

## Syllabus

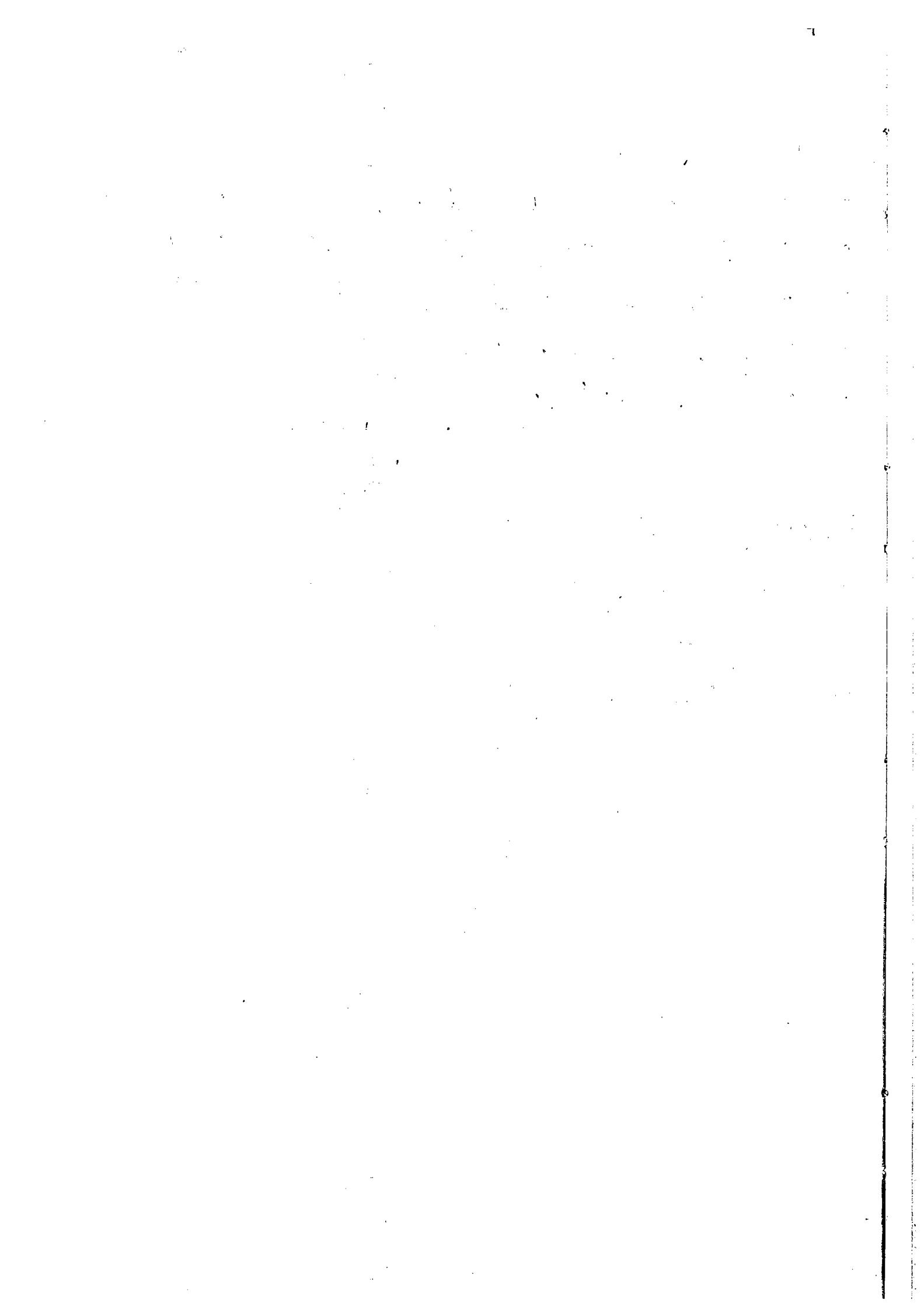
Dt- 16/01/2023-<sup>1</sup>

1. Introduction to Automation
2. Versions of Selenium
3. Selenium Web Driver Architecture
4. Selenium WebDriver Methods
5. Web Element & methods of Web Element
6. Locators & Types of Locators
7. Handling Multiple Web Elements
8. Handling Auto-Suggestions
9. Handling Hidden Auto-Suggestions
10. Handling Drop Down List
11. Sorting the Elements of a Drop Down List
12. Printing the Unique Contents of a Drop Down List
13. Finding the occurrence of Elements present in Drop Down List
14. Handling Mouse Operations
15. Synchronization
16. Handling Pop-ups
17. Handling Frames
18. Handling Chatbot
19. Page Load Timeout in Selenium
20. Java Script Execution in Selenium
21. Removing Chrome Warning messages
22. Handling Insecure Websites
23. Adding Chrome Extension
24. Headless Browser Testing
25. Capturing Selenium Log files

17. Different ways of Entering Character Sequence
18. Different ways of Refreshing a webpage
19. Special Locators in Selenium
20. Selenium Version 4 features
  1. Handling Web Table
  2. Page Object Model (POM)
  3. TestNG (Next Generation)
  4. Fetching Data from Excel Sheet
  5. Fetching Data from Properties file
  6. Fetching Data from HashMap
  7. Fetching Data from Object 1-D Array
  8. Fetching Data from Object 2-D Array
  9. Fetching Data from String Array
  10. Fetching Data from Iterator<Object> 1-D
  11. Fetching Data from Iterator<Object[]> 2-D
  12. Automation Framework
  13. Highlighting Elements of a webpage
  14. Cross Browser Testing
  15. Taking Screenshot of a webpage
  16. Verifying Error Messages in Selenium
  17. Maven
  18. Opening the Browsers without using Browser Drivers
  19. GitHub
  20. Jenkins

## Java Topics To Be Used In Selenium :-

1. Looping & Conditional Statements
2. OOPs (Polymorphism, Encapsulation, Abstraction, Inheritance)
3. String Class → equals(), equalsIgnoreCase(), contains(), contentEquals(), length(), toUpperCase(), toLowerCase(), isEmpty(), replaceAll(), toString(), split()
4. Thread Class → sleep()
5. Collections class → sort(), reverseOrder() → ArrayList
6. Collection → List, Set, Map
  - ↓
  - HashMap
  - HashSet
  - TreeSet
7. Array → 1D, 2D, Jagged
8. Exception Handling Mechanism → throws, try-catch
9. Iterator Interface
10. String Buffer & String Builder Class



## Automation Testing :-

→ Verifying the functionality of an application Automatically.

(a) We need to write automation codes / Scripts.

e.g:- Java / Python / C# / JS / Ruby / Perl .....

(b) We need an environment to write automation codes.

e.g:- Eclipse IDE / IntelliJ IDE / NetBeans IDE .....

(c) We need an automation tool to execute automation Scripts.

e.g:- Selenium / UFT / Test Complete / Tosca .....

## Advantages of Automation Testing :-

(i) Faster & saves our Testing time.

(ii) Reduce human effort.

(iii) Accurate & Consistent.

(iv) Reusability (Write the code once & execute it multiple times)

## Drawbacks of Automation Testing :-

(i) Extra skill set is required. (Programming Knowledge)

(ii) Initial investment is required.

(iii) Constant maintenance is required.

(iv) 100% automation is not possible. (OTP / QR Code / Captcha / Animations / Audio / Video / Games / GIF's .....

(v) Bug Count will be less compared to manual.

## Selenium :-

→ It is a

- (i) free → Selenium can be downloaded free of cost.
- (ii) Open-Source → We can view & download the source code of Selenium.
- (iii) Web-Application → Selenium can automate only Web-applications.

Drawback  
automation tool.

Can't automate Standalone appln  
& Client Server appln

↳ appln which  
opens through a  
web browser.  
URL

Can we Automate Mobile Applications?

Yes, we can automate the android & iOS mobile apps.

Automation Tools

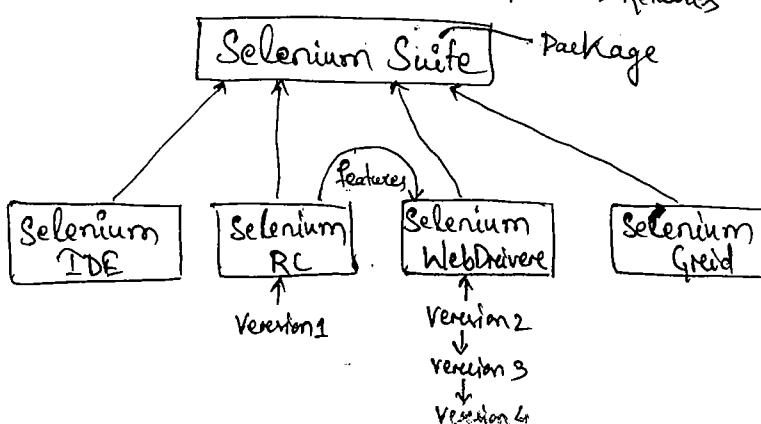
Selendroid → automate only android mobile apps.  
 Appium → automate both android & ios mobile apps.  
 ↳ most widely used mobile automation tool

## Versions of Selenium :-

- (i) Selenium IDE (Integrated Development Tool)
- (ii) Selenium RC (Remote Control)
- (iii) Selenium Grid
- (iv) Selenium WebDriver → latest version of Selenium

↳ <https://www.selenium.dev/downloads>

Previous Releases

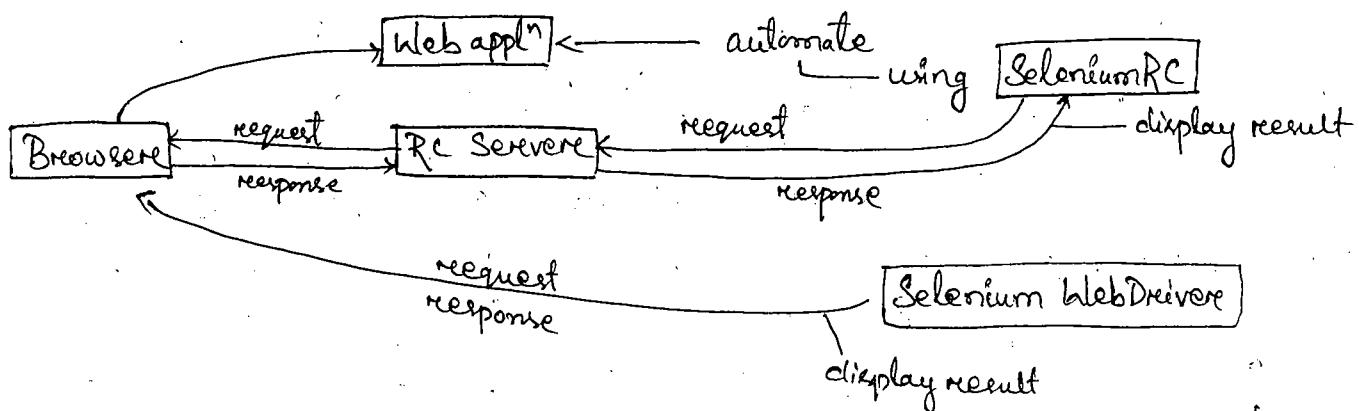


## Selenium IDE :-

- It has the simplest framework in the entire Selenium Suite.
- It is easy to install & use.
- It is available as a plugin for the Browsers.
- It does not support any programming languages.
- Looping & Conditional statements are not allowed.
- Execution is slower than both RC & WebDriver.

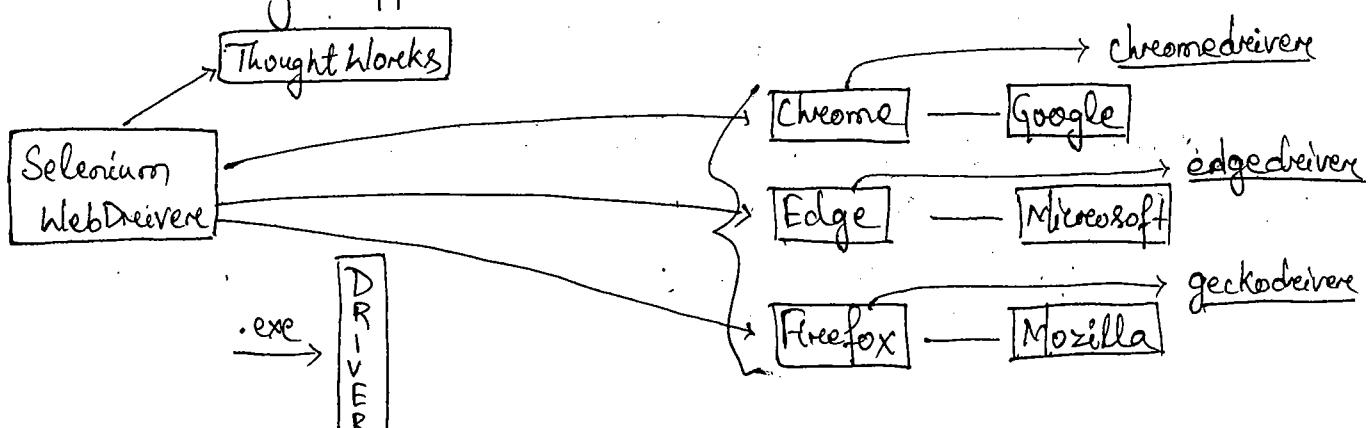
## Selenium RC :-

- It is the first Automation Tool that supported a programming language.
- Looping & Conditional statements are allowed.
- Execution is faster than IDE but slower than WebDriver.
- We need a RC Server to communicate with the Browser.



## Selenium WebDriver :-

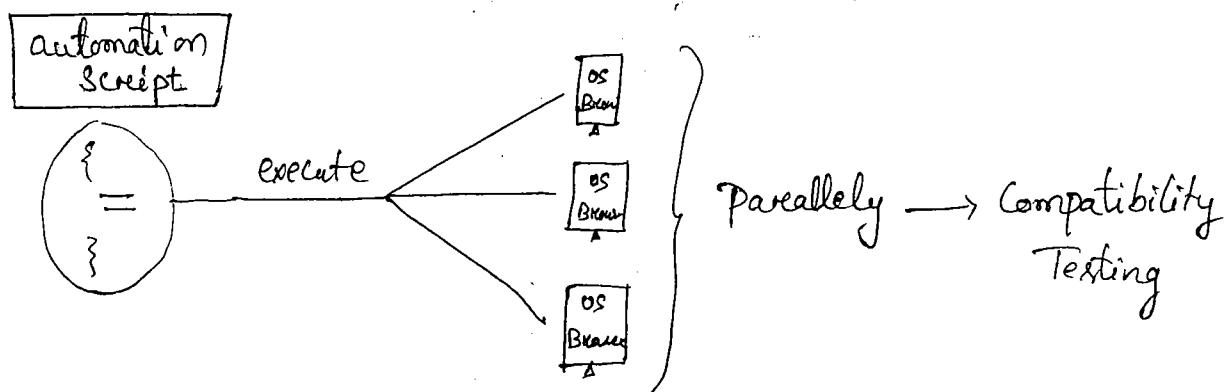
- It communicates with the browser directly without using any RC Server.
- Execution is faster than both IDE & RC.
- Does not readily support new browsers.



Dt-18/01/2023  
Wednesday

## Selenium Grid :-

- Using Selenium Grid we can execute the same code across different os's parallelly.



## Required Softwares :-

1. JDK - 8 & ABOVE till 17
2. ECLIPSE IDE → 2022 → eclipse.org/downloads/packages/release
3. SELENIUM JAR FILE → {
  - 2.52 → Selenium-dev/downloads
  - 3.141.59 → scroll
  - 4.0.0 → Previous Releases [click](#) releases
4. BROWSER → Chrome, Edge, Firefox → Selenium-Server-Standalone-2.52.0.jar, Selenium-Server-Standalone-3.141.59.jar, Selenium-Server-4.0.0.jar
5. BROWSER DRIVERS → {
  - ChromeDriver → Selenium.dev/downloads
  - EdgeDriver → scroll
  - GeckoDriver → Browser

download the drivers a/c to browser version

## Integrating Selenium Jar file into Java Project :-

- i) Create a Work Space or folder in any drive.
- ii) Switch to the newly created workspace.
- iii) Inside Eclipse create a Java project.
- iv) Add Selenium jar file to the Java project.
- v) Under SRC folder create a Java package.
- vi) Under the package create a Java class including main method.

## Explanation of point 4 :-

- (ii) Create a folder as "lib" inside the Java project.
  - (iii) Copy the Selenium jar file & paste it inside the created folder.
  - (iv) Right click on the Selenium jar file, go to Build path & click on Add to build path.
  - (v) Right click on the Java project, go to "Build Path", click on "Configure Build Path".
  - (vi) Click on Libraries Tab.
  - (vii) Under class path Selenium should be added.

## Creating an Instance of a Browser :—

```

graph LR
    A[Class] --> B[Reference Variable]
    B --> C[Assignment Operator]
    C --> D[new Keyword]
    D --> E[Constructor]
    E --> F[Statement Delimiter]

```

The diagram illustrates the components of a Java assignment statement. It starts with a 'Class' at the top left, which points down to a 'Reference Variable'. This variable then points down to an 'Assignment Operator'. The 'Assignment Operator' points down to a 'new Keyword'. This keyword points down to a 'Constructor'. Finally, the 'Constructor' points down to a 'Statement delimiter' at the bottom right.

## How to Set the Property of Browser Driver :-

```

graph TD
    Root[System . SetProperty (String key , String value);] --> Class[java.lang]
    Root --> Static[static method]
    Root --> Predefined[Predefined]
    Root --> Path[path of the browser driver]

```

## Assignment

Q1 Write a Script to open firefox & edge Browser.

Write a script to open chrome, Firefox & Edge.

// Chrome using Selenium Version 2.52.0

Package Practice;

import org.openqa.selenium.chrome.ChromeDriver;

public class Chrome

{  
    public static void main(String[] args)

        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

}

}

// Firefox using Selenium Version 2.52.0

Package Practice;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Firefox

{  
    public static void main(String[] args)

        System.setProperty("webdriver.gecko.driver", "./drivers/geckodriver.exe");

        FirefoxDriver fd = new FirefoxDriver();

}

}

// Edge using Selenium Version 2.52.0.

Package Practice;

import org.openqa.selenium.edge.EdgeDriver;

public class Edge

{  
    public static void main(String[] args)

        System.setProperty("webdriver.edge.driver", "./drivers/msedgedriver.exe");

        EdgeDriver ed = new EdgeDriver();

}

}

Dt-19/01/2023  
Thursday

### Note:-

- (i) If the browser driver key & value is not specified, we get IllegalStateException.
- (ii) If there is a mistake in the driver key & value, we get IllegalStateException.
- (iii) If the driver version is not compatible with the browser version, we get SessionNotCreatedException.

Selenium 2 → chromedriver, edgedriver, geckodriver



Selenium 3 → chromedriver, edgedriver, geckodriver <sup>use 0.30 version</sup>

Q. Can we open the Browser without specifying the path of the driver inside main()?

A. Yes, By using Static Block.

Package practice; // using Selenium Version 3.141.59

import org.openqa.selenium.chrome.ChromeDriver;

public class OpenBrowser

{

  Static

{

    System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

}

  public static void main(String[] args)

{

    ChromeDriver driver = new ChromeDriver();

}

}

Q: What is the difference between CamelCase characters & TitleCase characters?

CamelCase characters - SavvyRealSahu → methods & variables

TitleCase characters — SampreetiSahu → class, Interface.

## Eclipse Shortcut Keys :-

- i) minimizing & maximizing a Java class -  $\text{ctrl} + M$
  - ii) Deleting a line of code -  $\text{ctrl} + D$
  - iii) Auto importing the packages -  $\text{ctrl} + \text{shift} + O$
  - iv) Commenting multiple lines of code -  $\begin{matrix} /* \\ */ \end{matrix}$   $\text{ctrl} + \text{shift} + /$   
 $//$   $\text{ctrl} + /$
  - v) uncommenting multiple lines of code -  $\text{ctrl} + \text{shift} + \backslash$
  - vi) Indenting the lines of code -  $\text{ctrl} + \downarrow$

## Executing same code in Multiple Browsers :-

```
Package webdriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
Public class OpenMultiBrowsers
{
    Static
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        System.setProperty("webdriver.gecko.driver", "./drivers/geckodriver.exe");
    }
    Public static void test(ChromeDriver driver)
    {
        System.out.println("Chrome Browser is Opening");
    }
}
```

```

    public static void test(FirefoxDriver driver)
    {
        System.out.println("Firefox Browser is Opening");
    }

    public static void main(String[] args)
    {
        ChromeDriver driver1 = new ChromeDriver();
        OpenMultiBrowsers.test(driver1);

        FirefoxDriver driver2 = new FirefoxDriver();
        OpenMultiBrowsers.test(driver2);
    }
}

```

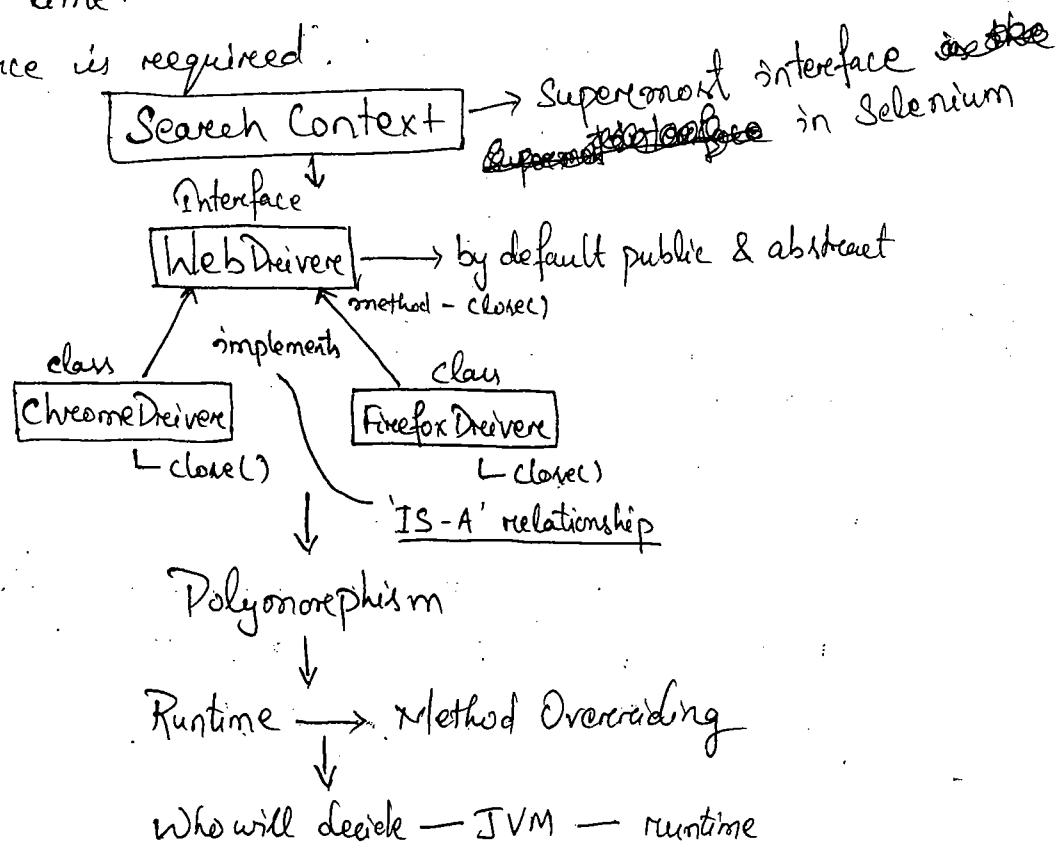
Body of the methods remains same

- Body of the methods remains same for all the browsers & it is a foolish way of writing the code again & again.
- Tomorrow if any changes occur, we must do the changes in all the methods.

#### Drawback :-

- Lengthy
- Consumes a lot of time.
- Lot of maintenance is required.

#### Solution:-



Can I create an object of WebDriver :-

No

Can I Create reference variable of WebDriver :-

WebDriver driver; → abstract method - has a head but no body. - e.g. public void on();

Concrete method - has a head & body. - e.g. public void on();

WebDriver driver;

ChromeDriver c = new ChromeDriver();

driver = c; // Auto Upcasting

Note :-

- (i) If Subclass has the same method as declared in the Parent class, it is known as Method Overriding.
- (ii) Method Overriding is used to provide the implementation of a method which is already provided by its Super class.
- (iii) Method Overriding is used for runtime polymorphism.

Rules of Method Overriding :-

- The method should have the same name as in the parent class.
- The method must have the same parameter as in the parent class.
- There must be a IS-A relationship (Inheritance).

Dt - 20/01/2023

Friday

Q. Write a Script to enter the URL of an application.

```
package WebDriver;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
```

Public Class EnterURL

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

```
{ System.setProperty("webdriver.chrome.driver", ".\drivers/chromedriver.exe");
```

```
WebDriver driver = new ChromeDriver(); // upcasting
```

```
driver.get("https://www.myntra.com");
```

}

}

Q. Write a Script to enter the URL of an application without using get().

Ans

```

    package webdriver;
    import org.openqa.selenium.WebDriver;
    import org.openqa.selenium.chrome.ChromeDriver;
    public class EnterURL
    {
        public static void main(String[] args)
        {
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.navigate().to("https://www.flipkart.com");
        }
    }
  
```

driver.navigate().to(args); → Method Chaining

```

graph TD
    A["driver.navigate().to(args)"] --> B["abstract method  
Navigation Interface  
void"]
    C["WebDriver Interface  
Navigation"] --> B
  
```

Q. What is Method chaining?

Ans Calling a method on a reference of either a class or interface where instance is referred by the immediate previous method.

Q. What is the difference between get() & navigate() ?

Ans Using get() we can enter the URL of an application.

Using navigate() we can enter the URL of an application, move forward & backward & refresh a webpage.

Q) Write a Script to perform Navigation Operations?

```

Package webDriver;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class BackForwardRefresh
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to("https://www.google.com");
        driver.navigate().to("https://www.myntra.com");
        driver.navigate().back();
        driver.navigate().forward();
        driver.navigate().refresh();
    }
}

```

Q) Write a Script to get the title of a webpage?

```

Package webDriver;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class GetTitle
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.myntra.com");
        System.out.println(driver.getTitle());
    }
}

```

Q. Write a script to verify or validate the title of a webpage.

*//using equals()*

```

A:-  

    Package webDriver;  

    import org.openqa.selenium.WebDriver;  

    import org.openqa.selenium.chrome.ChromeDriver;  

    Public class VerifyTitle  

    {  

        P S V m(-)  

        {  

            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  

            WebDriver driver = new ChromeDriver();  

            driver.get("https://www.google.com");  

            String actualTitle = driver.getTitle();  

            String expectedTitle = "google";  

            if(actualTitle.equals(expectedTitle))  

            {  

                System.out.println("Titles are matching");  

            }  

            else  

            {  

                System.out.println("Titles are not matching");  

            }  

            O/P:- Titles are not matching
        }
    }

```

*//using equalsIgnoreCase()*

```

    Package webDriver;  

    import org.openqa.selenium.WebDriver;  

    import org.openqa.selenium.chrome.ChromeDriver;  

    Public class VerifyTitle  

    {  

        P S V m(-)  

        {  

            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  

            WebDriver driver = new ChromeDriver();  

            driver.get("https://www.google.com");  

            String actualTitle = driver.getTitle();  

            String expectedTitle = "google";  

            if(actualTitle.equalsIgnoreCase(expectedTitle))  

            {  

                System.out.println("Titles are matching");
            }  

            else  

            {  

                System.out.println("Titles are not matching");
            }
        }
    }

```

```

package webDriver;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.ChromeDriver;
public class VerifyTitle
{
    public static void main()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
        String actualTitle = driver.getTitle();
        String expectedTitle = "google";
        if(actualTitle.contains("goo"))
        {
            System.out.println("titles are matching");
        }
        else
        {
            System.out.println("titles are not matching");
        }
    }
}

```

Op:- titles are not matching

```

//using ContentEquals()
package webDriver;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class VerifyTitle
{
    public static void main()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
        String actualTitle = driver.getTitle();
        String expectedTitle = "google";
        if(actualTitle.contentEquals(expectedTitle))
        {
            System.out.println("Titles are matching");
        }
        else
        {
            System.out.println("Titles are not matching");
        }
    }
}

```

Op:- Titles are not matching

Q. Write a Script to get the Current URL of a webpage .

A:- package webdriver;

```
import org.openqa.Selenium.WebDriver,
```

```
import org.openqa.Selenium.Chrome.ChromeDriver;
```

```
public class GetCurrentURL
```

```
{ public void main()
```

```
{
```

```
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
```

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

```
driver.get("https://www.gmail.com");
```

```
System.out.println(driver.getCurrentUrl());
```

```
}
```

```
}
```

Q. Write a Script to close a Browser .

A:-

```
package webdriver;
```

```
import org.openqa.Selenium.WebDriver,
```

```
import org.openqa.Selenium.Chrome.ChromeDriver;
```

```
public class CloseBrowser
```

```
{
```

```
public void main()
```

```
{
```

```
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
```

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

```
driver.close();
```

```
}
```

```
}
```

Q. Write a script to close the browser without using close().

```
11 package webdriver;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class GreetBrowser
{
    public void m()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.quit();
    }
}
```

Dt- 21/01/2023  
Saturday

Q: What is the difference between close() & quit()?

Log in Page

Forgot Password?  
← Clock

A hand-drawn diagram consisting of a large rectangle with a double-lined border. Two arrows point from the right edge of the rectangle towards its top-right corner, indicating a click or selection action.

- close();
  - ↳ closes a particular instance of browser where Selenium Control is present.
- quit();
  - ↳ closes all the instances of the browser.

? Write a Script to maximize a webBrowser.

driver.manage().window().maximize(); method chaining

↓  
abstract method  
WebDriver Interface  
Options

Q. Package WebDreiver;

```
import org.openqa.Selenium.WebDriver;
import org.openqa.Selenium.Chrome.ChromeDriver;
```

Public class MaximizeBrowser

```
{
    PSVM()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
    }
}
```

Q. Write a Script to delete all the Cookies of a Web Browser.

A- Cookie is nothing but a text that is stored in your browser whenever we visit any website.

```
driver.manage().deleteAllCookies();
↓           ↓
abstract method   abstract method
WebDriver Interface Options Interface
Options          void
```

P.

```
Package WebDreiver;
import org.openqa.Selenium.WebDriver;
import org.openqa.Selenium.Chrome.ChromeDriver;
```

Public class DeleteAllCookies

```
{
    PSVM()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe")
        WebDriver driver = new
        driver.get("https://www.google.com");
        driver.manage().deleteAllCookies();
    }
}
```

Q. Write a Script to get & Set the size of the Browser.

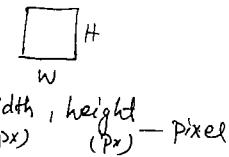
driver.manage().window().getSize();

↓  
abstract method  
WebDriver Interface  
Options

↓  
abstract method  
Options Interface  
Window

↓  
abstract method  
Window Interface

Dimension



driver.manage().window().setSize();

↓  
void

```
Package WebDriver;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
```

Public class GetSetBrowserSize

{  
    P S R m(—)

{  
    System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  
    WebDriver driver = new ChromeDriver();

    System.out.println(driver.manage().window().getSize());

    Dimension d = new Dimension(300, 300);

    driver.manage().window().setSize(d);

    System.out.println(driver.manage().window().getSize());

}

}

Q. Write a Script to get & Set the position of the Browser.

driver.manage().window().getPosition();

↓  
abstract method  
WebDriver Interface  
Options

↓  
abstract method  
Options Interface  
Window

(x-axis, y-axis)  
(px) (px)

↓  
abstract method  
Window Interface  
Point

driver.manage().window().setPosition();

↓  
void

P:

```

    package WebDrivers;
    import org.openqa.selenium.Point;
    import org.openqa.selenium.WebDriver;
    import org.openqa.selenium.chrome.ChromeDriver;
    public class GetSetBrowserPosition
    {
        public void main()
        {
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            System.out.println(driver.manage().window().getPosition());
            Point p = new Point(300, 300);
            driver.manage().window().setPosition(p);
            System.out.println(driver.manage().window().getPosition());
        }
    }

```

Q: Write a script to get the page source of an application.

A:

getPagesource();

↓  
abstract method  
WebDriver Interface

P:

```

    package WebDrivers;
    import org.openqa.selenium.WebDriver;
    import org.openqa.selenium.chrome.ChromeDriver;
    public class GetPageSource
    {

```

public void main()
 {

↓  
String

```

        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.getPageSource();
        driver.get("https://www.google.com");
        System.out.println(driver.getPageSource());
    }
}
```

## WebDriver Interface Methods :-

- 1. close() - void
- 2. quit() - void
- 3. getTitle() - String
- 4. get(webElement) - String
- 5. getPageSource() - String
- 6. navigate() - Navigation
- 7. manage() - Options
- 8. get() - void
- 9. findElement() - WebElement
- 10. findElements() - List<WebElement>
- 11. getWindowHandle() - String
- 12. getWindowHandles() - Set<String>
- 13. switchTo() - TargetLocator

## Navigation Interface Methods :-

1. to() - void
2. back() - void
3. forward() - void
4. refresh() - void

## Window Interface Methods :-

1. maximize() - void
2. getSize() - Dimension
3. setSize() - void
4. getPosition() - Point
5. setPosition() - void
6. fullScreen() - void
7. minimize() - void

## Options Interface Methods :-

1. window() — Window
2. timeouts() — Timeout
3. deleteAllCookies() — void
4. getCookies() — Set<Cookie>
5. ~~delete~~ deleteCookieNamed()
6. getCookieNamed() — String  
void Cookie

## String Class Methods :-

1. equals() — boolean
2. equalsIgnoreCase() — boolean
3. contains() — boolean
4. contentEquals() — boolean
5. length()
6. toUpperCase()
7. toLowerCase()
8. isEmpty()
9. replaceAll()
10. toString()
11. Split()

: what is the difference between equals() & ContentEquals() ?

String — String — equals() — true

String — String — ContentEquals() — true

String — StringBuffer — equals() — false

String — StringBuffer — ContentEquals() — true

Class EqualsContentEquals

{

    public void main()

{

        String a = "Sam";

        String b = "Sam";

        StringBuffer buf = new StringBuffer("Sam");

        StringBuilder bud = new StringBuilder("Sam");

        // if (buf.equals(bud))

        // if (buf.ContentEquals()) // method is not present

        // {

            System.out.println("Text is matching");

        // }

        // if (a.equals(b))

        // if (a.ContentEquals(b))

        // {

            System.out.println("Text is matching");

        // }

        // else

        // {

            System.out.println("Text is not matching");

        // }

        // if (a.equals(bud))

        // if (a.ContentEquals(buf))

        // {

            System.out.println("Text is matching");

        // }

        // else

        // {

            System.out.println("Text is not matching");

        // }

? ?

## WebElement :-

- Anything present on a Webpage is called as a WebElement.
- WebElement can be a checkbox, radio button, link, image, text field, etc.

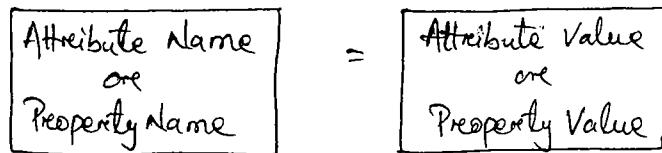
### \* Components of a WebElement :-

#### 1. HTML TAG -

- Immediate word which is present after the "<"

#### 2. ATTRIBUTE -

- It is a pair of words separated by assignment operator.



#### 3. TEXT -

- Any word which is present after the ">".

### \* How to view the HTML code of a WebElement ?

- Right click on the WebElement & select inspect.

## LOCATORS :-

- Locators are generally used to locate a WebElement present in the webpage.
- Selenium provides eight different types of locators to locate a WebElement uniquely in the Webpage.

## Types of Locators :-

1. id → All the locators are present under By class of Selenium
2. name → All the locators are static methods
3. linkText
4. PartialLinkText
5. className → All the locators take String as an argument
6. tagName
7. cssSelector → We use findElement() to locate/Search the WebElement.
8. xpath → abstract method  
WebElement Interface  
WebElement

1. id Locator :- We can locate a webElement using id locator if the webElement has an attribute as id.

```

P: Package Locator; 
    import org.openqa.Selenium.By;
    import org.openqa.Selenium.WebDriver;
    import org.openqa.Selenium.WebElement;
    import org.openqa.Selenium.ChromeDriver;

    Public class ID
    {
        P S v m(—)
        {
            System.setProperty("webdriver.Chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.get("https://en-gb.facebook.com/");
            // WebElement username = driver.findElement(By.id("email"));
            // username.sendKeys("Sam123");
            driver.findElement(By.id("email")).sendKeys("Sammy");
        }
    }
}

```

Tuesday

Note:-

- If the Specified location is not matching with any WebElement, we get NoSuchElementException.

2. name location :- We can use this location to locate a WebElement having name attribute.

P. Package Location;

```
import org.openqa.Selenium.By;
import org.openqa.Selenium.WebDriver;
import org.openqa.Selenium.WebElement;
import org.openqa.Selenium.Chrome.ChromeDriver;
```

{

PSVM(—)

{

```
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
WebDriver driver = new ChromeDriver();
```

```
driver.get("https://en-gb.facebook.com/");
```

```
driver.findElement(By.name("email")).sendKeys("Sammy");
```

{

{

3. LinkText location :- We can use this location to locate a WebElement by using its text value.

P. Public class LinkText

PSVM(—)

{

```
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
WebDriver driver = new ChromeDriver();
```

```
driver.get("https://en-gb.facebook.com/");
```

```
driver.findElement(By.linkText("Forgot your password?")).click();
```

{

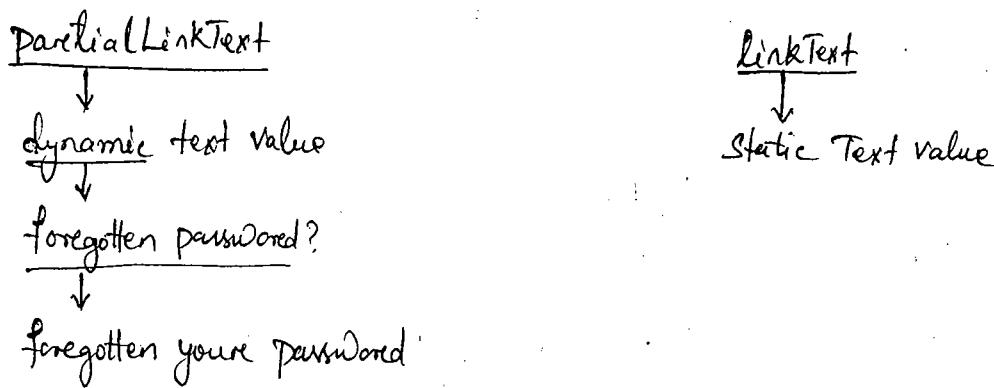
{

1. Partial LinkText locator :- We use this locator whenever there is a dynamic text value.

P. Public class partialLinkText

```

    {
        public void m()
        {
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.get("https://en-gb.facebook.com/");
            driver.findElement(By.partialLinkText("Forgotten password?")).click();
        }
    }
  
```



: Write a script to get the text of a webElement?

getText()

- abstract method
- WebElement Interface
- String

Public class GetText

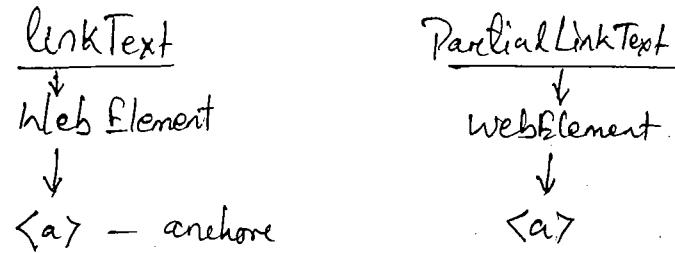
```

    {
        public void m()
        {
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.get("https://en-gb.facebook.com/");
            // WebElement link = driver.findElement(By.linkText("Forgotten password?"));
            // System.out.println("text of the webelement is :" + link.getText());
            System.out.println(driver.findElement(By.linkText("Forgotten Password?")));
        }
    }
  
```

## Assignment :-

1. Write a Script to login to <https://demo.actitime.com>.
2. Write a Script to login to <https://opensource-demo.orangehrmlive.com/>
3. Write a Script to Open actitime, print the text of forgot your password? link & click on it.
4. Write a Script to open orangehrm, print the text of forgot your password? link & click on it.
5. Write a Script to open actitime, click on actiTIME Inc link & use close() & quit().
6. Write a Script to open myntrea, click on Men link & print the title & current URL.
7. Write a Script to login to actitime & logout.
8. Write a Script to login to ~~orangehrm~~ & logout.
9. Write a Script to validate the title of login page & home page of actitime.
10. Write a Script to validate the title of login page & home page of orangehrm.
11. Write a Script to open Facebook (<en-gb.facebook.com>) & click on Create a page link.

Wednesday



<a href = "https://www.qspiders.com"> Qspiders </a>

Note:- LinkText & PartialLinkText locator will work for elements having anchor tag.

Dt- 27/01/2023

Freiday

### 5. className locator :-

→ We can use this locator to locate the element by using the class attribute.

↳ Class className

PSVM( )

```

    {
        System.setProperty("webdriver.chrome.driver", "/drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com");
        driver.findElement(By.className("inputtext")).sendKeys("Saroj");
        driver.findElement(By.className("inputtext")).sendKeys("abc");
    }
  
```

→ If the locator value is duplicate i.e. matching with multiple elements, then findElement() returns address of the first matching element.

## 6. tagname locator :-

→ Using this locator we can locate the webelement by using the HTML tag of the element.

P: class Tagname

{  
    P S V m (—)

{  
    System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  
    WebDiver driver = new ChromeDriver();  
    driver.get("https://en-gb.facebook.com/");  
    driver.findElement(By.tagName("input")).sendKeys("vishal");

}

}

→ If the element is an hidden element & we are trying to perform action on the element , we get ElementNotInteractableException or ElementNotVisibleException .

Developer's Tool Bar

Ctrl+F   

highlights all the input tags  
present in webpage.

input —

ElementNotInteractableException

input — hidden

By.tagName("input");

input — username

input — password

input — hidden

input — hidden

## 7. cssSelector Locator :-

→ Using this locator we can locate any webElement by using any attribute of the webElement.

Syntax:-

tagname [attributename = 'attributevalue']

### \* Steps to derive CSS or XPath Expression :-

- (i) Open the developer's tool bar
- (ii) Press  $\text{ctrl} + \text{F}$
- (iii) Enter the CSS or XPath expression.
- (iv) Check whether element is getting highlighted or not and  
element should be uniquely identified (one of one).

```

    Class cssSelector
    {
        public void (—) throws InterruptedException
        {
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.get("https://opensource-demo.orangehrmlive.com/");
            Thread.sleep(3000);
            driver.findElement(By.cssSelector("input[placeholder='Username']")).sendKeys("Admin");
            driver.findElement(By.cssSelector("input[type='password']")).sendKeys("admin123");
            driver.findElement(By.cssSelector("button[type='submit']")).click();
            driver.findElement(By.cssSelector("i[class='oxd-icon bi-caret-down-fill oxd-userdropdown-icon']")).click();
            Thread.sleep(3000);
            driver.findElement(By.linkText("Logout")).click();
        }
    }
  
```

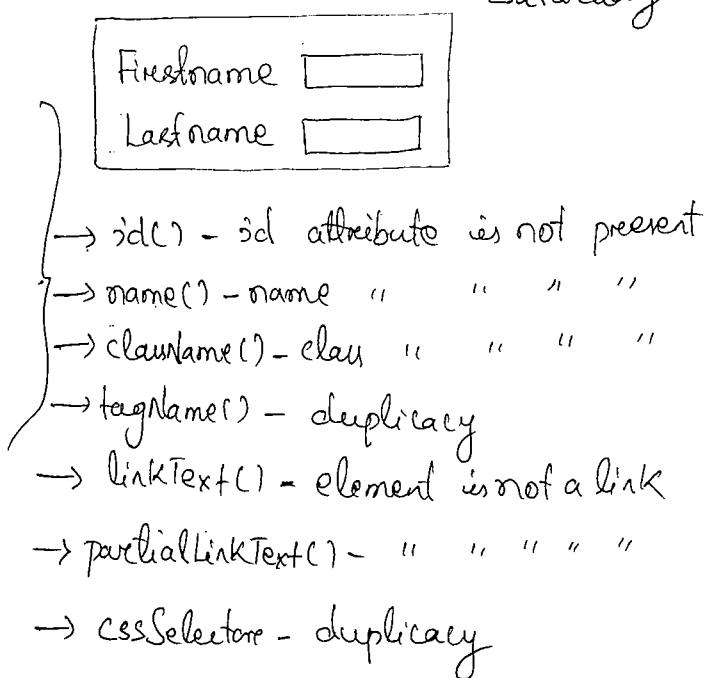
## Assignment - 2

- 1) Open Facebook , click on create new account , Enter first name , Surname , mobile num & new password.
- 2) Open <https://login.yahoo.com/account/create?> & enter first name , last name , email address , password & Birth year .
- 3) Open Instagram , Enter the username & password . & click on login button .
- 4) Open Instagram , click on Sign up & enter email address , full name , username , password .
- 5) Open <https://cremaccess.vtiger.com/log-in/> & enter username & password & click on ~~Login~~ <sup>Signin</sup> button .
- 6) Open ~~Netflix~~ <https://netflix.com/in/login> & enter username , password & click on Signin .

Dt-28/01/2023  
Saturday

### 8. Xpath :-

```
<html>
  <body>
    Firstname: <input type = "text">
    Lastname: <input type = "text">
  </body>
</html>
```



## \* Types of xpath:-

- (i) Absolute xpath
- (ii) Relative xpath
- (iii) xpath by attribute
- (iv) xpath by text
- (v) xpath by Contains
- (vi) xpath by group index
- (vii) xpath traversing
- (viii) Dependent Independent xpath

Q: What is the difference between cssSelector & xpath locator?

A:- Using cssSelector, we can locate a webElement by using its attributes.

Using xpath, we can locate a webElement by using its attributes or text value or partial text value or index value

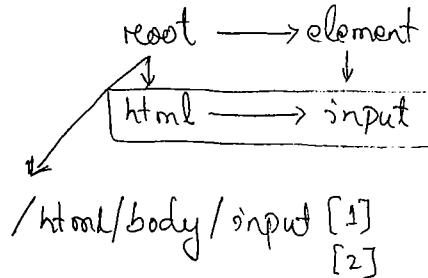
### (i) Absolute xpath:-

- Specifying the complete path of the webElement starting from root of the HTML Tree.
- Absolute xpath expression always starts from single forward slash (/) which represents root of the HTML Tree.
- The consecutive single forwarded slash represents immediate child tag.

```

<html>
  <body>
    <input type="text" />
    <input type="text" />
  </body>
</html>
  
```

Firstname	<input type="text"/>
Lastname	<input type="text"/>



## Class Absolute Xpath

```

    {
        PSvm()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Slot2/Automation/htmlpages/AbsoluteRelativeXpath.html");
        driver.findElement(By.xpath("//html/body/input[1]")).SendKeys("admin");
        driver.findElement(By.xpath("//html/body/input[2]")).SendKeys("manager");
    }
}

```

### Drawback :-

- Absolute xpath expression will be very lengthy & it takes a lot of time to derive the expression manually.
- Also it will consume a lot of time to locate the element as we are navigating via immediate child tags.

### (ii) Relative xpath :-

- In relative xpath we use double forward slash (//) which represents any or all the child tags.

## Class Relativexpath

```

    {
        PSvm()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Slot2/Automation/htmlpages/AbsoluteRelativeXpath.html");
        driver.findElement(By.xpath("//input[1]")).SendKeys("admin");
        driver.findElement(By.xpath("//input[2]")).SendKeys("manager");
    }
}

```

### Drawbacks :-

→ If there are multiple child elements with the same tagname, it is not recommended to use the index value as index value might keep on changing.

Q. What is the difference between Absolute xpath & Relative xpath ?

A:- In absolute xpath we use single forward slash which represents beginning of the HTML Tree & the consecutive forward slashes represents immediate child tag .

In relative xpath we use double forward slash which represents any child tag present in the HTML Tree .

<html>	//div - 3
<body>	//input - 6
<div>	/html - 1
<input>	//html - 1
<input>	//html - 1
</div>	/body - 0
<div>	//body - 1
<input>	//div/input - 6
<input>	//div//input - 6
</div>	//div[2]/input - 2
<div>	//div/input[1] - 3
<input>	//div[3]/input[2] - 1
<input>	//div[2]/input[3] - 0
</div>	/div - 0
</body>	
</html>	

Q. Write the relative xpath to highlight all the links, images, division, input, button, paragraph, table, table row, table division, ordered list, unordered list.

```
//a      //div    //p      //table   //td     //ul
//img    //input   //button  //tr     //ol
```

Dt - 30/01/2023

(ii) Xpath by Attribute :-

→ Using this locator we can locate the webelement by using its attributes.

Syntax :- //tagname[@attributename = 'attributevalues']  
 ↓  
 relative xpath

Class XpathByAttribute

```
{
  public void m() {
    System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get("https://en-gb.facebook.com/");
    driver.findElement(By.xpath("//input[@data-testid='royal_email']")).sendKeys("Sampreeti");
    driver.findElement(By.xpath("//input[@placeholder='Password']")).sendKeys("abc");
  }
}
```

Q. Can we use multiple attributes in a single xpath expression?

A. Yes we can use.

//tagname[@attributename = 'attributevalue'][@attributename = 'attributevalue'] . . . .

## Assignment - 3

Q. same question of Assignment - 2 using xpath.

### (iv) Xpath by groupindex :-

→ If the xpath expression is matching with multiple elements , we can locate the element uniquely by using the index value.

Syntax :-

(xpath expression) [int index]

Starts from 1

↓  
xpath by attribute

xpath by text

xpath by Contains

~~dependent~~ independent xpath

Class XpathBy

Class XpathByGroupIndex

{

P S v m (-)

{ System. SetProperty("webdriver.chrome.driver", "./driver/chromedriver.exe");

WebDrivee driver = new ChromeDriver();

driver.get("https://en-gb.facebook.com/");

driver.findElement(By.xpath("//a[@role='button'][2]")).click();

driver.findElement(By.xpath("//input[@type='text'][2]")).

Sendkeys("Samprati");

}

}

## Assignment - 4

1. Open Facebook , Click on Create a new account button & click on female radio button.
2. Login to Actitime , & click on settings .

### (v) Xpath by text :-

→ Using this locator we can locate the webelement by using its text value.

#### Syntax:-

//tagname[text() = 'textvalue']

or //tagname[. = 'textvalue']

class Xpathbytext

{  
    public void(→)

```
{ System.setProperty("webdriver.chrome.driver", "./driver/chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get("https://en-gb.facebook.com/");
    driver.findElement(By.xpath("//a[text()='Forgotten password?']")).click();
    driver.findElement(By.xpath("//a[. ='Forgotten password?']")).click();
}
```

### Assignment - 5

Same as Assignment 1.

Q: What is the difference between . & text() function in the syntax of xpath by text?

A: Using . we can highlight all the text values including hidden text value but not by using text() function.

(ii) xpath by Contains !-

→ We use xpath by Contains whenever there is a dynamic text value or dynamic attribute value.

dynamic text value

Forgotten password?

Forgotten your password?

→ //tagname[contains(text(), 'text value')]

dynamic attribute value

type = "Sampreeti Sahu"

type = "Sampreeti"

→ //tagname[contains(@attributename, 'attribute value')]

Class xpathByContains

{

P → r on (—),

{

```
System.setProperty("webdriver.chrome.driver", ".\drivers/chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("https://en-gb.facebook.com/");
driver.findElement(By.xpath("//input[contains(@placeholder, 'Email address')]")).
```

Sendkeys("Sampreeti");

```
driver.findElement(By.xpath("//a[contains(text(), 'Forgotten')]")).click();
```

}

{

→ We can also use xpath by Contains whenever there is a non-breakable space in text value or attribute value.

&nbsp; → text value

Forgotten&ampnbsppassword

→ Forgotten password

→ //tagname[contains(text(), 'text value')]

&nbsp; → attribute value

type = "arish&ampnbspsenapati"

→ arish senapati

→ //tagname[contains(@attributename, 'attribute value')]

```

class xpathbyContainsNbsp
{
    public void main()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://login.yahoo.com/account/create?");
        driver.findElement(By.xpath("//input[contains(@placeholder,'First')]")).sendKeys("admin");
        driver.findElement(By.xpath("//a[contains(text(),'Privacy')]")).click();
    }
}

```

### (Vii) Xpath Traversing :-

- Traversing means navigating from one element to another element.
- xpath supports two types of traversing
  - (i) forward Traversing — parent to any child
  - (ii) Backward Traversing — child to any parent

```

<html>
  <head></head>
  <body>
    <div>
      <input type="checkbox">
        "A"
      <input type="checkbox">
        "B"
    </div>
    <div>
      <input type="checkbox">
        "C"
      <input type="checkbox">
        "D"
    </div>
    <div>
      <input type="checkbox">
        "E"
      <input type="checkbox">
        "F"
    </div>
  </body>

```

#### (i) Forward Traversing

- (a) HTML to C → //html//div[2]/input[1]
- (b) BODY to F → //body//div[3]/input[2]
- (c) HTML to A → //html//div[1]/input[1]

#### (ii) Backward Traversing

- (a) F to HTML → //div[3]/input[2]/../../..
- (b) C to BODY → //div[2]/input[1]/../..
- (c) F to A → //div[3]/input[2]/../../..//div[1]/input[1]

## Class XpathTraversing

```

    {
        public void main()
        {
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.get("file:///C:/slot1/Automation/htmlPages/xpathTraversing.html");
            // HTML to C
            driver.findElement(By.xpath("//html//div[2]//input[1]")).click();
            //HTML to F
            driver.findElement(By.xpath("//body//div[3]//input[2]")).click();
            // HTML to A
            driver.findElement(By.xpath("//html//div[1]//input[1]")).click();
            // F to HTML
            System.out.println(driver.findElement(By.xpath("//div[3]//input[2]//..//..//..")).getTagName());
            // C to BODY
            System.out.println(driver.findElement(By.xpath("//div[2]//input[1]//..//..")).getTagName());
            // F to A
            driver.findElement(By.xpath("//div[3]//input[2]//..//..//div[1]//input[1]")).click();
        }
    }

```

Assignment - 6

- 1) Open Facebook & perform Forwarded Traversing from HTML to Username & enter data in Username.
- 2) Open Facebook & perform Forwarded Traversing from Body to Password & enter data in Password.
- 3) Open Facebook & perform Backward Traversing from Username to HTML & print the tagname.

(vii) Dependent Independent xpath :-

→ In order to handle completely dynamic elements, we use a static element & we use the concept of traversing.

\* Steps to derive Dependent Independent xpath expression:-

1. Identify the xpath of static element (independent element)
2. Perform Backward Traversing until both static & dynamic elements get highlighted.
3. Identify the xpath of dynamic element (dependent element)

## Class DependentIndependentxpath

```

    {
        public void m() {
            System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.get("https://www.selenium.dev/downloads/");
            driver.findElement(By.xpath("//p[.= 'Python']//a[.= 'Changelog']")).click();
        }
    }
  
```

Assignment - 7

- 1) Login to Actitime, click on settings, click on Types of Work, click on Create Type of Work, enter the work name, click on Create type of Work, click on Delete link of Created Work.
- 2) Open qspidern.com/contact & print the mobile numbers of qspiders USA
- 3) Login to orangehrm, click on directory & print the designation of Peter Mae Anderson.

## Methods of WebElement Interface :-

1. SendKeys() - It is used to enter the data into a text field  
(void) or text area.
2. click() - It is used to click on elements such as  
(void) image, checkbox or radiobutton, etc.
3. getText() - It is used to get the text of the WebElement  
(String)
4. getTagName() - It is used to get the HTML Tag of the  
(String) WebElement.
5. clear() - This method is used to remove the text present  
(void) in a text field or text area.

Q. Write a Script to remove the text present in a text field.

```

Ar class clear
{
    public void () throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");
        WebElement username = driver.findElement(By.id("email"));
        username.sendKeys("Swastik");
        Thread.sleep(3000);
        username.clear();
    }
}

```

Q. Write a Script to replace the text present in a text field.

A:- class ReplaceText

{

    public void (-)

    {  
        System.setProperty("webdriver.chrome.driver", "./drivers/Chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://en-gb.facebook.com/");  
        WebElement username = driver.findElement(By.id("email"));  
        username.sendKeys("Swastik");  
        username.clear();  
        username.sendKeys("Mohapatra");  
    }

}

Q. Write a Script to remove the text without using clear method.

A:- By using Keys Enumeration B.

Enum in Java is a collection of constant values

→ Keyboard Operation

↓                   ↓  
 Delete            Constant  
 Backspace

class DeleteBackspace

{

    public void (-)

    {  
        System.setProperty("webdriver.chrome.driver", "./drivers/Chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://en-gb.facebook.com/");  
        WebElement username = driver.findElement(By.id("email"));  
        username.sendKeys("Sampredi");  
        username.sendKeys(Keys.CONTROL, "a");  
        //username.sendKeys(Keys.BACK\_SPACE);  
        username.sendKeys(Keys.DELETE);  
    }

}

Q. Write a Script to perform Copy & Paste Operation.

Ans Class CopyPaste

```

    {
        P S V m ( - )
    {
        System . SetProperty ("webdriver . chrome . driver ", "/drivers/chromedriver . exe ");
        WebDriver driver = new ChromeDriver ();
        driver . get ("https : //en - gb . facebook . com /");
        WebElement username = driver . findElement (By . id ("email"));
        username . SendKeys ("Sampreeti");
        username . SendKeys (Keys . CONTROL , "A");
        username . SendKeys (Keys . CONTROL , "X"); // use "X" for cut
        driver . findElement (By . id ("pass")) . SendKeys (Keys . CONTROL , "V");
    }
}

```

Q. Write a Script to login to any application without clicking on login button.

Ans Class EnterReturn

```

    {
        P S V m ( - )
    {
        System . SetProperty ("webdriver . chrome . driver ", "/drivers/chromedriver . exe ");
        WebDriver driver = new ChromeDriver ();
        driver . get ("https : //demo . actitime . com");
        driver . findElement (By . id ("username")) . SendKeys ("admin");
        WebElement password = driver . findElement (By . name ("pwd"));
        password . SendKeys ("manager");
        password . SendKeys (Keys . ENTER);
    }
}

```

Q. Write a script to remove the text present in the textfield one by one.

Ans

```

class RemoveText
{
    public void main() throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");
        WebElement username = driver.findElement(By.id("email"));
        username.sendKeys("Sampreeti");
        for(int i=0; i<=7; i++)
        {
            Thread.sleep(1000);
            username.sendKeys(Keys.BACK_SPACE);
        }
    }
}

```

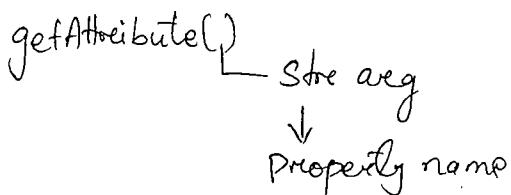
### Assignment - 8

#### Facebook username

1. Enter qspiders & remove \$ e & P.
2. Enter qspiders & then enter Pyspiders in the beginning.  
or pyspidersqspiders
3. Enter qspiders, give space & enter jspiders  
or qspiders jspiders
4. Enter qspiders in uppercase.
5. Enter ~~qspiders~~ in lowercase
6. Enter qspiders in username, press tab & enter abc in password.

6) getAttribute() - (String)

→ It is used to get the property value of the specified property name.



Q. Write a script to fetch the property value of a specified property name.

Ans

```

class GetAttribute
{
    public void run()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.facebook.com/");
        System.out.println(driver.findElement(By.id("m_login_email")).getAttribute("placeholder"));
    }
}
  
```

Q. Write a script to get the value present inside a text field.

Ans

```

class GetValue
{
    public void run()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.facebook.com/");
        WebElement username = driver.findElement(By.id("m_login_email"));
        username.sendKeys("areel");
        System.out.println(username.getAttribute("value"));
    }
}
  
```

Q. Write a script to fetch the length of the value present inside the text field.

```
Ar Class FetchLength
{
    PS VM C()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.facebook.com/");
        WebElement username = driver.findElement(By.id("m-login-email"));
        username.sendKeys("Sampredi");
        String value = username.getAttribute("value");
        System.out.println(value.length());
        for(int i=0; i<value.length(); i++)
        {
            username.sendKeys(Keys.BACK_SPACE);
        }
    }
}
```

Q. Write a script to perform Copy & Paste without using shortcut keys.

```
Ar Class CopyPaste
{
    PS VM C()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.facebook.com/");
        WebElement username = driver.findElement(By.id("m-login-email"));
        username.sendKeys("Sampredi");
        String value = username.getAttribute("value");
        WebElement password = driver.findElement(By.id("m-login-password"));
        password.sendKeys(value);
        if(password.getAttribute("value").equals("Sampredi"))
        {
            System.out.println("Successful copy paste");
        }
        else
        {
            System.out.println("Not successfully copy paste");
        }
    }
}
```

Q: Write a script to check whether the text field is empty or not.

```
Ans: class IsEmpty  
{  
    public static void main(String[] args)  
    {  
        System.setProperty("webdriver.chrome.driver", "./drivers/Chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.facebook.com/");  
        WebElement username = driver.findElement(By.id("email"));  
        username.sendKeys("");  
        String value = username.getAttribute("Value");  
        // System.out.println(value.isEmpty()); // false  
        System.out.println(value.isBlank()); // true  
    }  
}
```

Dt - 03/02/2023

Q: What is the difference b/w isEmpty() & isBlank().

A: isEmpty() will check whether the element is empty or not.

isEmpty() considers space as a character.

isBlank() will check whether the element is blank or not.

isBlank() <sup>does not</sup> consider space as a character.

(7) GetLocation() -

This method is used to get the location of the webElement.

abstract method

WebElement Interface

Point(x,y)

class GetLocation

{

    public void main()

    {  
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.facebook.com/");  
        WebElement username = driver.findElement(By.id("email"));  
        System.out.println(username.getLocation());  
    }

}

(8) GetSize :-

→ This method is used to get the size of the WebElement.

abstract method

WebElement Interface

Dimension (width, height)

class GetSize

{

    public void main()

    {  
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.google.com/");  
        WebElement image = driver.findElement(By.xpath("//img[@alt='Google']"));  
        System.out.println(image.getSize());  
    }

}

}

i) getCssValue(string arg) :- abstract method  
WebElement Interface  
String

→ This method is used to get the CSS properties of a WebElement.

class GetCssValue

{  
    public String()

```
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.facebook.com/");
        WebElement username = driver.findElement(By.id("m-login_email"));
        System.out.println(username.getCssValue("font-size"));
        System.out.println(username.getCssValue("font-family"));
        System.out.println(username.getCssValue("color"));
        System.out.println(username.getCssValue("background-color"));
    }
}
```

10) getRect() :- abstract Method, WebElement Interface, Rectangle.

→ This method is used to get the size of the WebElement as well as location of the WebElement.

class GetRect

{  
    public Rectangle()

```
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.facebook.com/");
        WebElement username = driver.findElement(By.id("m-login_email"));
        Rectangle re = username.getRect();
        System.out.println("width of the element is :" + re.width);
        System.out.println("height of the element is :" + re.height);
        System.out.println("x-axis position of the element is :" + re.x);
        System.out.println("y-axis position of the element is :" + re.y);
    }
}
```

Q. Write a Script to check whether element is displayed or not, enabled or not & Selected or not.

Ar Class IsDisplayed Enabled Selected

```
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("https://demo.actitime.com/");
WebElement checkbox = driver.findElement(By.id("keepLoggedInIncheckbox"));
if(checkbox.isDisplayed())
{
    System.out.println("element is displayed");
}
if(checkbox.isEnabled())
{
    System.out.println("element is enabled");
    //checkbox.click();
}
if(checkbox.isSelected())
{
    System.out.println("checkbox is selected");
}
```

### \* Handling Multiple WebElements :-

→ In order to handle multiple webElements , we use findElements() .

`findElements()` — abstract method

## WebDriver Interface

Diagram illustrating the inheritance relationship:

```

graph TD
    Interface[Interface] --> javaUtil[java.util]
    javaUtil --> ListWebElement[List<WebElement>]
    ListWebElement --> Interface
    ListWebElement --> getMethod["get() - returns the WebElement present"]
  
```

WebElement present  
inside the specific index.

inside the specific

inside the specific index.

↳ abstract method

↳ klebeElement

- Size() refers to the total no. of elements present in the List
- abstract method
- int

```

<html>
<body>
<input type = "checkbox"> A <br><br>
<input type = "checkbox"> B <br><br>
<input type = "checkbox"> C <br><br>
<input type = "checkbox"> D <br><br>
<input type = "checkbox"> E <br><br>
<input type = "checkbox"> F <br><br>
<a href = "https://www.google.com"> google </a> <br><br>
<a href = "https://www.myntra.com"> myntra </a> <br><br>
<a href = "https://www.flipkart.com"> flipkart </a> <br><br>
</body>
</html>

```

## Class MultipleCheckBox

```

{
    public void()
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///C:/Users/gsp/Desktop/Automation/Automation/HtmlPages/
                    MultipleElements.html");
        List<WebElement> checkboxes = driver.findElements(By.xpath(
            "//input[@type='checkbox']"));
        System.out.println("size of the list is : " + checkboxes.size());
        //click on all the checkboxes
        for(int i=0 ; i<checkboxes.size(); i++)
        {
            checkboxes.get(i).click();
        }
        //click on alternate checkbox
        for(int i=0 ; i<checkboxes.size(); i+=2)
        {
            checkboxes.get(i).click();
        }
        //click on alternate checkbox in reverse order
        for(int i=checkboxes.size() - 1 ; i>=0 ; i-=2)
        {
            checkboxes.get(i).click();
        }
    }
}

```

## Assignment - 9

- Write a Script to print the text value & the URL of all the links of Myntea.

Dt - 04/02/2023

## Assignment - 10

- Write a Script to print the text & URL of Men, Women, Kids, Home & Living, Beauty, Studio.
- Write a Script to store Men, Women, Kids, Home & Living, Beauty, Studio inside a list & later click on Men link.

## Handling AutoSuggestion :-

```

class AutoSuggestion
{
    public (--) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        driver.findElement(By.name("q")).sendKeys("Selenium");
        Thread.sleep(3000);
        List<WebElement> Suggestions = driver.findElements(By.xpath("//span[contains(text(), "Selenium")]"));
        System.out.println(Suggestions.size());
        // Using for loop & if cond
        for (int i=0; i < Suggestions.size(); i++)
        {
            if (Suggestions.get(i).getText().equalsIgnoreCase("Selenium testing"))
            {
                Suggestions.get(i).click();
                break;
            }
        }
    }
}

```

// using index value

// Suggestions . get(5).click();

// using Keys enum

```
for(int i=0; i<=5; i++)
```

{

Thread.Sleep(1000);

Search.SendKeys(Keys.ARROW\_DOWN);

{

Search.SendKeys(Keys.ENTER);

}

}

### Assignment-11

1. Write a script to handle the autosuggestion of Yahoo.com.
2. Write a script to handle the autosuggestion of Bing.com.
3. Write a script to open flipkart, handle the POPUP. Enter 'iphone14' & click on 'iphone14 Cover'.
4. Open Myntra, Enter "Tommy" ~~is~~ in search bar & click on 'Tommy Hilfiger Bags'.
5. Open Amazon, Enter 'iphonell4' in search bar & click on 'iphonell4 Cover with magSafe'.

Dt - 06/02/2023

## Handling Disappearing Suggestions :-

(i) Control + Shift + P

↳ Run > [focus]

↳ Emulate a focused page

(ii) developer tool bar

↳ Event Listener

↳ blur

↳ remove

findElement()

(i) Return type is WebElement.

(ii) If the specified location is matching with multiple elements, findElement() returns the address of 1st matching element.

(iii) If the specified location is not matching with any element, ~~findElement()~~ throws NoSuchElementException.

findElements()

(i) Return type is List of WebElement.

(ii) If the specified location is matching with multiple elements, findElements() returns the address of all the matching elements.

(iii) If the specified location is not matching with any element, findElements() throws EmptyList (size = 0).

## Handling DropDownList :-

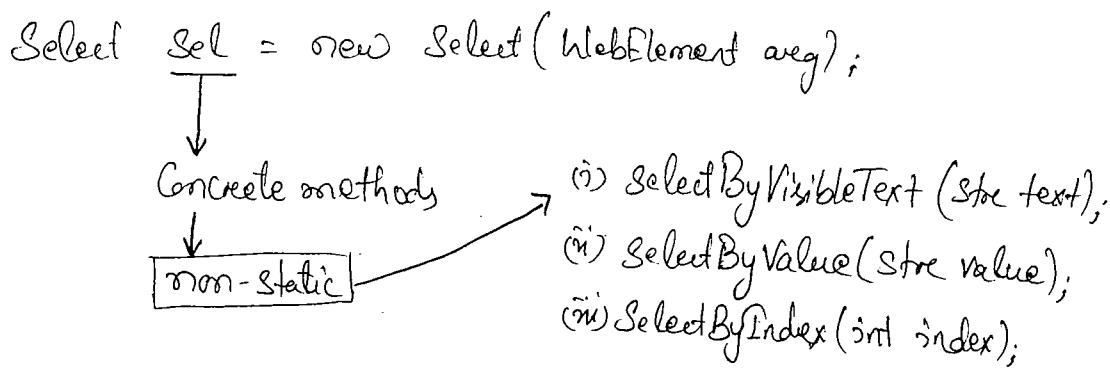
→ A DropDownList is generally developed using <Select>.

→ The Options of the DropDownList is generally developed using <option>.

→ In order to handle DropDownList we use Select class of Selenium.

→ Select class constructor takes an argument of type WebElement.

Select sel = new Select (WebElement arg);



<html>

<body>

```

<Select name="Students" single size=5>
  <option value="BS"> Biswanath </option>
  <option value="DM"> Debanish </option>
  <option value="AC"> Areal </option>
  <option value="PL"> Priyanka </option>
  <option value="SL"> Swendu </option>
</Select>

```

</body>

</html>

Class SingleSelect

{  
 public void ()  
 {  
 }

---

driver.get ("—");

```

WebElement dropdownlist = driver.findElement(By.xpath("//select[@name='Students']"));
Select select = new Select(dropdownlist);
//Select.selectByVisibleText ("Areal");
//Select.selectByValue ("DM");
Select.selectByIndex (3);
}

```

Note:-

- If multiple options are there with the same text or attribute value, Select class returns the address of first matching option.
- If the option is not matching with any element, we get NoSuchElementException.

☞

- \* We can also use Select class to handle multiple dropdownlist.
- \* We will use the above methods of Select class to select multiple options.
- \* We can use Select class methods to deselect an option of multiselected dropdownlist.
  - (i) deselectAll();
  - (ii) deselectByVisibleText(String text);
  - (iii) deselectByValue(String value);
  - (iv) deselectByIndex(int index);

Note:-

- If you try to deselect an option of a single select dropdownlist, we get UnsupportedOperationException.
- If we are trying to use the Select class methods & the tagname is not select, we get unexpected tagname exception.

&lt;html&gt;

&lt;body&gt;

&lt;select name="Student" multiple size=6&gt;

&lt;option value="BS"&gt; Biswa &lt;/option&gt;

&lt;option value="DM"&gt; Dev &lt;/option&gt;

&lt;option value="AC"&gt; Areej &lt;/option&gt;

&lt;option value="PL"&gt; Prayankha &lt;/option&gt;

&lt;option value="SL"&gt; Suwendu &lt;/option&gt;

&lt;option value="AC"&gt; Areej &lt;/option&gt;

&lt;/select&gt;

&lt;/body&gt;

&lt;/html&gt;

## Class Multiselect

{

---



---

```
driver.get("—");
```

```
WebElement dropdownlist = driver.findElement(By.xpath("//Select[@name='Students']"));
```

```
Select select = new Select(dropdownlist);
```

```
Select.selectByVisibleText("Areal");
```

```
Select.selectByValue("PL");
```

```
Select.selectByIndex(0);
```

```
//Select.deselectAll();
```

```
//Select.deselectByVisibleText("Preixanka");
```

```
//Select.deselectByValue("AC");
```

```
Select.deselectByIndex(0);
```

{

## Class IsMultiple

{

```
System.out.println("—");
```

{

---



---

```
driver.get("—");
```

```
WebElement dropdownlist = driver.findElement(By.xpath("//Select[@name='Students']"));
```

```
Select s = new Select(dropdownlist);
```

```
System.out.println(s.isMultiple());
```

{

{

## Assignment - 12

1. Open Facebook, click on Create new account & enter your D.O.B.
2. Open <https://www.americanairlines.in/intl/in/index.jsp?>, Enter 'Delhi' in From city & Select 'San Antonio'(SAT), Texas, USA', Enter 'Las' in To city & Select 'Las Vegas(LAS), Nevada, USA', Enter no.of adults as 4 & no.of childrens as 4 ..
3. Write a Script to check whether DropdownList is Single or multi Select .

`isMultiple()`

- non-static
- Select class
- boolean

4. Write a Script to print all the options of the DropdownList .

`getOptions()`

- non-static
- Select class
- `List<WebElement>`

5. Write a Script to print all the selected options of a DropdownList .

`getAllSelectedOptions()`

- non-static
- Select class
- `List<WebElement>`

## \* Class GetOptions

```
P S V m (→)
{
```

=====

```
driver.get("___"),
```

```
WebElement dropdownlist = driver.findElement(By.xpath("//select[@name='student']"));
```

```
Select select = new Select(dropdownlist);
```

```
List<WebElement> options = select.getOptions();
```

```
for(int i=0; i < options.size(); i++)
```

```
{
```

```
Sopln(options.get(i).getText());
```

```
}
```

```
}
```

## \* Class GetAllSelectedOptions

```
P S V m (→)
{
```

=====

```
driver.get("___");
```

```
WebElement dropdownlist = driver.findElement(By.xpath("//select[@name='student']"));
```

```
Select select = new Select(dropdownlist);
```

```
S.selectByVisibleText("Areal");
```

```
S.selectByVisibleText("Prayankar");
```

```
S.selectByVisibleText("Bhwanath");
```

```
List<WebElement> allselectedOptions = select.getAllSelectedOptions();
```

```
for(int i=0; i < allselectedOptions.size(); i++)
```

```
{
```

```
Sopln(allselectedOptions.get(i).getText());
```

```
}
```

}

```
3
```

## getFirstSelectedOption():-

non-static

Select class

WebElement

→ This method returns the element which is present in the first position of the DropdownList out of the selected options.

Class GetfirstSelectedOption

{

    public void ( )

{

    \_\_\_\_

    \_\_\_\_

        driver.get("\_\_\_\_");

        WebElement dropdownlist = driver.findElement(By.xpath("//select

[@name='Students']"));

        Select select = new Select(dropdownlist);

        Select.selectByVisibleText("Arul");

        Select.selectByVisibleText("Preyanka");

        Select.selectByVisibleText("Biswa");

        System.out.println(select.getFirstSelectedOption().getText());

}

}

## Select class Method :-

- 1) selectByVisibleText (String text) - void
- 2) selectByValue (String value) - void
- 3) selectByIndex (int index) - void
- 4) deselectAll() - void
- 5) deselectByVisibleText (String text) - void
- 6) deselectByValue (String value) - void
- 7) deselectByIndex (int index) - void
- 8) isMultiple() - boolean
- 9) getOptions() - List<WebElement>
- 10) getAllSelectedOptions() - List<WebElement>
- 11) getFirstSelectedOption() - WebElement

## Assignment - 13

- Write a Script to Sort the elements of the DropDownList in ascending & descending order.
- Write a Script to print the unique contents of the DropDownList
- Write a Script to print the unique contents of the DropDownList in ascending order.
- Write a Script to check whether the DropDownList has duplicates or not.
- Write a Script to print the occurrence of 'Aru' in the DropDownList.
- Write a Script to print the occurrence of all the elements present in the DropDownList.

## Actions Class:-

- In order to handle different mouse operations, We use actions class of Selenium.
- Actions class should be imported from interactions package.
- Actions class constructor takes an argument of type WebDriver.
- Whenever we use any methods of Actions class, at the end we must call perform() to perform the operation on the element.

Actions actions = new Actions(WebDriver arg);

we → click(WebElement arg); → Control goes to the element & perform click operation.

click(); → perform click operation on default Mouse location.

Q. Write a Script to Perform click operation on a WebElement ?

A:

Class Click

{

PS VM (-)

{

\_\_\_\_\_

driver.get(" ~~www~~ https://www.facebook.com/");

WebElement link = driver.findElement(By.xpath("//a[.=

'Forgotten password?'));

Actions actions = new Actions(driver);

actions.click(link).perform();

}

}

Q. Write a Script to perform DoubleClick on a WebElement ?

A:

Class DoubleClick

{

PS VM (-)

{

\_\_\_\_\_

driver.get("https://www.facebook.com/");

WebElement link = driver.findElement(By.xpath("//a[.=

'Forgotten password?'));

Actions actions = new Actions(driver);

actions.doubleClick(link).perform();

}

}

- Write a Script to perform right click on a WebElement.
- Right Click is also called as ContextClick in Selenium.

Class RightClick\_ContextClick

{

    PSVM (→)

{

        driver.get("https://www.facebook.com/");

        WebElement link = driver.findElement(By.xpath("//a

[. = 'Forgotten Password ? J"]));

        Actions actions = new Actions(driver);

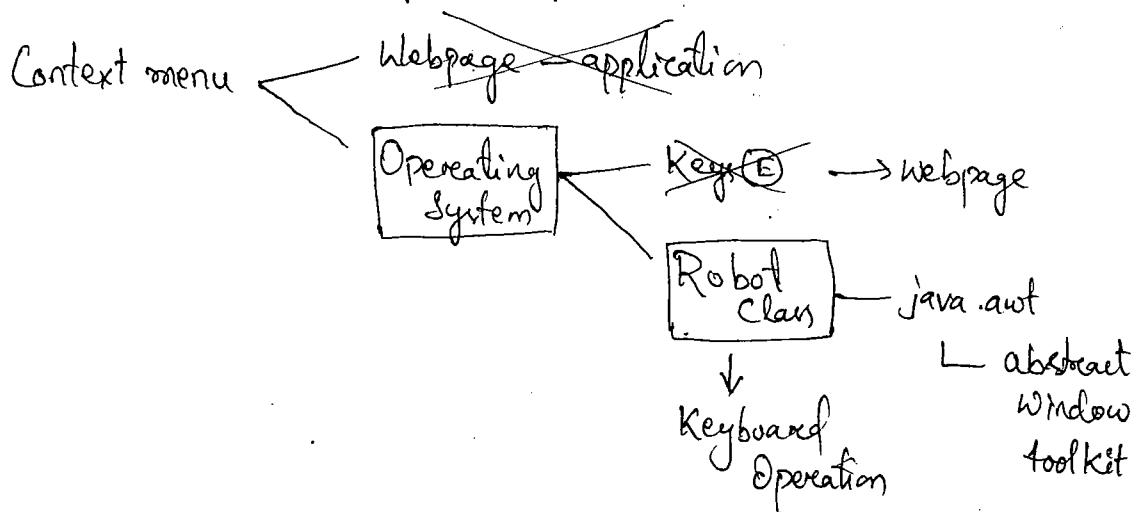
        actions.contextClick(link).perform();

}

}

Dt - 08/02/2023

- Write a Script to perform Operation on the Context menu.



Class ContextMenu

```
{ P S V m ( → ) }
```

---



---

driver.get("https://en-gb.facebook.com/");

WebElement link = driver.findElement(By.xpath("//a[.=

\*'Forgotten password?']"));

Actions actions = new Action(driver);

actions.contextClick(link).perform();

Robot r = new Robot();

r.keyPress(KeyEvent.VK\_DOWN);

r.keyPress(KeyEvent.VK\_DOWN);

r.keyPress(KeyEvent.VK\_ENTER);

r.keyRelease(KeyEvent.VK\_DOWN);

r.keyRelease(KeyEvent.VK\_ENTER);

Note:- Robot class is generally used to perform keyboard operations on the elements of the operating system.

→ Robot class should be imported from java.awt package.

→ Inside Robot class we have methods like keyPress() & keyRelease().

→ Both the methods take keycode as an argument.

→ Keycode means the physical key associated with the keyboard.

→ All the keycodes are present in the form of static variable inside KeyEvent class.

Q: Write a script to perform mouse hovering on an element.

A:-

Class Mousehovering

```
{ P S V m ( → ) }
```

---



---

driver.get("https://www.onyxta.com");

WebElement menLink = driver.findElement(By.xpath

70

(" //a [ . = 'Men' ] ));

WebElement sweatshirtLink = driver.findElement(By.xpath

(" //a [ . = 'Sweatshirts' ] ));

Actions actions = new Actions(driver);

actions.moveToElement(menLink).perform();

actions.click(sweatshirtLink).perform();

}

}

Q. Write a Script to perform Drag & Drop Operation?

R. Generally we drag the element from the source & drop the element onto the target.

Class DragDrop

{

PS von (-)

{

\_\_\_\_\_

\_\_\_\_\_

~~driver~~

driver.get("http://www.htmlgoodies.com/scripts/drag-drop  
-Custom/demo-drag-drop-3.html");

WebElement source = driver.findElement(By.xpath

(" //div [ . = 'Washington' ] [2] ));

WebElement target = driver.findElement(By.xpath

(" //div [ . = 'United States' ] ));

Actions actions = new Actions(driver);

actions.dragAndDrop(source, target).perform();

}

}

Q. Write a Script to perform click & Hold Operation.

Ans class ClickAndHold

{  
    public void(—)

=====

```
driver.get("https://www.facebook.com/");
WebElement link = driver.findElement(By.xpath
    ("//a[.= 'Forgotten password ?']"));
```

```
Actions actions = new Actions(driver);
actions.clickAndHold(link).perform();
actions.release(link).perform();
```

}

}

#### \* Methods of Actions Class :-

- 1) click(WebElement arg) - void Actions
- 2) doubleClick(WebElement arg) - void Actions
- 3) contextClick(WebElement arg) - void Actions
- 4) moveToElement(WebElement arg) - void Actions
- 5) dragAndDrop(WebElement source, WebElement Target) - void Actions
- 6) clickAndHold(WebElement arg) - void Actions
- 7) release(WebElement arg) - void Actions
- 8) perform() - void
- 9) sendKeys() - void Actions
- 10) scrollTOElement() - void Actions
- 11) dragAndDropBy(~~WebElement~~(webElement, min, max)) - void Actions
- 12) pause() - void Actions

## ArrayList

- ArrayList is an implementation of List Interface.
- ArrayList maintains the insertion order.
- ArrayList allows duplicate values.
- ArrayList allows any no. of null values.

## HashSet

- HashSet does not maintain the insertion order rather it follows hashCode.
- HashSet uses equals() to compare two objects.
- HashSet allows null object.
- HashSet does not allow heterogeneous objects.

## HashSet

- HashSet is an implementation of Set Interface.
- HashSet does not allow duplicate values.
- In HashSet we store the objects.
- HashSet uses add() to add the objects.

## HashSet

- HashSet is an implementation of Set Interface.
- HashSet does not maintain the insertion order rather it follows hashCode.
- HashSet does not allow duplicate value.
- HashSet allows only one null value.

## TreeSet

- TreeSet does not follow insertion order rather it follows sorted order.
- TreeSet uses Compare() to compare two objects.
- TreeSet does not allow null object, it throws NullPointerException.
- TreeSet allows homogeneous heterogeneous objects.

## HashMap

- HashMap is an implementation of Map Interface.
- HashMap allows duplicate value but does not allow duplicate key.
- In HashMap we store the key & value Pairs.
- HashMap uses put() to add the objects.

## \* Handling Popups :-

→ A Popup is a Window that displays one pops-up on the screen due to some activity.

1. Alert Confirmation
2. Hidden Division
3. Calendere
4. Notification / Web Push notification
5. Child Browser/ Window
6. File Download
7. File Upload
8. Prompt Popup

### 1. Hidden Division Popup -

- We can inspect the popup.
- We cannot move the popup.
- Popup will be colourless.
- Inorder to handle hidden division popup , we use `findElement()`.

Class HiddenDivision

```
{  
    public void m() {  
        _____  
        _____  
    }  
}
```

```
driver.get ("https://www.skillrary.com");
```

```
driver.findElement(By.xpath("//a[@class = 'closeCookie']"))
```

• `click()`,

```
}
```

## 2) Notification Popup

- We cannot inspect the popup.
- We cannot move the popups.
- It will have allow & block notification options.
- In order to handle Notification Popup, we use ~~ChromeOptions~~ class.

### ~~Class Notification~~

{

P S r m c →

```
ChromeOptions options = new ChromeOptions();
options.addArguments("disable-notifications");
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
WebDriver driver = new ChromeDriver(options);
driver.get("https://www.singaporeair.com/");
```

{

{

Note:- ChromeOptions class is a concept of Selenium WebDriver which is used to manipulate various properties of the ChromeDriver.

Dt-16/02/2023

## 3) Alert Confirmation Page & Popup :-

- We cannot inspect the pop-up.
- We cannot move the pop-up.
- It will have 'OK' & 'CANCEL' Button.
- In order to handle Alert Confirmation Popup, we need to Switch the control of Selenium from the webpage to the popup.
- Once the popup is handled, control automatically comes to the webpage.
- In order to click on 'OK' button, we use accept() of Alert interface.

→ In order to click on 'CANCEL' button, we use dismiss() of ~~Alert~~ Alert interface.

### Class AlertConfirmation

```
{  
    public void ()  
    {  
        _____
```

```
        driver.get("https://demo.actitime.com/");  
        driver.findElement(By.id("username")).sendKeys("admin");  
        driver.findElement(By.name("pwd")).sendKeys("manager");  
        driver.findElement(By.id("loginButton")).click();  
        driver.findElement(By.xpath("//div[@class='menu-icon'][2]")).click();  
        driver.findElement(By.xpath("//a[.= 'Types of Work']")).click();  
        driver.findElement(By.xpath("//span[.= 'Create Type of Work']")).click();  
        driver.findElement(By.id("name")).sendKeys("qspiders");  
        driver.findElement(By.xpath("//input[@type = 'submit']")).click();  
        driver.findElement(By.xpath("//a[.= 'qspiders']//a[contains( text(), 'delete')]")).click();  
        System.out.println(driver.switchTo().alert().getText());  
        driver.switchTo().alert().accept(); // Click on OK  
        //driver.switchTo().alert().dismiss(); // Click on Cancel
```

### Note :-

- If we are trying to handle the ~~Alert~~ Alert Confirmation Popup & the Popup is not present, We get ~~NoAlertPresentException~~.
- Without handling the Popup if we are trying to perform any operation on the elements of the webpage, We get ~~UnhandledAlertException~~.

### Class UnhandledAlertException

```
{  
    public void ()  
    {  
        _____
```

```
        driver.get("https://demo.actitime.com/");  
        driver.findElement(By.id("username")).sendKeys("admin");  
        driver.findElement(By.name("pwd")).sendKeys("manager");
```

```

driver.findElement(By.id("loginButton")).click();
driver.findElement(By.xpath("//div[@class='menu-icon'][2]")).click();
driver.findElement(By.xpath("//a[.= 'Types of Work']")).click();
driver.findElement(By.xpath("//span[.= 'Create Type of Work']")).click();
driver.findElement(By.id("name")).SendKeys("qspiders");
driver.findElement(By.xpath("//input[@type='Submit']")).click();
driver.findElement(By.xpath("//a[.= 'qspiders']//..//a[contains(text(), 'delete')]")).click();
driver.findElement(By.id("logoutLink")).click();
}
}

```

## Calendar Popup

→ It is a type of hidden division popup which is used to Select the dates from the Calendar.

class Calendar

```

{
    public void open() {
        driver.get("https://www.brightwhitelabels.com/");
        driver.findElement(By.id("ensCloseBanner")).click();
        driver.findElement(By.xpath("//div[.= 'Outbound']")).click();
    }
}
```

```

driver.get("https://www.brightwhitelabels.com/");
driver.findElement(By.id("ensCloseBanner")).click();
driver.findElement(By.xpath("//div[.= 'Outbound']")).click();
WebElement monthYear = driver.findElement(By.xpath(
    "//span[@class = 'month-name bold ng-binding']"));
String month = monthYear.getText();
String year = monthYear.getText().split(" ")[0];
String year = monthYear.getText().split(":")[1];

```

```

while (!month.equals("January") & year.equals("2024"))
{
    driver.findElement(By.xpath("//span[@data-icon='J'][2]")).click();
    String monthlyear1 = driver.findElement(By.xpath("//span[@class='month-name bold ng-binding']")).getText();
    month = monthlyear1.split(" ")[0];
    Year = monthlyear1.split(" ")[1];
}
driver.findElement(By.xpath("//span[.= '15']")).click();

```

### Assignment - 14

1. Open Americanairlines & enter departure & return date.
2. Open goindigo.in & enter departure (Jan 15) & return date.

Dt: 20/02/2023

### → Prompt Popup

- We cannot inspect the pop-up.
- We cannot move the pop-up.
- In order to handle this pop-up, we need to switch to the pop-up & then enter some data.

#### Class Prompt

```

{
    PSVM()
}
```

```

driver.get("https://the-internet.herokuapp.com/javascript-alerts");
driver.findElement(By.xpath("//button[@onclick='jsPrompt()']")).click();
Alert alert = driver.switchTo().alert();
alert.sendKeys("adown");
alert.accept();
String actualtext = driver.findElement(By.id("result")).getText();
if(actualtext.contains("adown"))
    System.out.println("entered the data into the prompt popup");
else
    System.out.println("didn't enter the data properly");
}

```

## 3) Authentication Popup

- We cannot ignore the popup.
- We cannot move the popup.
- In order to handle the popup, we need to authenticate the username & password.

class Authentication

```
{ PSVM(-)
{
```

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

```
"https://username:password@URL"
```

## 4) Child Browser / Window Popup

- We can ignore the popup.
- We can move the popup.
- We can close the popup.
- In order to handle child browser popup, we use getWindowHandle().
- Window Handle is nothing but the unique alpha numeric string value of the window.

getWindowHandle()

Absract method

WebDriver Interface

String

getWindowsHandles()

Absract method

Set<String> Interface

String

## P: Class WindowChildBrowser

```
{ public void m() {
    {
```

```
        driver.get("https://demo.actitime.com");
    }
```

```
    driver.findElement(By.xpath("//a[.= 'actiTIME Inc.']").click();
```

```
    System.out.println(driver.getWindowHandle());
```

```
    System.out.println(driver.getWindowHandles()); // Set<String>
```

```
}
```

```
}
```

## P:

## Class ActiTimeChildWindowPopUp

```
{ public void m() {
    {
```

```
        driver.get("https://demo.actitime.com");
    }
```

```
    driver.findElement(By.xpath("//a[.= 'actiTIME Inc.']").click();
```

```
    .click();
```

```
    Set<String> allWindows = driver.getWindowHandles();
```

```
    System.out.println("allWindows size is :" + allWindows.size());
```

```
    ArrayList<String> arraylist = new ArrayList<>()
                                    (allWindows);
```

```
    System.out.println("arraylist.size is :" + arraylist.size());
```

```
    for(int i=0 ; i < arraylist.size(); i++)
```

```
{
```

```
        driver.switchTo().window(arraylist.get(i));
```

```
        String actualTitle = driver.getTitle();
```

if(actualTitle.contains("Time Tracking"))

{

driver.findElement(By.xpath("//a[.= 'Trey Free']")) .  
click();

}

}

}

### Assignment - 15

- 1) Open Myntra, Enter 'Levis' in search bar & click on 'Levis Sneakers', click on Men filters radio button, click on 1st price range filter, click on Navy Blue & Black Colour, Click on the 1st product, switch to the new tab, Print the price of the product, discount percentage, <sup>Select size &</sup> add the product to bag. Click on bag, click on Enter Pincode, Enter Pincode & click on check. Click on Place Order.
- 2) Open Myntra, Open men, women & kids in new tabs. Switch to kids & print the title, print all the favorite brands, Switch back to homepage, Switch to men & print the ~~discount percentage~~ <sup>Title</sup> of top brands.

## 8) File Download Popup

- We cannot inspect the popup.
- We can move the popup.
- This popup is a OS popup, so we need to handle it using ~~Robot~~ class.

Class FileDownload

{

    public void() throws AWTException

{

```
        System.setProperty("webdriver.gecko.driver", "./drivers/geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.Selenium.dev/downloads/");
        driver.findElement(By.xpath("//img[@alt='Java']/..//a)[1]")).click();
```

    Robot r = new Robot();

    r.keyPress(KeyEvent.VK\_TAB);

    r.keyPress(KeyEvent.VK\_TAB);

    r.keyRelease(KeyEvent.VK\_TAB);

    r.keyPress(KeyEvent.VK\_ENTER);

    r.keyRelease(KeyEvent.VK\_ENTER);

}

}

## File Upload Popup

- We cannot inspect the popup.
- It is a Part of the OS.
- In order to handle file upload popup we use ~~String~~ StringSelection class.

Shift + Right click  
Copy as path

```
Class FileUpload
```

```
{  
    public void ()  
{  
    _____
```

```
        driver.get("https://www.maulikri.com/");  
        driver.findElement(By.xpath("//a[.= 'Register']"))  
            .click();  
        driver.findElement(By.xpath("//button[.= 'Upload Resume']"))  
            .click();
```

```
StringSelection file = new StringSelection("C:\\Users\\gupta\\  
Desktop\\SaveResume.docx");
```

// internally copy the path of the file

```
Toolkit.getDefaultToolkit().getSystemClipboard().setContents  
(file, null);
```

```
Robot r = new Robot();
```

```
r.keyPress(KeyEvent.VK_CONTROL);
```

```
r.keyPress(KeyEvent.VK_V);
```

```
r.keyRelease(KeyEvent.VK_CONTROL);
```

```
r.keyRelease(KeyEvent.VK_V);
```

```
r.keyPress(KeyEvent.VK_TAB);
```

```
r.keyPress(KeyEvent.VK_TAB);
```

```
r.keyRelease(KeyEvent.VK_TAB);
```

```
r.keyPress(KeyEvent.VK_ENTER);
```

```
r.keyRelease(KeyEvent.VK_ENTER);
```

```
}
```

- We use StringSelection class in order to store the path of the file that we want to upload.
- We use getSystemClipboard() in order to internally copy the path of the file.
- We use setContents() in order to pass StringSelection class object.
- Finally we use Robot class in order to navigate inside the pop-up.

### Assignment - 16

1. Handle fileupload popup of freshersWorld.com
  2. Handle fileupload popup of foundit.in
- Q: Write a script to perform Enter operation using Actions class.

A:

```

class Sendkeys
{
    public void main()
    {
        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");
        Actions actions = new Actions(driver);
        actions.sendKeys("qspiders").perform();
        actions.sendKeys(Keys.TAB).sendKeys("abc").perform();
        actions.sendKeys(Keys.TAB).sendKeys(Keys.TAB).perform();
        actions.sendKeys(Keys.ENTER).perform();
    }
}

```

}

Q. Write a Script to perform click operation using Actions class method but without passing any argument.

1. class ClickWithoutArgument

```
{  
    PSVM() {  
        _____  
    }
```

```
        driver.get("https://en-gb.facebook.com/");  
        WebElement link = driver.findElement(By.xpath  
            ("//a[.= 'Forgotten Password?']"));
```

Actions actions = new Actions(driver);

actions.moveToElement(link).click().perform();

}

}

2. Write a Script to perform keyboard Operation using Actions class

Class KeyUpKeyDown

```
{  
    PSVM() {  
        _____  
    }
```

```
        driver.get("https://en-gb.facebook.com/");
```

WebElement link = driver.findElement(By.xpath

("//a[.= 'Forgotten Password?']"));

Actions actions = new Actions(driver);

actions.moveToElement(link).keyDown(Keys.CONTROL)

.click().keyUp(Keys.CONTROL).perform();

}

}

Q. ContextClick

A/ Class ContextClick  
 {  
 PSVM(-)  
 }

driver.get("https://swisnl.github.io/jQuery-contextMenu/demo.html");

WebElement button = driver.findElement(By.xpath("//span [.= 'right click me ']"));

Actions actions = new Actions(driver);

actions.moveToElement(button).contextClick().perform();

driver.findElement(By.xpath("//span [.= 'Quiet ']")).click();

driver.switchTo().alert().accept();

}

}

Q. Write a Script to Scroll to an element using Actions class.  
 At (Selenium Version 4)

Class ScrollToElement

{  
 PSVM(-)  
 }

driver.get("http://www.orgontech.com/");

WebElement contactUs = driver.findElement(By.xpath

("(//a [.= ' Contact Us '] )[2]"));

Actions actions = new Actions(driver);

actions.scrollToElement(contactUs).perform();

}

}

1) Write a Script to perform drag & drop by Sliders.

```
class DragAndDropBySlider
{
    public void main()
    {
        _____
        _____
```

```
        driver.get("https://omayo.blogspot.com/p/page3.html");
```

```
        WebElement minPriceRange = driver.findElement(By.xpath
            ("//a[@aria-labelledby='price-min-label']"));
```

```
        Actions actions = new Actions(driver);
```

```
        actions.dragAndDropBy(minPriceRange, 100, 0).perform();
```

```
}
```

}

```
}
```

2) Write a Script to perform drag & drop by Resizer.

```
class DragDropResizer
{
    public void main()
    {
        _____
        _____
```

```
        driver.get("https://jqueryui.com/resizable/");
```

```
        WebElement frame = driver.findElement(By.xpath("//div[@class='demo
            -frame']"));
```

```
        driver.switchTo().frame(frame);
```

```
        Actions a = new Actions(driver);
```

```
        WebElement resizer = driver.findElement(By.xpath("//div
            [contains(@class,'ui-icon-gripsmall-diagonal-se')]]));
```

```
        a.dragAndDropBy(resizer, 100, 100).perform();
```

```
}
```

}

```
}
```

Q Write a Script to perform Pause operation using Actions class. 87

A2- Class Pause

```
{  
    pause(  
        {  
            duration  
        }  
    )  
}
```

```
driver.get("https://en-gb.facebook.com/");  
driver.findElement(By.xpath("//a[@role='button'][2]")).click();
```

```
Actions a = new Actions(driver);
```

```
a.pause(Duration.ofSeconds(3));
```

- SendKeys("Area1").pause(Duration.ofSeconds(3))

- SendKeys(Keys.TAB).pause(Duration.ofSeconds(3))

- SendKeys("Chakra").pause(Duration.ofSeconds(3))

- SendKeys(Keys.TAB).pause(Duration.ofSeconds(3))

- SendKeys("9878685848").pause(Duration.ofSeconds(3))

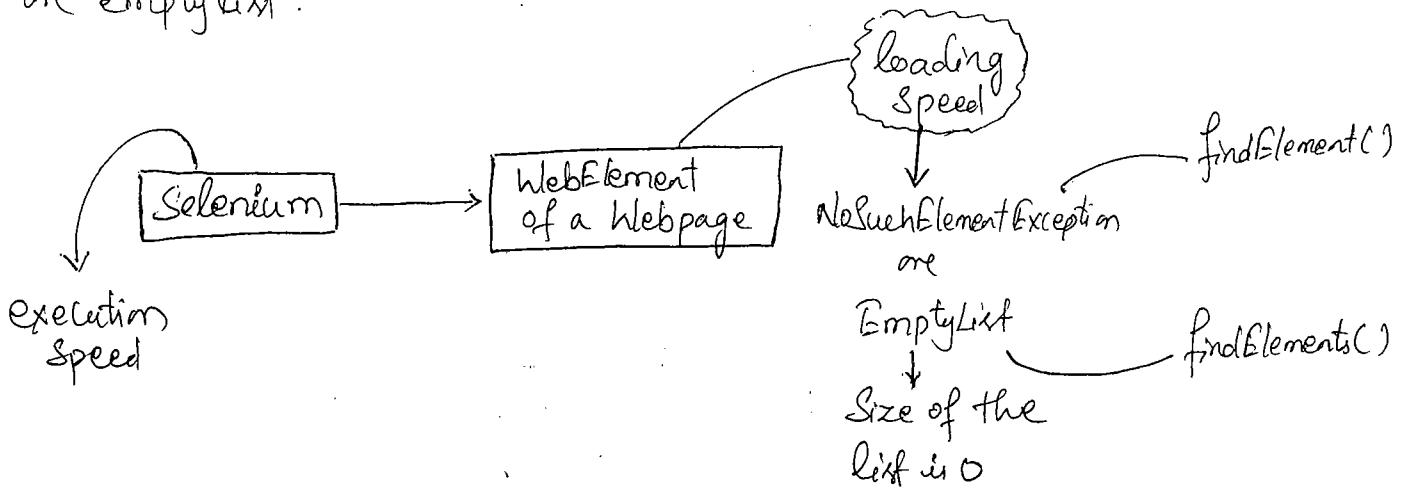
- SendKeys(Keys.TAB).pause(Duration.ofSeconds(3))

- SendKeys("abc").perform();

Dt - 22/02/2023

## Synchronization

- Synchronization is a process of matching the speed of Selenium with that of the application.
- During Runtime Selenium executes the scripts very fast and chances are there the WebElement might not be ~~not~~ loaded in the webPage & due to this we might get NoSuchElementException or empty list.



## Ways to achieve Synchronization:-

- 1) By using `Sleep()` of Thread class. — `Thread.Sleep()`
- 2) By using `implicitlyWait()` — `implicitlyWait()`
- 3) By using ~~explicitWait()~~ — ~~explicitWait()~~
- 4) By using ~~fluent~~ Fluent Wait

### 1) Sleep() of Thread class

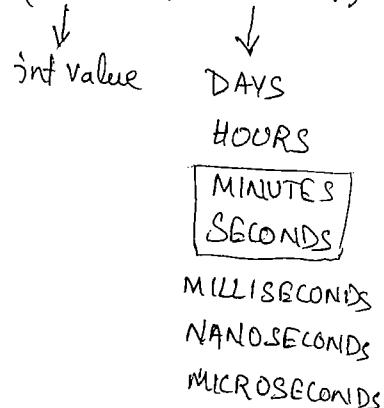
- `Sleep()` is a static method present inside Thread class which takes `millisecond` as an argument.
- Thread class should be imported from `java.lang` package.

#### Drawbacks:-

- (i) Lines of code will increase.
- (ii) It will wait for the complete duration so total execution time will be more.

### 2) implicitlyWait()

`driver.manage().timeouts().implicitlyWait(Duration, TimeUnit);`



- The above statement is written only once in the entire automation scripts.
- The above statement will be utilized for `findElement()` & `findElements()`.
- If the element is located before the duration, operation will be performed immediately & if the element is not located within the

duration, we get NoSuchElementException on empty list.

→ The above Syntax is generally written immediately after opening the browser.

P: Class ImplicitlyWait

{ PSVM( ) }

```
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.id("loginButton")).click();
driver.findElement(By.xpath("//div[@class='menu-icon'][2]")).click();
}
```

P: Class ImplicitlyWait Overloading

{ PSVM( ) }

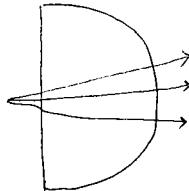
```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.manage().timeouts().implicitlyWait(10, TimeUnit.MINUTES);
driver.manage().timeouts().implicitlyWait(10, TimeUnit.MILLISECONDS);
driver.get("https://demo.actitime.com/");
driver.findElement(By.id("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.id("loginButton")).click();
driver.findElement(By.xpath("//div[@class='menu-icon'][2]")).click();
}
```

}

Note :-

→ During Runtime if the element is not located, it will keep on searching for the webElement every 500ms & this duration is called as Polling Period.

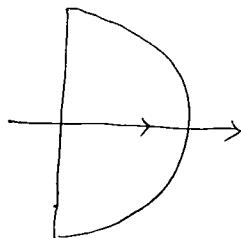
Method Overloading



TimeUnit.SECONDS

TimeUnit.MINUTES

Method Overriding



driver.get()

driver.navigate()

Drawback :-

(i) We can use this technique only for findElement() & findElements().  
→ and not for all the methods.

→

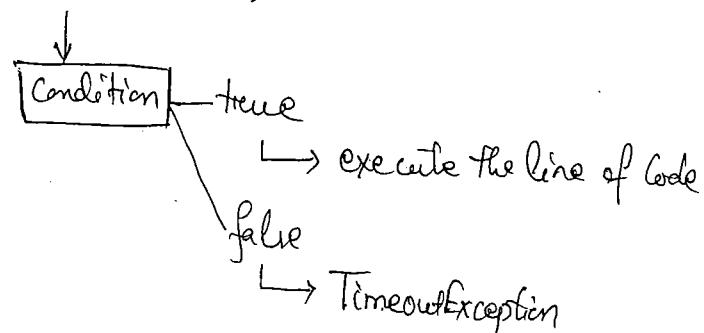
### 3) ExplicitWait

→ Whenever we can't use implicitlyWait, we go for explicitWait (other than findElement() & findElements()).

Syntax:- Selenium V3

```
WebDriverWait wait = new WebDriverWait(webDriver arg0, TimeoatinSeconds  
wait.until(ExpectedConditions.staticmethod()));
```

↓ arg1);  
int value



→ As we are specifying the waiting Cond^n explicitly, it is called as explicitWait.

→ If the Cond^n is true, operation will be performed, if the Cond^n is false we get TimeoutException.

## Class ExplicitWait

```
class ExplicitWait {
    public static void main(String[] args) {
        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.id("loginButton")).click();
    }
}
```

```
driver.get("https://demo.actitime.com/");
driver.findElement(By.id("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.id("loginButton")).click();

WebDriverWait wait = new WebDriverWait(driver, 10);
// wait.until(ExpectedConditions.titleIs("gridea"));
// wait.until(ExpectedConditions.titleIs("actiTIME - Enter Time - Track"));
// wait.until(ExpectedConditions.titleContains("gridea"));
// wait.until(ExpectedConditions.titleContains("Enter Time"));
// wait.until(ExpectedConditions.WebElementLocated(By.id("logoutLink")));
// wait.until(ExpectedConditions.WebElementLocated(By.xpath
// ("//div[@class='menu-icon'][2]")));
// wait.until(ExpectedConditions.alertIsPresent());
driver.findElement(By.xpath("//div[@class='menu-icon'][2]")).click();
}
}
```

### Note:-

→ If you are trying to switch to a window & the window is not present, we get NoSuchWindowException.

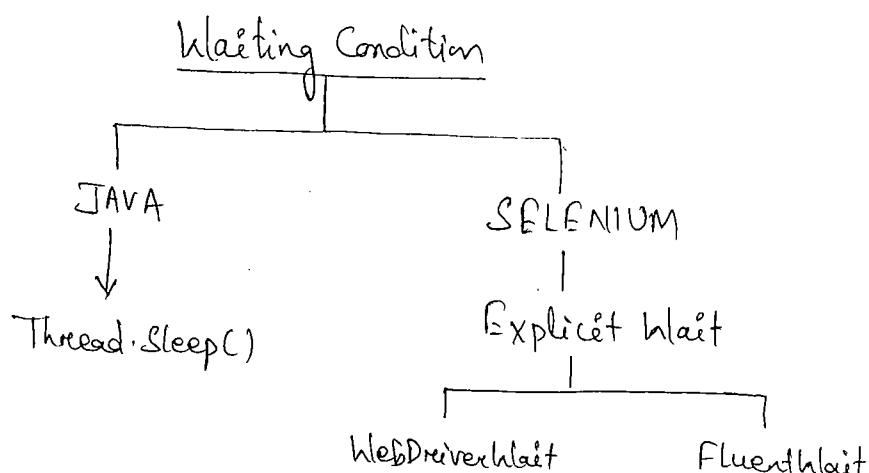
ImplicitWaitExplicitWait

- It can be used for findElement() & findElements().
- If the element is not located within the specified duration, we get NoSuchElementException or EmptyList.
- Duration can be in Days, Hours, Minutes, Seconds, Milliseconds, Nanoseconds & Microseconds.

- (i) It can be used for any methods including findElement() & findElements().
- (ii) If the element is not located within the specified duration, we get TimeoutException.
- (iii) Duration can be only in Seconds. (Version 3)

1) Fluent Wait

→ It is an advanced version of Explicit Wait which defines the maximum amount of time to wait for a condition to get satisfied as well as the frequency with which to check the condition.



Class FluentWaitDemo

```
{
  public void (-)
}
```

```

driver.get("https://demo.actitime.com/");
driver.findElement(By.id("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.id("loginButton")).click();
  
```

FluentWait<WebDriver> wait = new FluentWait<WebDriver>(driver)

- withTimeout(Duration.ofSeconds(10))
- pollingEvery(Duration.ofSeconds(2))
- ignoring(NoSuchElementException.class);

WebElement Settings = wait.until(new Function<WebDriver, WebElement>()

{ Public WebElement apply(WebDriver driver)

{ return driver.findElement(By.xpath("//div[@class='menu-item'][2]"));

}

} Settings.click();

\* } }

WebDriverWait

FluentWait

(i) Default polling period is 500ms.

(i) We can specify our own polling period.

(ii) We cannot ignore any exceptions.

(ii) We can ignore all exceptions.

Q. Write a Script to get all the Cookies of the Web Browser.

A. Class GetAllCookies

{ Service(-)

driver.get("https://www.google.com");

Set<Cookie> allCookies = driver.manage().getCookies();

for(Cookie cookie : allCookies)

{

System.out.println("Cookie name is :" + cookie.getName());

System.out.println("Cookie value is :" + cookie.getValue());

System.out.println("Cookie domain is :" + cookie.getDomain());

System.out.println("Cookie path is :" + cookie.getPath());

System.out.println("Cookie expiry is :" + cookie.getExpiry());

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

}

getCookies() → abstract method

→ Options Interf

→ Set<Cookie>

↓

class

↓ Selenium Pkg

Q. Write a Script to delete a particular cookie?

A) Class DeleteCookie

```

    {
        PSvm (—)
    }

    driver.get("https://www.google.com");
    Set<Cookie> allCookies = driver.manage().getCookies();
    for(Cookie cookie : allCookies)
    {
        System.out.println("Cookie name is :" + cookie.getName());
        System.out.println("-----");
    }
    driver.manage().deleteCookieNamed("OGPC");
    Set<Cookie> newCookies = driver.manage().getCookies();
    for(Cookie cookie1 : newCookies)
    {
        System.out.println("Cookie name is :" + cookie1.getName());
        System.out.println("-----");
    }
}

```

Q. Write a Script to get a particular cookie?

A) Class GetACookie

```

    {
        PSvm (—)
    }

    _____
    _____

```

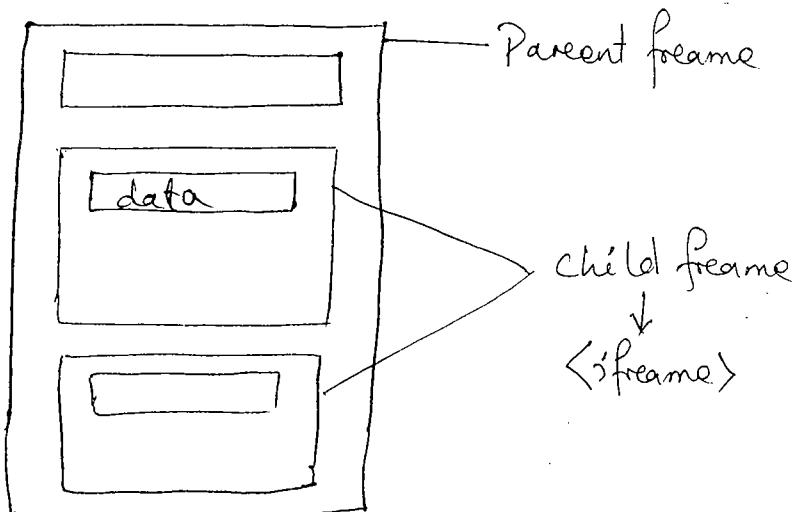
```

    driver.get("https://www.google.com");
    Cookie cookie = driver.manage().getCookieNamed("AEC");
    System.out.println(cookie);
}

```

## Handling Frames :-

- A Frame is a type of webElement which is developed using `<iframe>` tag.
- Generally the child frames are developed using `<frame>` tag.
- The default control is always present inside parent frame.



- In order to perform operation on the elements of a child frame, we need to switch the control inside child frame by using the following statement.

```
driver.switchTo().frame();
```

↓                              starts from 0

(i) frame index — int arg  
(ii) frame id — str arg  
(iii) frame name — str arg

- In order to Switch the Control to the parent frame, we can use the following Statement.

```
driver.switchTo().parentFrame();
```

```
driver.navigate().refresh(); // don't use
```

- We can also Switch to the child frame by taking childframe element reference.

Note :- If you are trying to perform operation on an element present inside child frame without switching to the child frame, we get NoSuchElementException.

> If frame id or frame name or frame index is incorrect, we get NoSuchElementException.

class HandlingFrames

```
{  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
driver.get("file:///C:/Users/gsp/Desktop/Automation/Automation/  
htmlpages/frames.html");  
driver.findElement(By.id("t1")).sendKeys("areul");  
// driver.switchTo().frame(0); // index of the frame  
driver.switchTo().frame("f1"); // id of the frame  
// driver.switchTo().frame("n1"); // name of the frame  
// driver.switchTo().frame("f3"); // NoSuchElementException  
driver.findElement(By.id("f2")).sendKeys("abani");  
driver.switchTo().parentFrame();  
// driver.navigate().refresh();  
WebElement childFrame2 = driver.findElement(By.xpath("//  
iframe[@id='f2']"));  
// driver.switchTo().frame("f2");  
driver.switchTo().frame(childFrame2);  
driver.findElement(By.id("t2")).sendKeys("haba");
```

```
{  
}
```

## Handling Chatbot :-

→ Chatbot is a AI Software which allows the user to communicate with the application through messages.

### Assignment - 17

1. Open Goindigo.in & handle both frame & chatbot.
2. Open AirAsia.co.in & handle the both frame & chatbot.

## Handling Tooltips :-

- A Tooltip is a text that appears whenever we mouse-hover on an element.
- Tooltips are traditionally developed using title attribute to the element.

title = "tooltip text"

```
class HandlingTooltip
```

```
{
```

```
    public void m1()
```

```

        driver.get("https://en-gb.facebook.com/");
        driver.findElement(By.xpath("//a[@role='button'][2]")).click();
        WebElement dateOfBirthTooltip = driver.findElement(By.id("birthday-help"));
        Actions a = new Actions(driver);
        a.moveToElement(dateOfBirthTooltip).perform();
        String tooltipText = dateOfBirthTooltip.getAttribute("title");
        if (tooltipText.equals("click for more information"))
        {
            System.out.println("tooltip text is matching");
        }
        else
        {
            System.out.println("tooltip text is not matching");
        }
    }
}
```

## PageLoad Timeout in Selenium :-

- PageLoad Timeout generally focuses on the time taken by the webpage to get loaded.
- If the page loads within the duration specified, execution continues otherwise we get Timeout Exception.
- Default page load timeout is 30 secs.

Class PageLoadTimeout

```
{ P S V m ( ) }
```

```
driver.manage().timeouts().pageLoadTimeout(10, TimeUnit.SECONDS),
```

```
driver.get("https://demo.actitime.com/"),
```

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

```
}
```

```
}
```

## Java Script Execution in Selenium :-

- Write a script to perform click operation using Java Script.
- Whenever the conventional click method of Selenium is unable to perform operation on an element, we might get exception such as ElementInterceptedException.

JavaScriptExecutor  
↳ Interface  
↳ selenium

```
js.executeScript(String arg[0] , Object arg1);
```

↓  
operation

↓  
webElement

("Argument[0].click()")

```

class Click
{
    public void click()
    {
        driver.get("https://login.yahoo.com/?");
        WebElement checkbox = driver.findElement(By.id("persistent"));
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript("arguments[0].click()", checkbox);
    }
}

```

Q Write a Script to perform Enter operation using JavaScript Executor.

```
(arguments[0].value = " ")
```

class Enter

```
{
    public void enter()
    {
}
```

```
driver.get("https://login.yahoo.com/?");
```

```
WebElement username = driver.findElement(By.id("login-username"));
```

```
JavascriptExecutor js = (JavascriptExecutor) driver;
```

```
js.executeScript("arguments[0].value = 'Sareej'", username);
```

```
}
```

```
}
```

#### \* Advantages of JavaScript

- We can perform operation on a disabled element.
- We can perform page scrolling.
- We can use JavaScript when Selenium pre-defined method don't work.

Q) Write a Script to perform Continuous Scrolling of a webpage using Java Script.

Class ScrollContinuous

{  
    public void main() {  
        \_\_\_\_\_

```
driver.get("https://www.myntra.com/");
JavascriptExecutor js = (JavascriptExecutor) driver;
for(int i=0; i<=9; i++)
{
    js.executeScript("window.scrollBy(0,100)");
}
for(int i=0; i<=9; i++)
{
    js.executeScript("window.scrollBy(0,-100)");
}
```

Write a Script to Scroll to a particular Element using Java Script.

Class ScrollToElement

{  
    public void main() {  
        \_\_\_\_\_

```
driver.get("https://www.wikipedia.org/");
WebElement wikiSource = driver.findElement(By.xpath
        ("//span[.= 'wikisource']"));
int x = wikiSource.getLocation().getX();
int y = wikiSource.getLocation().getY();
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("window.scrollBy("+x+", "+y+")");
}
```

Q: Write a Script to create Alert Popup using Java Script.

A<sup>2</sup> class Alert

{

driver.get("https://www.google.com/");

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("alert('Saroj is a bad guy!')");

driver.switchTo().alert().accept();

}

}

Q: Write a Script to create a prompt popup using Java Script.

A<sup>2</sup> class Prompt

{

PSVM(—)

{

\_\_\_\_\_

driver.get("https://www.google.com/");

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("prompt('What is your girlfriend name')");

}

}

Q) Write a Script to Create Confirmation Popup using Java Script?

## Class Confirmation

۳

P ⊃ v m (—)

۷

—  
—

```
driver.get("https://www.google.com/");
```

JavascriptExecutor js = (JavascriptExecutor) driver;

```
JS.executeScript("Confirm('Savej, Are you sure you don't have  
a girlfriend ? ')");
```

driver. SwitchTo(). alert(). accept();

3

3

Q. Write a Script to get the title of a webpage using JavaScript.

class GetTitle

۷

P S v m ( — )

۷

—

```
driver.get("https://www.google.com/");
```

JavaScript Executor js = (JavaScriptExecutor) driver;

```
SopIn(jse.executeScript("return document.title"));
```

1

3

Q. Write a Script to get the Current URL of a webpage using JavaScript.

Ans

Class GetURL

{

PSVM(—)

{

driver.get("https://www.google.com/");

JavascriptExecutor js = (JavascriptExecutor) driver;

String js.executeScript("return document.URL"));

}

}

Q. Write a Script to Enter a date into a calendar popup using Java Script.

Ans

Class CalendarPopup

{

PSVM(—)

{

=====

=====

driver.get("https://www.americanairlines.in/");

// JavascriptExecutor js = (JavascriptExecutor) driver;

// js.executeScript("document.getElementById('aa-leavingOn').value = '14/03/2023'");

WebElement departure = driver.findElement(By.id("aa-leavingOn"));

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].value='14/03/2023'", departure);

}

}

Q: Write a Script to perform operation on a disabled element using Java Script.

Ans Class DisabledElement

```
{  
    public void __()  
}
```

---

```
driver.get("https://omayo.blogspot.com/");  
WebElement textField = driver.findElement(By.id("t62"));  
System.out.println(textField.isDisabled());
```

```
JavaScriptExecutor js = (JavaScriptExecutor) driver;  
js.executeScript("arguments[0].value='areud'", textField);
```

}

}

Q: Write a Script to refresh a webpage using Java Script.

Ans Class Refresh

```
{  
    public void __()  
}
```

---

```
driver.get("https://www.google.com/");
```

```
JavaScriptExecutor js = (JavaScriptExecutor) driver;  
js.executeScript("history.go(0)");
```

}

}

DT - 28/02/2023 (05)

Q Write a Script to Scroll to a particular element using ScrollIntoView of Javascript?

A) Class ScrollIntoView

```
{  
    public void __()  
{  
        _____  
        _____  
        driver.get("https://www.wikipedia.org/");  
        WebElement wikiSource = driver.findElement(By.xpath  
            ("//span[.= 'wikiSource']"));  
        JavascriptExecutor js = (JavascriptExecutor) driver;  
        js.executeScript("arguments[0].scrollIntoView(true)", wikiSource);  
    }  
}
```

Q. Write a Script to Scroll to the end of the page using Java Script?

A) Class ScrollPageEnd

```
{  
    public void __()  
{  
        _____  
        _____  
        driver.get("https://www.myntea.com/");  
        JavascriptExecutor js = (JavascriptExecutor) driver;  
        js.executeScript("window.scrollBy(0, document.body.scrollHeight)");  
    }  
}
```

## Removing Chrome Warming Messages :-

→ Through this we have to remove the warning messages that get displayed in the console.

Class RemoveChromeWarning

{

PSVMC ->

{

System. SetProperty ("webdriver.chrome.SilentOutput", "true");

System. SetProperty ("webdriver.chrome.driver", ". /drivers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get ("https://www.myntra.com/");

}

}

## Handling Insecure Websites :-

Class InsecureWebsite

{

PSVMC ->

{

ChromeOptions options = new ChromeOptions();

options.setAcceptInsecureCerts(true);

System. SetProperty ("webdriver.chrome.driver", ". /drivers/chromedriver.exe");

WebDriver driver = new ChromeDriver(options);

driver.get ("https://entaxy.io/");

}

}

Q. Write a Script to open the Browser in Incognito mode?

A= Class IncognitoWindow

{  
PSVM(—)

{

ChromeOptions options = new ChromeOptions();

options.addArguments("incognito");

System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

WebDriver driver = new ChromeDriver(options);

driver.get("https://www.google.com");

}

}

Q. Write a Script to open the Browser in Maximized mode?

A= Class StartMaximized

{

PSVM(—)

{

ChromeOptions options = new ChromeOptions();

options.addArguments("start-maximized");

System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

WebDriver driver = new ChromeDriver(options);

driver.get("https://www.google.com");

}

{

Q. Write a Script to open the Browser in Full Screen.

A:-

Class FullScreen

{

PSVM(—)

{

```
driver.get("https://www.google.com");
driver.manage().window().fullscreen();
```

}

}

## Headless Browser Testing

→ Headless Browser is a Web browser without User Interface i.e. the program will execute in the background.

Advantage:-

- (i) Faster execution of Automation Script.
- (ii) Execution time will be less.
- (iii) Multi Tasking.

Class HeadlessBrowserTesting

{

PSVM(—)

{

```
ChromeOptions options = new ChromeOptions();
options.addArguments("headless");
```

```
System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
WebDriver driver = new ChromeDriver(options);
driver.get("https://demo.actitime.com");
```

```
Sopln("Login page title is :" + driver.getTitle());
driver.findElement(By.id("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.id("loginButton")).click();
Sopln("homepage title is :" + driver.getTitle());
```

```
}
```

## Capturing Selenium Log file :-

- Log file is just a simple file which keeps a track of all the execution history as well as default settings of the Browser. And the process is called as Logging.
- If we create a log file, we can debug our script easily.

```
class Logfile
```

```
{
```

```
PSVMC ->
```

```
{
System.setProperty("webdriver.chrome.logfile","./chromelogs/chromelogfile.txt");
System.setProperty("webdriver.chrome.driver","./drivers/Chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("https://demo.actitime.com");
Sopln("Login page title is :" + driver.getTitle());
driver.findElement(By.id("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.id("loginButton")).click();
Sopln("home page title is :" + driver.getTitle());
```

```
}
```

```
{
```

# Different ways of Entering Character Sequence

→ Class CharacterSequence

{  
    PS VM (—)  
    {

- // 1. by using Sendkeys() of WebElement Interface.
  - // 2. by using Sendkeys() of Actions class
  - // 3. by using JavascriptExecutor Interface
  - // 4. by using StringBuffer class - Sendkeys()
- 

```
driver.get("https://demo.actitime.com");
WebElement username = driver.findElement(By.id("username"));
StringBuffer buffer = new StringBuffer()
    .append("sa")
    .append("r")
    .append("o");
username.sendKeys(buffer);
StringBuilder builder = new StringBuilder()
    .append("sa")
    .append(" ")
    .append("o");
username.sendKeys(builder);
} } Stringbuffer class
```

## StringBuilder class

- i) StringBuffer is synchronized i.e. thread safe. It means two threads can't call the methods of StringBuffer class simultaneously.
- ii) StringBuffer is less efficient than StringBuilder.
- iii) StringBuffer was introduced in Java 1.0.

- (i) StringBuilder is Non-Synchronized i.e. thread not safe. It means two threads can call the methods of StringBuilder class simultaneously.
- (ii) StringBuilder is more efficient than StringBuffer.
- (iii) StringBuilder was introduced in Java 1.5.

## Class Performance Test

{

PSVM (-)

{

```
long startTime = System.currentTimeMillis();
```

```
StringBuffer buffer = new StringBuffer("Samprati");
```

```
for(int i=0; i<1000000; i++)
```

{

```
buffer.append("Sahu");
```

}

```
System.out.println("Time taken by StringBuffer is :" + (System.currentTimeMillis() - startTime + " ms"));
```

```
startTime = System.currentTimeMillis();
```

```
• startTime + " ms");
```

```
StringBuffer builder = new StringBuffer("Samprati");
```

```
for(int i=0; i<1000000; i++)
```

{

```
builder.append("Sahu");
```

}

```
System.out.println("Time taken by StringBuilder is :" + (System.currentTimeMillis() - startTime + " ms"));
```

}

{

Dt - 01/03/2023

## Different ways of Refreshing a Webpage

### Class RefreshWebPage

{

PSVM (-)

{

// 1. by using refresh() of Navigation Interface

// 2. by using JavaScriptExecutor Interface.

```

driver.get("https://www.google.com");
// driver.get(driver.getCurrentUrl());
// driver.navigate().to(driver.getCurrentUrl());
Robot r = new Robot();
r.keyPress(KeyEvent.VK_F5);
}
}

```

## Verifying Error Message in Selenium

→ Whenever we are performing any operation against the clients requirement, we get an error message.

Class VerifyErrorMessage

```
{
    PSVM()
}
```

---

```

driver.get("https://www.gmail.com");
WebElement nextButton = driver.findElement(By.xpath
    ("//div[@class='VfPpkd-RLmnJb'][2]"));

```

```

JavaScriptExecutor js = (JavaScriptExecutor) driver;
js.executeScript("arguments[0].click()", nextButton);
WebElement errorMessage = driver.findElement(By.xpath
    ("//div[@class='o6cumc TjGlae']"));

```

```

String errorMessageText = errorMessage.getText();
if(errorMessageText.equals("Enter an email or phone number"))
{
    System.out.println("error message text is matching");
}
else {
    System.out.println("error message text is not matching");
}
}

```

```

}

```

## Selenium Version 4 Features :-

113  
(4-7-2)

Q. Write a Script to minimize a Web Browser

## 12 Class MinimizeBrowser

{ P Sym (-) }

5

```
driver.get("https://www.google.com");  
driver.manage().window().minimize();
```

3

۷

Q: Write a Script to achieve Synchronization

## Ar Class Synchronization

{

P S Vm (—)

۲۷

// simplicet. Waef

```
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

## // Webdevverliefd

```
WebDreieckList wait = new WebDreieckList (dreiecke,
```

Duration. of seconds (10);

3

2

Q. Write a Script to Switch to a new Tab & new window.

Ans Class SwitchTabWindow.

```

{ P S V M ( — )
  {
    _____
    driver.get("https://www.google.com");
    // Switch to new tab
    // driver.switchTo().newWindow(WindowType.TAB);
    // driver.get("https://www.myntea.com");
    // Switch to new window
    driver.switchTo().newWindow(WindowType.WINDOW);
    driver.get("https://www.flipkart.com");
  }
}
  
```

## Special Locators in Selenium

Class Locators

```

{ P S V M ( — )
  {
    _____
    driver.get("https://en-gb.facebook.com/");
    // WebElement username = driver.findElement(new ByAll
    (By.id("email"), By.name("email"), By.xpath("//input[@data-testid"
      "= 'royal_email']")));
  }
}
  
```

WebElement username = driver.findElement(new ByIdOrName("email"));  
username.sendKeys("Sareoj");

```

  {
  }
}
  
```

## Handling Bootstrap Dropdown

Class BootstrapDropdown

{  
    PSVM(—)

{

```

driver.get("file:///C:/Users/gsp/Desktop/Automation/Automation/
html/pages/BootstrapDropDown.html");
driver.findElement(By.xpath("//button[@type='button']")).click();
List<WebElement> options = driver.findElements(By.xpath("//ul
[@class='dropdown-menu']//a"));
for(int i=0; i<options.size(); i++)
{
    if(options.get(i).getText().equals("Myntrea"))
    {
        options.get(i).click();
        break;
    }
}
if(driver.getTitle().contains("Online Shopping"))
{
    System.out.println("navigated to myntrea page");
}
else
{
    System.out.println("not navigated to myntrea page");
}
}

```

## Handling Web Table

- A Web Table in Selenium contains data in the form of rows & columns.
- A Web Table can be of two types :-
  - (i) Static Web Table
  - (ii) Dynamic Web Table

Q Write a Script to fetch all the headers of the Web Table.

A (For Static)

Class StaticWebTableFetchHeaders

```
{
  PSVM()
  {
    driver.get("https://omayo.blogspot.com/");
    List<WebElement> headers = driver.findElements(By.xpath
      ("//table[@id='table1']//thead//th"));
    for(int i=0; i<headers.size(); i++)
    {
      System.out.println("headers of the webtable is :" + headers.get(i).getText());
    }
  }
}
```

Q Write a Script to fetch all the data present inside the Web Table.

A (For Static)

Class StaticWebTableFetchData

```
{
  PSVM()
  {
    driver.get("https://omayo.blogspot.com/");
  }
}
```

117

```

List<WebElement> datas = driver.findElement(By.xpath
    ("//table[@id='table1']//tbody//tr"));
    for(int i=0 ; i<datas.size() ; i++)
    {
        System.out.println(datas.get(i).getText());
    }
}

```

Q: Write a Script to fetch first row data of the WebTable.

A: (For Static)

Class StaticWebTableFetchFirstRowData

```

{
    public void main()
    {
        _____
    }
}

```

```
driver.get("https://omayo.blogspot.com/");
```

```

List<WebElement> firstRowData = driver.findElement(By.xpath
    ("//table[@id='table1']//tbody//tr[1]/td"));

```

```
for(int i=0 ; i<firstRowData.size() ; i++)
{
    System.out.println(firstRowData.get(i).getText());
}
```

```

    System.out.println();
}
}

```

Q: Write a Script to fetch first Column data of the WebTable.

A: (For Static)

Class StaticWebTableFetchFirstColumnData

```

{
    public void main()
    {
        _____
    }
}

```

```
driver.get("https://omayo.blogspot.com/");
```

```

List<WebElement> firstColumnData = driver.findElements(By.xpath
    ("//table[@id='table1']//tbody//tr//td[1]"));

for(int i=0; i<firstColumnData.size(); i++)
{
    System.out.println(firstColumnData.get(i).getText());
}
}
}

```

Q. Write a Script to fetch the 2nd column data in 3rd row.

1) (For static)

```

class StaticWebTableSecondColumnThirdRowData
{
    public static void main()
    {
        driver.get("https://omayo.blogspot.com/");
        WebElement data = driver.findElement(By.xpath
            ("//table[@id='table1']//tbody//tr[3]//td[2]"));

        System.out.println("age of preaveen is :" + data.getText());
    }
}

```

Q. Write a Script to fetch all the column data of 3rd row.

2) (For static)

```

class StaticWebTableAllColumnDataThirdRow
{
    public static void main()
    {

```

```

        driver.get("https://omayo.blogspot.com/");
    }
}

```

```

List<WebElement> thirdRowData = driver.findElements(By.xpath
    ("//table[@id='table1']//tbody//tr[1]//td"));
for(int i=0; i < thirdRowData.size(); i++)
{
    System.out.println(thirdRowData.get(i).getText());
}
}
}
}

```

Q. Write a Script to fetch all the column data of 3<sup>rd</sup> column.

A) - (For static)

```

class StaticWebTableAllColumnDataThirdColumn
{
    public void __
    {
        driver.get("https://omayo.blogspot.com/");
        List<WebElement> CityData = driver.findElements(By.xpath("//table
            [@id='table1']//tbody//tr//td[1]"));
        for(WebElement city : CityData)
        {
            System.out.println(city.getText());
        }
    }
}

```

Q. Inwrite a Script to fetch all the data of the WebTable along with the Header.

A) - (For static)

```

class StaticWebTableAllDataWithHeaders
{
    public void __
    {
        driver.get("https://omayo.blogspot.com/");
        List<WebElement> header = driver.findElements(By.xpath("//table
            [@id='table1']//thead//tr//th"));
        List<WebElement> data = driver.findElements(By.xpath("//table
            [@id='table1']//tbody//tr"));
        for(WebElement headerCell : header)
        {
            System.out.print(headerCell.getText() + " ");
        }
        System.out.println();
        for(WebElement row : data)
        {
            for(WebElement cell : row.findElements(By.tagName("td")))
            {
                System.out.print(cell.getText() + " ");
            }
            System.out.println();
        }
    }
}

```

```
driver.get("https://omayo.blogspot.com/");
```

```
WebElement allDataWithHeaders = driver.findElement(By.xpath
    ("//Table[@id='table1']"));
```

```
Sopln(allDataWithHeaders.getText());
```

```
}
```

```
}
```

## Assignment -18

1. Write a Script to Count total no. of numeric values present in the web Table . (omayo.blogspot.com)
2. Write a Script to print the sum of all the numeric values present in the Web Table .

3.

## Dynamic Web Table

URL : <https://demo.opencart.com/admin/>

UN : demo

PWD : demo

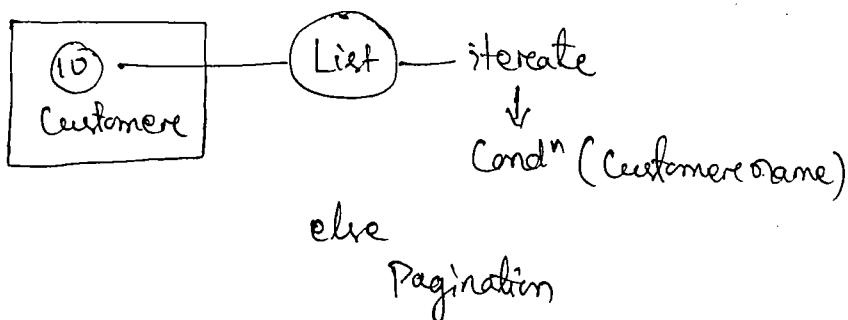
→ handle popup

→ click on Sales

→ click on Orders

## Create an order

demo.opencart.com

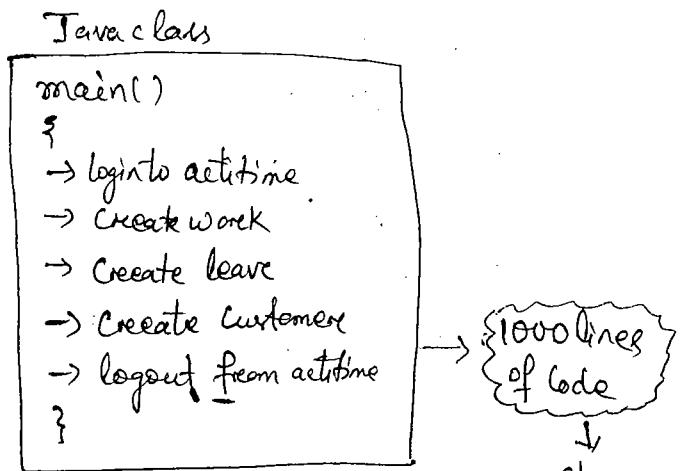


Dt - 03/03/2023

## Page Object Model (POM) :

Let's assume we need to execute the below scenarios —

- (i) login to actitime
- (ii) Create a type of work
- (iii) Create a leave type
- (iv) Create a customer
- (v) logout from actitime



→ For every webpage present in the application, we will create a separate class

→ POM class

Code maintenance will → Time Consuming  
be difficult

↓  
changes in the Script

↓  
difficult to do

we have many lines of codes.

→ Page Object Model is a Java Design Pattern in Selenium that creates element repository or object repository.

→ In Page Object Model Consider every webpage of the application as a Java Class file.

## Element Repository

LoginPage.java — POM CLASS

→ Store the WebElements

- Username Textfield
- Password Textfield
- Keep me logged in Checkbox
- Login Button
- Forgot password link
- Activation Link

Pattern in Selenium

### \* Advantages of POM :-

- Easy maintenance of Automation Scripts.
- Readability & Reliability of Automatic Scripts.
- Helps in reducing the lines of Code.

### \* Creating a POM class :-

→ Every POM class has three stages :-

- 1) Declaration
- 2) Initialization
- 3) Utilization

1) Declaration - We declare the web elements of the POM class by using @FindBy which means how you want to locate the element.

Syntax -

```
@FindBy(locatorname = "locatorValue")
```

```
private WebElement / List<WebElement> webelementname;
```

2) Initialization - We initialize the elements of POM class using initElements() of PageFactory class.

Syntax:-

public Constructor (WebDriver driver)  
{

Call to this

Pagefactory.initElements (WebDriver reference, this)

↓                    ↓  
Class                Static method

3) Utilization - In order to perform the utilization of the webElements, we create non-static methods & inside these non-static methods we have to perform the operation.

Note: (i) If the webElements are not initialized, we get **NULLPOINTEREXCEPTION**

(ii) In Page object model we create two types of classes  
 ↪ POM class  
 ↪ Test class

#### ↪ POM Class

- Here we store the elements & the methods.
- This class is called as Element Repository / Object Repository.
- We cannot execute this class as it has no main method.

#### 2) Test Class

- This class is used for execution purpose as it contains main method.
- Execution happens by calling the methods of POM class.

Example :-

// POM Class - LoginPage

class ActitimeLoginPage

{

// Declaration

@FindBy(id = "username")

private WebElement usernameTextField;

@FindBy(name = "pwd")

private WebElement passwordTextField;

@FindBy(id = "keepLoggedIncheckbox")

private WebElement keeploggedincheckbox;

@FindBy(id = "loginbutton")

private WebElement loginButton;

@FindBy(xpath = "//a [.= 'Forgot your password ? ']")

private WebElement forgotyourpasswordLink;

@FindBy(xpath = "//a [.= 'actiTIME Inc. ']")

private WebElement actitimeinlink;

// Initialization

public ActitimeLoginPage(WebDriver driver)

{

PageFactory.initElements(driver, this);

}

// Utilization

public void loginMethod() throws InterruptedException

{

usernameTextField.sendKeys("admin");

passwordTextField.sendKeys("manager");

keeploggedincheckbox.click();

loginButton.click();

Thread.sleep(3000);

public void forgotpasswordMethod()

{

forgotyourpasswordLink.click();

}

```

public void actitimeincMethod()
{
    actitimeInLink.click();
}

}

// POM Class - HomePage
class ActitimeHomePage
{
    @FindBy(id = "logoutLink")
    private WebElement logoutLink;

    public ActitimeHomePage(WebDriver driver)
    {
        PageFactory.initElements(driver, this);
    }

    public void logoutMethod()
    {
        logoutLink.click();
    }
}

// Test Class
class ActitimeTest
{
    @Test
    public void testActitime()
    {
        WebDriver driver = new FirefoxDriver();
        driver.get("https://demo.actitime.com");
        ActitimeLoginPage loginPage = new ActitimeLoginPage(driver);
        loginPage.loginMethod();

        ActitimeHomePage homepage = new ActitimeHomePage(driver);
        homepage.logoutMethod();
    }
}

```

## Assignment - 19

- > Write a Script to login to actitime, Create a type of work, Create a leave type, Create a Customer & logout from actitime using POM class concept.  
 www.(demo.actitime.com)

Dt- 04/03/2023

Q. What is the

Q. What is the difference between @FindBy , @FindBys ,  
 @FindAll .

Ans :- @FindBy :-

When the required WebElement object needs to match one single ~~one~~ Criteria , we use @FindBy .

@FindBys :- AND

When the required WebElement object needs to match all the given Criteria , we use @FindBys .

@FindAll :- OR

When the required WebElement needs to match atleast one of the given Criteria , we uses @FindAll .

Q. What is Pagefactory class in Selenium ?

A:- Pagefactory is a class provided by Selenium webdriver to support Page object design patterns .

Pagefactory class is generally used for initializing the webElements of the DOM class using initElements() .

### Example :-

```

class FindByFindBysfindAll
{
    @FindBy(id = "name")
    private WebElement usernameTextField;

    @FindBy
    {
        @FindBy(id = "email"),
        @FindBy(name = "email")
    }
    private WebElement emailTextField;

    @FindAll
    {
        @FindBy(id = "mobile"),
        @FindBy(name = "mobile")
    }
    private WebElement mobileTextField;
}

```

## TestNG (Next Generation)

→ In real world working in a project, we will have multiple test cases.

Test Case — Start from opening the browser & entering the URL.

end with closing the browser.

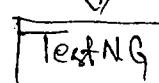
- 1. Login to actitime & logout.
- 2. Login to actitime, create a customer & logout
- 3. Login to actitime, delete the customer & logout.
- 4. Login to actitime, create type of work & logout
- 5. login to actitime, delete type of work & logout
- 6. login to actitime, create leave type & logout.

→ Individual Test result →  
for every test case

PASS

FAIL

Can we execute multiple java files having main() → No



↓  
instead of using main()  
we use test method

will be used for  
execution purpose.

← @Test

→ automatically the results will be displayed.

- TestNG is a unit testing framework which is used to execute multiple test cases.
  - Once all the TestCases are executed, TestNG will automatically generate a report called as TestNG report or HTML report which keeps a track of which TestCases got passed & failed.
  - Also in this report we can see the total execution time taken by the TestCases.
  - For every TestCase present, we should create a separate Java class and that class is called as Test Class.
- webpage → POM class
- Declaration → @FindBy  
Initialization → initElements()  
Utilization → non-static method
- Test Case → Test class
- ↳ Test Method
- ↳ @Test

#### What is a Test Class?

- It is a Java class which has ~~main()~~ instead of ~~Test~~ Test method instead of main().
- Test method is generally used for execution purpose.
- A Test method is a method which has a Test annotation (@Test).

#### Steps to download TestNG :-

1. Go to help & click on Eclipse market place.
2. Type TestNG in the search bar & hit enter.
3. Click on Install button of TestNG for Eclipse.
4. Uncheck the optional checkbox and click on Confirm.

## Adding TestNG to Java Project :-

1. Right click on the Java Project, Go to build path, Click on Configure Build Path, Under Libraries tab select class path, click on Add Library and select TestNG.
2. Click on Next & Finish.
- Q. Can we execute multiple TestCases or Test classes together?

A. Yes, by using TestNG Suite.

TestNG Suite is an XML file that contains the list of all the Test Cases or Test classes.

## How to create a TestNG Suite or TestNG.xml file ?

1. Right Click on the Java project, go to TestNG & select Convert to TestNG (Automatically TestNG.XML file will be created inside the Java project).

## How to execute TestNG Suite or TestNG.XML file ?

Right click on the TestNG.XML file and select Run as TestNG Suite (Automatically all the test cases will get executed one after another).

## How to generate TestNG report or HTML report ?

- (i) Refresh the Java project, inside the Project Test - output folder will be generated.
- (ii) Expand the Test-output folder and we can see a file available-report.html and index.html.
- (iii) To open the files, right click on it & select open with web browser.

## Assignment - 20

- Execute all the 6 Test Cases.

Dt - 07/03/2023

### Types of Execution :-

- Batch Execution
- Parallel Execution
- Group Execution

#### 1. Batch Execution

→ Execution of the Test Cases one after another in the order present inside TestNG.XML file is called as Batch Execution.

#### 2. Parallel Execution

→ Execution of the Test Cases or Test Classes parallelly at the same time is called as Parallel Execution.

→ The drawback of Parallel Execution is we cannot execute dependent test cases.

→ For dependent Test Case Execution we have to go for batch execution.

#### 3. Group Execution

→ Execution of the TestCases or Test Classes as a group is called as Group Execution.

→ Group Execution can be done in Batch or Parallel Mode.

→ In order to perform group execution, we use groups attribute of TestNG.

//testing.xml

```

<Suite name = "Suite" parallel = "classes">
  <test thread-count = "5" name = "Test">
    <groups>
      <run>
        <!--<include name = "Smoke"></include>-->
        <exclude name = "Smoke"></exclude>
      </run>
    </groups>
    <classes>
      <class name = "test.ActitimeLoginLogout"/>
      <class name = "test.ActitimeLoginCreateCustomerLogout"/>
    </classes>
  </test> <!--Test-->
</Suite> <!--Suite-->

```

//POM class - ActitimeLoginLogout

Class ActitimeLoginLogout

{ @Test(groups = "Smoke")

public void loginLogout() throws InterruptedException

{

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

ActitimeLoginPage LoginPage = new ActitimeLoginPage(driver);  
 LoginPage.loginMethod();

ActitimeHomePage HomePage = new ActitimeHomePage(driver);  
 HomePage.logoutMethod();

}

}

Q. Can we create multiple TestNG.XML files for multiple groups?

A: Yes.

Q. Can a Single Test Case or Test Class belong to multiple groups?

A: Yes, by specifying multiple group names.

### Advantages of TestNG :-

- (i) We can execute multiple Test Cases together.
- (ii) We can execute ~~one~~ Test Cases in Parallel and in Groups.
- (iii) We can generate detailed HTML report.

Dt- 09/03/2023

### \* How to Skip the execution of a Test class and Test Method?

→ In order to deliberately skip the execution of a Test class, we can comment the test class in our TestNG.XML file.

→ In order to deliberately skip the execution of a Test method, we use TestNG attribute as enabled = false.

// testng.xml

```
<?xml Version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE Suite SYSTEM "https://testng.org/testng-1.0.dtd">
<Suite name = "Suite">
    <test threadCount = "5" name = "Test">
        <classes>
            <!-- <class name = "extra.SkipClassMethod"/> -->
            </classes>
        </test> <!-- Test -->
    </Suite> <!-- Suite -->
```

## // Test Class

```
Class SkipClassMethod
{
    @Test
    Public void a()
    {
        System.out.println("Sampreeti");
    }

    @Test
    Public void b()
    {
        System.out.println("Preash");
    }

    @Test(enabled = false)
    Public void c()
    {
        System.out.println("Debasish");
    }
}
```

skip Sampreeti  
Preash

How to skip the execution of a Test class & Test Method without commenting the class & without using enabled = false?

Using @Ignore at class level & method level.

```
//@Ignore
Class IgnoreClassMethod
{
    @Test
    Public void a()
    {
        System.out.println("Sampreeti");
    }

    @Ignore
    @Test
    Public void b()
    {
        System.out.println("Preash");
    }

    @Test
    Public void c()
    {
        System.out.println("Debasish");
    }
}
```

skip Sampreeti  
Debasish

- \* How to reexecute all the failed Test Cases or Test Classes.
  - Sometimes our Test Cases gets failed due to Synchronization Problem. If we reexecute those failed test cases again, chances are there the test cases might get passed.
  - Whenever any TestCases gets failed, inside test output folder a file gets generated as ~~TestNG~~ testing-failed.xml & this file contains all the failed Test Cases.
  - In order to run this file, we right click on it & select Run as TestNG Suite.

- \* Can we have multiple Test methods in a single TestNG class?
  - Order of execution will be alphabetical.

Class MultipleTestMethods,

{

@Test

public void Saroj()

{

    System.out.println("Saroj");

}

@Test

public void Preteek()

{

    System.out.println("Preteek");

}

@Test

public void Seiman()

{

    System.out.println("Seiman");

}

}

~~order~~ Preteek

Saroj

Seiman

Q: How to execute the Test method in required order?

A: By using Priority attribute of TestNG

Note:- Default Priority is zero.

- Priority can be positive or negative value but cannot be decimals & variables.
- Order of execution will be in ascending order of priority.
- If two test methods have the same priority, those methods will get executed in alphabetical order.

L: Class TestMethodsPriority

```
@Test (priority = 3)
public void sarej()
{
    System.out.println("Sarej");
}
```

```
@Test (priority = 1)
public void prateek()
{
    System.out.println("Prateek");
}
```

```
@Test (priority = 2)
public void striman()
{
    System.out.println("Striman");
}
```

Op:- Prateek  
Striman  
Sarej

2: Class TestMethodsPriority

```
@Test
public void sarej()
{
    System.out.println("Sarej");
}
```

```
@Test (priority = 1)
public void prateek()
{
    System.out.println("Prateek");
}
```

```
@Test (priority = 2)
public void striman()
{
    System.out.println("Striman");
}
```

Op:- Sarej  
Prateek  
Striman

### 3. Class Test Methods Priority

{

@Test

public void Saroj()

{

System.out.println("Saroj");

{

@Test (Priority = 1)

public void Preetek()

{

System.out.println("Preetek");

{

@Test

public void Sreiman()

{

System.out.println("Sreiman");

{

{

o/p:- Saroj

Sreiman

Preetek

4.

### 4. Class Test Methods Priority

{

@Test (Priority = -93)

public void Saroj()

{

System.out.println("Saroj");

{

@Test (Priority = -89)

public void Preetek()

{

System.out.println("Preetek");

{

@Test (Priority = -102)

public void Sreiman()

{

System.out.println("Sreiman");

{

{

o/p:- Sreiman

Saroj

Preetek

- How do you execute a Test Method multiple times?
- InvocationCount.

Class TestMethodInvocationCount

```
{ @Test(invocationCount = 2)
```

```
    public void Saroj()
```

```
{     System.out.println("Saroj");
```

```
}
```

Op:- Preteek

Preteek

Preteek

```
@Test(invocationCount = 3)
```

```
    public void Preteek()
```

```
{
```

```
    System.out.println("Preteek");
```

```
}
```

Saroj

Saroj

Sreiman

Sreiman

Sreiman

Sreiman

```
@Test(invocationCount = 4)
```

```
    public void Sreiman()
```

```
{
```

```
    System.out.println("Sreiman");
```

```
}
```

Sreiman

- Can we use both Priority & Invocation Count?

Class PriorityInvocationCount

```
{ @Test(priority = 1, invocationCount = 2)
```

```
    public void Saroj()
```

```
{     System.out.println("Saroj");
```

```
}
```

Op:- Saroj

Saroj

Preteek

Preteek

Preteek

Saroj

Saroj

Sreiman

Sreiman

Sreiman

Sreiman

```
@Test(invocationCount = 3)
```

```
    public void preteek()
```

```
{
```

```
    System.out.println("Preteek");
```

```
}
```

```
@Test(priority = 2, invocationCount = 4)
```

```
    public void Sreiman()
```

```
{
```

```
    System.out.println("Sreiman");
```

```
}
```

```
}
```

Note :- Default invocation Count is 1.

- We cannot use invocationCount as decimals & variables.
- If invocationCount is kept as zero or negative, that particular Test method will not get executed.

### Assignment (Actitime)

1. @Test

Public void login() —— a()

@Test

Public void creatework() —— b()

@Test

Public void deletework() —— ~~c()~~

2. @Test (priority = 1)

Public void login()

@Test (priority = 2)

Public void creatework()

@Test (priority = 3)

Public void deletework()

3. @Test

Public void login()

@Test (priority = 1)

Public void creatework()

@Test (priority = 1)

Public void deletework()

4. @Test

Public void login()

@Test (priority = 1)

Public void creatework()

@Test (priority = 2)

Public void deletework()

Q: How to execute one test method when another test method is passed?

Ans: By using `dependsOnMethods`.

→ Whenever the main test fails, the dependent test will get skipped.

### Class DependsOnMethod

{

`@Test``public void login()`

{

`Sopln("Login");`

}

`@Test(dependsOnMethods = "login")``public void creatework()`

{

~~`soft`~~ `Assert.fail();``Sopln("Create Work");`

}

`@Test(dependsOnMethods = "creatework")``public void deletework()`

{

`Sopln("Delete Work");`

}

}

Q: Can we have multiple dependencies?

Ans: Multiple dependency means one test method is dependent on more than one method.

### Class MultipleDependency

{

`@Test``public void login()`

{

`Sopln("Login");`

}

@Test (dependsOnMethods = "login")

Public void CreateWork()

{

Sopln("Create Work");

}

@Test (dependsOnMethods = {"login", "CreateWork"})

Public void DeleteWork()

{

Sopln("Delete Work");

}

}

Q: How to deliberately skip the execution of a Test Method without using "enabled = false" & "@ignore".

A: If invocationCount is 0 or less than 0 (negative).

Q: How to deliberately fail the execution of a Test Method.

A: By using Fail() of Assert class.

Login → execute → pass → print

CreateWork → execute → Fail → No Print

DeleteWork → skip

Class FailTestMethod

{

@Test

Public void login()

{

Sopln("Login");

}

@Test (dependsOnMethods = "login")

Public void CreateWork()

{

Assert.Fail();

Sopln("Create Work");

}

@Test (dependsOnMethods = "CreateWork")

Public void DeleteWork()

{

Sopln("Delete Work");

}

- Q) Can we have two test methods dependent on each other?  
 ↴ We get TestNG Exception due to Cyclic dependency.

Class CyclicDependency

{ @Test

  Public void login()

{

    System.out.println("Login");

}

  @Test (dependsOnMethods = {"login", "deletework"})

  Public void creatework()

{

    System.out.println("Create work");

}

  @Test (dependsOnMethods = "creatework")

  Public void deletework()

{

    System.out.println("Delete work");

}

}

- Q) How not to skip the dependent test when the main test fails?

- Ans- Using always run attribute (alwaysRun=true)

Class AlwaysRun

{

  @Test

  Public void login()

{

    System.out.println("Login");

}

  @Test (dependsOnMethods = "login")

  Public void creatework()

{

    Assert.fail();

    System.out.println("Create work");

}

@Test (dependsOnMethods = "CreateWork", alwaysRun = true) 143

Public void deletework()

{

System.out.println("delete work");

}

}

Q. How to run a dependent test if the main test is excluded or missing?

A. By using ignoreMissingDependencies

Class IgnoreMissingDependencies

{

@Test

Public void login()

{

System.out.println("Login");

}

@Test (dependsOnMethods = "CreateWork", ignoreMissingDependencies = true)

Public void deletework()

{

System.out.println("Delete work");

}

}

Q. How to deliberately skip the test package, test class & test method at package level, class level & method level.

A. Under Src folder, create a new package with the same name & click on the checkbox, generate package-info.java.

@Ignore  
Package Extra;

Package Level

@Ignore

Public class A

{

}

}

Class Level

@Ignore

@Test

Public void b()

{

}

}

Method Level

Dt - 13/03/2023

Q: How to perform group execution of the Test Cases or Test Classes?

A: 1) by creating separate testing.xml files

Smoke.xml

Regression.xml

Functional.xml

2) by using groups attribute of TestNG.

```
@Test(groups = "Smoke")
```

```
@Test(groups = {"Smoke", "Regression"})
```

Q: Can one Test Method have multiple groups?

A: Yes.

```
@Test(groups = {"Smoke", "Regression"})
```

Q: Can we execute a Test Group when another Test Group is paired?

A: Class DependsOnGroups Attribute

```
@Test(groups = "Smoke")
```

```
public void login()
```

```
{ Assert.fail(); }
```

```
Sopln("login method is paired");
```

```
@Test(groups = "Regression", dependsOnGroups = "Smoke")
```

```
public void CreateWork()
```

```
{ Sopln("Create work method is paired"); }
```

```
@Test(dependsOnGroups = {"Smoke", "Regression"})
```

```
public void DeleteWork()
```

```
{ Sopln("Delete work method is paired"); }
```

Q How to execute a Test Group which is dependent on another Test Group?

A // testing.xml file

```

<Suite name="Suite">
  <test thread-count="3" name="Test">
    <groups>
      <dependencies>
        <group depends-on="Smoke" name="Sanity"></group>
        <group depends-on="Sanity" name="regression"></group>
      </dependencies>
    </groups>
    <clauses>
      <clause name="extra.DependsOnGroups"/>
    </clauses>
  </test>
</Suite>

```

// Program

```

class DependsOnGroups
{
  @Test(groups = "Smoke")
  Public void Validusername()
  {
    System.out.println("Valid Username");
  }

  @Test(groups = "Smoke")
  Public void Invalidpassword()
  {
    System.out.println("Invalid Password");
  }

  @Test(groups = "Sanity")
  Public void Invalid_username()
  {
    System.out.println("Invalid Username");
  }

  @Test(groups = "Sanity")
  Public void Validpassword()
  {
    System.out.println("Valid Password");
  }
}

```

```

  @Test(groups = "regression")
  Public void ValidUNValidPassword()
  {
    System.out.println("Valid Username Valid Password");
  }
}

```

### Output:-

```

  Invalid password
  Invalid Username
  Valid Password
  Valid Username
  Valid Username Valid Password

```

Q How to execute the Test Cases @ package Level ?

I (i) By using Packages Tag in TestNG.xml file

```
<Suite name = "Suite">
```

```
  <test thread-count = "5" name = "Test">
```

```
    <packages>
```

```
      <package name = "Selenium2"></package>
```

```
    </packages>
```

```
  </test> <!-- Test -->
```

```
</Suite> <!-- Suite -->
```

(ii) By Converting the package into TestNG suite / By creating separate xml files for different packages.

(iii) By using @Ignore in package level.

Q How to execute a Test Method in a Specified time ?

I → By using timeout attribute (millisecond)

→ If the test method is not executed within the timeout specified, we get ThreadTimeoutException.

Class TimeOutAttribute

```
{  
  public WebDriver driver;  
  @Test(timeout = 5000)  
  public void actitimeLoginLogout()  
  {  
    System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");  
    driver = new ChromeDriver();  
    driver.get("https://demo.actitime.com");  
  }  
}
```

ActitimeLoginPage LoginPage = new ActitimeLoginPage(driver); 147

LoginPage.LoginMethod();

ActitimeHomePage HomePage = new ActitimeHomePage(driver);

HomePage.LogoutMethod();

}

}

Q: How to specify the timeout of a Test Method having invocationCount?

A: → Using InvocationTimeout (milliseconds)

Class InvocationTimeout

{

public WebDriver driver;

@Test

Public void login()

{

ChromeDriver

// ChromeOptions options = new ChromeOptions(); } used

// options.addArguments("--remote-allow-origins=\*"); } for 4.7 & above

System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

driver = new ChromeDriver();

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

ActitimeLoginPage LoginPage = new ActitimeLoginPage(driver);

LoginPage.LoginMethod();

}

@Test(invocationCount = 2, invocationTimeout = 10000)

Public void CreateCustomer()

{

ActitimeHomePage HomePage = new ActitimeHomePage(driver);

HomePage.TasksMethod();

```

ActitimeTasksPage tasksPage = new ActitimeTasksPage(driver);
tasksPage.addnewMethod();
tasksPage.newCustomerMethod();
}

```

ActitimeCreateNewCustomerPage customerPage = new

```

ActitimeCreateNewCustomerPage(driver);

```

```

CustomerPage.entercustomernameMethod();

```

```

CustomerPage.createCustomerMethod();

```

```

HomePage.logoutMethod();
}
}

```

Dt-14/03/2023

Q Explain ThreadPoolSize attribute in TestNG.

A Class ThreadPoolSize

```

public WebDriver driver;

```

```

@Test(invocationCount = 10, threadPoolSize = 2)

```

```

public void login()
{

```

```

System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

```

```

driver = new ChromeDriver();

```

```

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

```

```

ActitimeLoginPage loginPage = new ActitimeLoginPage(driver);

```

```

loginPage.loginMethod();
}
}

```

```

ActitimeHomePage homepage = new ActitimeHomePage(driver);

```

```

homepage.logoutMethod();
}
}

```

## Verification In TestNG :-

- Comparing the actual result with the expected result is called as Verification.
- In order to perform Verification we use Assert class of TestNG.
- During runtime if Comparison fails, then TestNG will not allow the remaining statements of the current test method to get executed. Whereas other Test methods will get executed as it is.
- If Comparison fails, we get Assertion Error.

Class VerificationTestNG

```
{ public WebDriver driver;
```

@Test

```
public void VerifyTitle()
```

{

---

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

```
String expectedTitle = "Sampachi";
```

```
String actualTitle = driver.getTitle();
```

```
Assert.assertEquals(actualTitle, expectedTitle); //Assertion Error
```

```
Sopln("both title are matching");
```

}

@Test (dependsOnMethods = "Verify Title")

```
public void Zakore()
```

{

```
Sopln("Sakthi Lounda");
```

}

{

- Note) → If we want to continue the execution process even after comparison fails, we use SoftAssert class.
- SoftAssert class has all the methods of Assert class but all the methods are non-static.
  - Whenever we use the methods of SoftAssert class, at the end we must call assertAll() to update the result.
  - Any statements present after the assertAll() will not get executed if comparison fails.

Class SoftAssertDemo

```

{
    public WebDriver driver;
    @Test
    public void verifyTitle()
    {
        _____
        driver.get("https://demo.actitime.com");
        String expectedTitle = "Saaopreethi";
        String actualTitle = driver.getTitle();
        SoftAssert s = new SoftAssert();
        s.assertEquals(actualTitle, expectedTitle);
        Sopln("both titles are matching");
        s.assertAll();
        Sopln("getting executed");
    }

    @Test
    public void zaki()
    {
        Sopln("Sakth Launda");
    }
}

```

Assert / Hard Assert

- (i) Execution will not continue if comparison fails.
- (ii) All the methods are static.
- (iii) We don't use assertAll() as result gets automatically updated.

Verify / Soft Assert

- (i) Execution will continue even if comparison fails.
- (ii) All the methods are non-static.
- (iii) We use assertAll() to update the result.

Q: When we use Assert class & When we use Soft Assert class?

A)- For Critical Test Cases, we use Assert class.

For major & minor Test Cases we use Soft Assert class.

Assignment

@Test

public void login()

@Test (priority = 1, invocationCount = 2)

public void createCustomer()

•  $\hookrightarrow$  Verify Error Message  
using Assert class

Q: How to use ThreadCount for ~~parallel~~ parallel execution of Test Classes?

Ans

// testing.xml file

<Suite name = "Suite" parallel = "classes" thread-count = "2" >

<test name = "Test" >

<classes>

<class name = "threadCountClasses.A" />

<class name = "threadCountClasses.B" />

<class name = "threadCountClasses.C" />

<class name = "threadCountClasses.D" />

</classes>

</test>

## // Test class - A

```

package threadCountClasses;
class A
{
    @Test
    public void a()
    {
        System.out.println("a is executed");
    }

    @Test
    public void b()
    {
        System.out.println("b is executed");
    }

    @Test
    public void c()
    {
        System.out.println("c is executed");
    }
}

```

## // Test class - C

```

package threadCountClasses;
class C
{
    @Test
    public void f()
    {
        System.out.println("f is executed");
    }

    @Test
    public void g()
    {
        System.out.println("g is executed");
    }

    @Test
    public void h()
    {
        System.out.println("h is executed");
    }
}

```

## // Test class - B

```

package threadCountClasses;
class B
{
    @Test
    public void d()
    {
        System.out.println("d is executed");
    }

    @Test
    public void e()
    {
        System.out.println("e is executed");
    }
}

```

## // Test class - D

```

package threadCountClasses;
class D
{
    @Test
    public void i()
    {
        System.out.println("i is executed");
    }

    @Test
    public void j()
    {
        System.out.println("j is executed");
    }

    @Test
    public void k()
    {
        System.out.println("k is executed");
    }
}

```

Output:-

a is executed  
 d is executed  
 b is executed  
 e is executed  
 c is executed  
 f is executed  
 g is executed  
 i is executed  
 h is executed  
 j is executed  
 k is executed

Q. How to use threadCount for parallel execution of Test Methods?

A. //Testing.xml file

```
<Suite name="Suite" parallel="methods" thread-count="2">
  <test name="Test">
    <classes>
      <class name="threadCountClasses.A"/>
      <class name="threadCountClasses.B"/>
      <class name="threadCountClasses.C"/>
      <class name="threadCountClasses.D"/>
    </classes>
  </test>
</Suite>
```

Output:-

a is executed  
 b is executed  
 c is executed  
 d is executed  
 e is executed  
 f is executed  
 g is executed  
 h is executed  
 i is executed  
 j is executed

## Annotations in TestNG :-

→ There are three types of annotations i.e.

- (i) Pre Annotation → @Before
- (ii) Test Annotation → @Test
- (iii) Post Annotation → @After

### (i) @Before Method :-

This method will be executed before the execution of every Test Method present in the Test class.

### (ii) @AfterMethod :-

This method will be executed after the execution of every Test Method present in the Test class.

### (iii) @BeforeClass :-

This method will be executed before the execution of the Test Class.

### (iv) @AfterClass :-

This method will be executed after the execution of the Test class.

### (v) @BeforeTest :-

This method will be executed before the execution of Test tag present in TestNG.xml file.

### (vi) @AfterTest :-

This method will be executed after the execution of Test tag present in TestNG.xml file.

(vii) @BeforeSuite :-

This method will be executed before the execution of Suite tag present in TestNG.xml file.

(viii) @AfterSuite :-

This method will be executed after the execution of Suite tag present in TestNG.xml file.

`@ BeforeSuite`

`@ BeforeTest`

`@ BeforeClass`

`@ BeforeMethod`

`@ Test`

`@ AfterMethod`

`@ AfterClass`

`@ AfterTest`

`@ AfterSuite`

// Program

Class AnnotationsTestNG

{

`@Test`

Public void c()

{

    Sopln("c");

}

`@Test`

Public void b()

{

    Sopln("b");

}

`@Test`

Public void a()

{

    Sopln("a");

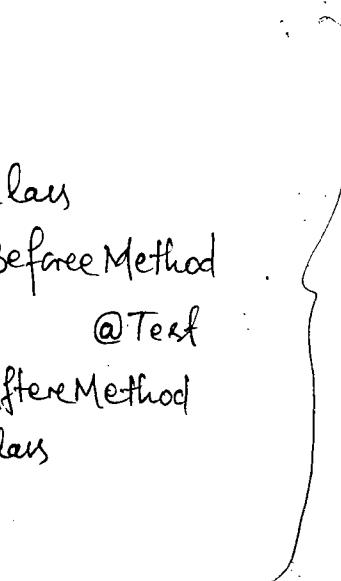
}

`@ BeforeClass`

Public void Chandra()

{

    Sopln("Chandra");



→ Order of  
execution

@BeforeClass

```
public void gireja()
{
    Sopln("gireja");
}
```

@BeforeMethod

```
public void Swastik()
{
    Sopln("Swastik");
}
```

@BeforeTest

```
public void Sampreeti()
{
    Sopln("Sampreeti");
}
```

@BeforeSuite

```
public void Preadosh()
{
    Sopln("Preadosh");
}
```

@AfterSuite

```
public void vishal()
{
    Sopln("vishal");
}
```

@AfterMethod

```
public void Shoeb()
{
    Sopln("Shoeb");
}
```

@AfterClass

```
public void debarish()
{
    Sopln("Debarish");
}
```

@AfterTest

```
public void Pretyusha()
{
    Sopln("pretyusha");
}
```

### Output:-

Preadosh

Sampreeti  
Chandrea  
gireja

Swastik

a  
Shoeb

Swastik

b  
Shoeb

Swastik

c  
Shoeb

Debarish  
Pretyusha  
Vishal

## Real Time Annotation Implementation of TestNG Annotations :-

// Base class

class BaseTest

{

    public WebDriver driver;

    @BeforeSuite

    public void executionStart()

{

        System.out.println("execution started");

}

    @BeforeClass

    public void openBrowser()

{

        // ChromeOptions options = new ChromeOptions();

        // options.addArguments("--remote-allow-origins=\*");

        System.setProperty("webdriver.chrome.silentOutput", "true");

        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

        driver = new ChromeDriver();

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

        ActitimeLoginPage loginPage = new ActitimeLoginPage(driver);

        loginPage.loginMethod();

}

    @AfterClass

    public void logoutFromApplication()

{

        ActitimeHomePage HomePage = new ActitimeHomePage(driver);

        HomePage.logoutMethod();

        driver.quit();

}

@AfterSuite

public void executionComplete()

{

System.out.println("execution completed");

}

}

// Test class - LoginLogout

Class ActitimeLoginLogout extends BaseTest

{

@Test

public void loginLogout()

{

System.out.println("loginLogout method is executing");

}

}

// Test class - LoginCreateCustomerLogout

Class ActitimeLoginCreateCustomerLogout extends BaseTest

{

@Test

public void createCustomer()

{

ActitimeHomePage homepage = new ActitimeHomePage(driver);

homepage.tasksMethod();

ActitimeTasksPage tasksPage = new ActitimeTasksPage(driver);

tasksPage.addNewMethod();

tasksPage.newCustomerMethod();

ActitimeCreateNewCustomerPage customerPage = new

ActitimeCreateNewCustomerPage(driver);

customerPage.enterCustomerNameMethod();

customerPage.createCustomerMethod();

{

}

Q: Can we declare parallel = classes at Suite level, Test level & classes level?

A:-

```

<Suite name = "Suite" parallel = "classes"> — Possible
<test name = "Test" parallel = "classes"> — Possible
<classes parallel = "classes"> — Not Possible
    <class>
        <class>
            </classes>
        </test>
    </Suite>

```

Q: Can we declare parallel = methods at Suite level, Test level & classes level?

A:-

```

<Suite name = "Suite" parallel = "methods"> — Possible
<test name = "Test" parallel = "methods"> — Possible
<classes parallel = "methods"> — Possible
    <class parallel = "methods"> — Particular class
        <class>
            </classes>
        </test>
    </Suite>

```

Q: Can we write parallel = false with multiple thread-count?

A:- Execution will happen sequentially.

Q: What will happen if we specify parallel = none?

A:- Execution will happen sequentially.

Q: Can we provide thread count as negative for parallel execution?

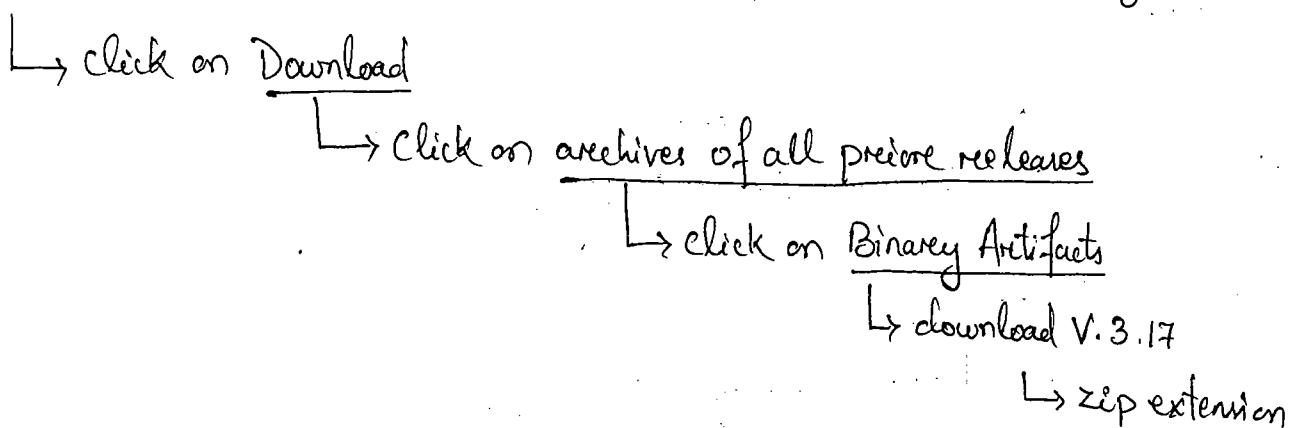
A:- We get illegalArgumentException.

- Q) What will happen if there are more thread count & less test classes tag , method tag & test tag ?
- If Execution will happen parallelly for no. of threads present but other threads will remain idle .  
 (Not a good practice to write)

## Excel File Handling

- In real time implementation of our automation Script, our Test Data should not be hard coded .
- It is always recommended to use some external resources like excel sheet, xml file, json file, properties file etc.
- This process is nothing but called as File Manipulation .
- To create and maintain Excel file, Java provides different classes and interfaces for file manipulation and we use Apache POI jar files to perform such operations .

Apache POI official website — <https://www.poi.apache.org>



## Steps to Read Data from Excel Sheet :-

- (i) Manually store all the data inside Excel sheet.
- (ii) Store the path of the Excel sheet.
- (iii) Enter inside the Excel sheet.
- (iv) Enter inside the respective sheet.
- (v) Enter inside the respective row.
- (vi) Enter inside the respective cell.
- (vii) Fetch the data from the cell.

Dt - 18/03/2023

Note:- → To create & maintain an excel file, Apache POI provides Workbook as a super interface of all classes.

→ There are two classes which implements workbook interface

- (i) HSSFWorkbook → .xls
- (ii) XSSFWorkbook → .xls or .xlsx

HSSF - Horrible Spread Sheet Format

XSSF - XML SpreadSheet Format

→ If the path of the Excel file is incorrect, we get FileNotFoundException.

→ If the sheet name or row index or cell index is incorrect we get NullPointerException.

→ If we are trying to fetch integer value using ~~get~~ getStringCellValue(), we get IllegalStateException.

→ In order to convert integer value to string, we should put a single quote in the beginning of the integer value.

## Class ReadDataFromExcel

@Test

public void readData() throws IOException

{

FileInputStream fis = new FileInputStream

("./testdata/Qspiders.xlsx");

XSSFWorkbook wb = new XSSFWorkbook(fis);

// Sheet s = wb.getSheet("Anil");

// Row r = s.getRow(3);

// Cell c = r.getCell(2);

// System.out.println(c.getStringCellValue());

// System.out.println(wb.getSheet("readash").getRow(4).getCell(5).getStringCellValue());

System.out.println(wb.getSheet("readash").getRow(4).getCell(5).toString());

}

Write data into excel sheet :-

1. Store the path of the excel sheet.
2. Enter inside the Excel sheet.
3. ~~Enter~~ Enter inside the existing sheet or create a new sheet.
4. ~~Enter~~ Create a new row.
5. Create a new cell.
6. Specify the cell value.
7. Write the data into Excel sheet.

## Class WriteDataToExcelSheet

@Test

public void writeData() throws IOException

{

FileInputStream fis = new FileInputStream("./testdata/Qspiders.xlsx");

Workbook wb = new XSSFWorkbook(fis);

// Sheet s = wb.getSheet("anil");

```

Sheet s = wb.createSheet("Kaushik");
Row r = s.createRow(10);
Cell c = r.createCell(10);
c.setCellValue("Kaushik is a good boy !!!");
FileOutputStream fos = new FileOutputStream
("./testdata/Gspiders.xlsx"),
wb.write(fos);
}

```

### Fetch Data from Properties File :—

→ A property file is a simple file which is used to store the data in the form of key & value pair.

#### // config.properties file

url = https://demo.actitime.com

usernameLocate = username

passwordLocate = pwd

loginLocate = loginButton

username = admin

password = manager

#### // Program

##### Class ReadDataPropertiesFile

{

public WebDriver driver;

@Test

public void propertiesfile() throws IOException

{

FileInputStream fis = new FileInputStream("./propertyfile/

config.properties");  
Properties properties = new Properties();  
properties.load(fis);

```
//ChromeOptions options = new ChromeOptions();
//options.addArguments("--remote-allow-origins=*");
System.setProperty("webdriver.chrome.silentOutput", "true");


---



```

```
driver.get(properties.getProperty("url"));
driver.findElement(By.id(properties.getProperty("usernameLocator"))).SendKeys(properties.getProperty("username"));
}
}
```

DT-20/03/2023

## Fetching Data from String 1D Array :-

Class FetchDataString1DArray

```
{ public WebDriver driver;
@DataProvider(name = "loginData")
public String[] provideData()
{
    String[] data = {"admin", "presha", "Samprati"};
    return data;
}
```

```
@Test(dataProvider = "loginData")
public void setup(String username)
```

```
{ //ChromeOptions options = new ChromeOptions();
//options.addArguments("--remote-allow-origins=*");
System.setProperty("webdriver.chrome.silentOutput", "true");


---



```

```
driver.get("https://demo.actitime.com");
}
```

```
driver.findElement(By.id("username")).SendKeys(username);
}
```

## Fetching Data from Object 1D Array :-

```

Class FetchDataObject1DArray
{
    Public WebDriver driver;
    @DataProvider(name = "loginData")
    Public Object [] pandata()
    {
        Object [] data = {"admin", "Praresh", "Sampreeti"};
        Return data;
    }
    @Test(dataProvider = "loginData")
    Public void setup(String username)
    {
        _____
        _____
        _____
        driver.get("https://demo.actitime.com");
        driver.findElement(By.id("username")).sendKeys(username);
    }
}

```

## Fetching Data from String 2D Array :-

```

Class FetchDataString2DArray
{
    Public WebDriver driver;
    @DataProvider(name = "loginData")
    Public String [][] pandata()
    {
        String [][] data = { {"admin", "manager"}, {"Praresh", "manager"}, {"Sampreeti", "Sampreeti"} };
        Return data;
    }
}

```

```

@Text(dataProvider = "loginData")
Public void setup(String username, String password)
{
    _____
    _____
}

```

```

driver.get("https://demo.actitime.com");
driver.findElement(By.id("username")).sendKeys(username);
driver.findElement(By.name("pwd")).sendKeys(password);
}

```

## Fetch Data from Object 2D Array :-

Class FetchDataObject2DArray

{

```

public WebDriver driver;
@DataProvider(name = "loginData")
public Object[][] provideData()
{

```

```

Object[][] data = { {"admin", "manager"},  

                    {"pradeesh", "pradeesh"},  

                    {"Sampreeti", "Sampreeti"} };

```

return data;

}

@Test(dataProvider = "loginData")

public void setup(String username, String password)

{

---



---



---



---



---



---



---



---

```

driver.get("https://demo.actitime.com");
driver.findElement(By.id("username")).sendKeys(username);
driver.findElement(By.name("pwd")).sendKeys(password);
}

```

167

Fetch Data from Strong 2D Array [having multiple Textfield]:

## Fetch Data from Object 2D Array [having multiple textfields] :-

Class FetchDataObject2DArryDemo

{

    Public WebElement driver;

    @DataProvider(name = "loginData")

    Public Object[][] provideData()

{

        Object[][] data = { { "admin", "manager" },  
                           { "preetesh", "preetesh" },  
                           { "Sampreeti", "Sampreeti" } };

    Return data;

}

@Test(dataProvider = "loginData")

Public void setup(Object[] obj)

{

    \_\_\_\_\_  
     \_\_\_\_\_  
     \_\_\_\_\_  
     \_\_\_\_\_

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

    driver.findElement(By.id("username")).SendKeys(obj[0]); //Error

    driver.findElement(By.name("pwd")).SendKeys(obj[1]); //Error

}

}

\* SendKeys return type is String so it will throw Error

Saying "The method SendKeys(CharSequence...) in the type WebElement is not applicable for the arguments (Object). So, we have to use String Arguments for integer values.

AssignmentObject [][] & String [][]Create a Separate  
'class'

# generics

BaseTest @Test  
DataProviderC@DataProvider  
(name = "loginData")@DataProvider  
(name = "RegisterData")@DataProvider  
(name = "CustomerData")

- Actitime - login using 5 different credentials & verify whether home page is displayed or not & quit the browser.
- Actitime - Create 5 different sets of customer by entering customer name & customer description & quit the browser.

3. Facebook - Firstname  
Surname

int - mobile no

new password

} 3 sets of data

Q: Can we give data with missing values & fetch data from Object/String 2DArray?

A: No, it will throw MethodMatcherException

Class FetchDataObject2DArrayException

{
 public WebDriver driver;

@DataProvider(name = "loginData")

public Object[][] pdata()

{ Object[][] data = { { "admin", "manager" },

{} {"pradash," } // MethodMatcherException

{} {"Sampreati", "Sampreati" } };

return data;

Dt - 21/03/2025

```

    @Test(dataProvider = "loginData")
    public void setup(String username, String password)
    {
        _____
        _____
        _____
        _____
        driver.get("https://demo.actitime.com");
        driver.findElement(By.id("username")).sendKeys(username);
        driver.findElement(By.name("pwd")).sendKeys(password);
    }
}

```

Q. How to run DataProvider in Parallel?

A. Class DataProviderParallel

```

{
    public WebDriver driver;
    @DataProvider(name = "loginData", parallel = true)
    public Object[][] provideData()
    {
        Object[][] data = { {"admin", "manager"},  

                           {"pradeesh", "pradeesh"},  

                           {"Sampreeti", "Sampreeti"} };
        return data;
    }
}

```

```

    @Test(dataProvider = "loginData")
    public void setup(String username, String password)
    {
        _____
        _____
        _____
        _____
    }
}

```

```

        driver.get("https://demo.actitime.com");
        driver.findElement(By.id("username")).sendKeys(username);
        driver.findElement(By.name("pwd")).sendKeys(password);
    }
}

```

Q: How to use DataProvider with a Thread Count ?

A: // testing.xml file

```
<Suite name="Suite" data-provider-thread-count="2">
  <test name="Test">
    <classes>
      <class name="fetchdata.DataProviderThreadCount"/>
    </classes>
  </test>
</Suite>
```

// Program

Class DataProviderThreadCount

{

```
  public WebDriver driver;
  @DataProvider(name="loginData", parallel=true)
  public Object[][] provideData()
  {
    Object[][] data = { { "admin", "manager" },
                       { "preetesh", "preetesh" },
                       { "Sampreeti", "Sampreeti" } };
    return data;
  }
```

@Test(dataProvider = "loginData")

Public void setup(String username, String password)

{

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

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

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

driver.findElement(By.name("pwd")).sendKeys(password);

}

{

## Fetching Data from Tagged Array :-

→ A Jagged Array is an array of arrays such that member arrays can be of different sizes i.e. we can create a 2D Array with a Variable no. of Columns in each row.

Class FetchDataJaggedArray

```
{
    @Test(dataProvider = "data")
    public void login(Object[] obj)
    {
        for(int i=0; i<obj.length; i++)
        {
            System.out.println(obj[i] + " ");
        }
        System.out.println();
        System.out.println("-----");
    }
}
```

```
@DataProvider(name = "data")
public Object[][] parseData()
{
    Object[][] data = { {"admin", "manager"},  

                       {"Samprati"},  

                       {"ashish", "Kumar", "Senapati"} };
    return data;
}
```

Op:-

admin	manager
Samprati	
ashish	Kumar
	Senapati

## Fetching Data from HashMap :-

→ HashMap implements Map interface wherein we store the data in the form of key & value.

Class FetchDataHashMap

```
{ public WebDriver driver;
```

@BeforeClass

```
public void setup()
```

```
{
```

```
=====
```

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

```
}
```

@AfterClass

```
public void teardown()
```

```
{
```

```
driver.quit();
```

```
}
```

Public static HashMap<String, String> getcredentialsmap()

```
{
```

```
HashMap<String, String> hash = new HashMap<>();
```

```
hash.put("admin", "admin:manager");
```

```
hash.put("distribution", "Sampreeti: Sampreeti123");
```

```
hash.put("Seller", "Debasish: Debasish123");
```

```
hash.put("delivery", "Shantnu: Shantnu123");
```

```
hash.put("User", "Disha: Disha123");
```

```
return hash;
```

```
}
```

@Test

```
public void login()
```

```
{
```

```

//driver.findElement(By.id("username")).sendKeys(credentialsMap().get("user")
    .split(":")[0]);
// driver.findElement(By.name("pwd")).sendKeys(credentialsMap().get("user")
    .split(":")[1]);
driver.findElement(By.id("username")).sendKeys(username("Seller"));
driver.findElement(By.name("pwd")).sendKeys(password("Seller"));
}

```

```
public static String getUsername(String role)
```

```
String username = credentialsMap().get(role);
return username.split(":")[0];
}
```

```
public static String getPassword(String role)
```

```
String password = credentialsMap().get(role);
return password.split(":")[1];
}
```

```
}
```

## Assignment

### ① HashMapDemo

```
@DataProvider(name = "loginData")
public HashMap<String, String>
getCredentialsMap()
```

DATA

Parallel → true

### Test Case

#### LOGIN

① Test  
UN  
PW

→ Parallel

Q. Why we don't use @DataProvider in Realtime project?

- Ans (i) Data Creation using @DataProvider is time consuming.
- (ii) Modification is a complex job & hence maintenance is very high.

Dt - 22/03/23

## Automation Framework

- Framework is a collection of reusable classes and interfaces that makes automation script development and execution faster.
  - Every Automation Framework has 3 stages :-
- (i) Framework Design
  - (ii) Framework Implementation
  - (iii) Framework Execution

### ~~Framework~~

#### Framework Design :-

- Our Framework Design was done by our Test Manager. Our Framework Design contains the following classes & interfaces :-
1. Auto Constant Interface
  2. BaseTest class
  3. BasePage class
  4. Excel Library class / Hashmap / Array
  5. Helper class
  6. Screenshot class

## 1. AutoConstant Interface

→ Inside this interface we have stored all the constant data that are going to remain same for the entire project like storing the Key & Value pairs of Browser Drivers, URL of the application, Path of the Excel sheet, Name of the sheet, etc.

### Public interface AutoConstant

{

```
String chrome_keys = "webdriver.chrome.driver";
```

```
String chrome_values = "./drivers/chromedriver.exe";
```

```
String gecko_keys = "webdriver.gecko.driver";
```

```
String gecko_values = "./drivers/geckodriver.exe";
```

```
String edge_keys = "webdriver.edge.driver";
```

```
String edge_values = "./drivers/msedgeedge.exe";
```

```
String url = "https://demo.actitime.com";
```

```
String excel_path = "./testdata/ActitimeData.xlsx";
```

```
String sheet_name = "LoginData";
```

{}

## 2. BaseTest Class

→ Inside this class we have stored all the reusable lines of codes like : Opening the Browser, Maximizing the Browser, Entering the URL, Closing the Browser, etc.

### Class BaseTest implements AutoConstant

{

```
public WebDriver driver;
```

```
@BeforeSuite
```

```
public void executionStart()
```

{

```
Sopln("execution started");
```

{

```
@BeforeClass
```

```
public void setup()
```

{

```

System.setProperty("chrome-key", "chrome-value");
driver = new ChromeDriver();
driver.manage().window().maximize();
driver.get(url);
}

@AfterClass
public void teardown()
{
    driver.quit();
}

@AfterSuite
public void executionComplete()
{
    System.out.println("Execution completed");
}

```

### 3. Excel Library class

→ Inside this class we have stored the lines of code to fetch the data from the Excel Sheet.

Class ExcelLibrary implements AutoConstant

```

{
    public String getCellValue(String sheet, int row, int cell)
    {
        FileInputStream fis = new FileInputStream(excel-path);
        XSSFWorkbook wb = new XSSFWorkbook(fis);
        String cellValue = wb.getSheet(sheet).getRow(row).getCell(cell)
            .getStringCellValue();
        return cellValue;
    }
}

```

### 4. BasePage Class

→ Inside this class we have stored all the reusable methods of different classes & interfaces like Select class, Actions class, Robot class, Alert interface, WebDriver interface, etc.

Class BasePage

178

{

//Select class

Public void SelectByVisibleText(WebElement element, String text)

{

Select select = new Select(element);

Select.selectByVisibleText(text);

}

//Actions class

Public void moveToElement(WebDriver driver, WebElement element)

{

Actions actions = new Actions(driver);

actions.moveToElement(element).perform();

}

//Robot class

Public void rebotTab()

{

Robot rebot = new Robot();

rebot.keyPress(KeyEvent.VK\_TAB);

rebot.keyRelease(KeyEvent.VK\_TAB);

}

//Alert Interface

Public void alertAccept(WebDriver driver)

{

driver.switchTo().alert().accept();

}

Public void alertEnter(WebDriver driver, String text)

{

driver.switchTo().alert().sendKeys(text);

}

//JavascriptExecutor Interface

Public void javascriptClick(WebDriver driver, WebElement element)

{

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click()", element);

}

## // WebDriver Interface

```
Public void getTitle(WebDriver driver, String text)
{
    String actualTitle = driver.getTitle();
    if(actualTitle.contains(text))
    {
        System.out.println("titles are matching");
    }
    else
    {
        System.out.println("titles are not matching");
    }
}
```

## // WebDriverWait class

```
Public void visibilityOfElementLocated(WebDriver driver, By locator)
{
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30));
    wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
}

Public void titleIs(WebDriver driver, String title)
{
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30));
    wait.until(ExpectedConditions.titleIs(title));
}
```

## // Package POM

```
Class ActitimeLoginPage extends BasePage implements AutoConstant
{
    Public WebDriver driver;
    @FindBy(id = "username")
    private WebElement usernameTextField;
    @FindBy(name = "pwd")
    private WebElement passwordTextField;
    @FindBy(id = "loginButton")
    private WebElement loginButton;
```

```
Public ActitimeLoginPage (WebDriver driver)
```

```
{ this.driver = driver;
```

```
    Pagefactory.initElements(driver, this);
```

```
}
```

```
Public void loginMethod()
```

```
{
```

```
    usernameTextfield.sendKeys(ExcelLibrary.getCellValue(sheet_name, 1, 0));
```

```
    PasswordTextfield.sendKeys(ExcelLibrary.getCellValue(sheet_name, 1, 1));
```

```
    getTitle(driver, "Time-Track");
```

```
}
```

```
}
```

// Package Test

```
Class ActitimeLoginLogout extends BaseTest
```

```
{
```

```
@Test
```

```
Public void loginLogout()
```

```
{
```

```
    ActitimeLoginPage LoginPage = new ActitimeLoginPage(driver);
```

```
    LoginPage.loginMethod();
```

```
}
```

```
}
```

## Framework Implementation

→ According to the Framework design, we will inherit different classes & interfaces in our POM class & Test class.

## Framework Execution

→ Once all the Test Cases are written, we will convert the Test Cases into TestNG Suite & finally run the testng.xml file.

## Assignment (Framework)

1. Login to Actifino & logout
2. Login to Actifino , create Customer & logout }
3. Login to Actifino , create type of work & logout }
4. Login to Actifino , delete type of work & logout }
2. Facebook Signup Page

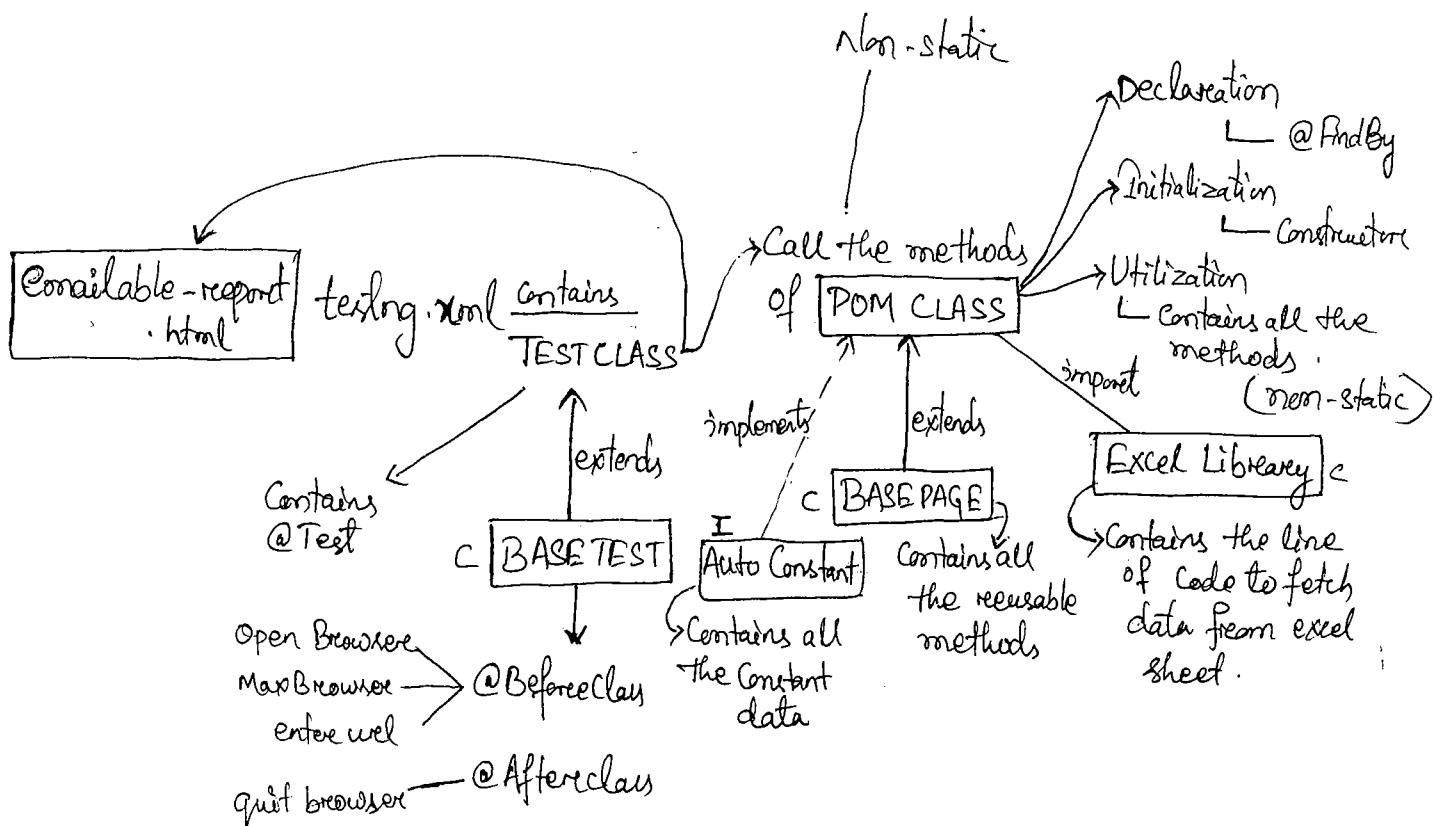
Dt-23/03/2023

## Types of Automation Framework :-

1. Data Driven Framework
2. Method Driven Framework
3. Hybrid Framework
4. Test Driven Framework
5. Keyword Driven Framework
6. Behaviour Driven Development Framework.

- In our Project we have used Hybrid Framework which is a combination of Data Driven and Method Driven Framework .
- Data Driven because we have stored all the Test Data of the project inside a separate class called as Excel Library .
- Method Driven because we have stored all the Reusable Methods of different classes and interfaces inside BasePage class .

## Architecture of Automation Framework :-



## Highlighting the Elements of a WebPage :-

// Package Generics - Helper class

class Helper

```
{
    public static void highlightElement(WebDriver driver, WebElement element)
    {
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript("arguments[0].setAttribute('style', 'background: red')", element);
        js.executeScript("arguments[0].setAttribute('style', 'background: white')", element);
    }
}
```

## // Package POM

class ActitimeLoginPage extends BasePage implements AutoConstant

```
{
    public WebDriver driver;
    @FindBy(id = "username")
```

```

private WebElement usernameTextField;
@FindBy(name = "pwd")
private WebElement passwordTextField;
@FindBy(id = "LoginButton")
private WebElement loginButton;
public ActitimeLoginPage(WebDriver driver)
{
    this.driver = driver;
    PageFactory.initElements(driver, this);
}
public void loginMethod()
{
    Helper.highlightElement(driver, usernameTextField);
    usernameTextField.sendKeys(ExcelLibrary.getCellValue(sheetName, 1, 0));
    Helper.highlightElement(driver, passwordTextField);
    passwordTextField.sendKeys(ExcelLibrary.getCellValue(sheetName, 1, 1));
    javascriptClick(driver, loginButton);
}

```

## Cross Browser Testing

- It is the process of executing the same Test Case across different different browsers.
- To execute to perform Cross Browser Testing we use @Parameters -

// BaseTest Class

class BaseTest implements AutoConstant

```

{
    public WebDriver driver;
    @BeforeSuite
    public void executionStart()
    {
        System.out.println("Execution Started");
    }
}
```

~~@BeforeClass~~

@Parameters("browser")

@BeforeClass

public void setup(String browser)

{

```

if (browser.equalsIgnoreCase("chrome"))
{
    ChromeOptions options = new ChromeOptions();
    options.addArguments("--remote-allow-origins=*");
    System.setProperty("webdriver.chrome.silentOutput", "true");
    System.setProperty(chromeKey, chromeValue);
    driver = new ChromeDriver(options);
    driver.manage().window().maximize();
    driver.get(url);
}

else if (browser.equalsIgnoreCase("edge"))
{
    EdgeOptions options = new EdgeOptions();
    options.addArguments("--remote-allow-origins=*");
    System.setProperty("webdriver.edge.silentOutput", "true");
    System.setProperty(edgeKey, edgeValue);
    driver = new EdgeDriver(options);
    driver.manage().window().maximize();
    driver.get(url);
}

else {
    FirefoxOptions options = new FirefoxOptions();
    options.addArguments("--remote-allow-origins=*");
    System.setProperty("webdriver.firefox.silentOutput", "true");
    System.setProperty(geckoKey, geckoValue);
    driver = new FirefoxDriver(options);
    driver.manage().window().maximize();
    driver.get(url);
}
}

```

```

@AfterClass
public void teardown()
{
    driver.quit();
}

@AfterSuite
public void executionComplete()
{
    Sopn("execution completed");
}
}

```

## //testng.xml file

```

<Suite name="Suite" parallel="tests">
    <test name="ChromeTest">
        <parameter name="browser" value="chrome"></parameter>
        <classes>
            <class name="test.ActitimeLoginLogout"/>
        </classes>
    </test>

    <test name="EdgeTest">
        <parameter name="browser" value="edge"></parameter>
        <classes>
            <class name="test.ActitimeLoginLogout"/>
        </classes>
    </test>

    <test name="FirefoxTest">
        <parameter name="browser" value="firefox"></parameter>
        <classes>
            <class name="test.ActitimeLoginLogout"/>
        </classes>
    </test>
</Suite>

```

# Assignment (Create Brewhere - Chrome, Edge, Firefox)

186

1. Login to Actitime & logout
2. Login to Actitime , CreateCustomer , <sup>@Test</sup> delete customer & logout
3. Login to Actitime , create work , delete work & logout
4. Login to Actitime , create leave type , delete leave & logout

Dt-24/03/2023

## Taking Screenshots in Selenium :-

- In order to take screenshot in Selenium, we use @Listeners of TestNG.
- We can take screenshot of Test classes that got passed, failed, skipped, untest start, untest finish, etc.
- But generally screenshots are taken whenever there is a failure of the test case.
- We generally don't take screenshot for everything as there will be a memory issue.

## Steps to take a Screenshot :-

1. Create an object of Specific Brewhere class & upcast it to webDriver reference.
2. TypeCast the same upcasted object to TakesScreenshot Interface.
3. Using the TypeCasted object we call getScreenshotAs() which returns sourcefile object.
4. Using file handler class we are copying the source file into destination file .

// Package generates - Screenshot class

Class Screenshot implements ITestListener

{

    Public void onTestFailure(ITestResult result)

{

        TakesScreenshot ts = (TakesScreenshot) BaseTest.driver;

        File Screenshot = ts.getScreenshotAs(OutputType.FILE);

        File destfile = new File("./failedScreenshot/failed1.png");

        try {

            FileHandler.copy(Screenshot, destfile);

}

    Catch (Exception e)

{

        e.printStackTrace();

}

}

}

## Taking Multiple Screenshot :-

Class Screenshot implements ITestListener

{

    Public void onTestFailure(ITestResult result)

{

        TakesScreenshot ts = (TakesScreenshot) BaseTest.driver;

        File Screenshot = ts.getScreenshotAs(OutputType.FILE);

        String name = result.getName(); // fetch the test Method that got failed.

        Date date = Calendar.getInstance().getTime(); // fetch the date &

        String today = date.toString().replaceAll(":", "-"); // time of the failure

        File destfile = new File("./failedScreenshot/" + name + today + ".png");

```
trey
{
```

```
    FileHandler.Copy (srcfile, destfile);
}
```

```
Catch (Exception e)
```

```
{  
    e.printStackTrace();  
}
```

```
}
```

```
}
```

## Q. How to implement @Listeners at Class Level?

Ans // Package test

```
@Listeners(generics.Screenshot.class)
```

```
Public class ActitimeLoginLogout extends BaseTest
```

```
{
```

```
@Test
```

```
Public void loginLogout() {  
    tearDown();
```

```
{
```

```
    ActitimeLoginPage loginPage = new ActitimeLoginPage(driver);  
    loginPage.loginMethod();
```

```
}
```

```
}
```

## Q. How to implement @Listeners at Suite Level? // testing.xml file

Ans <Suite name="Suite">

```
<listeners>
```

```
    <listener class-name="generics.Screenshot"></listener>
```

```
</listeners>
```

```
<test name="Test">
```

```
    <classes>
```

```
        <class name="test.ActitimeLoginLogout"/>
```

```
</classes>
```

```
</test>
```

```
</Suite>
```

Dt- 25/03/2023

# MAVEN

LC - .m2

Java project

→ manually downloading

(a) Selenium jar

(b) testing

(c) browser drivers

(d) apache poi jar

→ manually added to the project

automatically it will get  
added to the project.

maven project

Create

directory structure

Project object model

pom.xml

automatically all the  
jar files & plugins

will get downloaded.

Src/main/java

Src/test/java

Developers to store  
their Source code.

Testing team

Maven  
Repository

add the  
dependency code  
and plugins

Maven  
Surefire plugins

→ MAVEN is a build tool which is used to manage and build our Automation Scripts.

→ MAVEN build life cycle has 8 stages :-

(i) Validation

(ii) Compilation

(iii) Testing

(iv) Packaging

(v) Integration Testing

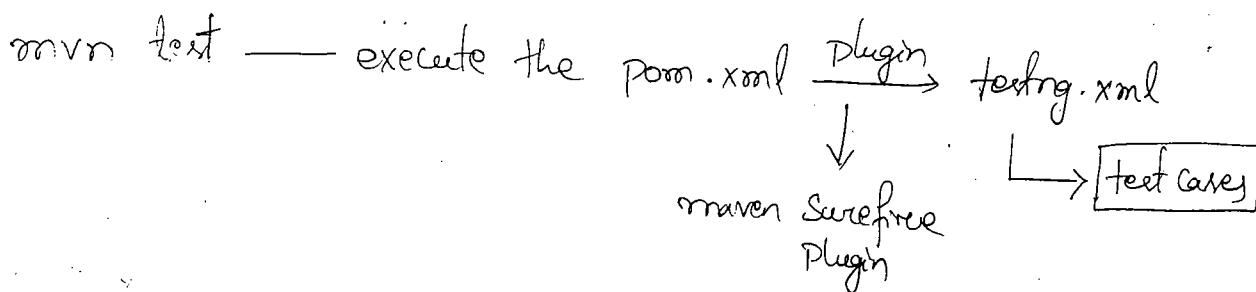
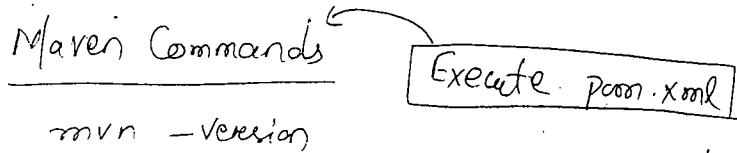
(vi) Verification

(vii) Installation

(viii) Deployment

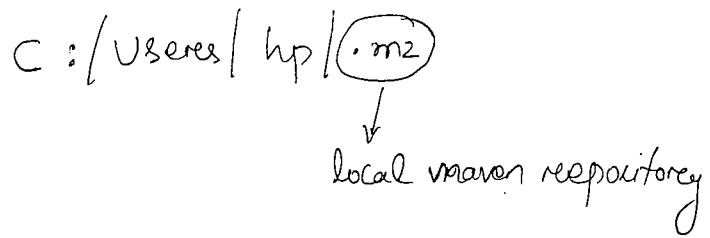
## Executing Maven Project in Command Line :-

1. Install MAVEN in your local machine.
2. Set the path of Maven in env. variables.



## MAVEN Commands :-

1. `mvn compile` - This command is used to compile the project source codes.
2. `mvn clean` - This command is used to clean or remove all the previous build generated files.
3. `mvn test` - This command is used to run the Test Cases of the project.
4. `mvn test-compile` - This command is used to compile the source codes present in `src/test/java`.
5. `mvn install` - This command is used to store the project jar file inside local repository.
6. `mvn package` - This command is used to create a package which contains all the project files.



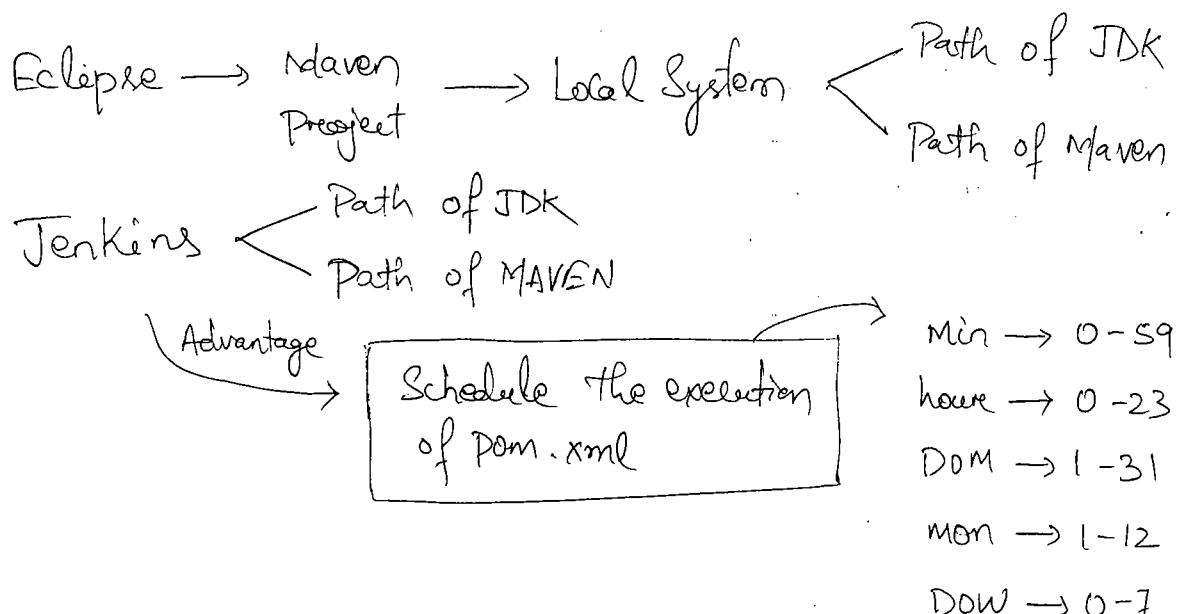
Dt - 28/03/2023

## Jenkins

Jenkins.io/download/

Port - 8080

access jenkins  $\xrightarrow{\text{URL}}$  localhost : 8080



Dt - 29/03/2023

- Jenkins is an Open-Source CI/CD Tools, which is used to implement CI/CD Pipeline.
- CI/CD stands for Continuous Integration & Continuous Deployment.
- A CI/CD pipeline is a series of steps that must be performed in order to deliver a new version of the s/w.

# GIT

## Commands of GIT :-

1. git version
2. git status
3. git init
4. git add .
5. git commit -m "Comments"
6. git config --global user.

# Git Hub

→ Github is an online repository which is used to push the source codes from local repository to remote repository.

## Steps to Push the Codes :-

1. Create an account in Github. ([github.com](https://github.com))

    └ Signup

2. Create a Repository in Github.

    └ Remote/Shared

    └ will have an URI

3. Download & install GitBash.

4. Right click on the project folder & Select GitBash Here.

5. Configure your username & email address.

git config --global user.name "SampreetiSahu"

git config --global user.email "sampreetsahu@gmail.com"

6. Initialize your git repository.

git init

7. Check the status of .git

git status

8. Add the files to local git repository.

git add .

9. Check the status

git status

~~enter email~~

Pub - Sampreeti@1234

UN - SampreetiSahu

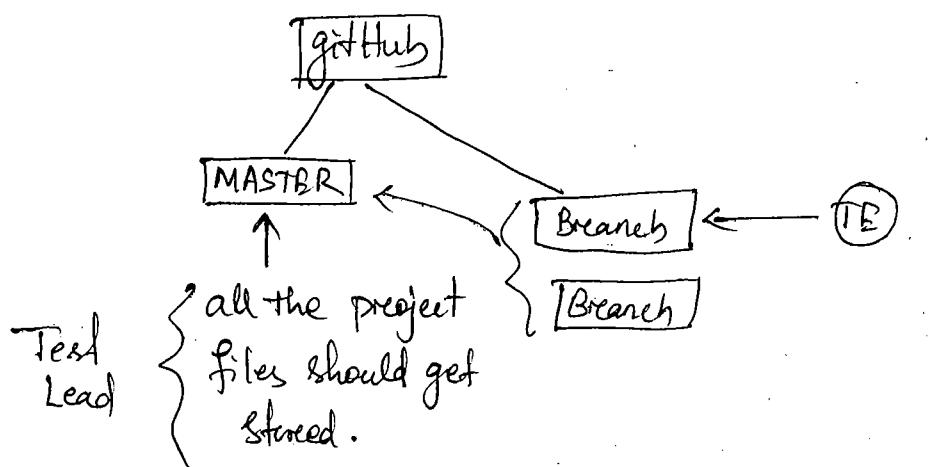
10. Commit the project files to remote repository.  
`git commit -m "message"`

11. Check the status

`git status`

12. Add the remote repository URL to push the codes.

`git remote add origin https://github.com/SampratiSahu/Activino.git`



13. Create a Branch.

`git branch Samprati`

`git branch`

`git checkout Samprati`

`git branch`

14. Push the project files to the respective branch.

`git push origin Samprati`

DT - 04/04/2023

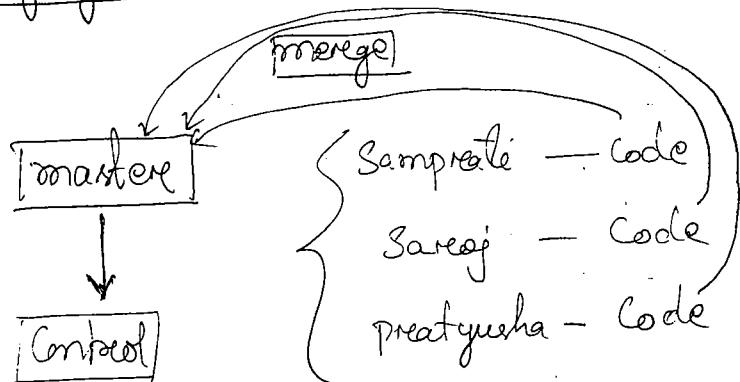
## Creating a new Branch & directly Switching to it :-

git checkout -b preatyusha

## Deleting a branch (Go to master / Go to other branch)

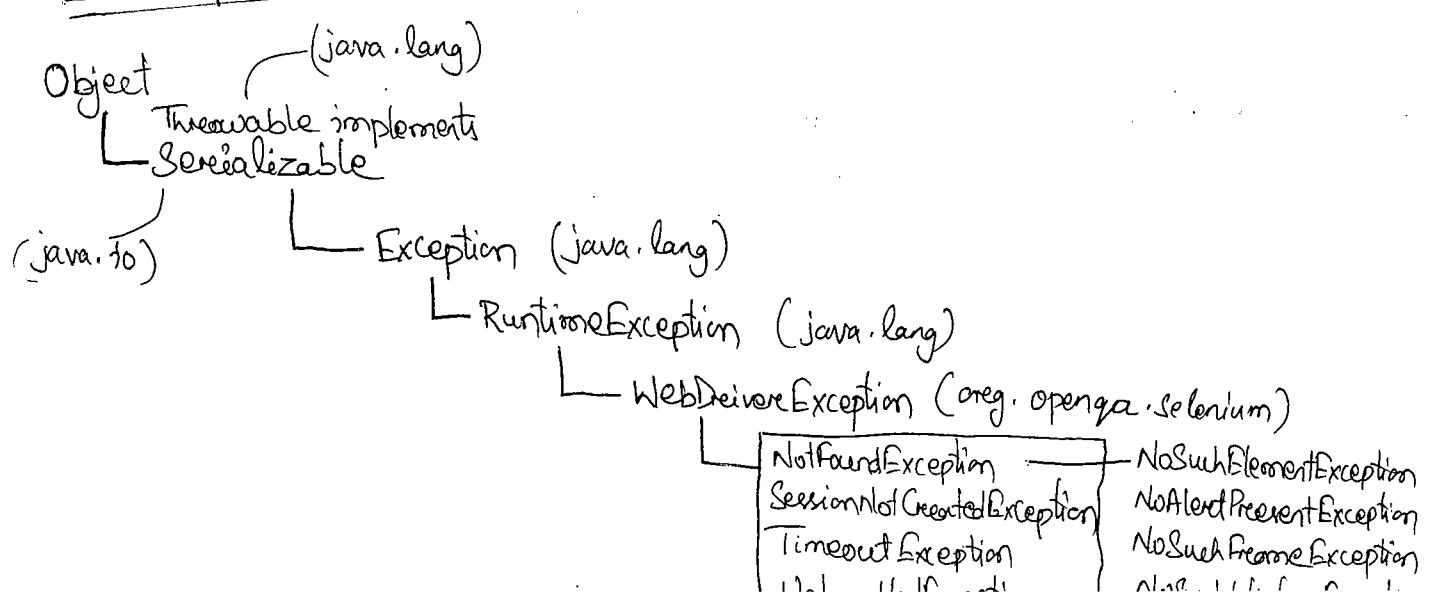
git branch -d preatyusha

## Merging two branches

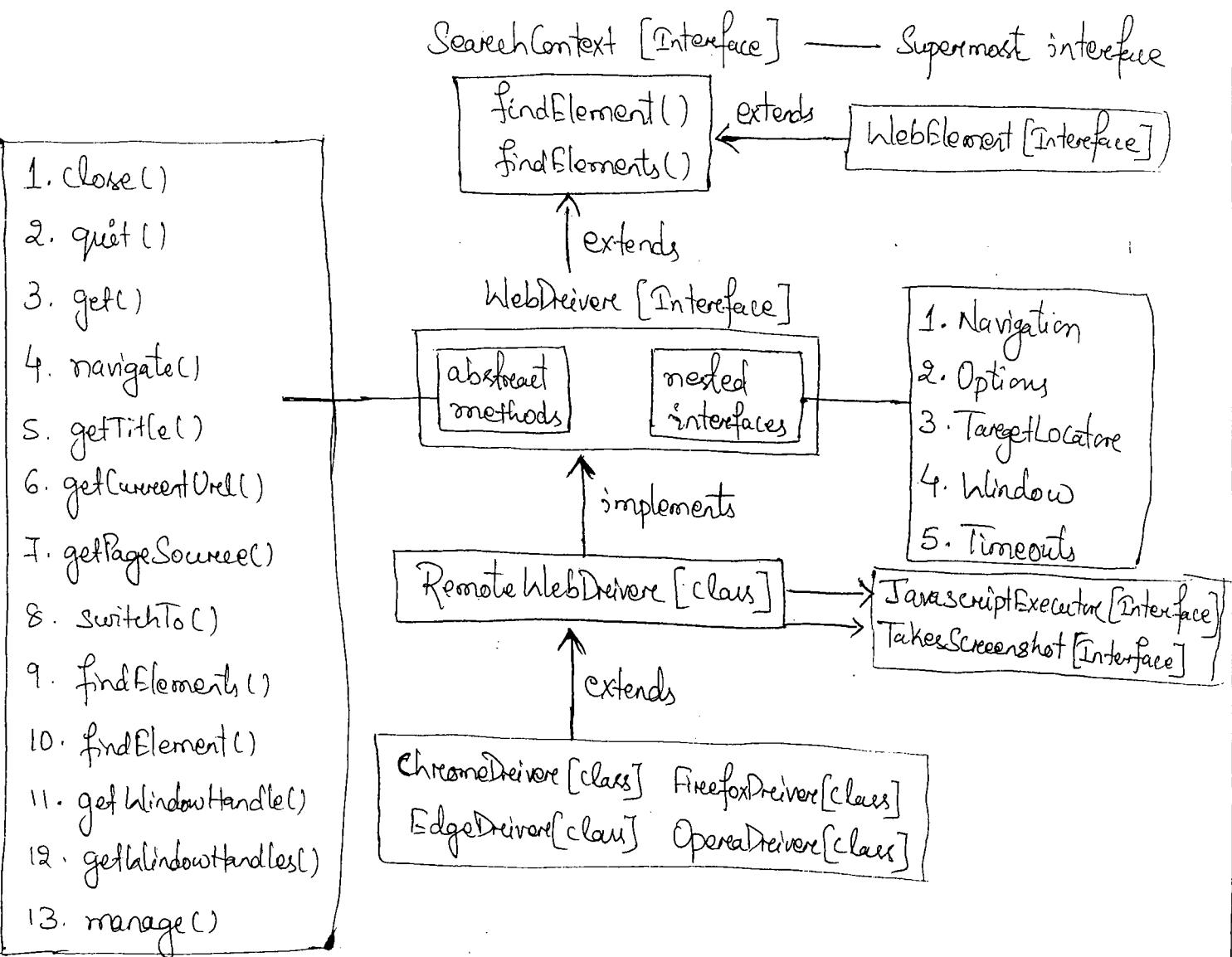


git merge Samprati

## Exception Hierarchy



# Architecture of Selenium WebDriver



# Selectors Hub