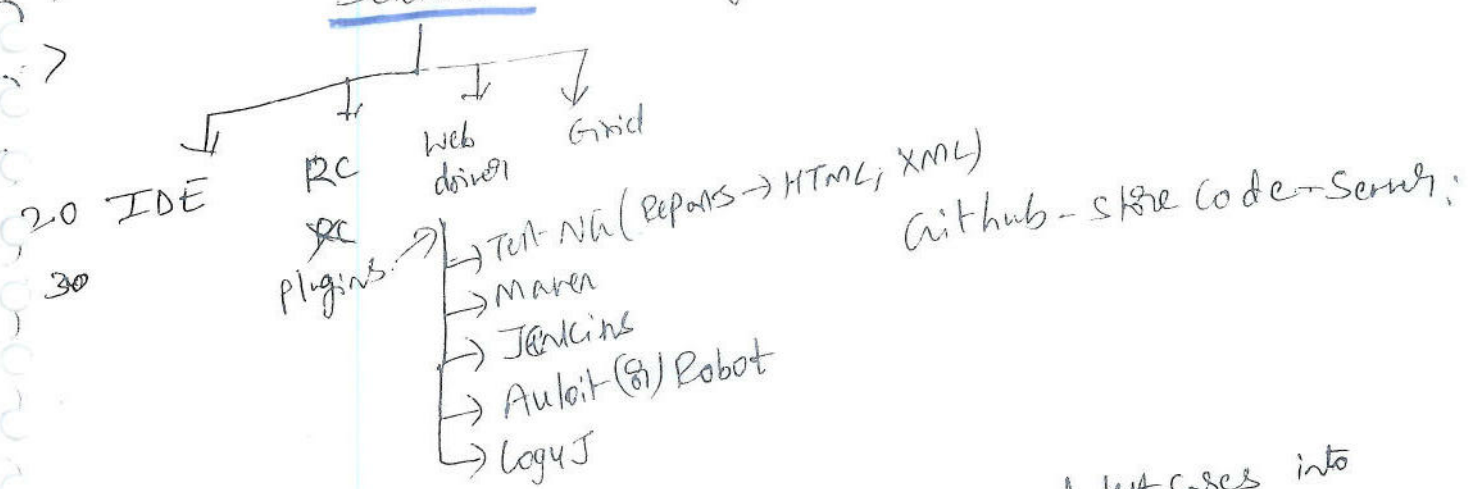
Selenium is a functional Testing tool
free/open source tool
Web based applications
Regression Testing

→ should provide both Actual & Expected Results

→ seleniumhq.org
→ licensed under APACHE 2.0
→ Selenium — Suite of tools ⇒ collection of Jar files



⇒

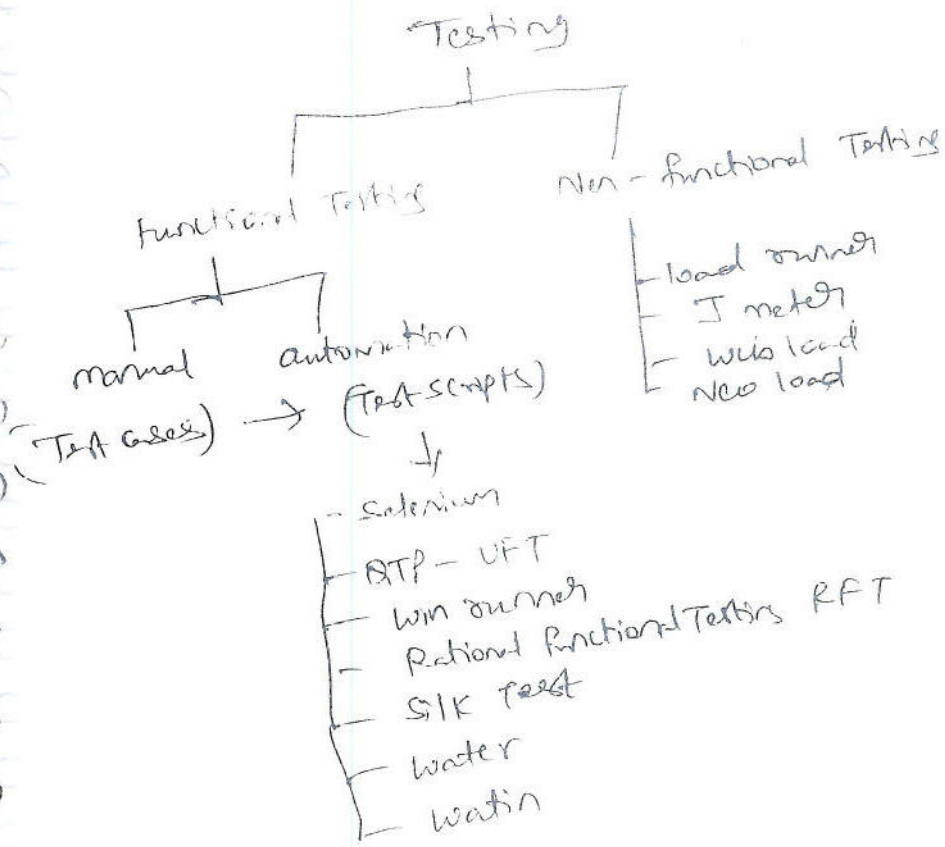
Automation :- It is a process of converting manual test cases into automated Test scripts Automation Testing is used to rerun test scenarios that were performed manually, quickly & repeatedly.

Apart from Regression testing automation testing is also used to test the application from load, Performance & stress point of view. It increases the test coverage improves accuracy & saves time & money in comparison to manual testing.

What is Automate?

→ It is not possible to automate everything in software. The areas at which a user can make transactions such as login form & registration forms, any area where large no. of users can access the software simultaneously should be automated. Furthermore, all GUI items connections with databases, field validations etc can be efficiently tested by automating the manual process

When to automate — Test automation should be used by considering the following aspects of software



Selenium	QTP
Open source	Commercial tool & costly
Java, Perl, PHP, ruby, Python, C#, html	VB script
Firefox, IE, chrome, Safari, opera	IE, Firefox, chrome
Windows, Mac, Linux	Windows
flexible & extentable	flexible & extentable less
Web apps & mobile apps	Desktop, web, mobile apps

Note: Selenium is for web + mobile apps. If any desktop Components & fields available in the application can handled with the help of SIKULI or AUTOIT.

→ Selenium 3.0

It is a open source tool.

Founder of selenium is Jason Huggins

It supports only web applications across diff. platforms & browsers.

Selenium has the below components

Press B – to insert/remove break point

→ Converting IDE scripts into other technologies

Go to File menu

Select Export test case as Java/testng/webdriver — Specific test case

Export test suite as " — Set of test cases

Target - IDE - webdriver by Ruby/xpath

→ Sel. RC = RC → Remote control.

It is selenium 1 & introduced in 2004 by Jason Huggins.

It is a server which is responsible to launch all the browsers with the help of third party RC servers.

RC has been merged into webdriver.

Sel. Webdriver:

- founder of webdriver is Simon Stewards.
- It is a functional testing tool which is used to automate only web apps (applications) but not desktop apps.
- Webdriver will not depends on 3rd party servers.
- Webdriver will launch application browsers from OS level.
- Limitations which are identified in RC are overcome in webdriver & also added advantages
- We can automate any app using webdriver.
- Webdriver supports programming languages like JAVA, .NET, PHP, HTML, C#, etc RUBY
- Webdriver supports invisible or headless browsers which is called as HTML Unit Browsers.
- This browser is used only for GUI testing like printing title of the page, (Direct URL, current Page source etc.
- This browser is fastest browser in executing because there is no loading time for browsers & URL
- We should not use this browser for validation.
- Using webdriver we can enchance our test scripts with help of programming languages like Looping, Conditions etc.
- Using w/driver we can perform parameterization (checking with multiple data) as well as database testing by using JDBC ~~testing~~ Java database connectivity.

(5) ✓

Downloading Selenium Standalone server Jar file:

It is a Common Jar file which is used for designing webdriver code, RC & Selenium GRID scripts.

Go to seleniumhq.org → click on download → for Selenium Standalone server click on " version 3.7.1 (which is latest)
This version no. will keep on updating

After downloading copy & paste this Jar file to library folder in automation framework folder.

Opening Eclipse:-

Double click on main folder (Automation Framework)
" " eclipse folder
" " (icon blue)

Give the path of main folder → ok.
click on workbench.

* Creating Java Project:

Go to File menu → New → select project → Java Project → Next →
Project name → (Health_Care) → Finish →

Adding Jar file to Project:

Expand ur project → Right click on JRE system Library → Build path → Configure Build path → click on External Jars button under libraries
~~click on~~ add Browse Jar files from main folder → Library (AFW)

click OK → OK.

Under Referenced Library make sure ur Jar file is added.
For every project need to add above Ref. Library (follow above)

Package: Package is a Collection of Java classes.
Packages will be created under SRC folder.
If no package name is given for a class the class will be stored into default package.

- Class:-
Class is a blue print of templates which contains objects.
Object is a Collection of bundles of properties & methods.
- Object can be anything which should be physically available.
In Java Object is nothing but data type / method / Instance.

- Public static void main:
It is a Main method which is responsible for executing class.
If a class does not have main method it cannot be executed.

String is a predefined class in Java which is used to store String in variables.

String Str = "Hello";

Non primitive / Reference data types:

Objects & Arrays.

Button a = new Button("ok")

" " " " in Java are

int a[] = new int[ ];

Variables:

A variable is a named memory location which can hold the program data.

Variables can be stored in 2 ways:

1. Primary (RAM)
2. Secondary (DVD, CD, USB etc) memory.

In general variables will be stored in to RAM.

Declaring variables:

datatype variable name; eg: int i; int a, b, c;

datatype variable name = value;

eg: int i = 10;

How to assign values into variables

→

Naming Restrictions: Conventions

JAVA variables are case sensitive, monkey is not the as Monkey & MONKEY

char monkey, Monkey, MONKEY;

Java variable name must start with a letter or $ or _

ex:

myvar

MYVAR

$myvar

_myvar

myvar1

myvar_1

1myvar - invalid.

variable names cannot be equal to java reserved words. (pre defined keywords)

ex: int, for, import.

> Must be unique in the scope of declaration

> Should not exceed 255 characters.

type
sysO = ctrl spacebar click enter

Ctrl + F11 → To run the class or right click mouse run as Java application

→ Type SYSO ctrl + spacebar + enter.

\t = tab space
\n = newline - (in

(3) Comparison / Relational operators :-

1) == equal to 2) != not equal to 3) > greater 4) >= greater equals to
5) < less than
This operators are used to compare 2 values & return either True or false. 6) <= less than equals

(4) Logical operators :-

1) Logical Not operator (!) — (true = false = true) — it uses (OR) operator. { TT = F, FT = T, TF = T, FF = T }
2) " AND " (&&)
3) " Or " (||)

Looping Statements :- This are used when u want to execute block of statement for several times. In Java there are 4 types :-

1) for loop 2) while loop 3) do... while loop 4) Enhanced for loop. (foreach)

1) For Loop : It repeats block of stents for a specified no. of times.

Syntax :-

```
for (strat value; end value; increment/decrement) {
    Statements
    ---
    ---
}
```

1)
```
for (int i=1; i<=10; i++)
{
    system.out.println(i);
}
```

2)
```
for (int i=1; i<=10; i+=4)
{
    syso (i);
}
```

3)
```
for (int i=10; i>=0; i--)
{
    syso (i);
}
```

4)
```
for (int i=10; i>=0; i-=3)
{
    syso (i);
}
```

→ String weekdays [] = {"mon", "tue", "wed", "thu", "fri", "sat", "sun"};

```
syso (weekdays [4]);
for (string eachday : weekdays)
{
syso (eachday);
}
```

(or)

```
for (int i = 0; i < weekdays.length; i++) {
    Syso (weekdays [i]);
  }
}
```

string weekdys [] = {"mon, tue, wed, ...}

for (string eachday : wee...

→ CONDITIONAL STATEMENTS:-

There are two " " in JAVA.

1) if
2) Switch statement

1) if :- checking whether given condition is True & False.

syso = s.o.p

```
if (condition)
a) syso ("True");

b) if (condition) {
   sop ("true");
   }
   else {
   sop ("false");
   }
```

To uppercase - It converts lower case to upper case

JAGANATH P

```
String str1 = "Hello";
Syso(str1.toUpperCase());
```

To lowercase - It converts to upper case to lower case

```
String str2 = "HELLO";
Syso(str2.toLowerCase());
```

```
Scanner obj = new Scanner(System.in);
for(int i=1; i<=5; i++){
System.out.println("enter course");
String course = obj.nextLine();
switch(course.toUpperCase()){
case "MANUAL";
Syso("yes it is available");
break;
case "QTP";
Syso("yes it is available");
break;
case "RPA", Not available
Syso("Contact Manager");
break;
case "JAVA";
Syso("hyt available");
break;
default:
Syso("check in gfiu");
break;
}}}.
```

ARRAYS: Array is a special type of variable which can store series of values of same data type. We assign values into array by using index number. There are 2 types of arrays.

1) Single Dimensional array
2) Multi "

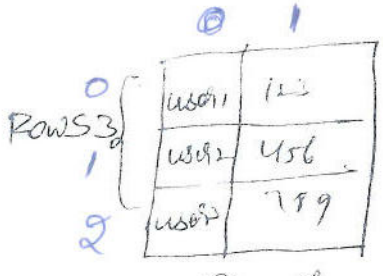
1) Single Dim. Arr: which can store single value of same data type.

Syntax: datatype [] array name = new datatype [size]; (or)

datatype array name [] = "

```
sysо(obj);
}
```

Two dimension Array:-



3*2 = 6

```
object str[][] = new object[3][2];
str[0][0] = "user1";   -> 1st row 1st column
str[0][1] = 123;       -> "  2nd colum
str[1][0] = "user2";   -> 2nd row 1st col
str[1][1] = 456;       -> 2nd row 2nd col
str[2][0] = "user3";
str[2][1] = 789;
syso("length of row:;"+str.length);
syso("length of Columns:;"+str[0].length);

// iterate all rows
for(int i=0; i<str.length; i++)
{
//iterate all columns
for(int j=0; j<str[i].length; j++)
{ syso("\t"+str[i][j]);
}
syso(i);
}
```

strong length of rows

```
-> int rows = str.length;
int cols = str[0].length;
Storing length of columns in first row.
for(int i=0; i<rows; i++)
{
for(int j=0; j<cols; j++)
```

Keyword - for, str, try, catch
class - should start with caps
Syso - S
Throw - T

EXCEPTION HANDLING IN JAVA:-

While developing the Java program there are some chances that the code which I have written can't be executed then such type of situation is known as exception. Here we have 2 types of exceptions:

-> Called exception (checked exception)
-> Unchecked " (unchecked ")

Second block of code

int a = 20, b = 5, c;
c = b + a;
sysl(c);
}

(*) Exception Handling automatically

Throws :- Throws is used for automatically throwing the exception whenever u don't want something specially to happen whenever exception occurs during run time instead of using try catch block it is better to use throws. Here the main reason is throws class can handle the same exception any no. of times in the program.

public static void main(String[] args) throws InterruptedException
sysl ("I am sleeping for 5sec");
Thread.sleep(5000);
sysl ("i am back");
sysl ("i am sleeping");

Thread.sleep() → Thread.sleep(5000); → sysl ("i am back"); }

Thread.sleep : It is a fixed waiting time before both taking tool & appn.

Syntax : Thread.sleep(milliseconds);

Throw :- It is used for throwing an error explicitly by the program. It is used to throw user defined messages if any error occurs.

STRING FUNCTIONS :-

Equals :- To compare 2 strings eg :- String str1 = "Hello"; friends
String str2 = "Hello";

① sysl (str1.equals(str2));
② sysl (str1.contains(str2));
③ sysl (str1.matches(str2)); It works on regular expression
④ syso(str1.equalsIgnoreCase(str2))

str1 = "hello";
str2 = ".*e.*";
True <= str1.matches(str2);
False <= str1.equals(str2);

Syntax
variable.length();

* ⑤ Length -- Returns length of the string

String str = "I will practise";
sysl (str.length());

⑥ Trim -- It removes white spaces on both left hand side & Right-HS

String str = " i am learning ";
sysl (str.trim());

```
String st1 = "Welcome to selenium world");
sys1 ("" + st1.substring(11,19)); // selenium
sys1 ("" + st1.substring(0,8)); // Welcome
```

(11) Substring :-

METHOD :-

A method is a group of statements which can perform some action or operation whenever they are called under main method. Code written inside a method cannot be executed by itself. To execute that method code block U need to call that method inside main method block.

Access modifiers in Java :-

It are keywords in Java which can be set the levels of access for class, methods, variables & constructors.

In Java there are 4 types:

1) Public :- We can access public methods or variables from all class of same package or other package.

2) Private :- private methods & variables can be accessed only from same class. We cannot access it from other classes or sub classes of even same package.

3) Protected :- protected methods can be accessed from classes of same package or sub classes of that class

4) No access modifier :- If method have not any access modifier then V can access it inside all class of same package only.

```
public class accessing {                  → static method
public static void add() {
  }
public static void mull() {

  }
  Public static void div() {
  }
              Public static void main(String) {
                              accessing.mull();
              }
  }
```

```
Public static void main (string [ ] args) { //main method
  // calling static methods
  static_method.add(23,3);
  static_method.div(23,4);
  static_method.sub();
  // creating object to call non static methods
  static_method obj = new static_method();
  obj.sub();
  }
}
```

both

we can access ↑ static & non static stuff

under	Non static method

Under static method, we can access only static method.

* we cannot access non static method in static method.

Return Types :-

You can use 2 kind of return types with methods.

1. Void
2. Any data types.

Void means returning nothing & when ur method is returning some values like integer, string, double etc you need to provide return type with method according to returning values data types.

— int, string, double etc Data types - If method is returning any value then u need to provide return type with method ie int, string, double etc.

— Void - If method is not returning any value then u can give void as return type. Void means method returning nothing.

```
— package Java-examples;
  Public class Return-type {
    Static int z;
    static double c;
  Public static void message() {
    sysout ("message");
  }
  Public static int add(int x, int y) {
    z=x+y;
    return z;
  }
```

```
Public static double div(int a, int b)
  { return c=a/b;
  }
Public static void main(string[] args){
  Return_type.message();
  add(23,43);
  div(20,40);
  sysout(c);
  sysout(z);
  } }
```

```
public void printdetails () {
// can access class variables of parent class directly inside child
sysout ("wheels of Audi =" +wheels);
    ("speed of Audi =" +speed);
// can access non static methods of parent class directly inside the
   seats();   //it is available in parent car class
// can not access private variable of parent class in child class
  // sysout ("wheels of Audi =" +type);
}
  public static void main (String [] args) {
      Audi A = new Audi();
      A.printdetails();
// can access instance variable of parent class using
// object reference of child class Inside static methods
      sysout ("color of Audi =" +A.color);
// can access non static method of parent class using object
// reference of child class Inside static methods.
      sysout ("Fuel of Audi =" +A.getfuel());
  }
}
```

Webdriver :-

Interface - It is a set of rules defined by us which has only declaration of functions, nobody of the function & it will force us to override all the functions in the class which implements this interface.

Syntax :- [modifier] interface [interface name {
[modifier] [Return type] method name1();
" " " 2();
}

** After downloading all drivers extract them into our project folder.
Health care project.

Downloading Addons to Firefox

elements = text box = objects — buttons (Login)

Firebug! It is addons to only firefox browser. which means first we need to download in firefox, install in firefox, install bload

Using firebug we can inspect ID, name, class name, locator values for any element in a webpage.

open firefox browser. enter below url

https://addons.mozilla.org/en-us/firefox/addon/firebug

click on Add to Firefox

click Install

use only firefox 47-2.3.1 version. Take

How to Inspect elements using firebug! → locator.

- Select & open an application
- Select one element in firefox — facebook, gmail.
- Right click mouse on element — Email/phone.
- Select Inspect element with firebug.
- In HTML tab values to be copied.
- Select property value & paste to eclipse.

* Property value is same for chrome/firefox/web browser.

NOTE!- using firebug we can inspect ID value, Name, class Name, Class property values.

→ Downloading Firepath!- using firepath we can inspect- XPath & CSS selector values

(For inspecting XPath & CSS selectors)

https://addons.mozilla.org/en-us/firefox/addon/firepath.

Find Elements() method

- First of all I don't use it frequently as its usage is very limited
- It gives back the whole list of all the elements matching the specified locator.
- If the element doesn't exist or not available on the page then the return value will be an empty list.

Its syntax is as follows:

```
List<WebElement> linkList = driver.findElements(By.tagName("a"));
```

Locator:-

locator is used for identifying elements in a web page.

There are 8 types of locators

Variation	Description	Sample
1) By.ID		
2) By.className	finds elements based on the value of the 'class' attribute	findElement(By.className(""))
3) By.cssSelector	finds elements based on the drivers underlying CSS Selector engine	findElement(By.cssSelector(""))
4) By.id	locates elements by the value of their "id" attribute	findElement(By.id(""))
5) By.linkText	finds a link element by the exact text it displays	findElement(By.linkText("REGISTRATION"))
By.name	locates elements by the value of the "name" attribute	findElement(By.name(""))

XPath :-

XPath is designed to allow the navigation of XML documents with the purpose of selecting individual elements, attributes, or some other part of an XML document for specific processing.

What is XML?

The Extensible Markup Language (XML) is the context in which the XML path language, XPath, exists. XML provides a standard syntax for the markup of data & documents.

XML documents contain 1 or more elements. If an element contains content, whether other elements or text then it must have a start tag and an end tag. The text contained betn the start tag & end tag is the elements content.

```
<Element >//start tag
  Element Content goes here // Element content
  </Element >// End tag
```

An element may have 1 or more attributes which will provide additional information about the element type & its content.

Sample XML :-

```
<?XML version='1.0'?>
<Catalog>
  <Book>
  <Title>Selenium Classes</Title>
  <author> Jafer </Author>
  </Book>
  </Catalog>
```

XPath can be viewed as a way to navigate around XML docs. Sit as street directions

Absolute XPath :-

// Starts with the root node or a forward slash (/).

The advantage of using absolute is it identifies the element very fast.

Disadvantage of this is, if anything goes wrong in some other tag

(3) getCurrentURL command:

Purpose:- This command is use to get the URL of the page currently loaded in the browser.

driver.getCurrentUrl();

(4) Get page source command:

P! This command is use to get the source of the last loaded page.

driver.getPageSource();

(5) close command:-

This " is use to close the cussrt window of the browser. if its the last window it will close the browser.

driver.close();

(6) Quit command:-

This " is use to quit the browser & all the opened windows in the browser.

driver.quit();

Ex:-

```
Package Webdriver_Examples;
* import org.openqa.selenium.*;        * it will cover multiple drivers of selenium like webdriver, sy*ck
import org.openqa.selenium.chrome.ChromeDriver;
Public class Chrome_Browser Example {
Public static void main (String [] args) throws InterruptedException {
    // create Instance object
Webdriver driver = new ChromeDriver();
    // launch url in a browser
driver.get("http://orangehrm.gedgetech.com/symfony/web/index.php/auth/login");
sysOL("First page title [1]:!" + driver.getTitle());
    " (" " url [1]:!" + driver.getCurrentUrl());
```

```
Sysout("current Url length is: " + obj.getCurrentUrl().length());
Sysout("------");
Thread.sleep(5000);
Sysout(" obj.getPageSource());
Sysout(" obj.getPageSource().length);

}
Public static void main(String[] args) throws InterruptedException {
  HtmlUnitDriver h = new HtmlUnitDriver();
  h.verifyGuiTests();

  }

}
```

SendKeys() — Void type — it does not return any value

sendKeys Command:

This method is used to send value in a textbox, list-box etc.

```
WebElement user = driver.findElement(By.id("user"));
User.sendKeys("Tech Beamers");
```

or follow below coding style

```
driver.findElement(By.id("user")).sendKeys("testing");
```

Web Element:-

It is a class which can store elements in a webpage.
It is a Java class which represents an object that holds the reference to an HTML element.

clear Command:-

void clear() - This method will clear the value of any text type element. It clears the existing text in a text box.

```
Public void verify_Protocol(){
String currenturl = driver.getCurrentUrl();
if(currenturl.contains("https://"))
{
Sysout("protocol is existing");
}else{
  sysout("protocol is not existing");
}}
Public void verifyTitleinSource(){
  String pagesource = driver.getPageSource();
  if(pageSource.contains("GMAIL"))
{ sysout("title is available in page source");
}
else{
  sysout("title is not available in page source");
  }}
public void Logout() throws Interrupted Exception {
  Thread.sleep(3000);
  driver.close();
}
public static void main(String[] args) throws Interrupted Exception{
  Get_Commands g = new Get_Commands();
  g.launch("http://gmail.com");
                 or FB
  g.verify_Get title();
  g.    "    _protocol();
  g.verifyTitleinSource();
  g.logout();
  }}
```

String attributevalue = driver.findElement(By.name("selenium")).getAttribute("value");

S.O.P.l("Available attribute value is:" + attributeValue);
(Sysout)

o/p — value selenium

(3)

getTagName();

It returns tag value of an Element in a webpage. like Textbox, Listbox, Radio button, links etc.

String text = driver.findElement(By.id("")).getTagName();

↓
variable.

(4)

getSize Command:

Dimension getSize() - This method returns the height & width of the given element. It doesn't allow to pass ~~any~~ any parameter but its return type is the Dimension object.

Command - element.getsize();

This method provides the size of the element on the webpage.

WebElement user = driver.findElement(By.id("user"));

Dimension dim = user.getsize();

SOP("height" + dim.height + "width" + dim.width)

(5)

GetLocation Command:-

Point GetLocation() - This method locate the location of the element on the page. It doesn't allow to pass any parameter but its return type is the Point object.

Command - element.getLocation();

This method provides the Point class object. U can easily find x & y Cordinates of a specific element.

```
Sopl(" text box value name is :: "+textbox);
String listbox = dr.findElement(By.id("day")).getTagName();
Sopl("list box value is :: "+ list box);
String linktag = dr.findElement(By.partialLinkText("Ter")).get
                                                    TagName();
Sopl(" link tag value is :: "+ link tag);
String button = dr.findElement(By.id("u-0-Y")).getTagName();
Sopl("button value name is :: "+ button);
String image = dr.findElement(By.XPath(".// * give xpath).get
                                                    TagName();
Sopl(" image value name is :: "+image);
Sopl(" -- -- ");
}
Public void verify - Getsize(){
WebElement user = dr.findElement(By.id("u-0-9"));
Dimension dim = user.getSize();
Sopl(" Height #" + dim.height + "width #" + dim.width);
dr.quit();
}
Public static void main(String[] args){
Get_Runtime Methods g = new Get_Runtime methods();
g.launch();
g.verify -Getetxt();
g.verify Get_attribute();
g.verify_Get-Size();
}
}
```

Navigate To Command

Void to (string arg 0) - This method launches a new web page in the browser window. It allows to apass a string parameter, & its return type is a void.

Command - driver.navigate().to(site URL);

It is almost similar to the driver.get(site URL) method. Here site URL is the website URL to open. It is the best practice to use a fully qualified string address.

```
driver.navigate().to(" ");
```
(1)

Forward Command

Void forward() - This method stimulates the browser's forward button action. It doesn't allow any parameter & its Return type is void.

Command - driver.navigate().forward();

It moves you forward by a single page into browsers history data.

```
driver.navigate().forward();
```
(2)

Back command!

Void Back() - This method simulates the browser back button action. It doesn't allow any parameter & its return type is void.

Command - driver.navigate().back();

It moves u back by a single page into the browsers history data.

```
driver.navigate().back();
```
(3)

Refresh Command!

Void refresh(): This method simulates the browsers refresh button action. It doesn't allow any parameter & its return type is void.

Command - driver.navigate().refresh();

It triggers the same action as does the F5.

```
driver.navigate().refresh();
```
(4)

(F3)

Implicitly Wait Commands :

Selenium Webdriver has borrowed the idea of implict waits from waitr. This means that v can tell Selenium that v would like it to wait for a certain amount of time before throwing an exception that it cannot find the element on the page. v should note that implicit waits will be in place for the entire time the browser is open. This means that any search for elements on the page could take the time the implict waits is set for.

driver.manage().timeouts().implicitly Wait (time in seconds, TimeUnit.SECONDS);

package is

import java.util.concurrent.TimeUnit;

// Write a script to check the borken links in the webpage.
checking which link is ok(available) which is not found in server.

'href' – is a property name which returns URL of the link

//getResponse code – It returns 200 – If the link is existing
" " 401 – If the " unauthorized
" " 404 – " " " not found.

~~Gets the~~

~~***~~

```
Package webdriver_Examples;
import java.net.HttpURLConnection;
import java.net.URL;
"      java.util.List;
"      "  .concurrent.TimeUnit;
import.org.openqa.selenium.*;
import.org.openqa.selenium.chrome.ChromeDriver;
Public class BrokenLinks {
Public static Void main(String[] args) throws InterruptedException {
// Instantiating chrome ~~Firefox~~ Driver
Webdriver driver = new ChromeDriver();
```

```
// The below function verifyLink(String urlLink) verifies any
   broken // Links & return the server status.
public static void verifyLink(String urlLink){
// Sometimes V may face exception "java.net.malformed
                                    URL Exception",
// Keep the code in try Catch block to continue the broken link
   analysis
try{
// use URL class - create object of the URL class & pass the
      //urlLink as parameter //
URL link = new URL(urlLink);
// create a connection using URL objects (ie. link)
HttpURLConnection httpConn = (HttpURLConnection) link.
                                    openConnection();
// set the timeout for 2 seconds
httpconn.setConnectTimeout(2000);
// connect using connect method
   httpconn.connect();
// use getResponseCode() to get the response code
if(httpConn.getResponseCode()==200){
   Sopl(urlLink+"_"+httpconn.getResponseMessage());
   }
   if(httpConn.getResponseCode()==404){
      Sopl(urlLink+"_"+httpconn.getResponseMessage());
   }}
// getResponseCode Method Returns = IOException - if an error occured
                                    connecting to the server //
- catch(Exception e){ }}}
```

200 ⇒ OK
401 ⇒ unauthenticated
404 ⇒ Not found

5) Using AND :- both property values should be matched

Syntax:

//tagname [@ attribute_name1 = 'attribute_value1' and @ attribute_name2 = 'attribute_value2']

//* [@ " = " " "

6) Using OR :- any one attribute value should match

Syntax:

//tagname [@ attribute_name1 = 'attribute_value1' or @ attribute_name2 = 'attribute_value2']

(or)

//* [@ " = " (or) " "

7) Contains() :- It is used to identify an element when we are familiar with some part of the attributes value of an element.

Syntax:-

//<HTML tag> [

(or)

//tagname [contains (@ attribute_name, 'attribute_value')]

(or)

//* ["]

8) starts_with() :- It is used to identify an element when we are familiar with the attributes value (starting with the specified text) of an element

syntax:

//tagname [starts_with (@ attribute_name, 'attribute_value')]

or

//* ["]

To identify the input field of type text after the FirstName field we need to use the below Xpath

//*[@id='First Name']/following::input[@type='text']

To identify just the input field after the firstname field we need to use the below Xpath

//*[@id='First Name']/following::input

preceding: selects all nodes that appear before the current node in the document, except ancestors, attribute nodes & namespace nodes

Xpath of the LastName field is as follows

//*[@id='Last Name']

To identify the input field of type text before the Last Name field we need to use the below Xpath.

//*[@id='Last Name']//preceding::input[@type='text']"

```
Package WebDriver_Examples;
* import org.openqa.selenium.By;
Public class Xpath_customise{
public static void main (String[] args) throws InterruptedException {
WebDriver driver = new ChromeDriver();
driver.navigate().to("file:///E:/LoginPage.html");
                    www.google.com
driver.manage().window().maximize();
// with single property
//driver.findElement(By.xpath("//input[@name='username']")).
                                    sendKeys("admin");
// with start with
driver.findElement(By.xpath("//*[starts-with(@name,'usernam')]")).
                                    sendKeys("admin");
```

// or write the code in the below style.

boolean is_displayed = driver.findElement(By.id("user")).isDisplayed();

Is Enabled Command :-

boolean isEnabled() - This method return true false depending on the state of the element (Enabled or not).

This method will usually return true for all items except those which are intentionally disabled.

WebElement user = driver.findElement(By.id("user"));
boolean is_enabled = user.isEnabled();

// or write the code in the below style

boolean is_enabled = driver.findElement(By.id("user")).isEnabled();

Is selected Command :-

boolean isSelected() - This method tests if the element is active or selected. It doesn't allow any parameter. but it does return a boolean status.

This method is only applicable for input elements like checkboxes, Radio Button & Select options. It'll return true when it finds the element is currently selected or checked.

WebElement lang = driver.findElement(By.id("Language"));
boolean is_selected = lang.isSelected();

// & u can rewrite the code as below

boolean is_selected = driver.findElement(By.id("Language")).isSelected();

package Webdriver_Examples;
import java.util.Scanner;
public class Conditions_Methods {
public static Webdriver driver;
public void Verify_Login() throws InterruptedException {

```
WebElement login = driver.findElement(By.xpath("//*[@name='Proceed']"));
boolean go_Displayed = login.isDisplayed();
S.o.p.l(go_Displayed);
log.n.click();
Thread.sleep(4000);
String message = driver.findElement(By.xpath("//*[@'login']/
                                    div[2]/div[1]");
S.o.P.l(message);
Thread.sleep(4K);
driver.quit();
}}
Public static void main(string[] args) throws InterruptedException{
Conditions_Methods m= new Conditions_Methods();
m.verify_Login();
}}
```

Copy xpath

// Working with list box & Dropdown Methods :-

Wherever any list box or dropdown has Select tag we need to create one object for select class.

List-Single selection
Select-class.
dropdown-multiple selection

Selecting Methods applicable for both dropdown & List box.

SelectbyVisibleText :

```
Select mydropdown = new
→ Select(driver.findElement(By.id("carlist")));
  mydropdown.selectByVisibleText("Audi");
```

(class, variable)

- It will select value from dropdown list using visible text value = "Audi"

Sent Gmail - 2 scripts
- examples - Listbox.

TestNG (Test Next Generation):

How to install TestNG :-

- Go to Help menu in Eclipse
- Select Eclipse Marketplace
- In find Text box type - testing click enter - under search tab
- click Install - TestNG for Eclipse
- click confirm
- Accept terms & conditions
- click finish.
- click ok to security loaning window.
- click yes to restart ur eclipse.

Adding TestNG Library for the project :-

- Right click on existing project
- Go to properties
- click on Java Build path
- Click on Libraries
- " Add Library
- Select TestNG
- click next
- " finish
- " OK

Note; U can create no. of projects in Eclipse & there is no need to Install TestNG. U need to Configure only TestNG Library.

> Annotations Available in TestNG

@ Before Test (5)
@ After Test (6)
@ Before class (3)
@ After class (4)
@ Before Method (1)
@ After Method (2)
@ Before Suite (7)
@ After " (8)
@ Before Groups (9)
@ After " (10)
@ Test (0)
@ Data provider (11)
@ parameters (12)
@ factory (13)
@ listeners. (14)

@ Before - Pre conditions - (Launch)
@ After - post conditions - (Logout)

@ Test - Marks a class or a method as a part of the Test

@ Before Method - A method which is marked with this annotation will be executed by every @ test annotated method

@ After Method - A method which is marked with this annotation will be executed after every @ test annotated method

@ Before class - A method which is marked with this annotation will be executed before first @ Test method execution. It runs only once per class.

@ After class - A method which is marked with this annotation will be executed after first @ Test method " " " "

@ Before Test - A " " " " " "
" before first @ Test annotated method

@ After Test - A " " " " "
" when all @ Test annotated methods completes the execution of those classes which are inside <test> tag in testng.xml file.

```
public void compose() {
Reporter.log("running compose", true);
}
@Test
public void reply() {
Reporter.log("running reply", true);
}
@after.Method
public void logout() {
Reporter.log("running logout", true);
}
}
```

① launch()
② compose()
③ logout()
④ launch()
⑤ reply()
⑥ logout()

Executing TestNG class :-

- Right click the mouse - Select Runas - TestNG test
- After executing TestNG class - Right click on Project - Refresh make sure you get 'test-output' folder.

Analysing HTML & XML Reports :-

Expand 'Test-output' folder

- Right click on ① Emailable report.html :
  - open with web browser.
  - View the report
- ② Index.html : Right click - open with web browser
- Expand Default Suite - Right click on ③ Default test.html - open with web browser
- View results in ④ testng-results.XML = It open in Design tab
  View in source tab. - check in source

NOTE: U can view these reports in local system under ur project

@ Test (Description = "   ", priority = 2, enabled = true)

Package Testing_ examples;

* import org. openqa. selenium .*;

"   "   "   " . chrome . chromedriver;

"   " . testng . *;

"   " . " . annotations . *;

Public class Second class{

Public static Webdriver driver;

@ BeforeTest

Public void Launch() {

driver = new ChromeDriver();

driver. navigate(). to ("http://newtours. demoat . com/*");

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

Reporter. log ("running launch", true);

}

@ Test (description = "verify RegisterLink", priority = 2, enabled = true);

Public void Register_Link() throws Interrupted Exception

driver . findElement (By. xpath(" //* [text() = 'Register'] ")). click();

Thread. sleep (5000);

Reporter. log (driver. getTitle());

Reporter. log (" running Register_Link", true);

}

@ Test (description = "verify Hotels_Link", priority = 0, enabled = true/false);

Public void Hotels_Link() throws Interrupted Exception

driver. findElement (By . xpath(" //* [text() = 'Hotels'] ")) . click();

Thread . Sleep (2000);

Reporter. log (driver. get Title ());

Reporter. log ("running Hotels_Link", true);

}

The following are the Expected Conditions that can be used in Explicit wait

↓ class

1. alertsPresent()
2. elementSelectionStateToBe()
3. elementToBeClickable()
4. elementToBeSelected()
5. frameToBeAvailableAndSwitchToIt()
6. invisibilityOfTheElementLocated()
7. invisibityOfElementWithText()
8. presenceOfAllElementsLocatedBy()
9. " "ElementsLocated()
10. textToBePresentInElement()
11. " " " located()
12. " " " elementValue()
13. titles()
14. titleContains()
15. visibilityOf()
16. " " allElements()
17. " " locatedBy()
18. " " located()

← methods

** (Methods do not contain spaces)

Actions in Webdrivers :-

1. Handling mouse Hovers & keystrokes using actions Concepts strokes.

In webdriver, handling keyboard events & mouse events (including actions such as Drag & Drop or clicking multiple elements with Control key) are done using the advanced user interactions API. It contains Actions & Action classes which are needed when performing these events.

In order to perform action events

// Configure the action

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

Package org.openqa.selenium.interactions.Actions

Method: keyUp(modifier_key)
Param: Modifier_key (Keys.ALT or Keys.SHIFT or Keys.CONTROL)
Purp: Performs a key release
parameters
purpose

Meth: moveByOffset(x-offset, y-offset)
Param: X-offset, Horizontal offset, a negative value means moving the mouse to left side.
y-offset, vertical offset, a negative value means moving the mouse to up.
Purp: moves the mouse position from its current position by the given offset.

Method: sendKeys(onElement, charSequence)
purp: It sends a series of keystrokes on to the element.

Keys.ARROW_DOWN
Keys.ARROW_UP
Keys.ARROW_LEFT
Keys.ARROW_RIGHT
Keys.TAB
Keys.ENTER
Keys.PAGE_DOWN
Keys.PAGE_UP
Keys.ESCAPE

Will fail the programs.

Robot class methods can be used to interact with keyboard/mouse events while doing browser automation. Alternatively AutoIT can be used, but its drawback is that it generates an executable file (exe) which will only work on windows so it is not a good option to use.

Some commonly & popularly used methods of Robot class during web automation.

class import
File - Java.io.File
FileUtils - org.apache.commons
put
OutType - org.openqa.selenium.OutputType;

File - throws IoException
& try/catch.

```
Package Testing_Examples;
import java.io.File;
 "    "   ".IoException;
 "    "   text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
 "      "      "    .TakeScreenshot;
 "      "      "    .WebDriver;
 "      "      "    .chrome.ChromeDriver;
 "      "      "    testng.annotations.Test;
 "      "      "    testng.annotations.Test;
Public class Take_Screenshot {
public static WebDriver driver;
@Test
Public void Screenshot() throws IOException {
 // java time stamp
DateFormat df=new SimpleDateFormat("dd_MM_YYYY hh_MM_SS");
Date d=new Date();
String datef = df.format(d);
driver = new ChromeDriver();
 ".navigate().To("http://facebook.com");
driver.manage().window().maximize();
```

To Execute XML file, Right click on it Run as TestNG Suite.

class folder created

After executing check ur outputs in testoutput folder.

If anyone Test case fail among multiple Test cases in a class TestNG will Generate Separate testng failed. xml file

To Execute failed Test case rightclick on testng failed. xml → run as testng test suite.

How to Separate class reports in xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="multiple" parallel="false">
  <test name="Test case 1">
    <classes>
      <class name="TestNG.examples.MouseEvents"/>
    </classes>
  </test> <!--Test case 1-->
  <test name="Test case 2">
    <classes>
      <class name="Testng examples.Mouse_hover1"/>
    </classes>
  </test> <!--Test case 2-->
</suite> <!--multiple-->
```

HANDLING ALERTS :- Confirmations & Prompts Popups

```
Alert alert = driver.switchTo().alert();
alert.accept();
alert.dismiss();
alert.sendKeys("  ");
```

```
Reporter.log("alert message is:: "+alertText, true);
Thread.sleep(3000);
dr.switchTo().alert().accept(); //accept alert
Thread.sleep(3000);
dr.close();
}}
```

⇒ package pack1;

* Sent Emails

⇒ **Handling windows / switching to windows**

⇒ Some web applications have many frames & multiple windows. Selenium WebDriver assigns an alphanumeric id to each window as soon as the WebDriver object is instantiated. This unique alphanumeric id is called window handle. Selenium uses this unique id to switch control among several windows.

Get Window Handle Command

Purpose:- To get the window handle of the current window.

```
String handle = driver.getWindowHandle(); // Return a string of alphanumeric window handle
```

Get window Handles Command

Purp: To get the window Handle of all the current windows.

```
Set<String> handle = driver.getWindowHandles(); // Return a set of window handle.
```

SwitchTo window Command

Purp: WebDriver supports moving between named windows using the "switchTo" method.

```
driver.switchTo().window("windowName");
```

(*)(*)(*)

```
package pack1;
* import java.util.Iterator;
public class windows_using Iterator {
public static WebDriver driver;

@Test
public void verify_Naukri() throws InterruptedException {
driver = new ChromeDriver();
driver.navigate().to("http://naukri.com");
driver.manage().window().maximize();
//get current window id
String [parent] = driver.getWindowHandle();
sopl ("parent window id is:" + parent);
// get all windows which are opened by webdriver
Set <String> S1 = driver.getWindowHandles();
//iterate all windows
Iterator <String> I1 = S1.iterator();
while (I1.hasNext()){
String [child_window] = I1.next();
//Here v will compare if parent window is not equal to child window then v will close child windows)
if (!parent.equals(child_window)){
//switch to child windows
driver.switchTo().window(child_window);
Sopl(driver.switchTo().window(child_window).getTitle());
Thread.sleep(5000);
driver.close();
}}
//switch to parent window
driver.switchTo().window(parent);
//click on Register
driver.findElement(By.xpath(".//*[@id='p /div/w../input"]).click();   } check if needed
                               give xpath
Thread.sleep(4000);
// click on I am Fresher
driver.findElement(By.xpath(".//*              ).click();   } check if needed
                            give xpath
Thread.sleep(4000);   }}
```

```
driver.close();
driver.switchTo().window(windows.get(0)); // switch to 1st window
Sop1(driver.getTitle());
Thread.sleep(5000);
driver.close();
}}
```

IFrame:-

- It is a webpage which is embedded in another webpage & an HTML document embedded inside another HTML document.
- The IFrame is often used to insert ~~form~~ content from another source such as an advertisement into a webpage.
- The <iframe> tag specifies

→ How 2 identify the iframe:

We cannot detect the frames by just Cing the page or by inspecting firebug, ~~Right click on element if u find the option like~~

→ Right click on element if u find the option like 'This frame' then it is an iframe.

- Right click on the page & click 'view Page Source' & search with the 'iframe' if u can find any tag name with the 'iframe' then it is meaning to say the page consisting an iframe.

Total no. of iframes:-

```
List<WebElement> allFrames = driver.findElement(By.tagName("iframe"));
    Sop1(allFrames.size());
```

→ How to switch over the elements in iframes using WebDriver Commands:

Basically u can switch over the elements in frames using 3 ways.

- By index
- By Name & Id
- By WebElement.

Properties file :- It is used for storing locator values like xpath, name & id etc and can be reused in any java file (class) just by loading it. The following steps need to be used in that particular class.

```
Properties p = new Properties();
FileInputStream i = new FileInputStream("path of the properties file");
p.load(i);
```

: retrieve any value from the properties file we need to follow the below syntax

```
p.getProperty("property name");
```

Object Repository :- In selenium there is no object repository concept but we will store all the objects information in the properties file & will treat it as object repository.

→ The same way how we load the properties file we can load the object repository.

→ The " " " " use " " " " " use the " " "

Navigation for Creating properties file :-

→ Right click on ~~Package~~ project → Go to new then click on ~~other~~ ~~Class~~ new file.

→ File name → Repository / properties
↓ it can be anything

Creating property file :- → click Finish.

- commenting in property file. - D

In above property file define locator values for login by using Xpath (it should be same throughout the file). Xpath.

```
# locator values for login
objuser = //*[@name='username']
objpass = //*[@name='password']
objsign = //*[@name='login']
```

7/11/17

Working with XL operations using Apache POI

Apache POI :- Handling excel files using " " in selenium web drivers.

As V all know Selenium supports only web browser automation. V need to get the help of 3rd party API like Apache POI to handle (read & write) excel files using Selenium WebDrivers.

Apache POI :- It is an open source library developed & distributed by Apache Software Foundation to design & modify Microsoft office using Java program. It is a popular API that allows working around Excel files using Java programs. In short, U can read & write MS Excel files using Java.

Apache POI is ur Java XL solution.

If U need to read or write an OOXML Excel file using Java (XLSX).

You'd use HSSF (Horrible Spread Sheet Format) if U needed to read & write an XL file using Java (XLS). Would use XSSF (XML Spread Sheet Format)

It has many predefined methods, classes & Interfaces.

List of different Java Interfaces & classes in POI for reading XLS & XLSX file

Interface	XLS class (Lower version 97-2003)	XLSX class higher version
Workbook	HSSF workbook	XSSF workbook
Sheet	HSSF sheet	XSSF sheet
Row	HSSF Row	XSSF Row
Cell	HSSF Cell	XSSF Cell

Downloading Apache POI Jar files :-

open → https://poi.apache.org/download.html → click on poi-bin-3.17-20170915.zip (28.65MB) → click on very first link under Binary Distribution

→ Save in Drive (C or D)

After downloading extract to ur Library folder created in D or C drive.

Adding Jar files to ur project.

Right click on JRE system library under ur project → Build path → Configure Build path.

click on

→ Add External Jars (under libraries tab)

→ Browse ur POI Jar files, go to library folder → Select all Jar files in POI → go back select OOX → select the file → " " → " " click OK.

FileOutputStream :- To write data in a file.

FileOutputStream fo = new FileOutStream(new file("c:\\newbook.xlsx"));
path of XL

write() :- Writing data in a file.

createcell() :- To write data in a specific column.
syntax :- creat-cell(column number)

setCellValue() :- To Set a value/text in a specified column/row.
setCellValue("write text");
"pass"

path - Single forward or Double Backward.

Reading Excell :-

```
package Reading;
import java.io.FileInputStream;
import org.apache.xssf.usermodel.XSSFRow;
"  "  "  "  "  .XSSFSheet;
"  "  "  "  "  .XSSFWorkBook;
"  "  "  "  "
public class Reading_Text {
public static void main(String [] args) {
  try{
// read file  f = data
FileInputStream fi = new FileInputStream("D:\\Automation_Framework\\Test-NG");
                                  give path - backward slash - 2
                                  forward  "  - 1
//create work book
XSSFWorkBook wb = new XSSFWorkbook(fi);
//get sheet from workbook
XSSFSheet ws = wb.getSheetAt(0);
//Count no of rows
int rc = ws.getLastRowNum();
S.O.P.l("no. of rows are :" +rc);
// count no of columns
XSSFRow row1 = ws.getRow(0);
int cc = row1.getLastCellNum();
S.O.P.l("no. of columns in first row :" +cc);
//Print data from username & password columns
  for(int i=1; i<=rc; i++){
String username = ws.getRow(i).getCell(0).
                  getStringCellValue();
String password = ws.getRow(i).getCell(1).
                  getStringCellValue();
S.O.P.l(username + " " + " " +password);
  }
  }catch(Throwable t){
  S.O.P.l(t.getMessage());
  }
  }
```

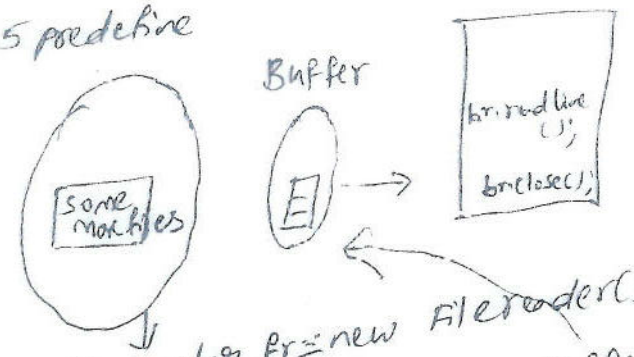
⇒ sent email;

⇒ Package Retetting;

14/11/17

⇒ Dynamic handling of file system;-

→ It is a concept provided in Java which is used for
→ creating a new file in a file system
→ writing data into a file
→ Reading " "
To do the same in Java we have 5 predefine
→ File
→ File Reader
→ Buffered reader
→ File writer
→ Buffered writer



File reader fr = new FileReader();
Buffered Reader br = new BufferedReader(fr);

Readline():- Reading line by line in a file.

→

```
Package Retetting;
import java.io.BufferedWriter;
public class Writing_Textinnotepad {
Public static void main (String[] args) throws IOException {
File f= new File("e://10clock.txt");
 f.createNewFile;
 FileWriter fr = new FileWriter(f);
 BufferedWriter bw = new BufferedWriter(fr);
 bw.write("I am good");
 bw.newline();
  bw.write("I am practicing");
  bw.newline();
  bw.write("so I am looking for aJob");
   bw.flush();
   bw.close();
  } }
```

→ write a script to parametrize data from notepad!

→ Paste data in notepad

Admin@admin

Admin@admin!

" " "

" " "!

" Admin

" " "!

Admin@admin!

Admin@Admin

→ Package Retesting;

import java.io.BufferedReader;

public class Reading_logindata {

public static webdriver dr;

" " Filereader fr;

" " BufferedReader br;

@Test

Public void verify_login() throws IOException{

fr = new FileReader ("e://clock.txt"); → use wbd doc excel.

br = new BufferedReader (fr);

String str=" ";

while((str = br.readline()) != null)

{ dr = new ChromeDriver();

dr.navigateTo("https://orangehrm.gedgetech.com/Symfony/web/index.pl");

dr.manage().window().maximize();

String arrayvar[] = str.split("@");

dr.findElement(By.xpath("//*[@id='txtusername']")).SendKeys(arrayvar[0]);

" " " " " name='txtpassword']")). " " [1]);

" " " " " 'submit' " " (Keys.ENTER);

String Expval = "http://orangehrm.gedgetech.com/Symfony/web/index

give url after login

String ACval = dr.getCurrentUrl();

if (Expval.equals(ACval)) {

Reporter.log("login Success",true);

```
@Test (description = "accessing data from DataProvider", dataprovider
                                                    = "Test-data_login")
Public void verify_login(string username, string Password) {
  WebDriver obj = new ChromeDriver();
obj. navigate(). to("http://newtours. demoaut. com");
 obj. manage(). window(). maximize();
obj. findElement (By. xpath ("//input[@name='username']")). sendkeys
                                                        (username);
obj. findElement(By. xpath("//input[@name='password']")). sendkeys
                                                        (password);
obj. findElement(By. xpath( "      " login) . sendkeys
                                                        (keys. ENTER);
 String Expval = "Find a Flight: Mercury Tours";
  "     Acval = obj. getTitle();
 if(Expval. equals(Acval)) {
  Reporter. log("login success", true); }
   else{ Reporter. log("login unsuccess", true); }
   obj. quit(); } }
```

POM: Page Object Model using Selenium webdriver

-> It is design pattern in which will help you to maintain the code & code duplication which is crucial thing in Test automation.

-> you can store all locators & respective methods in the separate class & call them from the test in which u have to use. so benefit from this will be if any changes in page then you do not have to modify the test simply modify the respective page & that's all.

-> you can create a layer betn ur test script & application page which u have to automate.

-> In other words, it will behave as Object repository where all locators are saved.

Implementation of POM using Selenium webdriver:

IF u want to implement POM then u have 2 choices & u can use any of it.

1- POM without page factory
2- POM with page factory

Define rest of the modules under same package.
Executing POM under src create a package as Test Script
under that package create a class as maintestcase

Main Test case ;- - Test NG class

```
package Test-Script
import org.openqa.selenium.webdriver;
public class Main Test case {
Public static WebDriver driver;
@BeforeMethod
Public void launch() throws Interrupted exception {
driver = new Chromedriver();
driver.navigate().to("http://primusbank.qedgetech.com/");
driver.manage().window().maximize();
//accessing locators & methods
Adminloginpage login = PageFactory.initElements(driver, Adminloginpage.class);
//call login method  -> Method
login.verify_login("Admin", "Admin");
}
@Test(description = "verify branch creation", priority=0, enabled=true)
Public void Branch() throws Interrupted Exception {
Branch creation creation = PageFactory.initElements(driver, Branch creation.class);
creation.verify_BranchCreation("pet1234", "Hightemabad", "Hgh7", "Srnagar",
                               "Kupally", "12345", "UK", "England", "London");

// get alert message
String alert message = driver.switchTo().alert().getText();
Reporter.log(alertmessage, true);
Thread.sleep(3000);
driver.switchTo().alert().accept();
}
@Test(description = "verify branchupdation", priority=1, enabled=true)
Public void Verify_Bupdation() throws Interrupted Exception {
Branch updation update = Pagefactory.initElements(driver, Branch updation.class);
update.verify_update("kadri", "Apur", "51578");
//get alert message
String alert message = driver.switchTo().alert().getText();
```

```
for (int i=1; i<=rc; i++){
driver = new ChromeDriver();
driver.navigate().to("http://primusbank.qedgetech.com/");
driver.manage().window().maximize();
Adminloginpage login = PageFactory.initElements(driver, Adminloginpage.class);
login.verify_login("Admin", "Admin");
String bname = sh.getRow(i).getCell(0).getStringCellValue();
"  add1 = sh.getRow(i)," (1)."
"  add2 = "  "  (2)."
"  add3 = "  "  (3)."
"  area = "  "  (4)."
"  zcode = "  "  (5)."
"  country = "  "  (6)."
"  state = "  "  (7)."
"  city = "  "  (8)."
BranchCreation createb = PageFactory.initElements(driver, BranchCreation.class);
createb.verify_BranchCreation(bname, add1, add2, area, zcode, country, state,
                              city);
//get alert message
String alertmessage=driver.switchTo().alert().getText();
Reporter.log(alertmessage, true);
Thread.sleep(2000);
driver.switchTo().alert().accept();
sh.getRow(i).createCell(9).setCellValue(alertmessage);
driver.quit();
}
fi.close();
fo=new FileOutputStream(new File("D://Automation-Framework
                                 new filepath
wb.write(fo);
fo.close();
wb.close();
}}
```

⇒ Get Email

Manual Testing	Keyword Driven framework TestNG
Precondition	Before
Test Scenario with Test scripts	Test
Post Condition	After

Advantages :- The same manual testing format v are using in automation so the maintainence is easy.

Junit & TestNG are predefined frameworks not necessary to implement the frameworks

Once the execution is completed it will provide the results also.

Dis Advantages :- V can choose keyword driven formework for suitable applications.

Hybrid Framework :-

- Combination of 2 (or) more frameworks is nothing but hybrid framework.
- V can utilize the advantages of both the frameworks in " "

DDF + MF (or) DDF + KDF
modular

Working with OrangeHRM project :-

Identify the scenarios in OrangeHRM :

1) Admin login 2) Employee creation 3) User creation

Preparing Test cases & Test steps in XL sheet.

Open XL sheet Rename sheet name as Test Cases & develop the Test cases.

Test case sheet :

Tcid	TestcaseDesc	Execute	Results
TC001	check Admin login	Y	
TC002	" New employee	Y	
TC003	" " User " Registration	Y	
TC004	check user login	N	

Rename sheet 2 as Test steps & define all the steps to execute the above test cases

Tcid	step id	stepDesc	Results	Keyword
TC001	step 1	Launch application		Launch app
TC001	step 2	Enter username, password & click login		Admin login
TC001	step 3	Logout		logout

public static String pwd = "Admin";
" " " Exp.val, Act.val;
" " " fn, Ln;
" " Properties P;
" " FileInputStream fi;

/*
* Module Name; verify launch application
* Designed By: Jagan
* Designed on: 08/09/17
*/
public boolean verify_adminlogin(Sent Email → incomplete prgm
Click "

→ Preparing Driver Script:-

Driver Script is nothing but executing all the test cases using TestNG.
To this driver script we need to associate XL File, Log4J File.
Under src Create a package Test Cases. Under that package
Create class name as Driver script with out selecting main method.
Develop the driver script under that class.

Sent Email:- Prgrms

~~SELENIUM GRID~~ MAVEN

MAVEN

Download Maven:- Open a browser - https://maven.apache.org/download.cgi
Click on Apache maven 3.5.2-bin.zip - In any drive Create a folder
Maven. Extract that zipfile to that folder.

Setting the class path of maven:- Right click on Computer → Properties →
Click on Advanced system settings → Environment variables →
under system variables click on New → Variable name = MAVEN_HOME
Variable value = Copy the maven Path
Paste it here
clickOK → Edit path variable → Copy the path of maven bin
(select path)
End of existing path add semicolon give path - ClickOK, OK, OK

Prerequisite: Add Java class Same as above.
JAVA_HOME

Surefire plugin is used during the "

Adding Jar files / dependencies :-

Adding dependency Jar files to pom.XML. open a browser enter URL https://mvnrepository.com/ → search type Selenium-server → search

2-- POI (2 Jars)

3- TestNG

click on very first link selenium server → click on 3.7.1 version → Uncheck Include Comments. → copy dependency → open pom.XML → Select pom.XML tab → paste to dependency (below add comments)

Add for TestNG, POI (2 Jar files POI, OOXML), Log4J → Refresh.

SRC/main/Java is used for maintaining Reusable scripts (property files)

SRC/Test/Java is used for executable scripts

Maven dependencies will show you all the Jar files

(By default Junit is added)

Creating maven test case :- Select SRC/Test/Java Right click the mouse → New → class

```
package Resources
Import org.openqa.selenium.By;
public class Orange {
    // static WebDriver dr;
    @Test
    Public void verify_login() {
        dr = new ChromeDriver();
        dr.navigate().to("https://orangehrm.redgetech.com/symfony/web/index/");
        dr.manage().window().maximize();
        dr.findElement(By.XPath("//*[@id='txtUsername']")).sendKeys("Admin");
        //                    txtPassword      "Admin"
        //                    btnLogin         Keys.ENTER
        dr.quit();
    }
}
```

Type the command
java -jar selenium-server-standalone-3.7.1.jar -role hub
click enter. make sure Selenium Grid Hub is up & running & also check the IP address to which nodes has to register.
open a browser — http://localhost:4444/grid/console to view grid console.
Configure Node & the Hub:- Configure Single IE browser, Single Firefox browser

select the folder in which u have Jar file — Shift right click the mouse select open command window here.

java -Dwebdriver.chrome.driver=D:\chromedriver.exe -Dwebdriver.gecko.driver=D:\geckodriver.exe -Dwebdriver.ie.driver=D:\IEDriverServer.exe -jar selenium-server-standalone-3.7.1.jar -role webdriver -hub http://localhost:4444/grid/register -port 5550 -browser browserName=firefox -browser browserName=explore -browser browserName=chrome

Give Continue no newline in command prompt → drivers should be available in respective folder eg: above D: drive.
Go & refresh ur Grid console:
To design test scripts in Grid that will run on the grid u need to use DesiredCapabilities & the RemoteWebDriver objects.
" " is used to set the type of browser & OS that u will automate.
RemoteWebDriver is used to set which node (or machine) that our test will run against.
To use the DesiredCapabilities object u must first import this package

```
import org.openqa.selenium.remote.DesiredCapabilities;
```

To use the RemoteWebDriver object u must import these packages

```
import java.net.MalformedURLException;
import java.net.URL;
import org.openqa.selenium.remote.RemoteWebDriver;
DesiredCapabilities capability = DesiredCapabilities.firefox();
capability.setBrowserName("firefox");
// platform (Platform.XP);
```

Using the RemoteWebDriver object: Import the necessary packages for RemoteWebDriver & then pass the DesiredCapabilities object that u created above as a parameter to the Remote WD obj.

```
<classes>
<test>
<testName = "Chrome Test">
<parameter name = "browser" value = "Chrome" />
<classes>
<class name = "Resources.AppTest" />
</classes>
</test>
<test name = "IE Test">
<parameter name = "browser" value = "ie" />
```

```
<classes>
<class name = "Resources.
AppTest" />
</classes>
</test>
</suite>
```

Creating Batch File:-

Create one Test NG class & define all ur browsers:

Sent Gmail!

- To execute this class right click on XML file run as Test NG Suite.

BF:- open a notepad

d: (driver)

cd selenium-Grid (Folder name)

java -Dwebdriver.chrome.driver=D:\chromedriver.exe -jar selenium-standalone-server-3.9.1.jar

↓ single line

↓ write command

click previous file for full command

Save notepad with .bat as .bat. To execute batch file double click on it.

Right click - edit for any changes.

AUTOIT :-

AutoIT is a freeware BASIC like scripting language designed for automating the windows GUI & general ||. It uses a combination of simulated keystrokes, mouse movement & window/control manipulation in order to automate tasks in a way not possible or reliable with other languages (e.g.: VBScript & SendKeys)

It is automation tool like Selenium but unlike selenium it is used for Desktop automation rather than web Automation.

It is a powerful tool & it Just not automate desktop windows button & form it automates mouse movements & keystrokes too.

Why Maven & Jenkins!- Selenium WebDriver is great for browser automation but when using it for testing & building a test framework it feels underpowered. Integrating Maven with Selenium provides following benefits Apache maven provides support for managing the full lifecycle of a test project

→ maven is used to define project structure, dependencies, build & test management

→ using pom.xml (maven) u can configure dependencies needed for building testing & running code. → Maven automatically downloads the necessary files from the repository while building the project.

Downloading Jenkins war/Jar file!-

Open ur browser - https://updates.jenkins-ci.org/download/war/ click 2.77 version. After downloading in D: drive create a folder as Jenkins, copy, paste Jenkins Jar file.

Running Jenkins!- Select a folder in which Jenkins Jar file is existing shift & right click the mouse. Select 'open command window here'

type: java -jar jenkins.war click enter.

Open a browser enter http://localhost:8080. Once u click enter button Copy the path provided, open window explorer & paste it. click ".

Select notepad & click OK. Copy the password & paste to Administrator Password text box in url. click continue. click on Install suggested plugins. wait for all plugins to install.

Create username & password & provide ur email id. click save & finish.

Login to Jenkins!

Select "folder right click on Jenkins folder - select open command window here type above command. java -jar jenkins.war click enter. wait for a msg Jenkins is fully up & running. Open a browser enter //localhost:8080/ click enter. Enter username & password.

Configuring Maven & Java to Jenkins!- To execute maven test-cases via Jenkins. Click on manage Jenkins → click on Global Tool Configuration.

Scheduling Project:- Expand ur project select Configure. Click on Build dropdown m

Triggers tab. Click checkbox Build periodically. Provide as
→ MINUTE HOUR DOM MONTH DOW —
MINUTE — Minutes within the hour (0-59)
HOUR — The hour of the day (0-23)
DOM — The day of the month (1-31)
MONTH — The month (1-12)
DOW — The day of the week (0-7) where 0 & 7 r Sunday.

eg:- 18 11 4 12 1

click save

Java Script Executor:- Creating object for javascript executor class.

Javascriptexecutor js = (Javascript Executor) driver;

Javascript:- Java script executor is an interface which provides mechanism to execute Javascript through selenium driver. It provides "executscript" & "executed Asyncscript" methods, to run Javascript in the context of the currently selected frame or window.

To enhance the capabilities of the existing scripts by performing Javascript injection into our class.

In simple words "Javascript can be executed with in the browser with of Java script Editor.

Packge: Import org.openqa.selenium.Javascript Editor;

Syntax:

Javascript Executor js = (Javascript Executor) driver;
js.executescript (script, Arguments);

Code:-

Packge Javascripts;
Import org.openqa.selenium.JavascriptExecutor;
public class Java-Executor {

```
Package Javascript;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

Public class login_facebook {
Public static void main (String[] args) throws InterruptedException {
WebDriver driver = new ChromeDriver();
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("window.location = 'http://facebook.com'"); → driver.manage().
                                                               window().maximize();
                ("document.getElementById('email').value='test@gmail.com'");
                ("                          ('pass').value = 'test543'");
                ("                          ('u_0_5').click();");

Thread.sleep(4000);
driver.navigate().back();
Thread.sleep(4000);
js.executeScript("window.location='http://facebook.com'");
                ("document.getElementById('email').style.
                    background color = 'green';");
                ("                  ('email').style.color = 'red';");
                ("                  ('email').value = 'test@gmail.com'");

Thread.sleep(4000);
js.executeScript("alert('Yogaruth');");
driver.switchTo().alert().accept();
Thread.sleep(4000);
js.executeScript("document.getElementById('u_0_4').checked=
                                                    ='true';
Thread.sleep(4000);
js.                ('day').value='23'");
    driver.quit(); } }
```

```
String coltext = cols.get(j).getText();
    S.o.p("\t" + coltext);
}
S.o.p.l();
}driver.quit(); }}
```

Sat Gmail

Database Testing using selenium:-

Selenium WebDriver is limited to testing Wr apps using Browser. To use " " for Database testing u need to use the JDBC("Java Database Connectivity")

JDBC (Java Database Connectivity) is a sql level API that allows U to execute SQL statmts. It is responsible for the connectivity betn the Java programming language & a wide range of databases. The JDBC API provides the following classes & interfaces.

-> Driver Manager -> Driver -> Connection -> Statement -> Result set -> SQL exception

In order to test ur Database using selenium U need to observe the following 3 steps

1 - make a connection to the DB.
2 - Send queries to the DB
3 - process the results

- make a connection to the DB: In order to make a connection to the DB the syntax is

```
Connection con = DriverManager.getConnection(db url, username, password);
```

U also need to load the JDBC driver using the code

```
Class.forName("com.mysql.jdbc.Driver");
```

2 - Send queries to the DB: Once connection is made U need to execute queries. You can use the stmt object to send queries.

```
Statement Stmt = con.createStatement();
```

Once the statement object is created use the executeQuery method to execute the SQL queries.

Extent Reports: - Add Extent Reports Jar file in maven pom.xml by searching in maven repository.

```
Package extent;
import org.openqa.selenium.By;
public class Reports_Extents{
public WebDriver Driver;
public static Extent reports Extent;
  "      "      "   test test;
  "      "      "   " Interrupted Exception test.Exception;
  "      "      "

@Before Method
public void Launch(){
driver = new Chromedriver();
" .navigate().to("http://newtours.demoaut.com/");
" .mange().window().maximize();

}
// Testcase1
@Test
public void testcase1()
{
extent = new ExtentReport("./extentsreports/TestReports.html");
test = extent.StartTest("test Case1");
test.log(Logstatus.Info, "enter username");
driver.findElement(By.name("username")).SendKeys("Admin");
test.log(Logstatus.Info, "enter password");
driver.findElement(By.name("Password")).Sendkeys("mercury");
  "      "   ("    "  , "click on Login");
  "      "      "   "  ("login")).click();

S.opl("in test case 2");
String Exp = "Find a flight: Mercury Tours:");
  "   Acv = driver.getTitle();
```