → div

3 → a → C

4 → a → D

(//a) [last () - 1] → C.

(//a) [last ()] → D

(//a)[1] | (//a)[last ()] → AD

//a → ABCD

//a[1] → AC

//a[2] → BD

(//a) [1] → A

(//a) [2] → B

(//a] [3] → C

(//a)[4] → D

(//a[1])[2] → C

(//a[1]) [1] → A

(//a[2])[1] → B

//a [2] → BD

(//a [2])[2] → D

(//a) [1] → A

//a [1] → AC

Xpath Array

1	A
2	B
3	C
4	D

Example : price of Second Selenium book.

Selenium	100
Selenium	100

Xpath : (//td [text() = 'Selenium'])[2]/../td[3

Html Code :

```
<table border = "1">
<tr>
<td> Selenium </td>
<td> 100 </td>
</tr>
<tr>
~~<tr>~~
<td> Selenium </td>
<td> 200 </td>
</tr>
</table>
```

1. write an Xpath to identify the price of 1st Soch Kurta present in myntra.com
2. write an Xpath to identify the price of HRX by hritik roshan t-shirt present in myntra.com
3. write an Xpath to identify the price of Cherokee jeans present under Kids Section.
4. WAXP to price of MI Smart Band 4 in flipkart.com.

1. (//h1 [. = 'Soch'][1]/../p[@ Class = 'pdp-discount-container']
2. (//h1 [. = 'HRX by Hrithik Roshan'])[1] /../p [@ class = 'pdp-discount-Container']
3. (//h1 [. = 'Cherokee'])[1]/../p[@ Class = 'pdp-discount-Container']
4. //div [. = 'Mi SmartBand 4'] /../../div [@ class = 'col col-5-12-207wAb']

Xpath By Group Index :

while writing Xpath Expression we can write the Expression within the brace & we can specify the index outside the brace which is called as group index.

(//a) Xpath Array.

1	A
2	B
3	C
4	D

when we use group index first it will execute expression present inside the brace, in this examples first it will execute //a & it will store matching elements i.e A, B, C, D links into xpath array where index starts from 1.

Then it will identify the matching element based on the index, which is specified outside the brackets, in this example it matches with first link A.

write an Xpath to identify oneplus 7T mobile phone present in amazone.in

Xpath :⇒ (//span[contains(Text(), 'Oneplus 7T')])[1]/../../../../../../div[2]/div[1]/div/div[1]/div[2]/div/a/span[1]/span[1]

⇒ (//span[contains(Text(), 'oneplus 7T')])[1]/../../../../../../

Note: we can use Contains function to handle 'Non Breakable Space' also.

How do you handle if there is 'Non Breakable Space' between the Strings?

Ex: <button> Sign   in </button>

Xpath:

//button[contains(text(),'sign')] or //button[contains(text(),'in')]

Q. write an Xpath to identify the price of iphone 11 pro in amazon.

Q. write a Script to remove the text present in the text box.

```
Public Class RemoveText {
Static {
System.Setproperty("webdriver.Chrome.driver", "./driver/
                                Chromedriver.exe");
}
Public Static void main(String [] args) {
webDriver driver = new ChromeDriver();
driver.get("file:C:\users\Samskruthi/onedrive/desktop/input.html")
driver.findElement(By.Xpath("//input[@type='text']) [1]"))
                                .Clear();
driver.close();
}
}
```

write a Script to Check whether actiTIME logo is displayed or facebook not.

```
public class LogoIsDisplayed {
Static {
System.Setproperty ("webdriver.Chrome.driver", "./driver/
                                    Chromedriver.exe");
}
public Static void main (String [] args) {
webDriver driver = new ChromeDriver();
driver.get ("https://www.facebook.com/");
boolean Logo = driver.findElement(By.Xpath ("//img[
                contains (@Class, 'fb')]")).isDisplayed();

if (Logo == true)
{ System.out.println ("Logo is displayed so pass");
}
else {
 System.out.println ("Logo is not displayed so fail");
 }
 driver.Close();
 }
}
```

write a Script to Check whether Login button is enabled or not for facebook.com

```
public Class LoginButtonDisplayed {
Static {
 System.Setproperty ("webdriver.Chrome.driver", "./driver
                                    /chromedriver.exe");
}
```

```
public static void main (String[] args) {
    WebDriver driver = new ChromeDriver ();
    driver.get ("https://www.facebook.com/");
    boolean loginbutton = driver.findElement (By.id ("u_0_b"))
                                .isEnabled ();
    if (loginbutton == true)
    {
     System.out.println (" Login button is Enabled so true");
    }
    else {
     System.out.println (" Login button is not Enabled so false");
    }
    driver.close ();
    }
}
```

write a Script to check whether Checkbox is Selected or not for actiTIME application.

```
Public class CheckBoxSelect {
Static {
 System.setproperty (" webdriver.Chrome.driver ", "./
                                driver/Chromedriver.exe");
 }
Public Static void main (String [] args)
{
  webDriver driver = new ChromeDriver ();
driver.get ("https://demo.actitime.com/login.do");
```

```
boolean Checkbox = driver. findElement (By. id ("KeepLoggedIn
                    CheckBox")). isSelected ();
 if (Checkbox = true)
  {
   System. out. println ("Checkbox is Selected");
  }
 else {
    System. out. println ("Checkbox is not Selected");
   }
  }
}
```

write a Script to login to the actiTIME application.

a. Enter the username.

b. Enter the pwd

c. Click on login button

d. close the browser.

```
Public class ActiLogin {
  Static {
  System. setproperty ("webdriver. Chrome. driver", ". /
          driver / Chromedriver. exe");
  }

  Public Static void main (String [] args)
  {
   webDriver driver = new ChromeDriver ();
   driver. get ("https : // demo. actitime. com / login. do");
   driver. findElement (By. id ("username")). SendKeys ("admin");
```

```
driver. findElement (By. name ("pwd")). SendKeys ("Manager");

driver. findElement (By. Xpath ("// div [. = 'Login']")). Click(

driver. Close ();

  }

}
```

11/09/20

Write a Script to Check whether with width of the user name & pwd Checkbox are equal or not.

```
Public Class DimensionWidth {

Static {

 System. Setproperty ("webdriver. Chrome. driver", "./
driver / Chromedriver. exe");

 }

 Public Static void main (String [] args) {

webDriver driver = new ChromeDriver ();

driver. get ("https : // demo. actitime. com / login. do");

int unwidth = driver. findElement (By. id ("username"))
                                  . getSize(). getwidth ();

int pwwidth = driver. findElement (By. name ("pwd"))
                                  . getSize (). getwidth ();

if (unwidth == pwwidth) {
```

```
System.out.println("width of both the text box are equal");
}
else {
System.out.println("width of both the text box are not equal");
}
driver.close();
}
}
```

write a Script to check whether height of the both the text box are equal or not.

```
public class DimensionHeight {
static {
System.setproperty("webdriver.Chrome.driver", "./driver/Chromedriver.exe");
}
public static void main (String [] args) {
webDriver driver = new ChromeDriver();
driver.get("https://demo.actitime.com/login.do");
int UNHeight = driver.findElement(By.id("username")).getSize().getHeight();
int pwHeight = driver.findElement(By.name("pwd")).getSize().getHeight();
if (UNHeight == pwHeight)
{
System.out.println("Height of both textbox are equal");
}
```

```
else {
System.out.println("both the textbox height are not
                                equal");
driver.Close();
}
}
}
```

write a script to check whether both the text boxes are properly alligned or not.

```
Public Class Alligment {
Static {
System.Setproperty("webdriver.chrome.driver", "./driver/
Chromedriver.exe");
}
public Static void main (String [] args) {
webDriver driver = new ChromeDriver();
driver.get("https://demo.achtime.com/");
int UNX = driver.findElement (By.id ("Username")).get
                        Location().getX();
int pwx = driver.findElement (By.name ("Pwd")).get
                        Location().getX();
if (UNX == Pwx) {
System.out.println(" Both are equal and aligned");
}
else {
System.out.println("Both are not equal");
}
driver.Close();
}
}
```

write a Script to click on the Submit present in open Souce billing application. without using .Click method.

```
public class withoutClick {
Static {
System. setproperty ("webdrive. Chrome. driver", "./
                    driver/ Chromedriver. exe");
}
public Static void main (String [] args) {
webDriver driver = new ChromeDriver ();
driver. get ("http://demo. opensourcebilling. org/");
driver. findElement (By. id ("user_login-btn")). Submit()
driver. Close ();
}
}
```

Note: Submit method will work on when the type = "Submit" attribute is present.

write a Script to Copy the text present in 1st text box & Paste it into 2nd text box.

```
public class Copypaste {
Static {
System. setproperty ("webdriver. Chrome. driver", "./driver/
                    Chromedriver. exe");
}
public Static void main (String [] args) {
webDriver driver = new ChromeDriver ();
driver. get ("file : c: // users / samskruthi / onedrive/
                    desktop / input. html");
```

```
driver. findElement (By. Xpath ("//input [@type = 'text' and @value='A']")
                                .SendKeys(Keys. CONTROL+"ac");
webElement SecondTbx = driver. findElement (By. Xpath ("// input
                                [@type = 'text' and @value = 'B']"));
  SecondTbx .Clear();
  SecondTbx. SendKeys (Keys. CONTROL + "v");
 }
}
```

write a Script to Check whether Skills textbox & Location textbox is properly alligned or not in naukri. Com

```
Public Class AllignmentNaukri {
 Static {
   System. Setproperty ("webdriver. Chrome. driver", "./
                                driver / Chromedriver. exe");
 }
 public Static void main (String [] args) {
   webDriver driver = new ChromeDriver ();
   driver. get ("https : // www. naukri. Com/");
   int Skilly = driver. findElement (By. id ("qsb-keyword-Sugg")).
                                getLocation (). getY ();
   int locationy = driver. findElement (By. id ("qsb-location-Sugg"))
                                getLocation (). getY ();
   if ( Skilly == locationy) {
     System. out. println (" Skill textbox and Location textbox
       are alligned");
   }
```

```
else {
System.out.println ("Skill textbox and location textbox are
                    not alligned");
  }
driver.Close();
  }
}
```

Write a Script to Copy the text in email textbox and paste it in into password textbox for opensource billing application.

```
public class CopyPaste2 {
Static {
System.Setproperty ("webdriver.Chrome.driver", ":/driver/
                    Chromedriver.exe");
  }
public Static voidmain (String [] args) throws Interrupted
                    Exception {
WebDriver driver = new ChromeDriver();
driver.get ("http://demo.opensourcebilling.org/users/
                    Sign-in");
driver.findElement (By.id ("email")).SendKeys (Keys.
                    CONTROL + "ac");
Thread.Sleep (2000);
driver.findElement (By.id ("password")).Clear();
Thread.Sleep (2000);
driver.findElement (By.id ("password")).SendKeys (
                    Keys.CONTROL + "v");
```

```
driver. Close ();
  }

}
```

17/9/20

write a Script to print the value present in the textbox.

```
public Class printTextBox {
Static {
System. setproperty ("webdriver. Chrome. driver", "./driver/
                                    Chromedriver. exe");
 }
public Static void main (String [] args) {
webDriver driver = new ChromeDriver ();
driver. get ("https : // demo. opensourcebilling. org / ");
String email value = driver. findElement ( By. id ("email")) .
                                    getAttribute ("value");
System. out. println (email value);
driver. Close ();
 }
}
```

what are the frequently used Locators in Selenium?

id()

name ()

linkText ()

Xpath ()

write a Script to printtext of the link "Forgot your Password

```
public Class print2 {

    Static

  {
```

```
System.setproperty("webdriver.Chrome.driver", "./driver/
                                Chromedriver.exe");
}
public static void main (String [] args) {
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    String forgottext = driver.findElement (By.id
                    ("topasswordRecoverypageLink")).getText()
    System.out.println (forgottext);
    driver.close();
}
}
```

write a Script to verify that the email textbox present in facebook login page is empty or not.

```
public class VerifyLoginpage {
  Static {
System.setproperty ("webdriver.Chrome.driver", "./
                    driver / Chromedriver.exe");
  }
  public Static void main (String [] args) {
  webdriver driver = new ChromeDriver();
```

```
driver.get ("https://www.facebook.com/login/");
driver.findElement(By.Xpath ("//input [id='email']")).Sendkeys ("Sam")
String a = driver.findElement (By.Xpath ("//input [@id='email']")).
                                        getAttribute ("Value");
boolean b = a.isEmpty ();
if (b == true) {
    else {      S.o.p ("box is empty"); }
                S.o.p ("box is not empty"); } }
```

write a Script to remove the value present in textbox without using Clear method. (open source billing application)

```
public class DeleteText {
Static {
System.Setproperty ("webdriver.Chrome.driver", "./driver/
  Chromedriver.exe");
}
public Static void main(String [] args) {
webDriver driver = new ChromeDriver ();
driver.get ("http://demo.opensourcebilling.org/");
webElement emailtx = driver.findElement (By.id ("email"));
emailtx.Sendkeys (Keys.CONTROL+"a");
emailtx.Sendkeys (Keys.DELETE);
}
}
```

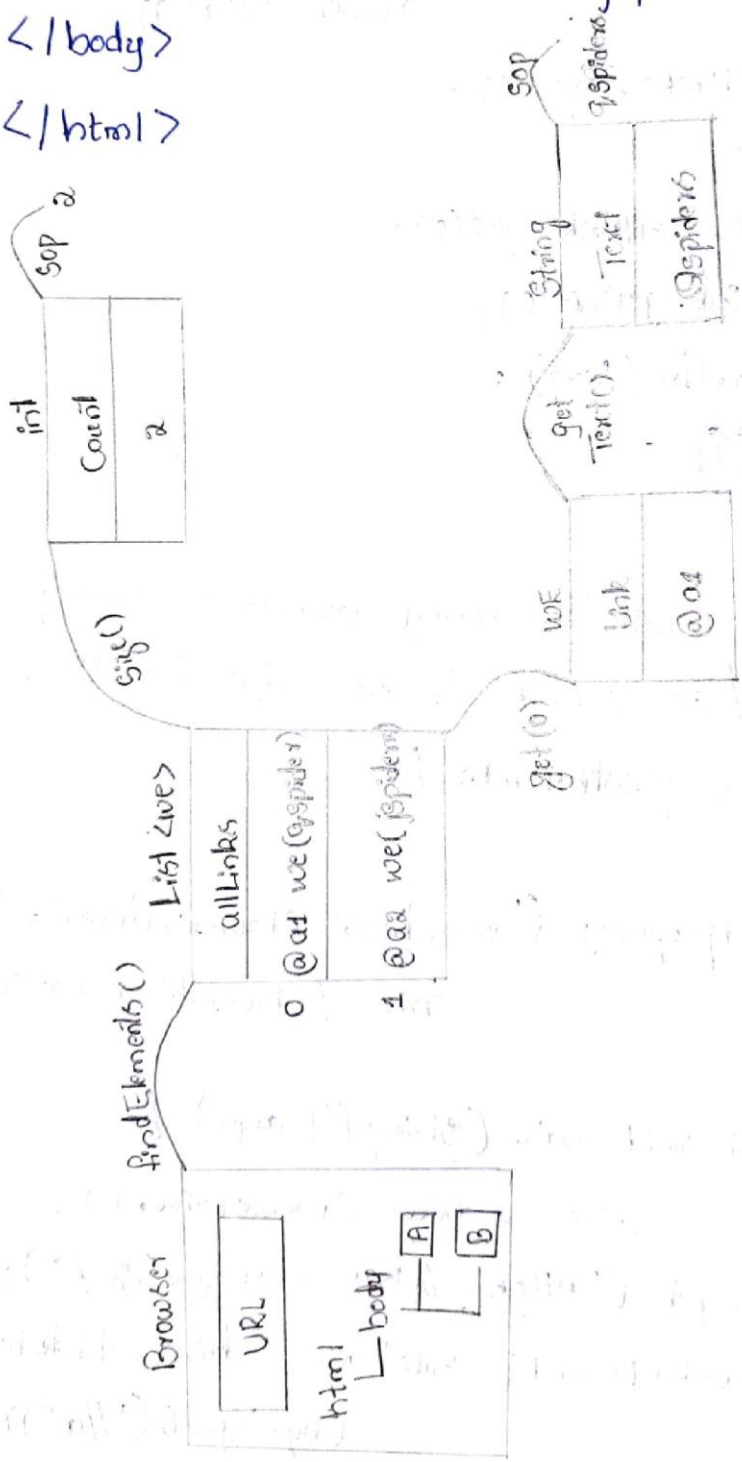
Handling Multiple Elements : In order to handle multiple element, we use find elements methods. Return type of find Elements is "list of web elements". List should be imported from java.util package.

If locators are matching with multiple element, it will return address of all matching element.

If locators are not matching with any of the ~~multiple~~ element, then find elements method will return empty list.

HTML Code of link.html

```
<html>
<body>
<a id="d1" name="n1" Class="c1" href="http://www.
        qspiders.com/" > Qspiders</a><br>

<a id="d2" name="n2" Class="c2" href="http://
        www.jspiders.com/" >jspiders</a><br>
</body>
</html>
```



```
public class HandlingMultipleElements {
Static {
System. Setproperty ("webdriver. Chrome. driver", ". / driver/
                        Chromedriver . exe");
  }
Public Static void main (String [] args) {
  WebDriver driver = new ChromeDriver ();
  driver. get (" file : /// C: / users / Samskruthi / Desktop / Demo.html)
  List <WebElement> AllLinks = driver. findElements ( By.
                                  Xpath ("//a"));
  int Count = AllLinks. Size ();
  S.o.pln( Count);
  WebElement link = AllLinks. get(0);
  String text = link. getText ();
  System. Out. Println (text);
  driver. Close ();
  }
}
```

15/09/20
Write a Script to Count how many number of links present in Flipkart.com / amazon.in & the print the links on Consol

```
public Class PrintAllLinks {
  Static {
  System. Setproperty ("webdriver. Chrome. driver", "./
                        driver / Chromedriver. exe");
  }
  public Static void main (String [] args) {
      WebDriver driver = new ChromeDriver ();
      driver. get ("https : // www. amazon .in / ");
      List <WebElement> AllLinks = driver. findElements
                        (By. Xpath ("//a"));
```

```
int Count = AllLinks.Size();
System.out.println(count);
for(int i=0; i< Count; i++) {
WebElement link = AllLinks.get(i);
String text = link.getText();
System.out.println(Text);
}
driver.Close();
}
}
```

WAS to Count how many number of links present in flipkart.com & print the links on the console. (Input should be taken from user (using Scanner Class).

```
Public Class HandlingMultipleElements1 {
Static {
System.Setproperty("webdriver.Chrome.driver", "./driver/
                                    Chromedrive.exe");
}
Public Static Void main (String [] args) {
    System.out.println("enter the Url");
    Scanner Sc = new Scanner(System.in);
    String URL = Sc.nextLine();
    WebDriver driver = new ChromeDriver();
    driver.get(URL);
List<WebElement> AllLinks = driver.findElements(By.Xpath("//a"));
    int Count = AllLinks.Size();
    System.out.println(Count);
    for (int i=0; i< Count; i++) {
    WebElement link = AllLinks.get(i);
    String text = link.getText();
    System.out.println(text);
    }
    driver.Close();
    }
}
```

Auto Suggestion:

Write a program to print all the auto suggestion in google.com after typing java in search textbox and count the number of auto suggestion.

or

Automate the following scenario:

1. open Chrome browser
2. Go to www.google.com
3. Type 'java' in search textbox
4. Find all the auto suggestions & print count of auto suggesti
5. Print the text of auto suggestion.
6. Select the first auto suggestion.

```
public class AutoSuggestion {
Static {
System.setproperty("webdriver.Chrome.driver", ".//driver/
                    Chromedriver.exe");
}
public static void main (String [] args) throws InterruptedException {
// open the Chrome browser
WebDriver driver = new ChromeDriver();
// Go to www.google.com
driver.get("https://www.google.com/");
// Type 'java' in the search text box
driver.findElement(By.name("q")).SendKeys("java");
Thread.Sleep(3000);
// Find all the auto suggestions & print count of auto suggestion
```

```
List <WebElement> AllSugg = driver. findElements (By. Xpath (
                    " //span [Contains (text (), 'java')] "));
int Count = AllSugg . Size ();
// print all the autoSuggestion
System .out .println (count);
for (int i=0 ; i< Count ; i++) {
    String text = AllSugg . get (i) . getText ();
        System. out. println (text);
  }
  // Select the first AutoSuggestion
    AllSugg . get (0) . Click ();
    driver . Close ();
  }
}
```

* Difference between find element and find Elements.

Sl.no	Find element	Find elements.
1.	Return type of find element is web element.	List of web elements.
2.	If the locator is not matching it will throw no such element exception.	Returns empty list
3.	If locators are matching with multiple elements it will return the first matching element.	It returns all the matching elements
4.	used to handle single element.	Handle multiple elements.

Auto-Suggestion for Selenium by for-each loop & to print la
auto Suggestion.

```
public Class AutoSuggSelenium {
Static {
    System.setproperty ("webdriver.Chrome.driver", "./driver/
                        Chromedriver.exe");
}
public static void main (String [] args) throws InterruptedExce
                                                    ption {
    webDriver driver = new ChromeDriver ();
    driver.get ("https://www.google.com/");
    driver.findElement (By.name ('q')).SendKeys ("Selenium");
    Thread.Sleep (4000);
List < webElement > AllSugg = driver.findElements (By.xpath ("//
                        Span [Contains (text(), 'Selenium')]"));
    int Count = AllSugg.Size();
    System.out.println (count);
    for (webElement Sugg : AllSugg) {
        String text = Sugg.getText();
        System.out.println (text);
    }
    AllSugg.get (count-1).Click ();
        driver.Close ();
    }
}
```

16/9/20

Automate the following Scenario.

1. Open Chrome browser
2. Go to www.flipkart.com
3. Type 'iphone' in the Search text box
4. Find all the autoSuggestion & print Count of auto Suggestion.

5. print the text of auto Suggestion (Note you have to use "for each loop")

6. Select last auto Suggestion.

Go to makemytrip.com & type "ban" in from textbox, then print all the auto Suggestion & select the "bangalore" in it.

ways to select all the checkbox. Url should be taken from the user.

Synchronization : process of matching the selenium speed with the application is called synchronization.

Most of the times selenium is faster than the application, because of this reason, we may not get expected result or we may get exception such as "NoSuchElementException".

Example: using Thread.Sleep method.

```
public class Synchronization {
Static {
System.setproperty ("webdriver.Chrome.driver", "./driver/
                                Chromedriver.exe");
}
public static void main (String [] args) throws InterruptedExc
                                                    eption {
webDriver driver = new ChromeDriver ();
driver.get ("https://demo.actitime.com/");
driver.findElement (By.id("username")).SendKeys ("admin");
driver.findElement (By.Name ("pwd")).SendKeys ("manager");
driver.findElement (By.Xpath ("//div [text() = 'Login']")).
                                    Click ();

Thread.Sleep (5000);
driver.findElement (By.id ("logoutLink")).click ();
driver.Close ();
}
}
```

Implicit wait : In Selenium there are different ways to synch-ronize the script, one of the frequently used option is implicitly wait.

Syntax :

```
driver.manage().timeouts().implicitlywait (10, TimeUnit.SECONDS);
```

It takes two arguments, first one is long (duration) second one is TimeUnit.

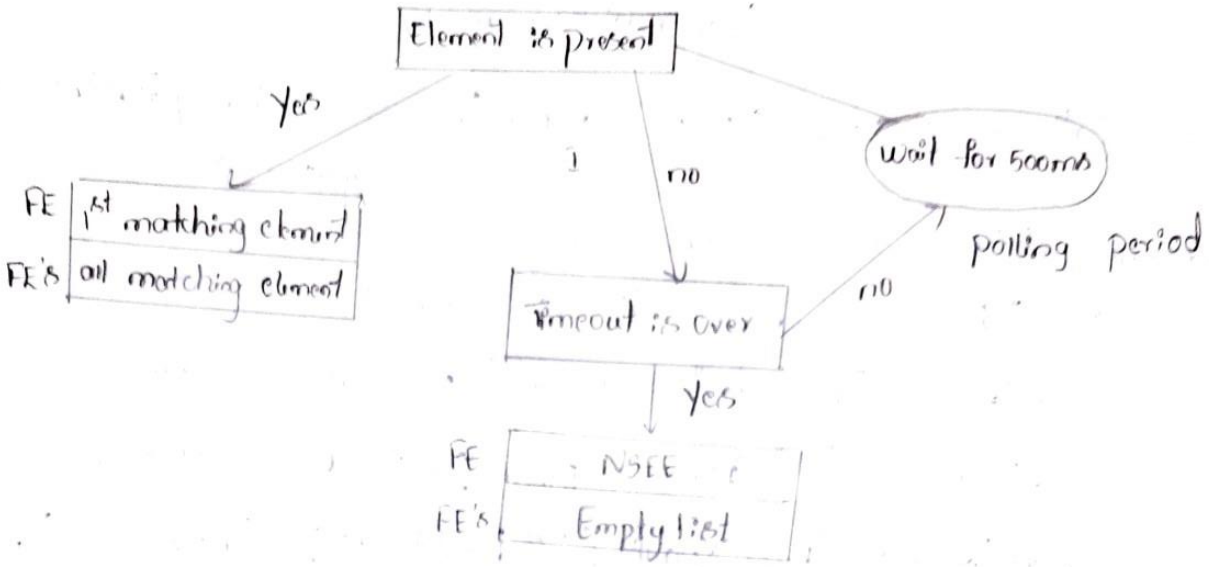
TimeUnit can be Days, Hours, Minutes, Seconds, MiliSeconds, MicroSeconds, NanoSeconds etc.

The specified duration is used only by findElement & findElements statement.

Means implicit wait will work only for find element/elements. not for any other functions.

Default implicit wait is zero Seconds.

Flow Diagram :



→ when the Control comes to any findElement or find Elements Statement, it will check whether the element is present or not.

→ If the element is present the findElement method return first matching element where as findElements returns all the matching elements.

→ if the specified element is not present then it will Check for the timeout.

→ if the time is over findElement method will throw NSEE (No Such Element Exception) where as find Elements method returns Empty List.

→ if the time is not over it waits for 500ms (1/2 Sec) which is called as polling period.

Then it will Continue to Check whether the element is present or not.

```
⇒ public Class Synchronization 1 {
  Static {
  System. Setproperty ("webdriver. Chrome. driver", "./driver/
                                    Chromedriver. exe");
  }
```

```
public static void main (String [] args) throws InterruptedException {
    webDriver driver = new ChromeDriver ();
    driver. manage (). timeouts (). implicitlywait (70, TimeUnit. SECONDS);
    driver. get ("http://localhost/login.do");
    driver. findElement (By. id ("username"). SendKeys ("admin");
driver. findElement (By. name ("pwd")). Sendkeys ("manager");
driver. findElement (By. Xpath ("//div[text()='Login']")). Click ();
driver. find Element (By. id ("logoutLink")). Click ();
    driver. Close ();
  }
}
```

17/09/20

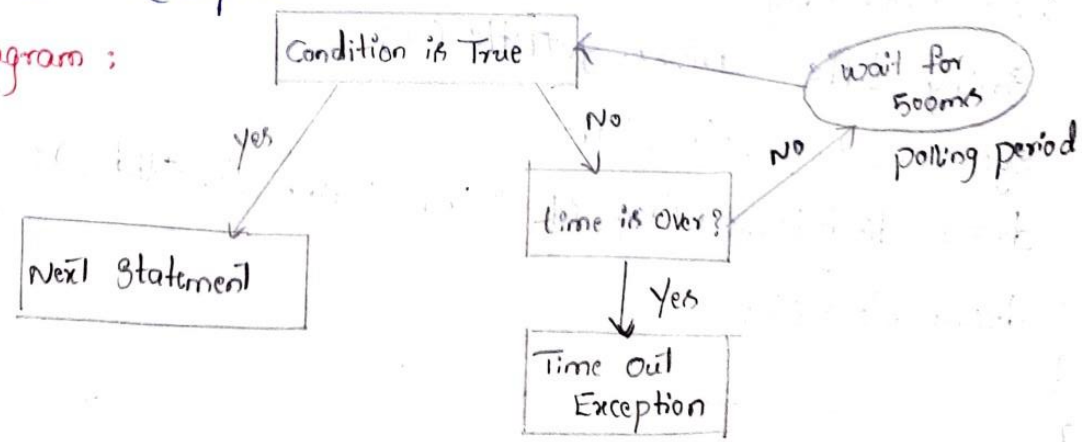
1. Can we specify implicitlywait statement multiple times in the Selenium Script? Yes.

2. Is it necessary to write Implicitlywait statement multiple times? No

Explicit wait: In order to handle the synchronization of any method, we can use explicit wait. WebdriverWait itself called as Explicit wait. Because we have to specify the waiting condition explicitly.

Syntax: WebDriverWait wait = new WebDriverWait (driver, 10);
wait. until (ExpectedConditions. titleIs ("actiTIME - Enter Time-Track"));

Flow Diagram:



→ When the Control Comes to wait. Until Statement it will Check the specified Condition.

→ If Condition is true, it will go to the next statement, if the Condition is false, it will Check for the timeout.

→ If time out is over, it will throw Time out exception, else it will wait for 500ms & it will Continues to Check the Condition.

Note : If the above flow diagram titles ("actiTIME - Enter Time Track" is the Condition.

By using Explicit wait we handle the Synchronization of any Statement but only one at a time.

⇒

```
public Class SynchronizationCls {
    Static {
  System.. Setproperty ("webdriver. Chrome. driver", "./ driver /
                                    Chromedriver. exe");
  }
  public Static void main (String[] args ) {
    webDriver driver = new ChromeDriver ();
    driver. get ("http:// localhost / login. do");
    driver. findElement (By. id ("username")). Sendkeys ("admin");
    driver. findElement (By. name ("pwd")). Sendkeys ("manager");
    driver. findElement (By. Xpath ("// div [text () = 'Login']")). Click()
    webDriverWait wait = new webDriverWait (driver, 10);
    wait. until (ExpectedConditions. title Contains ("Enter"));
     String title = driver. getTitle ();
     System. out. println (title);
     driver. findElement (By. id ("logout link")). Click ();
     driver. Close ();
  }
}
```

Note :

ExpectedCondion is a interface & ExpectedCondion is a Class.

Can we handle the Synchronization of findElement method using Explicit wait ? Yes.

⇒ Example :

```
public class Synchronization1 {
Static {
 System.Setproperty ("webdriver.Chrome.driver","./driver/
                                    Chromedriver.exe");
}
public Static void main (String[] args) {
 webDriver driver = new ChromeDriver();
 driver.get ("http://localhost/login.do");
 driver.findElement (By.id ("username")).Sendkeys ("admin");
 driver.findElement (By.name ("pwd")).Sendkeys ("manager");
driver.findElement (By.Xpath ("//div[text() = 'Login']")).Click()
webDriverwait wait = new webDriverwait (driver, 10);
wait.Until (ExpectedConditions.visibilityofElementLocated (By.id
                                    ("logoutlink")));
driver.findElement (By.id ("logoutlink")).Click();
driver.Close();
}
}
```

Difference b/w Implicit and Explicitwait.

	Implicitwait	Explicitwait
1.	we do not specify the waiting Condition explicitly	we should specify the waiting Condition explicitly
2.	we can handle Synchronization of all 'Findelement' and 'Findelements'.	we can handle Synchronization of any method But only one at a time.

3.	After the time duration we get 'NoSuchElement' exception	After the duration we get 'Timeout' exception.
4.	Time Duration can be Days, Hours, Minutes, Seconds, etc...	Time duration will be only seconds.

18/09/20

Customwait : Handling the Synchronization of the automation Script by writing out own code is called as Custom wait.

write a Script to handle the Synchronization by using Customwait.

```
try {
driver. findElement (By . id("logout link")). click();
System . out. println("hi");
}
catch ( Exception e) {
    // e. printStackTrace ();
System . out. println ("bye");
}
```

⇒

```
public class CustomWait {
Static {
  System . Setproperty ('webdriver. Chrome. driver", "./driver
                                    /Chromedriver. exe");
}
public Static void main (String [] args) {
webDriver driver = new ChromeDriver ();
driver. get ("http : // demo.actitime / login . do");
driver. findElement (By. id ("username")). SendKeys ("admin");
driver. findElement (By. name ("pwd")). SendKeys ("manager");
driver. findElement (By. Xpath ("// div [text () = 'Login:]")). Click(
int i = 0;
while (i <= 200) {
```

```
try {
  driver.findElement (By.id ("logoutLink")).click ();
  break;
}
Catch (Exception e) {
    i++;
}
driver.Close();
}
}
```

write a script to check whither login page is loaded within 7sec or not.

Note : Timeouts().pageloadTimeout() is used only by get method.

```
public Class pageLoadTime {
  Static {
System.Setproperty ("webdriver.Chrome.driver", ":/driver/
                                  chromedriver.exe");
}
Public Static Void main (String [] args) {
WebDriver driver = new ChromeDriver ();
driver.manage().timeouts().pageLoadTimeout (7, TimeUnit.
                                              SECONDS);
try {
driver.get ("https://demo.actitime.com/");
System.out.println ("page is loading within 7 Seconds");
}
Catch (TimeOutException e) {
System.out.Println ("page is not loading within 7 Seconds")
}
driver.Close();
}
}
```

What are the different ways of Synchronization?

→ Implicit wait, Explicit wait, Custom wait and Fluent wait (not widely used in industry. it is mainly used to Change the polling period (Means we can Change default polling period i.e half sec (500 milli sec)) to any time.

What is default value of polling period?

→ 500ms (1/2 Second)

Handling ListBox: (Drop Down or Combo Box)

ListBox is Created using Select Tag.

There are two type of list box.

a. Single Select Listbox (also Called as dropdown or Combobox)

b. Multiselect Listbox.

Content of the listbox is Created using option tag.

To handle the listbox, we used Select class of Selenium. it Should be imported from the following packages. import org.openqa.selenium.Support.ui.Select.

Select Class has parameterized Constructor it takes an argument of type webElement (i.e address of the listbox) In order to Select the required option present in the listbox we can use any one of 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 then it will select first matching option (in dropdown list) & if the Specified option is not present (text, value or index), we get NoSuchElement Exception.

Select class can also be used to handle multiselect listbox.

In Select Class we can also have the following 9 Methods.

This can be used on multiselect listbox.

1. deselectByVisibleText (Str)
2. deselectByIndex (int)
3. deselectByValue (Str)
4. deselectAll ()
5. isMultiple ()
6. getAllSelectedoption ()
7. getFirstSelectedOption ()
8. getoptions ()
9. getwrappedElement ()

If the specified option is duplicate in multiselect listbox, it selects all the matching option.

⟹ Example 1

```
public class SingleSelect {
  Static {
  System.Setproperty("webdriver.Chrome.driver","./driver/
                        Chromedrider.exe");
  }
  public Static void main (String [] args) throws Interrupted
                                                Exception {
  WebDriver driver = new ChromeDriver ();
  driver.get ("https://www.SkyScanner.co.in/");
  driver.findElement (By.Xpath ("(//span[Contains@class,
                                    'DateInput')])[1]")).click(
  WebElement MonthListbox = driver.findElement (By.name("months"))
  Select s = new Select (MonthListbox);
  s.SelectByIndex (4);
  Thread.Sleep (3000);
```

```
S. SelectByValue ("2021-06");
Thread Sleep(3000);
S. SelectByVisibleText ("August 2021");
}
}
```

→ Example 2

```
public Class SingleSelect1 {
Static {
System. setproperty ("webdriver. Chrome. driver", ". / driver
                                    /chromedriver. exe");
}
public Static Void main (String [] args) throws Interrupted
                                    Exception {
WebDriver driver = new ChromeDriver ();
driver. get ("https :// www. coreinsurance. com / thicl / proposalc
                                    /renew / index - care");
driver. find Element (By. id ("dob")). Click ();
WebElement monthListbox = driver. findElement ( By. Class
                                    Name ("ui - date picker - month"));
Select S = new Select (Month Listbox);
S. SelectBy Index (8);
S. SelectBy Value ("5");
S. SelectBy VisibleText ("oct");
}
}
```

→ Example 3.

```
Public Class SingleSelect2 {
Static {
System. Setproperty ("webdriver. Chrome. driver", ". / driver/
                                    Chromedrive. exe");

}
```

```
public static void main (String[] args) throws Interrupted
                                        Exception {
webDriver driver = new ChromeDriver ();
driver.get ("file ;///C:/Users/Shwetha/Samskruthi/Desktop/
                                        Dob.html");
WebElement MonthListbox = driver.findElement (By.id ("Dob"));
Select S = new Select (Month Listbox);
S.SelectByIndex (8);
S.SelectByValue ("5");
S.SelectByVisibleText ("oct");
}
}
```

write a Script to Select and deselect in MultiSelect

```
public class MTRmultiSelect {
static {
System.Setproperty ("webdriver.Chrome.driver", ".//driver/
                                        Chromedriver.exe");
}
public static void main (String [], args) throws Interrupted
                                        Exception {
webDriver driver = new ChromeDriver();
driver.get ("file ;///C:/Users/Samskruthi/OneDriver/Desktop
                                        /Hotel.html");
WebElement MtrListBox = driver.findElement (By.id ("mtr"));
Select S = new Select (MtrListBox);
S.SelectByIndex (0);
S.SelectByValue ("d");
S.SelectByVisibleText ("Vada");
Thread.Sleep (3000);
S.SelectByVisibleText ("Vada");
Thread.Sleep (3000);
boolean V = S.isMultiple ();
System.Out.println (V);
```

```
s.deselectByIndex(1);
s.deselectByValue("v");
s.deselectByVisibleText("idly");
}
}
```

write a Script to get firstSelected option present in SLV listbox

```
public class FirstSelected {
Static {
System.Setproperty("webdriver.Chrome.driver", "./driver
                                /chromedriver.exe");
}
public Static Void main (String[] args) {
  webDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/Samskruthi/Desktop/Hotel.htm
webElement SLVListbox = driver.findElement(By.id("slv"));
  Select s = new Select(SLVListbox);
webElement FirstSelectedoption = Select.getFirstSelectedoption()
String text = FirstSelectedoption.getText();
System.out.println(text);
  driver.close();
}
}
```

write a Script to print all the Selected option in SLV listbox.

```
public class AllSelected {
Static {
System.Setproperty("webdriver.chrome.driver", "./driver/
                                Chromedriver.exe");
}
public Static void main (String[] args) {
webDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/Samskruthi/Desktop/Hotel.html");
```

```
webElement SlvListbox = driver. findElement (By. id ("Slv"));
Select S = new Select (SlvListbox);
List <webElement> AllSelectedoptions = S.getAllSelectedoptions();
for (webElement option : AllSelectedoptions) {
    String text = option. getText();
      System. out. println(text);
  }
  driver. close ();
 }
}
```

write a Script to print all the options present in SLv listbox along with the Count.

```
public class Alloptions {
  Static {
System. Setproperty ("webdriver. Chrome. driver", "./driver/
                                    Chromedriver. exe");
 }
 public Static Void main (String [] args) {
webDriver driver = new ChromeDriver ();
driver. get ("file :///C : /users / Samskruthi /Desktop/ Hotel. html")
webElement SLvListbox = driver. findElement (By. id ("Slv"));
    Select S = new Select (SLvListbox);

List < webElement > Alloptions = S. getoptions ();
  int Count = Alloptions. Size ();
 System. out. println (Count);
 for (int i=0 ; i< Count ; i++) {
   String text = Alloptions. get (i). getText();
      System. out. println (text);
  }
   driver. close ();
  }
}
```

write a Script to print all the option in Sorted order.

```
Public Class SortedOrderMtr {
Static {
System. Setproperty ("webdriver. Chrome. driver", "./driver/
                                                Chromedriver. exe");
}
public Static void main (String [] args) {
webDriver driver = new ChromeDriver();
driver. get ("file:///C:/Users/Sanskruthi/Desktop/Hotel. html")
webElement Mtrlistbox = driver. findElement (By. id ("mtr"));
ArrayList<String> alltext = new ArrayList<String>();
  Select S = new Select (Mtrlistbox);
List<webElement> alloptions = S. getoptions();
    for (webElement option : alloptions) {
      String text = option. getText();
    alltext. add (text);
  }
  Collections. Sort (alltext);
  for (String Stext : alltext) {
    System. out. println (Stext);
  }
  driver. Close();
}
}
```

23/9/20

write a program to Select all the option and deselect in reverse Order for MTR listbox.

```
Public Class AllSelectoptions{
Static {
System. Selproperty ("webdriver. Chrome. driver", "./driver/
                                                Chromedriver. exe");
}
public Static Void main (String [] args) throws Interrupted
                                                Exception {
webDriver driver = new ChromeDriver();
```

```
driver. get ("file : /// C : Users / Samskruthi / Desktop / Hotel.html");
webElement mtrlistbox = driver. findElement (By. id ("mtr"));
Select S = new Select (mtrlistbox);
List < webElement > Alloptions = S.getoptions ();
int Count = Alloptions. Size();

for (int i=0 ; i < Count ; i++) {    // selecting all the options
    S. SelectByIndex (i);
  Thread. Sleep (1000);
}

for (int j= Count-1 ; j>=0 ; j--) { // deselecting all options.
    S. deselectByIndex (j);
    Thread. Sleep (1000);
  }
}
```

write a Script to print only Duplicate option for MTR listbox.

```
Public Class OnlyDuplicate {
Static {
System. Setproperty ("webdriver. Chrome. driver", "./driver/
                                        Chromedriver. exe");
}
public Static void main (String [] args) throws Interrupted Exception {
webDriver driver = new ChromeDriver ();
driver. get ("file : /// C : Users / Samskruthi / Desktop / Hotel. html");
webElement mtrlistbox = driver. findElement (By. id ("mtr"));
Select S = new Select (mtrlistbox);
HashSet < String > allitext = new HashSet <> ();
List < webElement > Alloptions = S. getoptions ();
int Count = Alloptions. Size ();
for (int i= 0 ; i< Count ; i++) {
  String text = Alloptions. get (i). getText ();
```

```
if (alltext.add(text) == false) {
    System.out.println(text);
  }
}
  driver.Close();
 }
}
```

Handling pop up's :

In Selenium, depending on popup we write different types of Code to perform action on the popup

popup are generally Catagorised as follows.

1. Java Script or alert popup or Confirmation popup.
2. Hidden division or Calender popup
3. File upload popup
4. File download popup
5. Print popup
6. Child window popup
7. Notification popup
8. Authentication popup

1. Java Script popup or alert popup : or (Confirmation popup)

Characteristics : a. we Cannot inspect this popup.

b. we Cannot move this popup

c. This popup Contains ok button (alerts) or it Contains ok and Cancel button (Confirmation popup)

d. It will be present below the address bar in the middle section of the browser.

Example :

```
Public Class Alertpopup {
Static {
System.Setproperty("webdriver.Chrome.driver",".//driver/
                                    Chromedriver.exe");
  }
```

```
public static void main (String [] args) {
WebDriver driver = new ChromeDriver ();
driver.get (" https://www.seleniumeasy.com/test/javascript-
                                alert-box-demo.html");
driver.findElement (By.Xpath ("//button [text() = 'Click me!']) [1]"))
                                .Click ();
WebDriverWait wait = new WebDriverWait (driver, 10);
wait.until (ExpectedConditions.alertIsPresent ());
Alert a = driver.SwitchTo().alert ();

String text = a.getText ();

a.accept ();
System.out.println (text);

driver.Close ();

}

}
```

Solution : To handle java script popup we can use "Switch to. Alert()" statement.

After switching to alert popup we can use,

a. getText (): to get the text present on the popup.

b. Accept (): to click on OK button.

c. Dismiss (): to click on Cancel button.

d. Sendkeys (): to type the text in the popup.

All the above code work on Confirmation popup also

* If the popup is alert then there will not be any difference b/w accept or dismiss (both of them clicks on OK)

24/9/2020.

2. Hidden Division or Calender popup:

Characterstics: a. we can inspect this popup

b. we cannot move this popup.

Solution: we handle hidden division popup using "driver.findElement().method".

Example:

```
Public class HiddenDivision {
 Static {
 System.Setproperty ("webdriver.chrome.driver", "./driver/
  }                                          Chromedriver.exe");
 public static void main (String[] args) throws InterruptedExcepti
 WebDriver driver = new ChromeDriver();
 driver.manage().timeouts().implicitlywait (10, TimeUnit.SECONDS);
 driver.get ("https://www.flipkart.com/");
 Thread.Sleep (3000);
 driver.findElement (By.Xpath ("//button)[2]")).Click();
  }
}
```

write a Script to Automate the following Scenario.

a. open the browser

b. Enter the URL - https://www.religarehealthinsurance.com/thic1/proposalcp/renew/

c. Enter the policy number - 123

d. Click on DOB

e. Select the required date - (year, month & date)

f. Specify the Contact number - 9845098450

g. Click on let's renew button.

```
Public Class HiddenDivisionScenario {
Static {
System.Setproperty ("webdriver.Chrome.driver", ".\driver\
                                    Chromedriver.exe");
}
public Static void main (String [] args) {
// a. open the browser.
webdriver driver = new ChromeDriver();
driver.manage().timeouts().implicitlywait (10, TimeUnit.SECONDS)

//b. Enter the URL
driver.get ("https://www.careinsurance.com/shicl/proposalcp/
                                    renew/index-care");

// c. Enter the policy-number 123
driver.findElement (By.id ("policynumber")).Sendkeys ("123");

// d. Click on DOB.
driver.findElement (By.id ("dob")).click();

// e. Select the required date.
webElement yearlistbox = driver.findElement (By.ClassName ("ui-datepicker-
                                    Year"));
Select S = new Select (yearlistbox);
S.SelectByVisibleText ("1995");
webElement Monthlistbox = driver.findElement (By.ClassName
                                    ("ui-datepicker-month"));
Select S1 = new Select (Monthlistbox);
S1.SelectByVisibleText ("Dec");
driver.findElement (By.LinkText ("12")).click();
```

```
// f. Specify the Contact number - 9845098450.
driver.findElement(By.id("alternative-number")).sendKeys("9845098
// g. Click on lets renew Button
driver.findElement(By.id("renew-policy-Submit")).click();
}
}
```

Note : In the most of the cases Calender popup will be hidden division popup which will be handled using findElement Statement.

3. File uplode popup :

Characteristics :
1. we can move this popup
2. we cannot inspect this popup.
3. this popup will be having two buttons
   a. open , b. Cancel
4. Title of the popup will be either open or file uplode.

Solution : To handle file uplode popup, we specify the absolute p of the file as an argument for sendkeys method. sendkeys method should be useful for the element on which we click, it displays upload popup.

Example :

```
public class FileUplode {
Static {
System.setproperty("webdriver.Chrome.driver", "./driver/
                                Chromedriver.exe");
}
public Static void main (String[] args) throws InterruptedExce
                                -ption {
```

```
WebDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/Samskruthi/Desktop/
                                    Naukri.html");
Thread.Sleep(3000);
// give absolute path of the file as an argument for SendKeys
driver.findElement(By.id("cv")).SendKeys("C:\\Users\\
                    Samskruthi\\Desktop\\Resume.docx");
  }
}
```

25/09/2020

html Code : `<input type="file" id="cv"/>`

```
⇒ Public Class FileUploade1 {
  Static {
System.Setproperty("webdriver.Chrome.driver", ".\drivers\
                                    Chromedriver.exe");
  }
Public Static void main (String [] args) throws InterruptedEx
                                    ception {
WebDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/~~Sheetha~~ Samskruthi/Desktop/Naukri.html")
Thread.Sleep(3000);
File f = new File(".\data\Resume.docx");
String Absolutepath = f.getAbsolutepath();
driver.findElement(By.id("cv").SendKeys(Absolutepath);
  }
}
```

Robot Class :

Robot Class in Java Class present in Java.awt package (awt = Abstract window Toolkit)

(Robot class works similar to SendKeys. when ever u want to perform Key board operation in windows we go for Robot Class. we can use two methods in robot class they are KeyPress - KeyRelease)

Write a program to demonstrate Robot class (for typing Qsp in the notepad)

```
public class RobotCls {

public static void main (String [] args) throws AWTException,
                                                IOException {
Runtime.getRuntime().exec("notepad");
Robot r = new Robot();
r.KeyPress(KeyEvent.VK_SHIFT);
r.KeyPress(KeyEvent.VK_Q);
r.KeyRelease(KeyEvent.VK_SHIFT);
r.KeyPress(KeyEvent.VK_S);
r.KeyPress(KeyEvent.VK_P);
}
}
```

Runtime in java class and is a single-ton class. Single-ton means it will allow to create only one object at any instance of time.

~~Automate the following Scenario.~~

4. File download popup :

Characteristics with respect to firefox browser.

1. we can move this popup
2. we cannot inspect this popup
3. It will having open with and save file radio button along with OK & Cancel button.

Solution: by using robot class.

Automate the following Scenario.

a. open the firefox browser

b. Go to download page of Selenium

c. Click on download link of java (which will display file download popup)

d. select Save file radio button by pressing alt + S

e. Click on OK button by pressing Enter Key.

Solution :

```
public class FileDownloadpopup {
Static {
System. Setproperty ("webdriver. gecko. driver", ". / driver /
                                        geckodriver. exe");
}
public Static void main (String [] args) throws InterruptedExce
                                        -ption, AWTException {
WebDriver driver = new FireFoxDriver();
driver. get ("https: // www. Selenium. dev / downloads /");
driver. findElement (By. Xpath (" // td [text() = 'java'] / .. //a[. =
                                        'Download']")). Click();
Thread. Sleep (3000);
Robot r = new Robot();
r. Keypress (KeyEvent. VK_ALT);
r. Key press (KeyEvent. VK_S);
r. Key Release (KeyEvent. VK_ALT);
Thread. Sleep (3000);
r. key press (KeyEvent. VK_ENTER);
r. KeyRelease (KeyEvent. VK_ENTER);
Thread. Sleep (9000);
driver. Close();
}
}
```

Note: when we click on any type of download option in Chrome browser, it will not display file download popup, insted of that it will start downloading file directly, hence we don't need to handle file download popup in chrome browser. In all the other browsers we use robot class.

28/9/2020.

5. Print popup:

Characterstics of print popup with respect to firefox browser.

1. we cannot inspect this ~~browser~~ popup
2. we can move this ~~browser~~ popup
3. This popup will be having print and cancel button.
4. Along with the print title.

Solution: we handle this popup by using Robot Class.

Automate the following Scenario:

1. Open firefox browser and go to google.com.
2. Select the second radio button in pages
3. Specify the pages from 2-4
4. Take 2 copies.
5. Click on print button.

```
⟹ public class PrintPopup {
    static {
        System.Setproperty ("webdriver.gecko.driver", ".\driver\
                                                geckodriver.exe");
    }
    public static void main (String[] args) throws AWTExceptio,
                                                InterruptedException {
    WebDriver driver = new FireFoxDriver();
    driver.get ("https://www.google.com/");
```

```
Robot r = new Robot();
Thread.Sleep(3000);
r.KeyPress(KeyEvent.VK_CONTROL);
r.KeyPress(KeyEvent.VK_P);
r.KeyRelease(KeyEvent.VK_CONTROL);
Thread.Sleep(3000);
r.KeyPress(KeyEvent.VK_ALT);
r.KeyPress(KeyEvent.VK_G);
r.KeyRelease(KeyEvent.VK_ALT);
Thread.Sleep(2000);
r.KeyPress(KeyEvent.VK_2);
r.KeyPress(KeyEvent.VK_MINUS);
r.KeyPress(KeyEvent.VK_H);
Thread.Sleep(2000);
r.KeyPress(KeyEvent.VK_TAB);
r.KeyRelease(KeyEvent.VK_TAB);
Thread.Sleep(2000);
r.KeyPress(KeyEvent.VK_ENTER);
r.KeyRelease(KeyEvent.VK_ENTER);
}
Thread.Sleep(2000);
r.KeyPress(KeyEvent.VK_ENTER);
r.KeyRelease(KeyEvent.VK_ENTER);
}
}
```

Note: we use robot class to handle print popup in all the browsers Except Chrome. in Chrome browser we can inspect print button which we can handle it by using findElement method.

Automate the following Scenario.

1. Open the Chrome browser
2. goto google.com
3. Press control p
4. Click on print button.

6. Notification popup :

Characteristics :

1. we cannot inspect this popup
2. we cannot move this popup
3. It will have two buttons allow and block.
4. It is displayed below the address bar in the beginning similar to alert popup.

Note : Chromedriver() → it will open the chrome browser in default setting.

ChromeOption class having non static method called 'addArguments

```
→ public class NotificationPopup {
  static {
  System.setproperty ("webdriver.chrome.driver", ".\driver\chromedriver
                                                    .exe");
  }
  public static void main (String[] args) {
  ChromeOptions option = new ChromeOptions();
  Option.AddArguments ("--disable-notifications");
// open the browser in modified settings.
      webDriver driver = new ChromeDriver(option);
      driver.get ("https://www.cleartrip.com/");
  }
}
```

To handle this popup we change the settings of the browser. So that notification popup will not be displayed. To change the settings of the browser we use Add Arguments () method of ChromeOptions () class. and it is a non static method.

Note : addArguments () is an example for overloaded method.

For every browser we have respective options class Ex: Chrome options, Firefox options, InternetExplorer options etc.

In order to open the browser with modified settings we use parametrised Constructor in respective browser class.

new ChromeDriver () → will open the Chrome browser in default settings.

new ChromeDriver (option) → will open the Chrome browser in modified settings.

The above statement is an example for Constructor overloading.

7. Authentication Popup :

Characteristics :

1. we Cannot move this popup
2. we Cannot inspect this popup
3. This popup will be having username & password textbox along with sign in and Cancel button.
4. This popup will be displayed just below the address bar middle section of the browser similar to alert popup.

Solution : To handle authentication popup we send username & password along with the URL inside the get method.

Example : user name is admin and password is admin,
URL is : https://the-internet.herokuapp.com/basicauth.

→

```
public class Authentication popup{
 Static {
System.setproperty ("webdriver.chrome.driver",".//driver/
                                    Chromedriver.exe");
 }

 public static Void main (String[] args) {
      WebDriver driver = new ChromeDriver();
```

```
driver.get ("https://admin:admin@the-internet.herokuapp.
                                com/basic_auth");
}
}
```

29/09/2020

8. Child window popup :

Characteristics :

1. we can move this popup
2. we can inspect this popup
3. This popup will have minimize, maximize & close button along with the address bar.

Solution : To handle child browser popup we use getwindowHandle () method & Switchto.window statement.

Note : address of the browser present on the desktop is also called as window Handle. (or session ID).

In order to retrieve it we use getwindowHandle method.

```
⇒ public class Childwindow {
  static {
System.setproperty("webdriver.chrome.driver", "./driver/Chromedriver.exe");
  }
  public static void main (String[] args) {
    webDriver driver = new ChromeDriver();
  driver.get ("https://www.naukri.com/");
  Set<String> Allwh = driver.getwindowHandles();
  int Count = Allwh.size();
  System.out.println(Count);
  driver.quit();
  }
```

Difference b/w getwindowHandle() & getwindowHandles().

getwindowHandle() will return the window handle (address) of the current browser where as getwindowHandles() return the windowhandles (address) of all the browsers.

Example :

```
public static void main (String[] args) {
webDriver driver = new ChromeDriver();
String wh1 = driver.getwindowHandle();
System.out.println(wh1);
driver.get ("https://www.naukri.com/");
String wh2 = driver.getwindowHandle();
System.out.println(wh2);
Set<String> Allwh = driver.getwindowHandles();
int count = Allwh.size();
System.out.println(count);
for (String wh: Allwh) {
    System.out.println(wh);
}
driver.quit();
}
```

write a script to print the titles of all the browsers.

```
public class printAllTitle {
static {
System.setproperty ("webdriver.chrome.driver", "./driver/chromedriver.exe");
}
public static void main (String[] args) {
webDriver driver = new ChromeDriver();
driver.get ("https://www.naukri.com/");
Set<String> Allwh = driver.getwindowHandles();
for (String wh : Allwh) {
```

```
driver. switchTo(). window (wh);
String title = driver. getTitle ();
System. out. println (title);
}
driver. quit();
}
}
```

30/9/2020

*** write a Script to close all the browsers without using Quit method.

```
public Class CloseWithoutQuit {
Static {
System. Setproperty ("webdriver. Chrome. driver", "./ driver/
                                    Chromedriver. exe");

}
public Static void main (String [] args) {
webDriver driver = new ChromeDriver()
driver. get ("https ://www. naukri. com /");
Set <String> Allwh = driver. getwindowHandles();
for (String wh ; Allwh) {
    driver. switchTo. window (wh);
        driver. close ();
    }
}
}
```

write a Script to close the specific browser.

```
Public Class CloseSpecificWindow {
Static {
System. Setproperty ("webdriver. Chrome. driver" ; "./ driver/
                                    Chromedriver. exe");

}
Public Static void main (String [] args) {
    System. out. println ("enter the title which you want to close");
    Scanner Sc = new Scanner (System. in);

String ExpectedTitle = Sc. nextline();
```

```
WebDriver driver = new ChromeDriver();
  driver.get("https://www.naukri.com/");
  Set<String> Allwh = driver.getWindowHandles();
  for (String wh : Allwh) {
      driver.SwitchTo().window(wh);
    String ActualTitle = driver.getTitle();
    if (ActualTitle.equals(ExpectedTitle)) {
            driver.close();
      }
    }
}}
```

write a Script to close all the Child browser except parent br

```
Public class CloseChildWindows {
  Static {
  System.Setproperty("webdriver.Chrome.driver", "./driver/
                                        Chromedriver.exe");
  }
 public Static void main (String [] args) {
  webDriver driver = new ChromeDriver();
 driver.get("http://popuptest.com/jpopuptest1.html");
 String Expectedtitle = driver.getTitle();
  Set<String> Allwh = driver.getwindowHandles();
 for (String wh : Allwh){
    driver.SwitchTo().window(wh);
    String ActualTitle = driver.getTitle();
  if (ActualTitle.equals(Expectedtitle)) {
  }
  else {
      driver.Close();
  }
 }
}
}
```

write a Script to Close only parent browser.
(user just driver. Close after getting url)

write a Script to Close a Specific browser.

Summary of the popup.

	Popup	Solution
1.	Alert or javascript Popup	driver. SwitchTo().alert() - accept(), dismiss(), getText()
2.	Hidden Division or Calender popup	By using find element method
3.	File upload popup	BrowserButton. SendKeys ("absolutepath")

4.	File download popup	By usin Robot class (For Chrome no need to handle)
5.	print popup	Robot Class (For Chrome - find Element()
6.	Notification popup	Browser option class (chrome options class - add Argument()
7.	Chilwindow/Child browser popup	Get window handle() & Switchto().window(
8.	Authentication popup	By sending user name and password along with url in get() method.

1/10/2020

Handling Tabs : Tabs is also treated as new window in Selenium. Here we are going to handle the tab same ways as we handle browser window or Child browser popup.

write a Script to count the number of tabs present in actitime application. after clicking actitime link.

```
Public class HandlingTabs {
Static {
System.Setproperty ("webdriver.Chrome.driver", "./driver/
                                        Chromedriver.exe");
}
Public Static void main (String [] args) throws InterruptedException {
    webDriver driver = new ChromeDriver();
    driver.get ("https://demo.actitime.com/");
    driver.findElement (By.linkText ("actiTime Inc.").click();
    Thread.Sleep (5000);
    Set <String> Allwh = driver.getWindowHandles();
    int Count = Allwh.Size();
    System.Out.println (Count);
    driver.quit();
}
}
```

how do you Close the Current tab?

driver.Close (Control will be always present in the Current tab or parent tab Similar to Child browser popup)

How do you Close all the tabs?

driver.quit.

How do you Close all the tabs without using quit.

```
public Class CloseWithoutQuit1{
Static {
System.Setproperty ("webdriver.Chrome.driver", ". /drivers /
                                        Chromedriver.exe");

}
public Static void main (String [] args ) throws InterruptedException {
webDriver driver = new ChromeDriver ();
driver. get ("https : // demo. actitime. com /");
driver. findElement (By. linkText ("actitime Inc.")). Click ();
Set < String > Allwh = driver. getwindowHandles ();
int Count = Allwh. Size ();
System. out. println (count);
for (String wh : Allwh ) {
  driver. SwitchTo (). window (wh);
  driver. Close ();
}
}
}
```

To Close all tabs without using forloop.

```
public Class withoutForloop {
Static {
System. Setproperty ("webdriver. Chrome. driver", ". / drivers /
                                        Chromedriver. exe");

}
public Static void main (String [] args )
    webDriver driver = new ChromeDriver ();
    driver. get ("https : // demo. actitime. com /");
```

```
driver.findElement(By.linkText("actiTime Inc")).click();

Set<String> Allwh = driver.getWindowHandles();

int count = Allwh.size();

System.out.println(count);


Iterator<String> it = Allwh.iterator();
    String parentwh = it.next();
    String childwh = it.next();
    driver.switchTo().window(parentwh);
    driver.close();
    driver.switchTo().window(childwh);
    driver.close();
}
}
```

Automate the following Scenario.

1. open the Chrome browser.
2. Enter the URL of actitime (http://demo.actitime.com/)
3. Login to the application (UN: admin PW: manager)
4. Click on 'About your actitime' present in help dropdown
5. Click on read Service agreement link present in the popp
6. print all the header present in licence agreement tab
7. Close the aggrement tab
8. close the main tab.

→
```
public class HandlingTabs2 {
Static {
System.setProperty("webdriver.Chrome.driver", "./drive
                                    Chromedriver.exe");
}
```

```
public static void main (String [] args) {
    webDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.get("https://demo.actitime.com/");
driver.findElement(By.id("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.Xpath("//div[text()='Login']")).click();
driver.findElement(By.Xpath("(//div[@Class='menu_icon'])[4]")).
                                                    click();
driver.findElement(By.LinkText("About your actiTIME")).click();
driver.findElement(By.LinkText("Read Service agreement")).click();
Set <String> Allwh = driver.getwindowHandles();
Iterator <String> it = Allwh.iterator();
    String ptab = it.next();
    String ctab = it.next();
    driver.switchTo().window(ctab);
List <webElement> Allheadings = driver.findElements(By.Xpath("//u"))
for (int i = 0; i < Allheadings.size(); i++) {
    String HeadingsText = Allheadings.get(i).getText();
        System.out.println(HeadingsText);
  }
    driver.close();
    driver.switchTo().window(ptab);
    driver.close();
  }
}
```

Handling Mouse actions :

By using mouse we can perform the following actions.

1. Handling dropdown menu (By using mouse Hover)
2. Context Click (Right Click)
3. Drag and drop
4. Double Click

How to handle drop down menu? (or handling mouse hover) Mouse hover means your mouse pointer to a particular position.

Dropdown menu is an element on which if we move the mouse pointer it will display the list of options. To handle drop down menu we use 'moveToElement()' method of Actions class. moveToElement() is nothing but mouse hover.

In Selenium 'Action' is a interface & 'Actions' is a Class. 'Actions' Class is present inside the 'Interactions package'. 'Actions' Class is mainly used for mouse actions. Actions class has a parametrised Constructor, where it takes webdriver as an argu whenever we call any method of actions class, we have to use 'Perform method' at the end of the statement.

imp note : all the methods of Actions Class are overloaded method.

Automate the following Scenarios :

a. Open the browser

b. Go to vtiger.com

c. mouse hover to resource tab (mouse pointer to resources)

d. Click on Contact in dropdown menu

e. get the bangalore phone number & print it on the Console.

02/10/2020

```
Public Class MouseHover {
static {
System.Setproperty ("webdriver.chrome.driver", ".\driver\chromedriver
.exe");
}
public static void main (String[] args) {
WebDriver driver = new ChromeDriver(); // opening browser
driver.manage().timeouts().implicitlywait (10, TimeUnit.SECONDS);
// Go to vtiger.com
driver.get ("http://www.vtiger.com/");
// mouse hover to resource tab
WebElement Resources = driver.findElement (By.partialLinkText
("Resources"));
Action a = new Actions (driver);
a.moveToElement (Resources).perform();
// Click on Contact in dropdown menu.
driver.findElement (By.Xpath ("//h6[contains (text(), 'contact')]")
.click();
// get the bangalore phone number and printing on Console.
String phonenumber = driver.findElement (By.Xpath ("//p[
Contains(.,'Bengaluru')]/../p[2]")).getText()
System.out.println (phonenumber);
// Close the browser
driver.close();
}
}
```

? How do you handle Context click (Context menu):

Right click in mouse is called Context link

1. when we right click on any element we get list of options which is called as Context menu.

2. To right click on the element we use 'Context Click' method of action class.

3. To select the required option in the Context menu we prove the Shortcut Keys such as 't' for new tab, 'w' for new window.

write a Script to open the actitime link in new window.

```
public class RightClick {
Static {
System.Setproperty ("webdrivers.chrome.driver", "./drivers/Chromedriver.exe");
}
public Static void main (String [] args) throws AWTException {
  webDriver driver = new ChromeDriver();
  driver.get ("https://demo.actitime.com/");
webElement target = driver.findElement (By.LinkText("actiTIME Inc."))
    Actions a = new Actions (driver);
      a.ContextClick (target).Perform();
    Robot r = new Robot();
    r.KeyPress (KeyEvent.VK_w);
      driver.quit();
  }
}
```

write a Script to perform drag and drop.

```
public class DragDrop {
Static {
System.Setproperty ("webdriver.chrome.driver", "./driver/Chromedriver.exe");
}
public Static void main (String [] args) throws InterruptedException
    WebDriver driver = new ChromeDriver();
```

```
driver.get("http://dhtmlgoodies.com/submitted-scripts/i-google-
                like-drag-drop/index.html");

WebElement Source = driver.findElement(By.xpath("//h1[text()='Block 1']"));
WebElement target = driver.findElement(By.xpath("//h1[text()='Block 4]"));
    Thread.Sleep(3000);

Action a = new Actions(driver);
 a.dragAndDrop(Source, target).perform();
 }
}
```

How do you perform double click in Selenium.

We can use Actions Class to do double click on the element.

Example : a.doubleClick(webElement).perform();

Assignment :

Go to the vtiger.com application

Click on 'Customers' option present under 'resource' dropdown menu

Double click on 'read full story' button and check whether

'Hacker Earth' page is displayed or not.

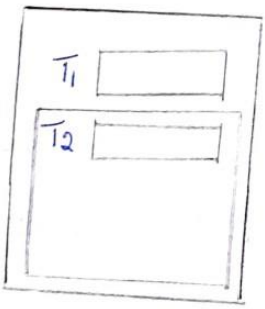
5/10/2020

Handling Frames :

Webpage inside the another webpage is Called as Embedded webpage

Developers Creates embedded webpage using iframe tag.

While automating the element, if it is present inside the frame, we should transfer the Control in to the frame using SwitchTo().frame() Statement.



page1.html

HTML Code to Create first frame : (Content of page 1.html)

T1 : <input id = "d1" type = "text" />
<iframe id = "f1" Src = "page2.html" > </iframe>

HTML Code to Create Second frame : (Content of page 2.html)

T2 : <input id = "d2" type = "text" />

⇒

```
Public class HandlingFrames {
 Static {
  System.setproperty ("webdriver.Chrome.driver", "./driver/
                                        Chromedriver.exe");
 }
public Static void main (String [] args) {
    WebDriver driver = new ChromeDriver();
driver.get ("file:///C:/users/Sambkruthi/Desktop/page1.html")
```

```
driver. switchTo(). frame(0);
driver. findElement (By.id ("d2")). sendKeys ("jsp");
driver. switchTo() . parentFrame();
driver. findElement (By. id ('d1")). sendKeys ("qsp");
}
}
```

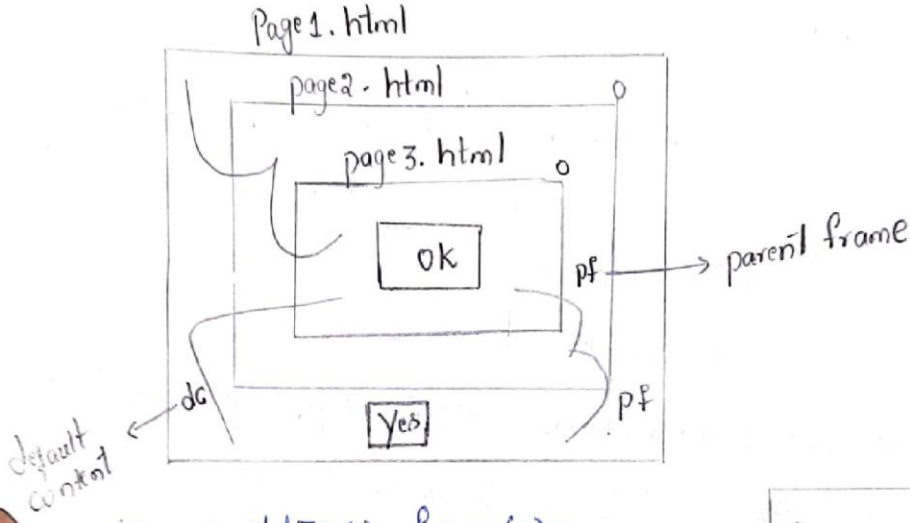
⇒

```
public class HandlingFrames1 {
static {
System. setproperty ("webdriver. chrome. driver", ". /driver/chromedriver.exe");
}
public static void main (String[] args) {
webDriver driver = new ChromeDriver();
driver. get ("file:///c:/users/samskruthi/Desktop/page1.html");
driver. findElement( By. id ("d1")). sendKeys ("q")
driver. switchTo() . frame (0);
driver. findElement (By. id ("d2")). sendKeys ("j");
driver. switchTo(). defaultContent();
driver. findElement (By. id ("d1")). sendKeys ("s");
driver. switchTo(). frame ("f1");
driver. findElement (By. id ("d2")). sendKeys ("s");
driver. switchTo() . parentFrame();
driver. findElement (By. id ("d1")). sendKeys ("p");
webElement frameWE=driver. findElement (By. Xpath ("//iframe"));
driver. switchTo() . frame (frameWE);
driver. findElement (By. id ("d2")). sendKeys ("p");
}
}
```

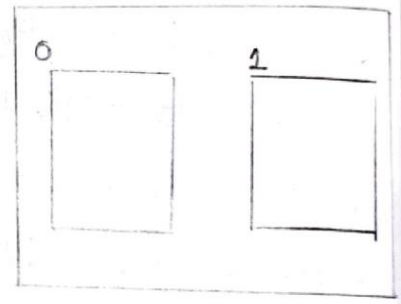
Note: In the above example frame method is overloaded. it takes only one argument of any of the following 3 types.

1. int (index of the frame and starts from zero)
2. String (id of the frame)
3. webElement (address of the frame)

Nested frame:




```
driver.switchTo().frame(0);
driver.switchTo().frame(0);
driver.fE.click();
```
→ OK

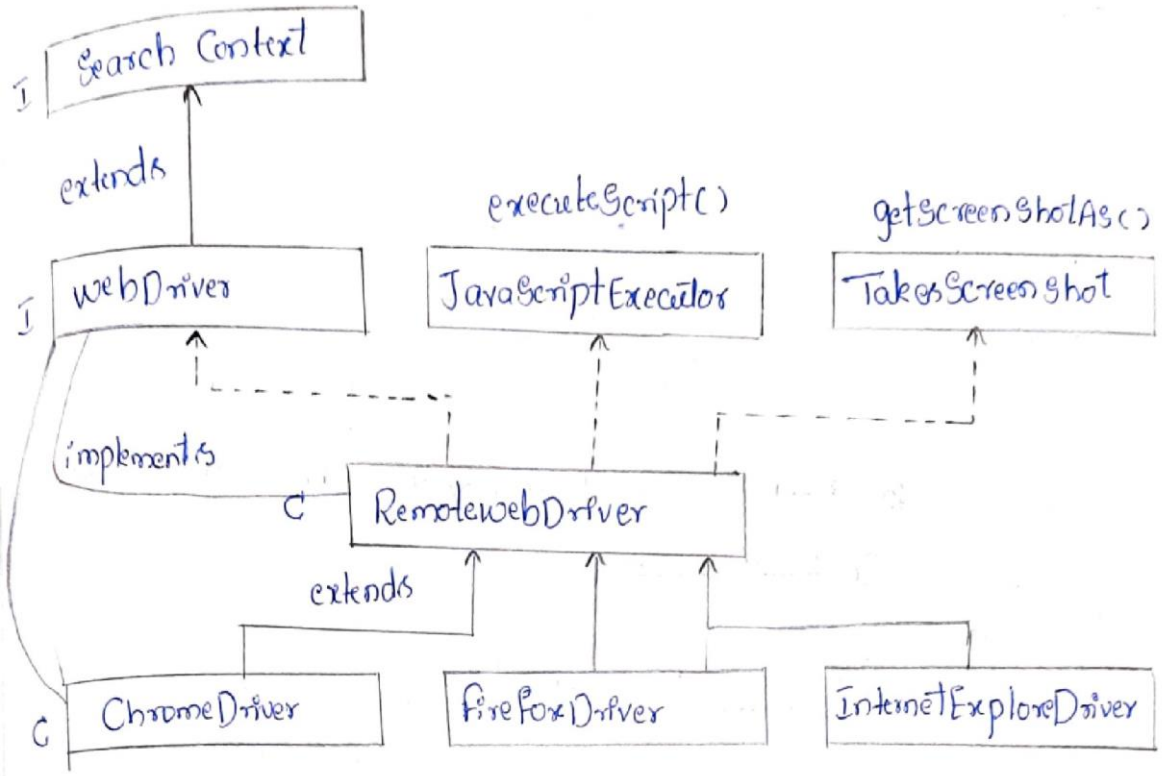



```
driver.switchTo().parentFrame();
driver.switchTo().pF();
driver.findElement.click();
```
→ Yes

6/10/2020

Handling disabled elements and scroll bar.

The executeScript() method is declared in JavaScriptExecutor interface which is implemented in RemoteWebDriver class. Since we already upcasted browser specific classes to webDriver interface, this (executeScript()) method will be hidden, in order to access this method we should downcast it (remotewebdriver) or typecast the object to javaScriptExecutor.



Ex: html code

UN ~~pw~~ : <input type = "text" id = "t1" disabled / >

pw : <input type = "text" id = "t2" disabled >

<input id = "d1" type = "button" value = "Login"/>

```
=> public class HandingDisabled {
    static {
    System.Setproperty ("webdriver.chrome.driver","./drivers/
                                                Chromedriver.exe");
    }
    public static void main (String[] args) {
      webDriver driver = new ChromeDriver();
    driver.get ("file:///C:/users/Samskruthi/Desktop/disabled.html")
    driver.findElement (By.id ("t1")).sendkeys ("admin");
    JavaScriptExecutor j = (JavaScriptExecutor) driver;
     j.executeScript ("document.getElementById ("t2").value="manager")
    }
}
```

In order to validate the java script in the browser inspect the element & click on console tab present in the developer tool bar & type the script.

document.getElementById('d3').click() → for clicking on element

document.getElementById('d2').value='' → for deleting the value present in textbox

Write a script to scroll 2000 pixel vertically in bbc.com/news application

```
Public Class ScrollingVertically {
  Static {
  System.Setproperty ("webdriver.chrome.driver", ".\driver\
                                                  chromedriver.exe");
  }
  Public Static Void main (String[] args) {
    webDriver driver = new ChromeDriver();
    driver.get ("https://www.bbc.com/news");
    JavaScriptExecutor j=(JavaScriptExecutor) driver;
    j.executeScript ("window.ScrollBy (0, 2000)");
  }
}
```

Write a script to scroll to the particular element in BBC application

```
Public Class ScrollingVertically {
  Static {
  System.Setproperty ("webdriver.chrome.driver", ".\driver\
                                                  chromedriver.exe");
  }
  Public Static Void main (String[] args) {
    webDriver driver = new ChromeDriver();
    driver.get ("https://www.bbc.com/news");
    int y = driver.findElement (By.Xpath ("//h2[text()='Most read']")
                                          getLocation().getY();
    System.out.println(y);
    JavaScriptExecutor j = (JavaScriptExecutor) driver;
    j.executeScript ("window.ScrollBy (0,"+y+")");
  }
}
```

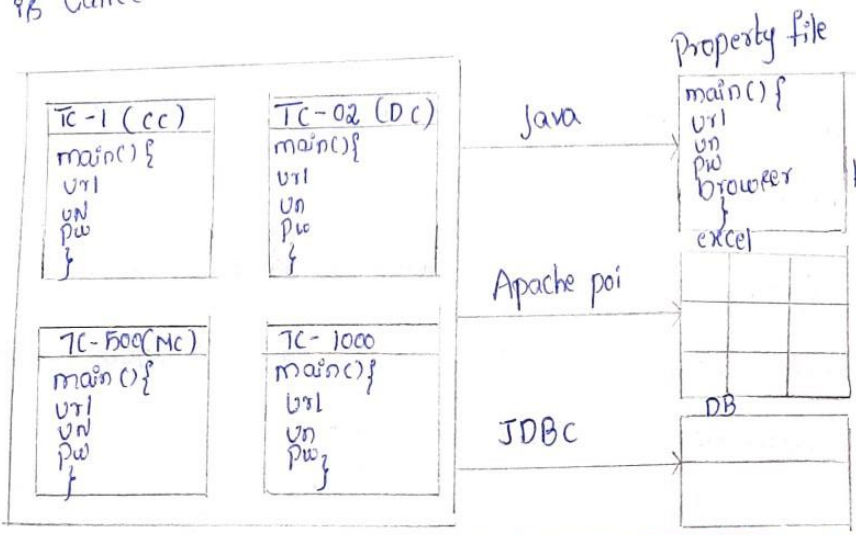
Write a Script to Scroll till the end of the webpage

```
Public Class ScrollEnd{
Static {
System.Setproperty ("webdriver.Chrome.driver", ".| driver|
                                        Chromedriver.exe");

}
public Static void main (String [] args) throws InterruptedException {
  webDriver driver = new ChromeDriver();

driver.manage().window().maximize();
driver.get ("https://www.bbc.com/news");
~~JavaScriptExecutor~~ j = (~~JavaScript Ex~~ win
JavaScriptExecutor j = (JavaScriptExecutor) driver;
j.executeScript ("window.ScrollTo(0,document.body.ScrollHeight)")

thread.sleep (3000);
j.executeScript ("window.ScrollTo (0,0)"); // again to top

  }
}
```

07/10/2020

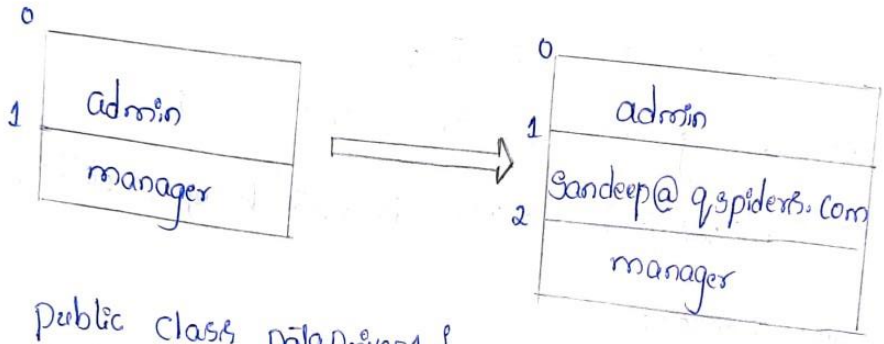
Data Driven Testing:

Testing the application with multiple test data which is kept in external resource file like excel, property file and database etc is called as data driven testing.



⇒ public class DataDriven {
public static void main (String [] args) {
ArrayList < String > lst = new ArrayList < String > ();
lst. add ("admin");
lst. add ("manager");
System. out. println (lst. get (1));
}
}

// if try to add here lst. add ("Sandeep@qspiders. com");
instead of getting password we get email.



⇒ public class DataDriven1 {
public static void main (String [] args) {
HashMap <String, String> hs = new HashMap <String, String> ();
hs. put ("username", "admin");
hs. put ("email", "Sandeep@qspiders. com"); // even though email is added b/w un & pw.
hs. put ("password", "manager");
System. out. println (hs. get ("username")); // still get same data
System. out. println (hs. get ("password"));
}
}

→ If we use index, we will not retrieve the same data, if any data is added or deleted tomorrow.

→ we get same o/p even if new data is added or deleted in between tomorrow if we use key value pair (of maps). Same concept is used to store in property file.

→ As per the rule of automation, test data should not be hard coded within the test script. Because modification & maintainance of test data is difficult & whenever we want to run the script with different set of data. Insted of hard coding, get the data from external resource file & run the test script. This is called as Data Driven testing. external resource might be property file (.property) or excel (.xlsx) or DataBase

Handling property file :

Copy this code in notepad & save it as any file name with '.property' extension.

```
url htt://demo.actitime.com/
username admin
password manager.
```

→ In property file data will be stored in the form of key value pair. Key & value should be seperated by single space.

→ By default all the data available in property file is string.

```
⇒ public class HandlingProperty {
  Public static void main (String[] args) throws IoException {
    // get java representative object of the physical file
    FileInputStream fis = new FileInputStream("./data/commondata.property");
    Properties p = new properties (); // Create object property class
    p.load(fis); // load the file
    String url = p.getproperty("url"); // get data(value) by passing k
    String un = p.getproperty ("username");
    String pw = p.getproperty ("password");
    System.out.println (url);
    System.out.println (un);
    System.out.println (pw);
  }
}
```

Advantages of property file :

1. It is very faster in execution

2. It is very light weight when compared to any other external resource file.

→ In order to read the data from the property file we should get the java object of the physical file by using FileInputStream class.

→ Then by taking the help of property file, load the file & get the value by using the key. (by using getproperty()).

Assignment.

8/10/2020

Data Driven from Excel file :

Apache poi (poor obfuscation implementation)

we need to use apache poi plugin or third party tool in order to read the data & write the data from all Microsoft documents like .xls, .xlsx, .ppt, .docx, .outlook. Apache poi is a free & open source library tool similar to Selenium.

Installation of Apache POI :

1. Go to google and search for Apache POI download.
2. Click on first link & navigate to apache Community.
3. under binary distribution click on poi-bin-4.1.2 zip.
4. Click on first URL and download the zip file.
5. unzip the folder & copy all the jar's (20 jar files) & paste it inside the jar folder in the Eclipse
6. Add all the jar's to the build path.

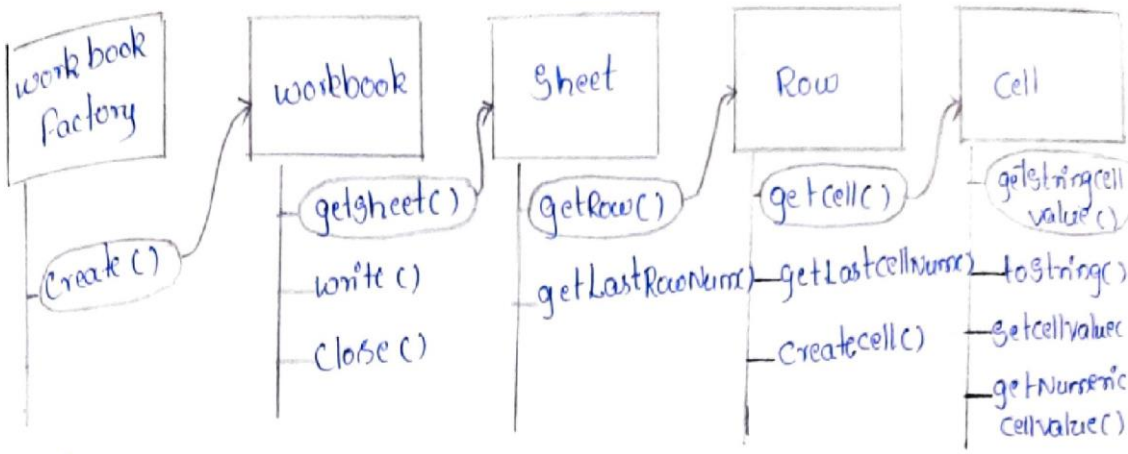
Note : Make sure that all the jars available in lib folder, ooxml-lib should also be added to jars folder in eclipse.

```
=> public Class ExcelData {
   public static void main (String[] args) throws EncryptedDocumentException, IOException {

// get java representative object of the physical file
   FileInputStream fis = new FileInputStream ("C:\\users\Samskrut\\Desktop\\testscript.xlsx");

// open the excel in read mode or Create workbook.
   workbook wb = workbookFactory.Create (fis);
```

```
// get the control of the sheet, row & cell then read the data
String value = wb.getSheet("CreateCustomer").getRow(1).
                                    getcell(0).getstringValue();

// print the data on the Console.
System.out.println(value);
}
}
```

Another method

```
Public class ExcelData {
public static void main (String[] args) throws EncryptedDocu
                                    mentException, IoException {

// get java representative object of the physical file

~~FileStream~~
FileInputStream fis = new FileInputStream(".\data\TestScript.xlsx")
// open the excel in read mode or Create workbook
workbook wb = workbookFactory.Create(fis);
// get the Control of the sheet
Sheet sheet = wb.getsheet("CreateCustomer");
// get Control of the row
Row row = Sheet.getRow(1);
// get Control of the Cell
Cell cell = row.getcell(0);
// Read the String data
String value = Cell.getstringcell value();
// print it on the Console.
System.out.println(value);
}
}
```



09/10/2020

write a program to write the data back to excel sheet

[To convert External file into java readable format we use "FileInputStream".]

```
public class writeExcelData {
public static void main (String[] args) throws EncryptedDocumentException, IOException {

// read the data from external resource or get the java object ... in read mode

FileInputStream fis = new FileInputStream ("./data/testscript.xlsx")

// open or go to the workbook & write the data.
workbook wb = workbookFactory.Create (fis);
wb.getsheet ("Create Customer").getRow(1).getcell(4).setcellvalue ("Pai");

// open the file in write mode.
FileOutputStream fos = new FileOutputStream("./data/testscript.xlsx")
// Save the workbook in the write mode (actual writting happes)
wb.write (fos);
// Close the workbook
wb.Close();
}
}
```

write a program to read multiple data from excel.

```
Public class MultipleExcelData {
public static void main (String [] args) throws EncryptedDocumentException , IOException {
FileInputStream fis = new FileInputStream ("./data/testscript.xlsx")
workbook wb = WorkbookFactory. Create (fis);
// get the row Count
int rowCount = wb. getSheet ("InvalidLogin"). getlastRowNum();
// Execute the for till the last row.
for (int i=1 ; i<= rowCount ; i++) {
String un = wb. getSheet ("InvalidLogin"). getRow (i). getcell (0).
                                        getStringcellvalue ();
String pw = wb. getSheet ("InvalidLogin"). getRow (i). getcell (i).
                                        getstringcellvalue();
System. out. println (un+"\t"+pw);
}
}
}
```

Note: whenever we are reading data from excel the file extension should be .xlsx. All the apache files API's should be imported from org. apache. poi. ss. usermode package while writing data back to excel, specific excel file should be closed compulsory i.e. you should close the excel file ~~before~~ after doing any modification or addition.

Advantage of data driven testing

1. Reusability or common data and test script data.
2. Modification of data in excel file or external resource file is easier.
3. Maintenance of data in excel file or external resource file is easier.

4. Test data can be created explicitly prior to the test execution
5. we can test the application with the huge data.

Generic Library :

Coding Standard :

```
/**
* This is the generic class for data driven testing
* @author Sandeep Samskruthi
*
*/
```

To get above lines type /** above the class name (then enter key)
To get below lines type /** above the method (then enter key).

```
/**
* This is the generic class for data driver testing
* @ author Samskruthi
*
*/
public class FileLib {

/**
* Reading the data from property file
* @ param key
* @ return value
* @ throws IOException
*/
public String getPropertyValue (String key) throws IOException {
FileInputStream fis = new FileInputStream ("./data/Commondata.properties")
Properties p = new Properties ();
p. load (fis);
String value = p.getproperty (key);
return value ;
}
}
```

10/10/2020

```
/**
* to read the data from excel
* @param Sheetname
* @param row
* @param cell
* @return String Data
* @ throws EncryptedDocumentException
* @ throws IoException
*/
public String getExcelData (String Sheetname, int row, int cell) throws
                EncryptedDocumentException, IOException {
  FileInputStream fis = new FileInputstream ("./data/testscript.xlsx
  workbook wb = workbookFactory.Create (fis);
  String data = wb.getSheet (Sheetname).getRow (row).getcell (cell)
                                getString (cell value ());
  return data;
}
/**
* to write the data to excel
* @param Sheetname
* @param row
* @param Cell
* @ ~~param~~ return value
* @ throws EncryptedDocumentException
* @ throws IOException
*/
public void SetExcelData ( String Sheetname, int row, int cell,
                String value) throws EncryptedDocumentException {
FileInputStream fis = new FileInputstream ("./data/testscript.xlsx");
workbook wb = workbookFactory. Create (fis);
wb.getSheet (Sheetname).getRow (row).getcell (cell).Setcellvalue (value)
```

```
Fileoutputstream fos = new Fileoutputstream (".\data\testscript.xlsx")
wb.write(fos);
wb.close();
}
}
```

12/10/2020.

```
/**
* Generic Class for all webdriver related elements
* @author Samskruthi
*
*/
public class webDriverCommonLib {
/**
* implicit wait
* @param driver
public void waitForPageToLoad (webDriver driver){
  driver.manage().timeouts().implicitlywait (10, TimeUnit.SECONDS)
}
/**
* Explicit wait for visibility of the element
* @param driver
* @param element
*/
public void waitForElement (webDriver driver, webElement element) {
  webDriverwait wait = new webDriverWait (driver, 10);
  wait.until (ExpectedConditions.visibility of (element));
}
/**
* CustomWaitForElementIsEnabled
* @param element
*/
```

```
Public void CustomWaitForElementIsEnabled (webElement element){
    int i=0;
    while (i<100){
        try {
            element.isEnabled();
            break;
        }
        Catch (Exception e) {
            i++;
        }
    }
}

/**
* Select by using index
* @ param element
* @ param ~~elem~~ index
*/
Public void Select (webElement element, int index) {
        Select s = new Select (element);
        s. SelectByIndex (index);
    }
    /**
    * Select by using visible text
    * @ param element
    * @ param text
    */
    public void select (webElement element, String text) {
        Select s = new select (element);
        s. SelectByVisibleText (text);
    }
}
```

To Call this above method Create an object of the Class & by using reference variable Call the method.

```
webDriver driver = new ChromeDriver();// web driver Calling
webDriverCommonLib w = new webDriverCommonLib(); // Creating
                                                    object of generic Class
w.waitForpageTo Load (driver); // implicit wait Calling.
```

Advantages :

1. Generic library is one of the component in automation framework, it contains common classes which can be reusable for all the test script & any project as well.

2. Generic library class contains reusable methods which is created by framework developer, & it is being shared to the team members via GitHub.

3. All the generic classes are available in seperate package. the package name should be com. projectName. generics

4. It contains two classes.

FileLib

- getpropertyValue (String key) : String → it is used to read the data from property file based on the key.
- getExcelValue (String sheetname, int rownum, int cellnum) : String → It is used to read the data from excel.
- SetExcelData (String sheetname, int rownum, int cellnum, String val : void → it is used to write the data to excel.

webDriver CommonLib

- waitForPageToLoad () : void → used for implicit wait
- waitforElement (webDriver driver, webElement) : void → used for explicit wait.
- CustomwaitForElement (webElement element) → used for Custom wait
- Select (webelement element, String text)
  Select (webelement element, int index)
  → used to select value from drop down

Note : we should not make every method as static because it will take more time to execute. So make it as non static.

Advantages of generic Library:

1. Common Classes which can be used to any test script & any project.
2. Reusability of Codes
3. Test script development is faster
4. No need to put effort to rewrite the same program
5. Code optimization.
6. No need to remember the syntax for every operation (instead, we should call reusable call reusable custom methods available in generic package)
7. It can easily shared to all the team members via GitHub.

13/10/2020.

unit testing Frame work Tool.

unit testing Frame work tool

TestNg < java, .net

Junit – java

Nunit – .net

PyDev – python

Rspec – ruby

TestNG (TestNextGeneration):

TestNG is a unit testing framework tool which is mainly used for batch execution. (it is third party open source tool). Basically testNG is used by developers to perform unit testing and it is also used in Selenium to perform black box testing. (testNG is a plugin for eclipse & it is inspired by junit & Nunit with some additional frames).

Advantages or additional features of TestNG:

1. Batch execution (Run multiple test cases).
2. Group Execution (ex. To run only Smoke test cases or integration test cases)
3. parallel execution
4. Generate reports (HTML reports)
5. Listeners ~~annotations~~ features
6. Additional annotations.
7. Run only failed test Scripts.

Installation steps of TestNG.

i. Go to eclipse & click on help & Select eclipse market place.
ii. Type TestNG in Search textbox & click on Search button
iii. Click on install button present under testNG for eclipse
iv. Click on Confirm (Click on next).
v. Select the radio button 'I agree to licence agreement'
vi. And click on finish.

After installation in order to check or verify testNG tool is installed or not, go to eclipse → window → Show view → others → expand java folder → Search for testNG.

Another way of installing testNG (Select install anyway if it gives any warning)

Eclipse → help → install new Software → work with → type "http://dl.bintray.com/testng-team/testng-eclipse-release/."
→ click add and then follow the instructions till finish.

Note: After testNG installation to the eclipse add testNG library for all the projects in eclipse (Required projects).

Right click on the project folder, go to build path, & Select add libraries.

Select TestNG & click on next & click on Finish.

Note : while Creating testNG class we should not use following thing.

a. Do not use default package

b. Not use main() method

c. No System.out.println()

In Case of automation, unit testing framework tool like TestNG will be used to achieve batch execution without any manual interruption. And it will generate HTML report & also provide Screen shot for failed test Scripts. In order to achieve batch execution every test Script automation is executed using testNG annotation. TestNG also handle framework Components during batch execution & it is main Controller of the framework.

ex :-
```
public Class Demo {
    @test
    public Class TestDemo () {
    Reporter. log ("welcome to testng", true);
    }
}
```

when we execute the above Code it will automatically generate the execution result (report) in HTML Format. In order to See it do the following procedure.

1. Refresh the java project (F5) which will displays test output folder.

2. Expand the folder & do right click on emailabe-report.html file.

3. Go to open with & Select web browser.

TestNG Suite (To perform batch execution):

TestNG Suite is a Xml file which Contains list of TestNG classes which are to be executed. TestNG Classes are test Classes means a Class which Contains test methods (@Test). Test method means a method which is having '@test' annotation Suite file is used for batch execution. To Create it

i. Right Click on java project.

ii. Go to TestNG Select Convert it into TestNG.

iii. click on finish, it Creates TestNG.xml Suite file inside the java project.

To execute it:

a. right Click on Xml file.

b. go to ~~Test~~ run as and Select testNG Suite.

(or open the Suite file (testng.xml) & Click on run button..

Batch execution: Collection of multiple test Scripts are Called as batch or Suite. Execute all of them together via testNG Suite.xml file is Called as batch execution.

14/10/2020

Content of testNG.xml Suite file:

```
<Suite name="Suite">
  <test name="Test">
    <classes>
      <class name="Com.actitime.test.CreateCustomer"/>
      <class name="Com.actitime.test.modifyCustomer"/>
    </classes>
  </test>
</Suite>
```

1. If the Class Contains multiple test methods in which order they are executed ?

→ Alphabetical order (ascending)

2. How to execute the test methods in required order ?

→ using Priority

Syntax : @Test (priority = 1)

Note : Default priority is 0 (Zero).

- If the priority is duplicate then those methods will be executed in alphabetical order.
- we can specify -ve value for priority & it will execute them in ascending order.
- Variables and decimals numbers are not allowed.

3. How do you run a test method multiple times ?

→ using InvocationCount.

Syntax : @Test (invocationCount = 1)

- Defaut InvocationCount is 1 (one).
- If we specify '0' or -ve number it will not execute that test method. Variables & decimals (fraction) are not allowed.

4. How do you make a test depends on other test ?

→ By using DependsOnMethod option.

Syntax : @Test (dependsOnMethods = "CreateCustomer")

Note : If both priority and dependency are specified it will consider the dependency.

5. What if 2 methods are ~~m~~ depends on each other ?

→ we get TestNG Exception

→ Error is cyclic dependencies.

6. How do you disable the test method (test case) ?

→ @Test (enable = false) or (invocation Count = 0 or -ve value)

7. How do you intentionally fail the test?

→ using Assert.fail();

Note: In order to open the report in Excel format.

a. open the report & Right click on the report & select export to microsoft excel.

b. Then click on import and ok.

```
⇒ public class Demo {
  @Test (priority = 2)
  public void CreateCustomer() {
  Reporter.Log ("CreateCustomer", true);
  }
  @Test (priority = 1, dependsOnMethods = "CreateCustomer")
  public void DeleteCustomer() {
  Reporter.Log ("DeleteCustomer", true);
  }
}
```

Annotations: Annotation is a meta data which provides a special instruction to the java compiler during runtime (it is one of the block in java which is used to develop / provide custom instruction).

Optional Annotations:

@parameters

@AfterGroups

@BeforeGroups

@Factory → design pattern

@Guice → dependency injection

@ignore → ignore the test cases - skips.

@Listeners

@Notification

@objectFactory

@TestInstance.

Default order of TestNG annotation:

1. @Before Suite
2. @Before Test
3. @Before Class
4. @Before Method
5. @Test
6. @After method
7. @After Class
8. @After Test
9. @After Suite.

⇒
```
public class Demo{
  @Test
  Public Void Create Customer(){
  Reporter. Log ("Create Customer", true);
 }
 @Test
  Public Void DeleteCustomer(){
  Reporter. Log ("Delete Customer", true);
 }
 @BeforeMethod
  Public void login (){
  Reporter. log ("login", true);
 }
 @AfterMethod
  public void logout (){
   Reporter. Log ("logout", true);
 }
}
```

O/P :- login
Create Customer
Logout
Login
DeleteCustomer
Logout.

⇒ public class Demo {

```
@Test (priority = 1, invocationCount = 2)
public void editCustomer () {
Reporter. log ("editCustomer", true);
}
@Test
public void registerCustomer () {
Reporte . Log ("registerCustomer", true) ;
}
@Test
public void deleteCustomer () {
Reporter. Log ("DeleteCustomer", true);
}
@BeforeMethod
public void login () {
Reporter. log ("login", true);
}
@AfterMethod
public void logout () {
Reporter . Log ("logout", true) ;
}
@Before Class
public void openBrowser () {
Reporter. log ("openBrowser", true);
}
@After Class
public void closeBrowser () {
Reporter. log ("closeBrowser", true);
}
}
```

o/p :

openBrowser
Login
Delete Customer
Logout
Login
Register Customer
Logout
Login
editCustomer
Logout
Login
editCustomer
Logout
Close Browser.

15/10/2020

@ Before method : before method annoation will be executed, before executing of each @ test method within a Class, in real time, before method annotation will be used to develop Common pre Conditions which is required for all the test scripts like login program.

@ After Method : After method annotation will be executed, after executing every @ test method in a class. in real time after method annotation will be used to develop post Condition like logout program.

@ Before Class : Before Class method annotation will be executed before every test Class. in real time @ Before Class method will be used to open the browser.

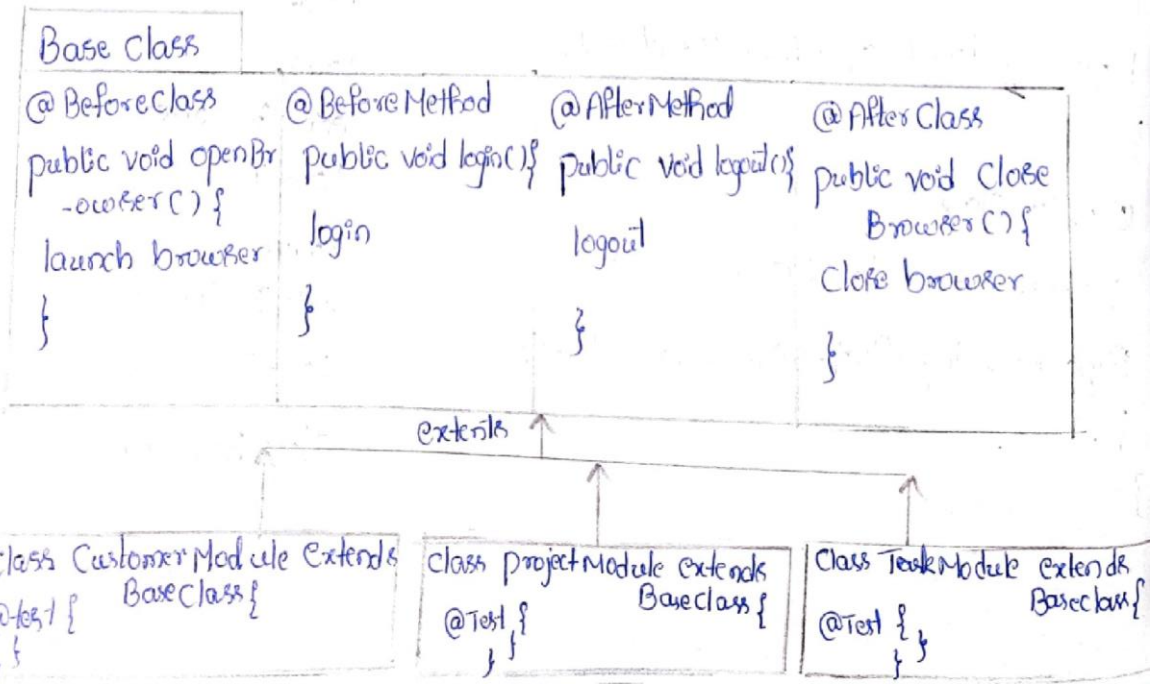
@ After Class : After Class annotation method will be executed after every test Class (after executing all methods in a class) in real time it will be used to close the browser.

Note : Both the annotations method name shouldn't we same name.

@ Test : This indicates the test method.

Annotation usage in real framework :

Base Class :



⇒ BaseClass: Base class is the parent class in framework. It contains Configuration as shown below.

```
⇒ public Class BaseClass {
  Static {
  System. Setproperty ("webdriver. Chrome. driver", ". / driver /
                                                Chromedriver. exe");
  }
public webDriver driver;
@Beforeclass
public Void OpenBrowser () {
Reporter. log ("openBrowser", true);
driver = new ChromeDriver ();
driver. manage (). window (). maximize ();
driver. manage (). timeouts (). implicitlywait (10, TimeUnit. SECONDS);
}
@Afterclass
public void CloseBrowser () {
Reporter. log ("CloseBrowser", true);
driver. Close ();
}
@ BeforeMethod.
public void login () {
Reporter. Log ("login", true);
driver. get ("http: // localhost/ login. do");
driver. findelement (By. id ("username")). Sendkeys ("admin");
driver. findElement (By. name ("pwd")). Sendkeys ("manager");
driver. findelement (By. Xpath ("// div [. = 'Login' ]")). Click ();

}
@ After Method
Public Void logout () {
   Reporter. log ("logout", true);
   driver. findElement (By. id ("logoutlink")). Click ();
} }
```

```
public Class CustomerModule extends BaseClass {
@Test
public void testCreateCustomer() {
    Reporter.Log("CreateCustomer", true);
  }
}
```

Note: BaseClass will be Created under Com.actitime.generics package. it Contains all the Configuration methods like open browser, login, logout, Close browser etc. all the test classes will be Created under com.actitime.test package. Every test Case Should extends base Class.

16/10/2020

Group execution: Execute Collection of Similar-testScripts is Called group execution.

```
<Suite name = "Suite">
    <groups>
        <run>
            <include name = "Smoketest"/>
        </run>
    </groups>
    <test name = "Test">
        <classes>
        <Class name = "Com.actitime.test.CustomerModule"/>
        <class name = "Com.actitime.test.TaskModule"/>
        <class name = "Com.actitime.test.projectModule"/>
    </classes>
  </test>
</Suite>
```

```
@Test (groups = "regressiontest")
@Test (groups = {"Smoketest", "regressiontest"})
```

In Automation regression testsuite will be divided into two types.

a. Smoke testscripts b. regression testscripts.

In order to achieve Smoke testing we go for grouping execution in testing.

To achieve group execution every testscript should have a group name along with @Test, one testscript can have multiple group name (as shown above).

In case of group execution every Configuration method availab in baseclass should have group name, otherwise those annotation will not participate in group execution.

Ex : @BeforeMethod (groups = {"Smoketest", "regressiontest"})

In order to achive group execution, we should declare group key within testng.xml file. Group key should be declare before test tag & after the Suite tag.

How to achive Regional regression :

In Regional regression we will execute only impacted area or testscript.

In order to achieve regional regression testing we use methods and include tag inside the class tag as shown below.

```
<Suite name = "Suite">
  <test name = "Test">
    <classes>
      <class name = "Com.aditime.test.CustomeModule">
        <methods>
          <include name = "testCreateCustomerwithTaskname"/>
          <include name = "testCreateCustomerwithDescription"/>
        </methods>
      </class>
    </classes>
  </test>
</Suite>
```

How do you execute only failed test cases (test methods)?

By using testng-failed.xml present in the test-output folder.

Assertion:

Assertion is feature available in testing which is used to verify expected result of the test script.

As per the rule of the automation every expected result should be verified with assert statement instead of java if-else statement bcz if-else block doesn't have capacity to fail the test script.

There are two types of Assert statements available in TestNG

1. Assert (Hard Assert)
2. Soft Assert

How do we compare actual value with expected value without using if else statement.

Using assertEquals() of Assert class.

Example: Assert.assertEquals(actual, Expected);

```
⇒ public class Assertion {
        static {
            System.setproperty("webdriver.chrome.driver", "./driver
                                                /chromedriver.exe");
    }
  @Test
  public void testVerifyTitle() {
    webDriver driver = new ChromeDriver();
    driver.get("https://www.google.com/");
    String ExpectedTitle = "Google";
    String ActualTitle = driver.getTitle();
    if (ActualTitle.equals(ExpectedTitle)) {
            Reporter.log("title matching-pass", true);
    }
```

```
else {
  Reporter.log ("title not matching - fail", true);
 }
driver.close ();
}
```

```
public class Assertion1 {
  Static {
  System.setproperty ("webdriver.Chrome.driver", "./driver/
                                        Chromedriver.exe");
  }
 @Test
 public void testVerifyTitle (){
 webDriver driver = new ChromeDriver();
 driver.get ("https://www.google.com/");
 String ExpectedTitle = "Google";
 String ActualTitle = driver.getTitle()
 Assert.assertEquals (ActualTitle, ExpectedTitle);
 driver.close ();
 }}
```

19/10/2020

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

In the above example it will not close the browser if Comparison fails

What are the important methods available under Assert Class (or Soft Assert)?

assertEquals ()

assertNotEquals ()

assertTrue ()

assertFalse ()

assertSame ()

assertNotSame ()

assertNull ()

assertNotNull ()

fail ()

All the above methods are static methods of Assert Class (Hard Assert). In order to continue the execution even after failure of the comparison, we can use SoftAssert. But here all the methods are non static. (Both are having same no methods except assertAll).

```
public class Assertion {
    static {
    System.setProperty ("webdriver.chrome.driver",".\driver\
                                                chromedriver.exe");
    }
  @Test
  public void testVerifyTitle () {
    webDriver driver = new ChromeDriver ();
    driver.get ("https://www.google.com/");
    String ExpectedTitle = "Google";
    String ActualTitle = driver.getTitle ();
    SoftAssert s = new SoftAssert ();
    s.assertEquals ( ActualTitle , ExpectedTitle);
    driver.close();
    s.assertAll ();
  }
}
```

Note : To update status of the comparison in to the result window we should use assertAll () method at the last.

Any statement after assertAll () method will not be executed if comparison fails.

Assert (Hard Assert)	Soft Assert (Verify)
1. All the methods are Static	All the methods are non Stat
2. If the Comparison fails remai-ning statement will not be executed in Current method	Executes remaining statements if even if the Compailation fails.
3. we do not Call assertAll method.	we should Call assertAll method at the last.

How do you execute only failed test Cases (test methods)?

By using testng-failed.Xml present in the test-output folder.

Parallel Execution:

Executing the multiple testscripts with multiple browser Concurrently or at the same time is Called parallel execution.

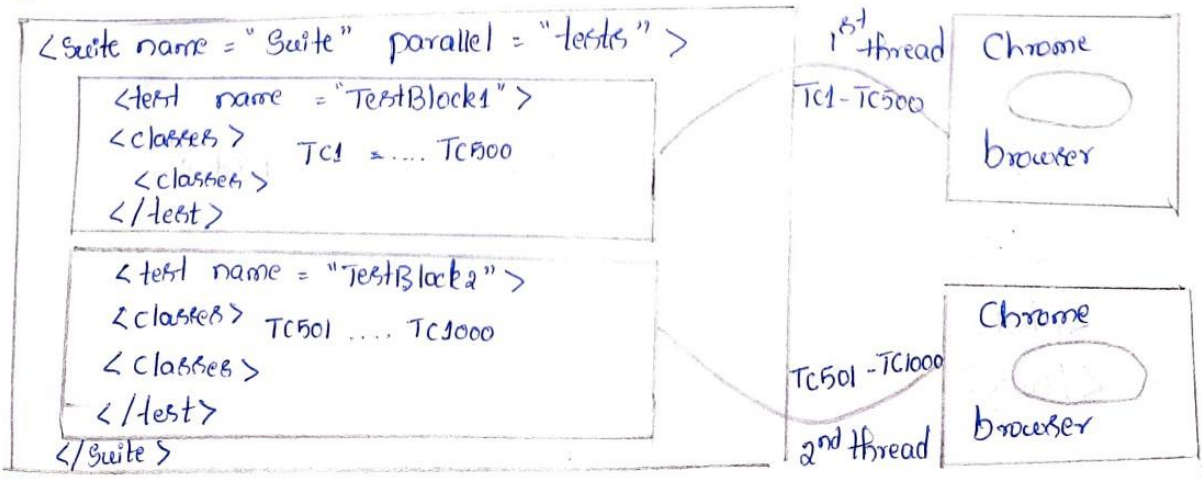
Testng provide two type of parallel execution.

1. Distributed parallel execution
2. Cross browser testing or Compatibility parallel execution.

1. Distributed parallel Execution:

If we execute 1000 Scripts Sequentially it will take 10 hrs to get the result.

In order to get result in early stages, we should go for distributed parallel Execution.



Note : To achieve Distributed parallel execution we should Customize testNG.xml Suite file as shown.

Create multiple test block or test runner & distribute the testscripts & enable the attribute

parallel = "tests" in Suite tag.

Cross Browser Testing or Compatibility parallel execution :

```
<Suite name = " Suite" parallel = "tests">
  <test name = "TestBlock1" >
  <parameter name = "browser" value = "chrome"/>
  <classes>
     <class name = "com.elf.test.CreateCustomer"/>
  </classes>
  </test>

  <test name = "TestBlock2">
  <parameter name = "browser" value = "firefox"/>
  <classes>
  <class name = "com.elf.test.CreateCustomer"/>
   <classes>
  </test>
  </Suite>
```

browser = "chrome"

```
Base Class
@parameters ("browser")
@BeforeTest
  public void
  {....
  }
```

Chrome

Firefox

browser = "firefox"

```
<Suite name = " Suite" parallel = "tests">
  <test name = "TestBlock1">
    <parameter name = "browser" value = "chrome"/>
      <classes>
        <class name = "com.actitime.test.CreateCustomerModule"/>
      <classes>
  </test>
  <test name = "TestBlock2">
    <parameter name = "browser" value = "firefox"/>
      <classes>
        <class name = "com.actitime.test.CreateCustomerModule"/>
      </classes>
  </test>
</Suite>
```

In order to achieve Cross browser testing we should Create multiple test runner (test block) and provide browser parameter for each test runner, whenever we execute xml file each test runner launches the specific browser using threads, then executes all the test scripts w.r.t specific browser for both test runner runs parallely.

In case of parallel execution we should mandatorily use @parameters annotation along with @BeforeTest annotation inside the BaseClass.

@parameters annotation will be used to receive browser parameter from xml file into test script or base class.

In real time to perform Crossbrowser or Cross platform testing, we should go for Selenium grid.

Selenium grid is another tool available in Selenium Community which is used to run in another Virtual machine.

20/10/2020

write a script to take the screenshot of the webpage.

```
public class ScreenShot {
Static {
System.setproperty("webdriver.Chrome.driver", "./driver/
                                                Chromedriver.exe")
}
@Test
public void testScreenShot() throws IOException {
    webDriver driver = new ChromeDriver();
    driver.get("https://www.google.com/");
    TakeScreenShot t = (TakeScreenShot) driver; // take screenshot

    File Src = t.getScreenShotAS(OutputType.FILE); // open empty fil
in Current java project under Screen shot folder
    File dest = new File("./ScreenShot/ss.png"); // Copy paste
the Screen shot to required location from ram.
```

```
Fileutils. CopyFile (Src, dest);
driver. Close ();
  }
}
```

Note : In order to use Fileutils. CopyFile statement we need to download Commons-io-jar from apache Commons (Search in google after typing Commons io download. Click on first link and download Zip file present under binaries).

Listener : Testing Listner is used to monitor the test execution in runtime & perform appropriate action or operation whenever the test case is failed (passed or skipped etc).

⇒ public class ListenerImplementation extends BaseClass implements ITestListener {

```
  public void onTestStart (ITestResult result) {
  }
@Override
public void onTestSuccess (ITestResult result) {
}
@override
public void onTestFailure (ITestResult result) {
String name = result. getName ();
TakesScreenShot t = (TakesScreenShot) driver;
File Src = t. getScreenShotAs( outputtype. FILE);
File dest = new File ("./ ScreenShot /"+ name + ".png");
try {
    Fileutils. CopyFile (Src, dest);
  } catch (IoException e) {
        e. printStackTrace();
  }
}
```

```
@Override
public void onTestSkipped (ITestResult result) {
}
@Override
public void onTestFailedButWithinSuccessPercentage (ITestResult Result){
}
public void onStart (ITestContext Context) {
}
public void onFinish (ITestContext Context) {
}
}
```

whenever we have to implement the Listener feature, we should create a class which implements ITestListener & provide implementation (override) to all methods present in ITestListener.

ITestListener is an interface which is used to receive the failure (all) event from @Listener annotation.

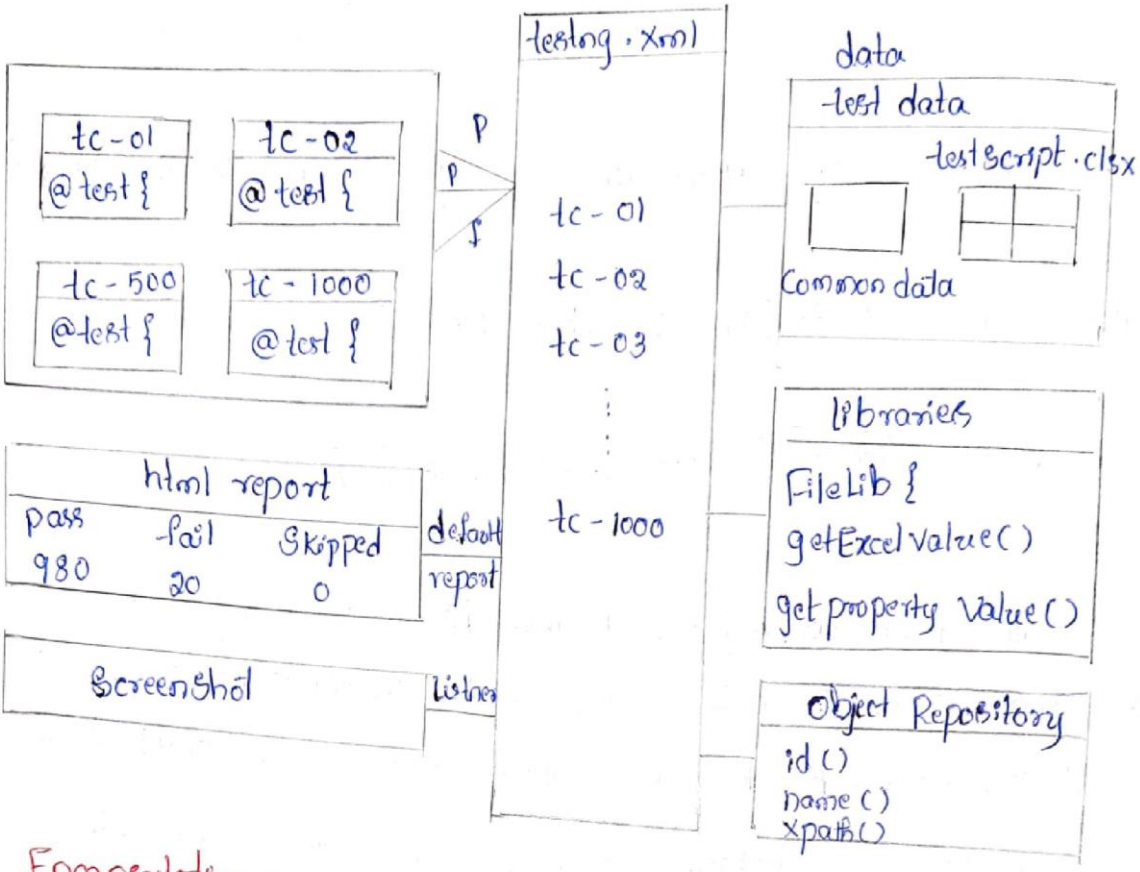
@Listener annotation should be declare before the class declaration block in every test script which is used to monitor the test execution in the runtime and generate event if any test script is getting failed.

(Else those test script will not be monitored while execution)

ex :

```
@Listeners (Com.actitime.generics.ListenerImplementation.class)
public class CustomerModule extends BaseClass {
@Test
public void testCreateCustomer ()
{
Reporter.log ("CreateCustomer", true);
Assert.fail ();
}
}
```

Architecture of testing :



Encapsulation :

Hiding the data & binding with methods in order to hide internal implementation is called encapsulation.

We achieve this by making variable as private and access outside the class with the help of getters & setters methods.

⇒
```
Public class A {
    private int i ;          // Declaration

    public A(int j) {
        i = j;               // initialisation
    }

    public int getvalue () {
    return i;                // we can only read the value
    }

  public void setvalue (int k) {   // we can set the value
    i = k ;                        means write access
  }
}
```

```
public class B {
public Static void main (String[] args) {
A a = new A(10);
int x = a.getvalue();                    // utilization
System.out.println(x);
a.SetValue(20);                          // utilization
System.out.println(a.getvalue());
}
}
```

Data will be stored in a variable in java, for any given variable we should perform following steps.

1. Declaration
2. Initialization
3. utilization.

There are two classes in the above example. The purpose of Class A is only to manage the variable 'i', where as 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 be written as

```
WebElement UNTBX = driver.findElement(By.id("username"));
UNTBX.Sendkeys("admin");
```

Or

```
WebElement UNTBX;                        // declaration
UNTBX = driver.findElement(By.id("username"));   // initialization
UNTBX.Sendkeys("admin");                 // utilization.
```

```
=> public Class Loginpage {
   Private webElement untbx ;   // declaration
   Public Loginpage (webDriver driver) {
    Untbx = driver . findElement (By . id ("username")); //initiliza
   }
  Public void SetUser (String un) {
      Untbx . Sendkeys (un) ;   // utilization
  }
}
```

= = = = =

```
public Class MainMethodclass {
Static {
System . Setproperty ("webdriver . chrome . driver", ". / driver /
}                                                  Chromedriver . exe");
Public Static void main (String [ ] args) {
webDriver driver = new ChromeDriver ();
driver . get (" https : // demo . actitime . com /");
Loginpage l = new Loginpage ( driver);
  l . SetUser ("admin");
 }
}
```

22/10/2020

```
public Class Loginpage {
private webElement untbx ;      // declaration
private webElement pwtbx ;
private webElement lgbtn ;

public Loginpage (webDriver driver) {
untbx = driver . findElement (By . id ("username")) ; // initalization
pwtbx = driver . findElement (By . name ("pwd")) ;
lgbtn = driver . findElement (By . xpath ("// div [. = 'Login ']")) ;
}
```

```
public void Setuser (String un) {
    untbx . Sendkeys (un);                    // utilization.

}
public void Setpassword (String pw) {
    pwtbx . Sendkeys (pw);

}
public void ClickLogin () {
    lgbtn . Click ();

}
public void Setlogin (String un, String pw) {
    untbx . Sendkeys (un);
    pwtbx . Sendkeys (pw);
    lgbtn . Click ();
}
}
```

not mandatory (Business Logic)

= = = = =

```
public Class MainMethodClass {
  Static {
System. Setproperty ("webdriver. chrome. driver", ". / driver / Chromedriver. exe");

}
public Static void main (String [] args) {
  webDriver driver = new ChromeDriver ();
  driver. get ("https : // demo. actitime. Com / ");
  Loginpage L = new Loginpage (driver);
  L. Setlogin ("admin", "manager");
}
}
```

= = =

```
⟹ public class MainMethodClass {
   Static {
   System . Setproperty ("webdriver. Chrome. driver", ". / driver / Chromedriver. exe");

   }
```

```
public static void main (String[] args) throws InterruptedException
webDriver driver = new ChromeDriver();
driver.get ("https://demo.actitime.com/");
Loginpage L = new Loginpage (driver);
L.setLogin ("admin1", "manager1");
Thread.Sleep (3000);
L.setLogin ("admin", "manager");
}
}
```

when we execute the above code we may get StaleElementReferenceException (Selenium unchecked) because when it clicks on login button after entering invalid user name & password, page will be reloaded & address of the element will be changed. But the reference variables such as untbx (pwtbx, lgbtn) will be holding old address. it will try to enter valid user name using old address which is no longer exists (invalid) hence we get the exception.

Script to explain StaleElementReferenceException :

```
public class Demopom {
static {
System.Setproperty ("webdriver.chrome.driver", "./driver/Chrome
                                                driver.exe");
}
public static void main (String[] args){
webDriver driver = new ChromeDriver();
driver.get ("https://demo.actitime.com/");
// Stores the username textbox address as @p1 in unbtbx
webElement untbx = driver.findElement (By.id ("username"));
// refresh & username textbox gets new adress like @p2 when we
   refresh the page.
driver.navigate().refresh();
```

```
// try to enter admin using old adress i.e @p1
    untbx. Sendkeys ("admin");

}
}
```

Page Object Model (POM):

POM is one of the java designed pattern which is used to store the objects (Elements). POM concept is used by both developers & automation test engineers to develop & test web pages. it is also used to avoid StaleElementReferenceException.

In POM Class we declare the elements by using @FindBy annotation & we initialise the element by using pageFactory.initElements method.

@FindBy annotation should be imported from org.openqa.selenium.support.pageFactory;

Syntax 1 : For Single element

```
@FindBy (locator = "locator value")
private webElement ElementName ;
```

Syntax 2 : For Multiple elements.

```
@FindBy (locator = "locator value")
private List <webElement> ElementName ;
```

To initialise the element we use initElements () method of PageFactory Class, it will take two arguments.

1. webDriver - driver
2. Object of POM Class - this

initElement method will only loads the element (only declaration of reference variable) but it will not initialise actually. element are actually initialised during the runtime when we try to perform any action on ~~init~~ elements. This process is called lazy initialization

This will avoid StaleElementReference Exception.

Note : The class in which elements of the web page are stored by using @FindBy annotation is called as POM Class (page Class)

Ex :- Loginpage.

⇒

```
public class Loginpage {
@ FindBy (id = "username")
private webElement untbx;

@ FindBy (name = "pwd")
Private webElement pwtbx;
@ FindBy (Xpath = "// div [. = 'Login']")
Private webElement lgbtn;

public Login pag.
Public Loginpage (webDriver driver){
 page Factory . initElements (driver, this);
}
public void SetUser (String un){
    untbx. SendKeys (un);
}
Public void Setpassword (String pw){
  pwtbx . SendKeys (pw);
}

public void clickLogin () {
    lgbtn . click ();
}

public void SetLogin (String un, String pw){
      untbx . SendKeys (un);
      pwtbx . SendKeys (pw);
    lgbtn . click ();
}
}
```

= = =

```
public class MainMethodClass {
Static {
System.setproperty ("webdriver.Chrome.driver", "./driver/
                                        Chromedriver.exe");
}
public Static void main (String [] args) throws InterruptedException {
WebDriver driver = new ChromeDriver();
driver.get ("http://localhost/login.do");
Login_page L = new Login_page (driver);
L.setLogin ("admin1", "manager1");
Thread.Sleep (3000);
L.setLogin ("admin", "manager");
}
}
```

23/10/2020.

What happens if we dont use initElements method in POM Class (PageFactory.initElements (driver, this);

we get null pointer exception.

Can we develop POM Class without Constructor? Yes.

Can we use initElements method in main method class?

Yes, we should explicitly call initElements () method in test class.

```
⇒ public class Loginpage
  {
  @FindBy (id = "username")
  private WebElement untbx;

  @FindBy (name = "pwd")
  private WebElement pwtbx;   // declaration.

  @FindBy (Xpath = "//div [. = 'Login']")
  private WebElement lgbtn;
```

```
// Business Logic method
public void GetLogin (String un, String pw) {
  untbx. SendKeys (un);
  pwbtbx. SendKeys (pw);          // utilization
  lgbtn. Click ();
 }
}
```

= = = = =

```
public class MainMethodClass {
Static {
System. Setproperty ("webdriver. Chrome. driver", "./ driver/
                                        Chromedriver. exe");
 }
public Static void main (String[] args) throws InterruptedException{
WebDriver driver = new ChromeDriver ();
driver. get ("https : // demo. actitime . com /");
  Loginpage L = new Loginpage ();
pageFactory . initElements (driver, L);
  L. SetLogin ("admin", "manager");
 }
}
```

What is object repository ?

It is the location where we store the objects or elements. page object model (POM) is called as object repository or page object repository.

Note; In POM we store the element present on the page in a particular page class.

What is the difference between page object model & page factory

Page object model is a java designed concept where as page factory is a class which implements this POM concepts.

Advantages of page object models :

a. web Elements are maintained based on the web pages, so maintainance of the web elements are easy.

b. Modification of web element locators is easy whenever UI is getting Changed.

c. Xpath & locators of the web elements are reusable so that no need to put the same effort to rewrite the xpath again & again.

d. we can avoid StaleElementReference Exception.

e. object repository can be easily shared across the team members

In pom we can maintain all the web element locators based on the web pages. Number of POM classes will be equal to number of webpages. POM Classes will be Created under Com.actitime.pom package or Com.actitime.objectrepository.

How do you declare an element in POM ?

By using @FindBy annotation.

How do you initalize the element in POM ?

By using page factory. initElements () method.

Frame work :

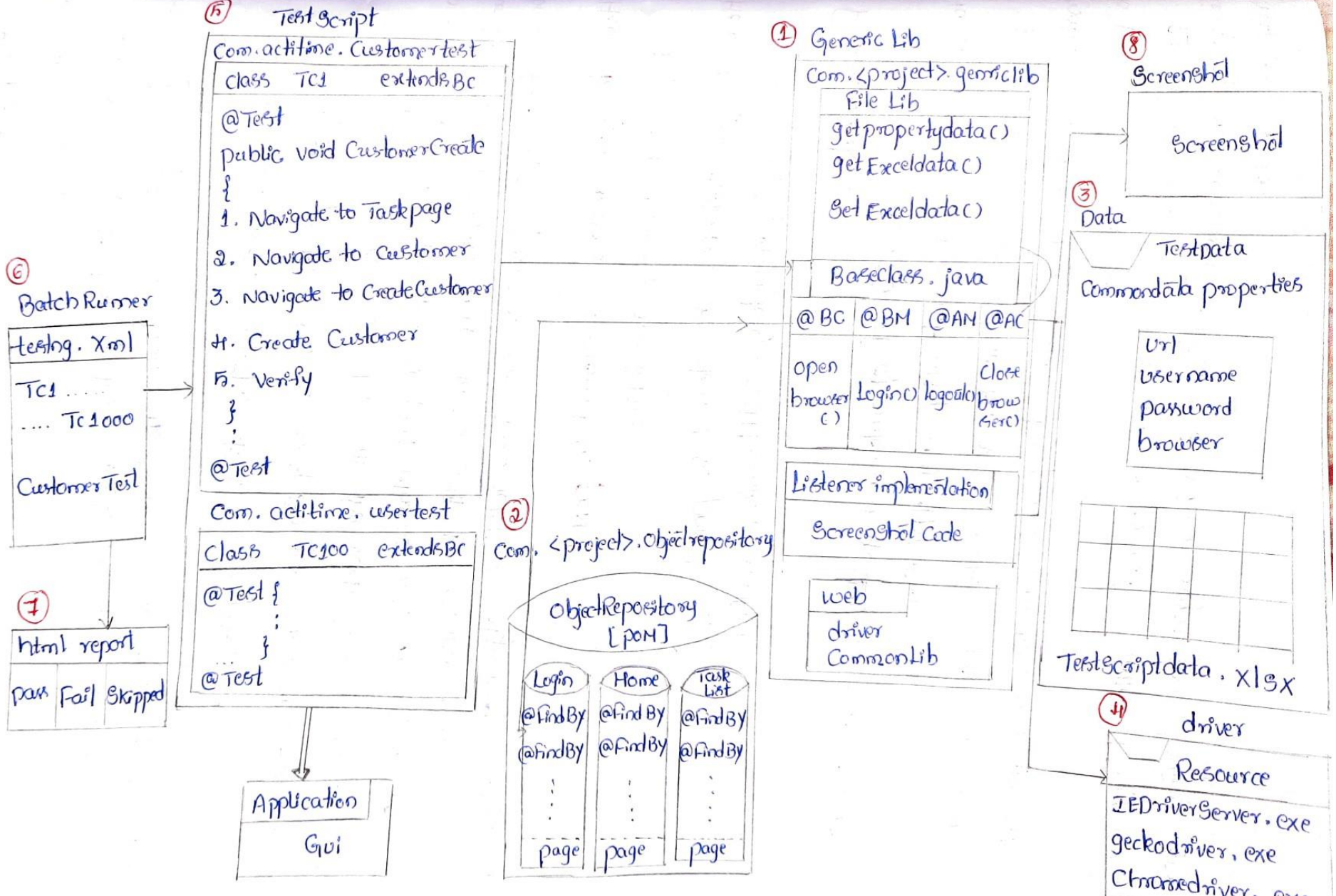
Frame work is a set of rules and guidelines or best practice to be followed while automating any application.

<u>or</u> Frame work is a Collection of reusable Components that makes automation test Scripts development, execution & modification to be easier & faster.

<u>or</u> Framework is a instruction or procedure followed by every Company to automate test Script & achieve regression testing for every new build.

or Frame work is a well organised structure of reusable components, where one drive script (Testng.xml) will take core of entire batch execution without any manual interaction.

Architecture of Hybrid Framework :



1. The framework is developed using TestNG, POM & excel library. it is a Combination of Data-Driven & Method-Driven framework, which we Called 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 & also extend from base Test Class which has @BeforeClass, @After Class.

4. First @BeforeClass is executed which opens the browser and @BeforeMethod will be executed which Contains enter the URL and login to the application.

```
webDriver driver = new ChromeDriver();
```

5. After executing Before Method will start the execution of testmethod. The testMethod takes the data from excel sheet & Perform the action by Calling the method present in the POM Class.

Example:

```
String un = Excel.getCellData(xlpath, sheet, 1, 0);
Loginpage l = new Loginpage(driver);
l.setUserName(un, pw);
```

6. After executing testmethod, it will executes AfterMethod which Contains logout & AfterClass which Closes the browser.

```
driver.Close();
```

7. After executing all the Scripts it will generate the result in html format (test-output folder) of the framework.

8. Since we implimented Listener it will take the ScreenShot of failed test Cases.

Framework Folder Structure.

- Actitime
  - SrC
    - com.actitime.generics
      - Base class
      - FileLib
      - ListnerImplementation
      - webDriverCommonLib
    - Com.actitime.objectrepository
      - Loginpage.java
      - Homepage.java
      - Tasklistpage.java
    - Com.actitime.testscript
      - CreateCustomer.java
      - Delete Customer.java
      - Modify Customer.java.
  - data
    - Testscript.xlsx
    - Commondata.property
  - Screenshot
  - test-output
  - testng-Xml

27/10/2020

@ Listeners (Com.actitime.generics.ListenerImplementation.Class)

public Class CreateCustomer extends BaseClass {

@ Test

public void testCreateCustomer () throws EncryptedDocumentException, IOException, InterruptedException{

String

Xpath axes : Xpath Axes are Special Keywords of Xpath which has following Syntax.

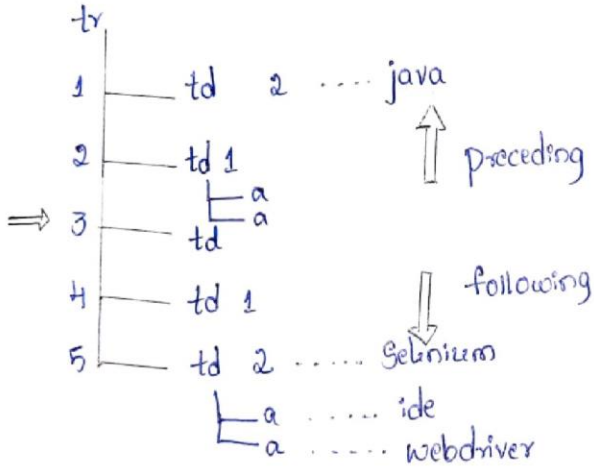
Syntax : // AxesName ::tag

Axes - Names :

a. ancestor
b. descendant
c. Child
d. parent
e. following - Sibling
f. preceding - Sibling.

Ex : / descendent :: a → //a

Ex2 : // td [text() = 'Selenium'] / following - Sibling ::td



// td [3] / following - Sibling ::td [2] — Selenium

// td [3] / preceding - Sibling ::td [2] — java

// td / parent ::tr

// td [5] / child ::a [2] - webdriver

Xpath for Starts - with function : we can use Starts - with function for either text or attribute value.

Syntax :

// tag [starts - with (text (), 'text value')]

// tag [starts - with (@ AN , 'AV')]

~~30/10/2020~~

Source Control tool :

Auto It :

Autoit is a free windows based application automation tool, which can be downloaded free from following website.

Url : https : // www. autoScript. 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 below code
3. WinWaitActivate ("opening Selenium - Server - Standalone - 3.141.59 . jar")
4. Sleep (2000)
5. Send ("{LEFT}")
6. Sleep (2000)
7. Send ("{ENTER}")

3. Go to file and Select Save
4. Navigate to required location ex : D Specify the name ex : Script1 & click Save

   Extension will be .au3
5. Go to tools & Select Compile & it Creates new . exe file (Same location
6. Double click to execute it (to execute manually)

To run the Complete Code programmatically in java we Should write the code as Shown below.

Q. write a Script to perform the following Steps.

1. open firefox browser
2. Go to download page of Selenium
3. Click on link 3.141.59
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 ;

Selenium Script :

```
⇒ public Class FileDownloadpop {
  Static {
  System. Setproperty ("webdriver. gecko. driver", ". / driver /
                                        geckodriver. exe");
  }
  public Static void main (String [] args) throws InterruptedException,
                                        IOException {
  webDriver driver = new FirefoxDriver ();
  driver. get ("https : // www. Selenium. dev / downloads / ");
  driver. findElement (By. linkText ("3.141.59")). click ();
  Thread. Sleep (2000);
  Runtime. getRuntime (). exec ("c: \\ Users \\ Samskruthi \\ Desktop
                                        \\ FileDownload. exe");
  Thread. Sleep (10000);
  driver. Close ();
  }
```

Exceptions :

1. Illegal State Exception : (Selenium / unchecked)

we get this exception whenever the path of driver execut-able files is not set. (Run time exception)

2. Interrupted Exception : (Java Checked / Exception).

we get this exception whenever we use Thread-sleep.

3. No Such Element Exception : (Selenium / unchecked)

we get this exception whenever locators are not matching with any of the element.

4. Invalid Selector Exception : (Selenium / unchecked)

we get this exception whenever there is a Syntax error in Css Selector or xPath.

5. Index Outof bound Exception : (java / unchecked)

whenever we use i <= count insted of i < Count we get this Exception.

6. Time Out Exception : (Selenium / unchecked)

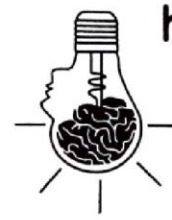
we get this exception whenever the element is not found within a specific time given in explicit wait.

7. Unsupported Operation Exception : (Selenium / unchecked)

whenever we use deSelect Method ~~Option~~ on a Single Select textbox we get this exception

8. unexcepted Tag Name Exception : (Selenium / unchecked)

we get this Exception whenever we try to use Select class on Button, ~~List~~ Checkbox etc. (other than List box) or when the tagname is not Select.



Space for Creativity

9. NoAlertpresentException : (Selenium / unchecked)

we get this Exception whenever we try to handle alert popup when popup itself is not present.

10. UnhandledAlertException : (Selenium / unchecked) we get this Exception whenever we try to perform action on browser without handling the alert popup.

11. AWT Exception : (java / checked)

we get this exception whenever we use robot Class

12. NoSuchwindowException : (Selenium / unchecked)

we get this exception whenever we try to perform actions on the window which is already Closed. or no longer available.

13. NoSuchframe Exception : (Selenium / unchecked)

we get this exception whenever we give wrong id or index of the frame.

14. ElementNotInteractableException : (Selenium / unchecked)

we get this exception whenever we try to perform action on disabled elements

15. FileNotFoundException : (Java / checked)

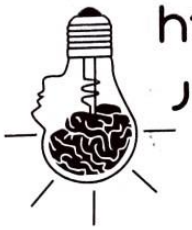
we get this exception whenever we use FileInputStream. Class.

16. IoException : (Java / checked)

we get this exception whenever we use

17. EncryptedDocumentException : (Apache poi / Checked)

whenever we try to read the data from the excel we get this exception.



→ open Github $\xrightarrow{\text{Click}}$ Create Github account $\xrightarrow{\text{Click}}$ new repository in dropdown present at right Corner. $\longrightarrow$ Verify email.

→ Create a new repository:

Repository name: SeleniumActiTime.

↓ ~~After Create repository~~

- Select public

↓

☑ Add a README file.

↓

Click Create repository

→ under Code

↓

Copy: https://github.com/Samskruthi20/ActiTimeSelenium.git (uri)

↓

In Eclipse right click on project

↓

team

↓

push branch master.

↓

Paste uri, UN, pwd, Click on Checkbox ☐ Store in Secure Store

↓

Click on force overwrite branch Checkbox.

↓

next

↓

finish.