Important Java concepts required for Selenium :-

1). Conditions :- (if, if-else, Switch)

if-else :

In if-else statement, if the condition is true then we will execute if-block, else we will execute else block.

Syntax:

```
if (condition)
{
  if-statement
}
else
{
  else-statement
}
```

Ex:-

```
int a=10, b=20;
if(a>b)
{
  a is greater
}
else
{
  b is greater
}
```

Switch :

```
int n=10;
switch (n)
{
  case 0 : s.o.p("value is 0");
           break;
  case 1 : s.o.p("value is 1");
           break;
  case 2 : s.o.p("value is 2");
           break;
  default: s.o.p("Null value");
           break;
}
```

"Main method cannot be inherite because, main method is static"

Scanned by CamScanner

2) Looping Statements (loops) :-

(for, while, for each/enhanced for loop)

for :-

Syntax :-
```
for (initialization, condition, inc/dec)
{
  ==
}
```

Ex :-
```
for (int i=0; i<=10; i++)
{
  s.o.p(i);
}
```

while :- In while loop if the condition is true, it will execute the statements present in the loop, else it will break the loop.

To break the loops explicitly, we use the statement break.

Syntax :-
```
while (condition)
{
  =
  ==
}
```

```
while (i<=10)
{
  s.o.p(i);
  i++;
}
```

```
while (true)
{
  s.o.p(10);
  break;
}
```

3). OOPs :-

(Inheritance, polymorphism, Encapsulation, interface/abstraction class)

4) Constructors

Scanned by CamScanner

12/01/18

Exceptions :-

Exceptions are the special events which will terminate the execution of the code

2 Types of Exceptions :

1). checked Exceptions

2). Unchecked Exceptions

1). checked Exceptions :-

Exceptions which occurs during the compilation it called checked exceptions.

2) Unchecked Exceptions :-

The exceptions which occurs during runtime it called unchecked exceptions.

Ex :- ArrayIndexOutOfBoundsException.

Exceptions can be handled by using throws keyword

(or) try-catch blocks & finally block.

Collection :-

* - It is a framework
* - collection is also called as container where we can store any type of objects
* Basically collection itself is an interface.

Scanned by CamScanner

Code Optimization :-

Reducing the number of statements, but still getting same output is called as code optimization.

Ex :-1 ① int a=10;
int b=20;
int sum=a+b;
s.o.p(sum);

② int a=10;
int b=20;
s.o.p(a+b);

③ int a=10;
s.o.p(a+20);

④ s.o.p(10+20);

Ex :2 :-

```
class Dog
{
  String name = "tiger";
}
```

step-1 :- Dog d=new Dog();
String s=d.name;
int n=s.length();
s.o.p(n);

step-2 : Dog d=new Dog
String s=d.name;
s.o.p(s.length())

step-3 :- Dog d=new Dog();
s.o.p(d.name.length());

step-4 : s.o.p(new Dog().name.length());

Scanned by CamScanner

System.out.println :-

In this statement 'System' is a class.

'out' is a static variable of type printStream

& 'println' is the overloaded method of printstream class

Ex :-

```
public PrintStream
{
  public void println()
  {
    ==
  }
  public void println(String)
  {
    ===
  }
  public void println(int)
  {
    ==
  }
}
```

```
class System
{
  static printStream out;
  ==
}
```

Scanned by CamScanner

Method chaining :-

* One method calling another method is called method chaining.

Ex:-

```
package qsp;
class Dog
{
  public String dogName()
  {
    String name = "tiger";
    return name;
  }
}
public class Demo
{
  public static void main (String[] args)
  {
    Dog d = new Dog();
    //method chaining
    int n = d.dogName().length();
    S.o.p(n);
  }
}
```

Scanned by CamScanner

Upcasting :-

16/01/18

- Converting sub-class object into super-class type is called as upcasting.
- When we convert sub-class object into super-class type, then we can access the methods which are present in super class only. All the sub-class methods will be hidden
- If both sub-class & Super class contains same method then JVM will execute the method present in sub-class (overridden method).

Ex :-

```
class A
{ public void testAC)
  {
    s.o.p("testAC) of class A");
  }
}
class B extends A
{
 public void testAC)
  { s.o.p("testAC) of class B");
  }
 public void testBC)
  { s.o.p("testBC) of class B");
  }
```

Scanned by CamScanner

```
    public void testC()
    {
      s.o.p("testC() of class B");
    }
}
public class Demo
{
  p.s.v.m()
    {
      A a = new B();
      a.testA();
    }
}
```

o/p :- testA() of class B
(Overrided method will be executed);

```
interface A
{ public void testA();
}
class B implements A
{ public void testA()
    { s.o.p("testA() of class B");
    }
    public void testB()
    {
      s.o.p("testB() of class B");
    }
}
```

Scanned by CamScanner

```
public class Demo
{ public static void main (String [] args)
  {
    A.a = new B();
    a.testA();
  }
}
```

Generic Methods :-

* Generic methods are Reusable methods.
* Generic methods contains the set of logical statements which are commonly repetative in multiple places.
* In generic method we should not hard code any values, everything should be entered dynamically.

Factory class :-

* Factory class is a java class which contains generic methods which is called as factory method (or) helper method.
  This factory method will return the object of implementation class.

Ex :-

```
interface Switch
{ public void on();
  public void off();
}
class Television implements Switch
{
```

```
class Airconditior implements Switch
{ public void on()
  { s.o.p("switch on AC");
  }
  public void off()
  { s.o.p("switch off AC");
```

Scanned by CamScanner

```
    public void on()
    { s.o.p("Switch on TV");
    }
    public void off()
    { s.o.p("switch off TV");
    }
}

class FactoryDemo
{ public Switch getObject(String str)
  { switch s = null;
    if (str.equalsIgnoreCase("TV"))
    { s = new Television();
    }
    else if (str.equalsIgnoreCase("Ac"))
    { s = new AirCondition();
    }
    return s;
  }
}
public class Demo
{ public static void main(String[] args)
  {
    FactoryDemo f = new FactoryDemo();
    Switch s = f.getObject("tv");
    s.on();
    s.off();
  }
}
```

o/p :-

Switch on TV

Switch off TV.

Scanned by CamScanner

Automation:

* Testing the functionality of an application [Desktop (or) Web based ap
using some automation tools [QTP & Selenium]. it called
as Automation.

Advantages of Automation:

*. The human effort will be less.
*. We can save the time.
*. There will be more consistancy in the project.

Difference b/w Selenium & QTP:

Selenium	QTP
1) It is Free	1) It is a licensed tool
2) It is an open source tool	2) It is not an open source tool.
3) Using Selenium we can auto-mate web-based application.	3) Using QTP we can automate both Desktop & web-based application
4) It supports for 12 different types of programming languages.	4) It supports for only VB scripts.
5) It supports for almost all the browsers [IE, FF, GC, opera, ...]	5) It supports for IE, Firefox & Google Crome browsier
6) It supports for multiple platforms such as windows, max, linux, ... etc	6) It supports for only window operating systems.

Scanned by CamScanner

Selenium :-

* Selenium is a free and open source automation tool, which is used to automate the functionality of Web-base application.

* To make use of selenium for commercial purpose, we need not to buy any licence. It is freely available in the following website.

http://www.seleniumhq.org/download/

* We can view the source code of selenium. It is available in the following website.

http://github.com/seleniumHQ/selenium

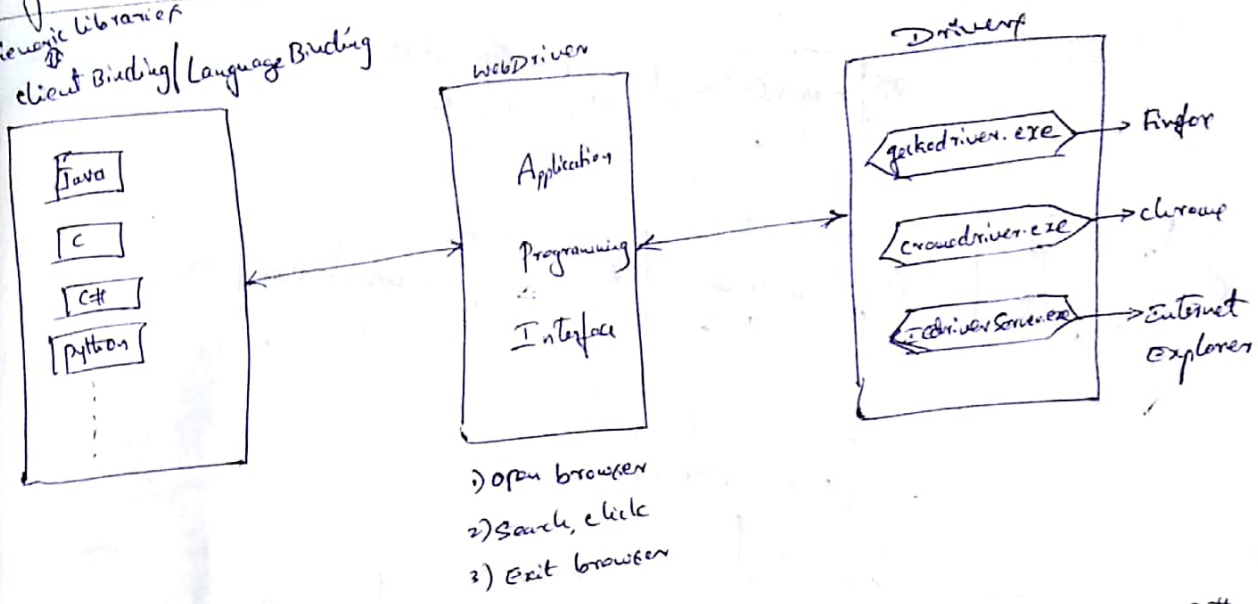
* Using Selenium we can Automate the web-based applications such as gmail, FB, Amazon, flipkart .... etc

* Selenium supports for 12 different types of programming languages such as.

1). Java
2). C
3). C#
4). Javascript
5). python
6). ruby
7). perl
8). php
9). dart
10). R
11). Tcl
12). Haskell

Scanned by CamScanner

High level Architecture of Selenium :-

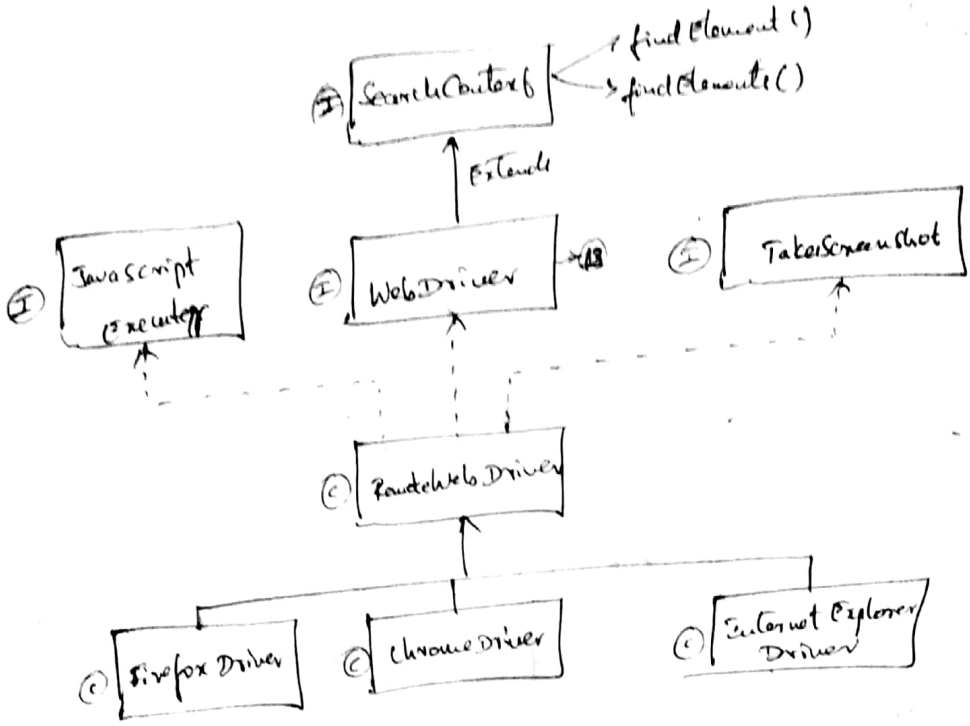


* Selenium supports for multiple languages such as java, c, c# .. etc.
* Each of these languages contains its own client bindings (or) we also called it as language binding.
* In each of these programming languages some Generic libraries are already developed. These Generic libraries are called as client bindings (or) language binding.
* These client binding will communication with API which is called as Web drivers - Here, the client bindings will communicate what action has to be performed on the browser.
* In order to perform the action on the browser, webdriver will communicate with driver executable files, such as geckodriver.exe, chromedriver.exe, IEdriverServer.exe.
* geckodriver.exe is used to perform action in Firefox browser
  chromedriver.exe is used to perform action in chrome browser.
  IEdriverServer.exe is used to perform action in Internet Explorer browser.

Scanned by CamScanner

(#11)

Java Selenium High level Architecture :-



* Search Context is the supermost Interface, which contains two methods. i.e. 1) findElement() method
2) findElements() method

* Search context is Inherited by another interface which is called as WebDriver

* WebDriver Interface contains 13 different types of methods. They are

1) get()
2) getTitle()
3) getCurrentUrl()
4) getPageSource()
5) findElement()
6) findElements()
7) getWindowHandle()
8) getWindowHandles()
9) SwitchTo()
10) manage()
11) navigate()
12) close()
13) quit()

Scanned by CamScanner

* It also contains other interfaces such as
  1. JavaScriptExecuter &
  2. TakesScreenShot
* All these interfaces are implemented in a class called RemoteWebDriver.
* All the methods of RemoteWebDriver class are overridden in respective browser classes such as
  → FirefoxDriver
  → ChromeDriver &
  → InternetExplorerDriver
* ChromeDriver class is used to work with chrome browser.
* FirefoxDriver class is used to work with firefox browser.
* InternetExplorerDriver class is used to work with InternetExplorer browser

Tools required for Selenium :-

1. jdk [1.8].
2. eclipse IDE.
3. selenium jar file.
4. driver executable files.
5. browsers.
6. Application under Testing.

18-01-18

Installing Selenium :-

* Download selenium jar file & driver executable files which are available in following website.

http://www.seleniumhq.org/download/

1. Selenium server standalone 3.8.1
2. chromedriver.exe

Scanned by CamScanner

iii). geckodriver.exe

iv). IEDriverServer.exe

* Extract all the driver executable files.
* In eclipse create a project with a name Automation.
* Under java project [Automation] create two folders and name it as i). drivers
  ii). jars
* Store all the extracted driver executable files under drivers folders.
* Store all the jar files under jars folders.
* Associate jar file with current java project.

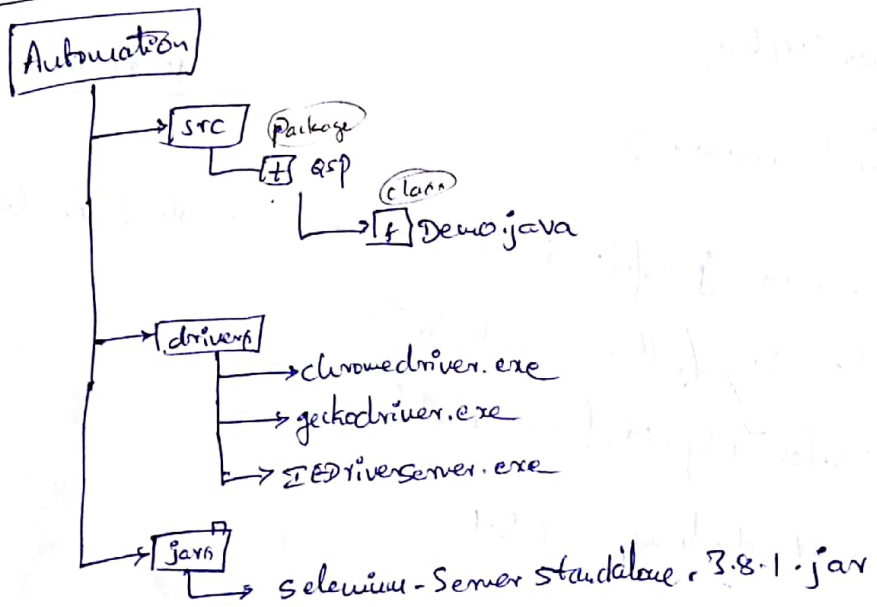
To Associate the jar file, right click on the jar file, go to Build path, click on Add to Build Path.

Note : 1). All the jar files should be associated with java project.

2). If we do not associate the jar files we cannot use the classes or interfaces which are present in the jar files.

3). Associated jar files will be present under Reference Libraries.

Standard Folder structure :-



Scanned by CamScanner

Handling Firefox Browser :-

* To work with the firefox browser we the class Firefox driver.
* Before launching the browser we have to specify the path of the driver executable files by using setproperty method of system class.
* Setproperty method takes two argument of type String.
* key for geckodriver is "webdriver.gecko.driver". and the value is path of the driver executable files.

Note :- If we donot specify the path of the driver executable file then it will throw IllegalStateException.

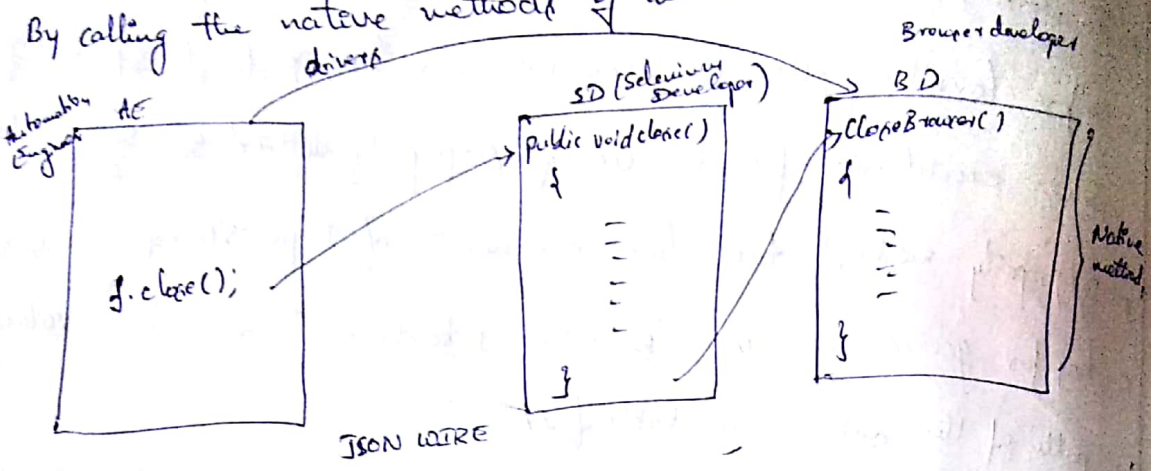
Ex :-

```
public class Demo
{
    public static void main (String[] args)
    {
        String key = "webdriver.gecko.driver";
        String value = "./drivers/geckodriver.exe";
        System.setProperty(key, value);
        // To open the browser
        FirefoxDriver f = new FirefoxDriver();
        // To close the browser
        f.close();
    }
}
```

Scanned by CamScanner

① How does selenium performs the action on the browser?

By calling the native methods of the browser.



② How does selenium calls native methods of the browser?

By using driver executable files.

③ Why do we need driver executable files?

to call the native methods of the browsers.

④ How does selenium communicates with Driver executable files.

by using JSON WIRE [Java Script object notation] protocol.

22/01/18

Handling chrome Browsers:-

* In order to handle the chrome browser we use a class called "chromedriver".
* The key for chromedriver is "webdriver.chrome.driver"

Ex:-

```
public static void main(String[] args)
{
    //To specify the path of the drivers
    String key = "webdriver.chrome.driver";
    String value = "./drivers/chromedriver.exe";
```

Scanned by CamScanner

```
System.setProperty(key, value);
// To open chrome browser
ChromeDriver c = new ChromeDriver();
Thread.sleep(3000);
// To close the browser
c.close();
}
```

Handling InternetExplorer Browser :-

* Before launching the IE Browser we have to do the following settings.

step-1 :- Open IE Browser

step-2 :- Go-to tools & click on internet options

step-3 :- Go-to security tab

step-4 :- Select enable protected mode checkbox in
- -> Internet
- -> local Intranet
- -> Trusted sites
- -> Restricted sites

step-5 :- If not change the zoom level to 100%. [ctrl+0]

* If we do not do the above setting, then it will throw "SessionNotCreatedException".

Ex :-
```
public static void main(String[] args)
{
    String key = "webdriver.ie.driver";
    String value = "./drivers/IEDriverServer.exe";
```

Scanned by CamScanner

```
System.setProperty(key, value);
InternetExplorerDriver ie = new InternetExplorerDriver();
Thread.sleep(2000);
ie.close();
}
```

Settings by using Selenium :-

```
public class Demo
{
  public static void main(String[] args)
  {
    String key = "webdriver.ie.driver";
    String value = "./drivers/IEDriverServer.exe";
    System.setProperty(key, value);
    DesiredCapabilities dc = new DesiredCapabilities();
    dc.setCapability(InternetExplorerDriver.IGNORE_ZOOM_SETTING, true);
    dc.setCapability(InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_
                     IGNORING_SECURITY_DOMAINS, true);
    InternetExplorerDriver ie = new InternetExplorerDriver(dc);
    Thread.sleep(2000);
    ie.close();
  }
}
```

Scanned by CamScanner

Write a Script to open & close the browser based on user input.

```
public class Demo
{
public static void main (String[] args)
  {
    S.o.p("Enter browser:");
    Scanner scan = new Scanner(System.in);
    String browser = scan.nextLine();
    WebDriver driver = null;
    if (browser.equalsIgnoreCase("ff"))
    {
      String gecko-key = "webdriver.gecko.driver";
      String gecko-value = "./drivers/geckodriver.exe";
      System.setProperty(gecko-key, gecko-value);
      driver = new FirefoxDriver();
    }
    else
    if (browser.equalsIgnoreCase("gc"))
    {
      String chrome-key = "webdriver.chrome.driver";
      String chrome-value = "./drivers/chromedriver.exe";
      System.setProperty(chrome-key, chrome-value);
      driver = new ChromeDriver();
    }
    Thread.sleep(2000);
    driver.close();
  }
}
```

Scanned by CamScanner

* The above script is an example for Run-time Polymorphism.
* To execute the same scripts on multiple browsers we are creating a object of browser classes and upcasting into webdriver.
* According to Selenium Coding Standard, we will create the object of browser classes and upcast into webdriver.

Ex:-

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

23/01/18

Methods of Webdriver Interface:-

1. How do you enter URL?
   by using get() method
2. How do you get title of the webpage?
   by using getTitle() method
   * Return type of getTitle() method is String.
3. How do you get URL of a webpage?
   by using the method getCurrentUrl();
   * Return type of get current URL method is String.
4. How do you get source code of webpage.
   by using getPageSource() method.
   * Return type of getPageSource() method is String.

Scanned by CamScanner

Ex:-

```
public static void main (String[] args)
{
    // To open the browser
    String key = "webdriver.chrome.driver";
    String value = "./drivers/chromedriver.exe";
    System.setProperty(key, value);
    WebDriver driver = new ChromeDriver();

    // To enter URL
    driver.get("https://www.google.co.in/");

    // To get the title of the webpage
    String title = driver.getTitle();
    System.out.println(title);

    // To get the url of current webpage
    String url = driver.getCurrentUrl();
    System.out.println(url);

    // To get the source code of current webpage
    String pageSrc = driver.getPageSource();
    System.out.println(pageSrc);

    // To close the browser
    driver.close();
}
```

o/p:

⇒ Google
https://www.google.co.in/
source code .....

Scanned by CamScanner

Note :-

If the specified Url is invalid, then it will throw exception.

Write a script to get the size & position of the window and to maximize the window.

```
=> public static void main(String[] args)
{
    String key = "webdriver.chrome.driver";
    String value = "./driver/chromedriver.exe";
    System.setProperty(key, value);
    WebDriver driver = new FirefoxDriver();

    // To get the size of the window
    Dimension d = new Dimension(300, 300);
    driver.manage().window().setSize(d);

    // To get the position of the window
    Point p = new Point(250, 350);
    driver.manage().window().setPosition(p);

    // To maximize the window
    driver.manage().window().maximize();

}
```

Scanned by CamScanner

What are the difference between get() method & navigate() meth

* Using get() method we can just enter the Url.

* Using navigate method we can
  → enter the url
  → navigate to previous page
  → navigate to next page
  → refresh the webpage

Ex:-

```
public static void main(String[] args)
{
    String key = "webdriver.gecko.driver";
    String value = "./drivers/geckodriver.exe");
    System.setProperty(key, value);
    WebDriver driver = new FirefoxDriver();

    driver.get("https://www.google.co.in/");
    Thread.sleep(1000);

    // To enter url
    driver.navigate().to("https://www.facebook.com/");
    Thread.sleep(1000);
    // To navigate to previous page
    driver.navigate().back();
    Thread.sleep(1000);

// To navigate to next page
    driver.navigate().forward();
    Thread.sleep(1000);
```

Scanned by CamScanner

```
//Refresh the webpage
driver.navigate().refresh();
}
```

What is the difference between close() & quit() method

*. close() method will close only current browser.
*. quit() method will close all the browsers open by Selenium

Ex:-

```
public static void main(String[] args)
{
    String key = "webdriver.gecko.driver";
    String value = "./drivers/geckodriver.exe";
    System.setProperty(key, value);
    WebDriver driver = new FirefoxDriver();
    driver.get("https://www.naukri.com/");
    driver.quit();
}
```

14/01/18

Scanned by CamScanner

24/01/18

-: HTML :-

* Anything which is present on the webpage is called as web elements.

Ex :- Text field, button, list box, ... etc.

* These elements are developed by using HTML.
* HTML stands for HiperText Markup Language.

Components of HTML :-

* There are three different types of components

1) Tag
2) Attributes
3) Text

Tag :-

* Tags defines the type of the elements.

Ex :- input, a, div, body, etc...

<tag> → opening tag
</tag> → closing tag

Attributes :-

* Attributes modifies the default behaviour of elements.

Ex :- id, name, class, href .. etc.

* Each attribute will be having its own value.

Text :-

* Anything which is present inbetween opening tag & closing tag is called as text.

Ex :- Facebook, gmail ... etc.

* We can develope the webpage by using Notepad.

Scanned by CamScanner

* The Different types of locators are

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

Note :-

Out of these 8 locators, id, name, className are available as Attributes of an element.

Inspecting the Elements :-

* Fetching the source code which is used to develope an element is called as inspecting the elements.
* In order to inspect the elements in firefox browser we use an Add-on called Try-xPath
* To install the Add-on → to go menu & click on Add-on's.
  → click on extensions & search for Try-xPath.
  → click on instal button of Try-xPath & follow the default instructions.

~~→ To inspect~~

* To inspect the element right click on the element & click on inspect element.
* For security purpose in some of the application right click option will be disabled in such cases perform the following steps.

Scanned by CamScanner

Step-1 :- Press F12, which will open Developer tools

Step-2 :- Side click on pick an element icon { }.

Step-3 :- click on the element which is to be inspected.

Inspecting Elements in Chrome Browser :-

Here we use ChroPath Add-on to inspect the elements.

Step-1 :- In chrome browser go to menu, more tools & click on extensions

Step-2 :- click on get more extensions and search for chropath

Step-3 :- Click on Add to chrome & follow the default instructions

26/01/18

```
<html>
    <head>
        <title> welcome </title>
    </head>
    <body>
        UserName : <input type="text">
        password : <input type="password">
        <input type="submit" value="Login">
        <a href="https://gmail.com"> Inbox (3000) </a>
        <a href="https://facebook.com" id="link"
name="fb" class="ab" > Forgot Password ?? </a>
    </body>
</html>
```

Scanned by CamScanner

* In order to handle the single element we use findElement() method.
* Return type of findElement() is WebElement.
* If the specified locator is not matching with any element then findElement() will throws "NoSuchElementException".
* If the specified locator is matching with multiple elements then findElement() will return address of first matching element.

Ex:-

```
public static void main(String[] args)
{
    String key = "webdriver.gecko.driver";
    String value = "./drivers/geckodriver.exe";
    System.setProperty(key, value);
    WebDriver driver = new FirefoxDriver();
    driver.get("file:///C:/Users/Admin/Desktop/sample.html");

    // By using id
    driver.findElement(By.id("link")).click();
    driver.navigate().back();

    // By using name
    driver.findElement(By.name("fb")).click();
    driver.navigate().back();
    // By using ClassName
    driver.findElement(By.className("ab")).click();
    driver.navigate().back();
```

Scanned by CamScanner

```
// By using tagName
// performs action on matching element ie Zubox (10)
driver.findElement(By.tagName("a")).click();
driver.navigate().back();

// By using LinkText
driver.findElement(By.linkText("Forgot Password ??")).click();
driver.navigate().back();
// By using partialLinkText
driver.findElement(By.partialLinkText("Zubox")).click();

}
}
```

29/01/18

CssSelector :-

* If we cannot identify the element by using id, name, classname, linktext, partiallinktext & tagname, then we can identify that element by using cssSelector.

* The syntax for css selector is

tagname[attributeName = 'attributevalue']

Ex:- input[type = 'password']

String css = "input[type='password']";
driver.findElement(By.cssSelector(css)).sendKeys("rakesh123");

* Here, if we specify the tag name, it returns all the matching elements.

Scanned by CamScanner

* To verify the cssSelector Expression → go to try-xpath → select querySelectorAll option from way left box specify the cssSelector Expression in Expression field & click on Enter, which gives number of matching elements

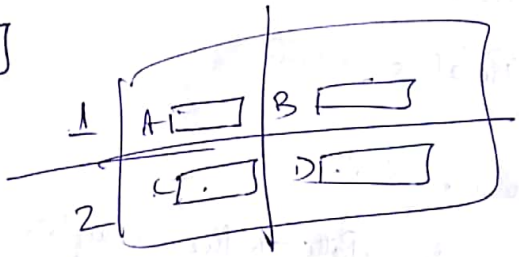
Note :-

* In cssSelector id can be represented by using #

Ex :- input [id='pw']
⇕
input #pw

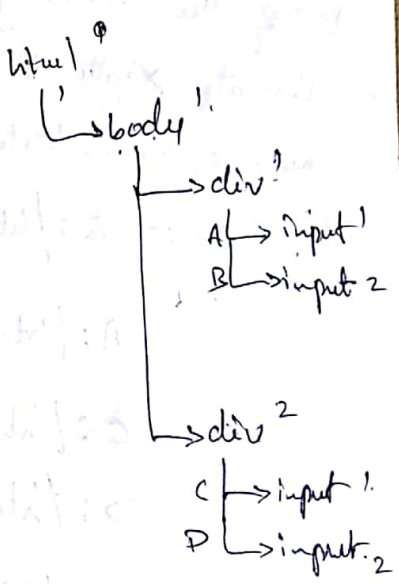
* In cssSelector class can be represented by using .

Ex :- input [class='pass']
⇕
input.pass



sample web page :-

```
<div>
    A: <input type="text">
    B: <input type="text">
</div>
<div>
    C: <input type="text">
    D: <input type="text">
</div>
```



Scanned by CamScanner

* Here we can represent the webpage in the form of html tree structure.

* In html tree each and every element contains the index value and the index value starts from 1.

* If there are similar type of childs under the same parent then the index value will be increased by 1, otherwise starts from 1.

Xpath :-

* XPath is the path of the element present in the webpage

* There are two types of XPath, they are
  1) Absolute XPath
  2) Relative XPath.

Absolute XPath :-

* Absolute XPath is the complete path of the element from the root of the webpage.

* Absolute XPath is represented by single forward slash (/) it means immediate child.

Ex :- B : /html/body/div[1]/input[2]

A : /html/body/div[1]/input[1]

C : /html/body/div[2]/input[1]

D : /html/body/div[2]/input[2]

AB : /html/body/div[1]/input

CD : /html/body/div[2]/input

AC : /html/body/div/input[1]

BD : /html/body/div/input[2]

Scanned by CamScanner

AD: /html/body/div[1]/input[1] | /html/body/div[2]/input[2]

BC: /html/body/div[1]/input[2] | /html/body/div[2]/input[1]

ABC: /html/body/div[1]/input | /html/body/div[2]/input[1]

ABD: /html/body/div[1]/input[1] | /html/body/div[2]/input[2]

ACD: /html/body/div[1]/input[1] | /html/body/div[2]/input

30-01-18

Note:-

*. we can concatinate the multiple xPath expression by using pipeline (symbol (|).

Relative xPath:-

*. It represents the any child which is present on webpage.

*. It is denoted by double forward slash (//), which means any child.

Ex:-

//input --- all the matching elements

//input[1] --- all 1st matching elements

A: //div[1] /input[1]

B: // div[1] /input[2]

C: // div[2] /input[1]

D: // div[2] /input[2]

AB: // div[1] /input

Scanned by CamScanner

CD: //div[2]/input

AC: //input[1]

BD: //input[2]

AD: //div[1]/input[1] | //div[2]/input[2]

BC: //div[1]/input[2] | //div[2]/input[1]

ABC: //div[1]/input | //div[2]/input[1]

ABD: //div[1]/input | //div[2]/input[2]

ACD: //input[1] | //div[2]/input[2]

BCD: //input[2] | //div[2]/input[1]

ABCD: //input

(1) What are the different types of xPath?

→ Absolute xPath

→ Relative xPath

(2) What is the difference between single forward slash (/) & double forward slash (//)?

*. / → represents the immediate child.

*. // → represents the any child present on the webpage

(3) Derive an xPath expression, which matches with all the links & all the images, present on the webpage?

*. //a → all the links present on the webpage

*. //img → all the images present on the webpage

Scanned by CamScanner

Q) what is the difference between //a and //img/a ? //img//a

* //a → represents all the links
* //img/a → represents all the immediate links present under all the images
* //img//a → represents all the links present under all the images

XPath by Attributes :-

Sample webpage :-

```
<div>
    A : <input type = "text" value "A" >
    B : <input type = "text" value "B" >
</div>
<div>
    C : <input type = "text" value "C" >
    D : <input type = "text" value "D" >
</div>
```

* To identify the specified elements, if we use the index it may not work properly, because whenever the position of the element changes, then the index value will also changes dynamically.
* To overcome the above problem, we can include attributes in place of index. which is called as XPath by Attributes

Scanned by CamScanner

* It is applicable for both absolute xPath and Relative xPath.

Syntax :-

tagName[@attributeName = 'attribute value']

Ex :-

Absolute → /html/body/div/input[@value = 'B']

Relative → //input[@value = 'B']

Assignment :-

① Derive the xPath to identify all the elements present in facebook login page.

⇒ ⓐ Email or phone :
//input[@name = 'email']

ⓑ password :
//input[@name = 'pass']

③ Login :
//input[@value = 'Login']

④ Forgotten Account :
//a[@href = 'https://www.facebook.com/recover/initiate?ev=110']

⑤ First Name :
//input[@name = 'firstname']

⑥ Surname :
//input[@name = 'lastname']

⑦ Mobile number or Email Address :
//input[@name = 'reg_email__']

⑧ New password :
//input[@name = 'reg_passwd__']

⑨ Date of Birth : ⓘ Date :
//select[@name = 'birthday-day']

ⓘⓘ Month :
//select[@name = 'birthday-month']

ⓘⓘⓘ Year :
//select[@name = 'birthday-year']

⑩ Radio Button : ⓘ Female :
//input[@value = '1']

ⓘⓘ Male :
//input[@value = '2']

⑪ create Account :
//button[@name = 'websubmit']

⑫ create a Page :
//a[@href = '/pages/create/?ref_type=registration_form']

Scanned by CamScanner

Xpath by Text :-

31/01/18

* If the specified element doesn't contain any Attribute and if it containing the text then we can identify the element by using xpath by text.

Syntax :-

```
tagName [text() = 'textvalue']
```

* It is applicable for both relative & Absolute xpath.

Ex :- `//td [text() = 'Java']`

* Text function can be replaced by using . (Dot).

Ex :-

```
//td [. = 'Java']
```

* If the textvalue ~~contains~~ contains space, the space is considered as a text

Ex :-

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

Xpath by contains() :-

* If the specified element is partially dynamic then we can identify the element by using xpath by contains().

Syntax :-

① If the text is partially dynamic

```
tagName [contains (text(), 'textvalue')]
```

② If the attribute is partially dynamic

```
tagName [contains (@attributename, 'attributevalue')]
```

Scanned by CamScanner

Handling   (non breakable space) :-

* While developing the applications developers can give the space in multiple ways such as spacebar, tab,   (non-breakable space)
* If the space is developed by using   statement then we can handle that element by using xpath by contains methods.

Sample web page :-

 facebook

Ex:- //a[contains(text(), 'facebook')]

① When we do use xpath by contains ?

~~We use~~ can use xPath by contains() if

→ the element partially dynamic

→ the element contains any special character such as nbsp

② How do you handle partial dynamic element ?

* By using xpath by contains()

Assignment :-

① Derive the xpath to hiddafy the below specified element present on selenium download page

C#
Ruby
Python
JavaScript Node
3.81

② Derive the xpath expression to identify the below specified element present on facebook login page

Scanned by CamScanner

Traversing :-

* Navigating from one element to another element in webpage using xpath is called Traversing.
* here, xpath user axis to navigate from one element to another element

There are 6 different types of axis :

1. child.
2. Parent
3. descendant
4. ancestor
5. following-sibling
6. Preceding-sibling

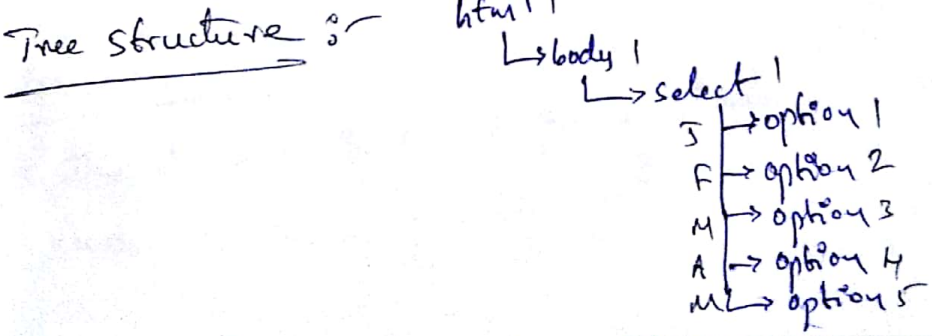
The Syntax for axis is :

```
/axis :: tagname
```

sample webpage :-

```
<select id = "Month">
    <option value = "J"> Jan </option>
    <option value = "F"> Feb </option>
    <option value = "M"> Mar </option>
    <option value = "A"> Apr </option>
    <option value = "M"> May </option>
</select>
```

Tree structure :-



Scanned by CamScanner

Child :-

*. It represents the immediate child of specified ele...

Ex:- /html/child::body/child::select/child::option

*. The shortcuts for child axis is / (forward slash)

Ex:- /html/body/select/option

Parent :-

*. It represents the immediate parent of specified element

Ex:- //option/parent::select/parent::body/parent::html

*. The shortcuts for parent axis is /..

Ex:- //option/../../../..

Assignment :-

① //td[text()='c#']
//td[text()='Ruby']
//td[text()='Python']
//td[text='JavaScriptNode']
//~~td~~a[contains(text(),'3.81')]

② //div[@class='521r fsm feq']

Scanned by CamScanner

01/08/18

Descendent :-

* It represents the any child present on the webpage.

Ex :- /html/descending::option[1]

* The shortcut for descendent is //

Ex :- /html//option[1]

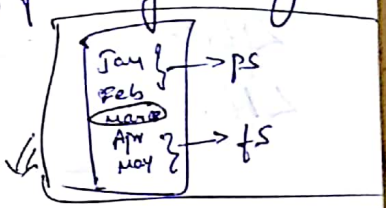
Ancestor :-

* It represents any parent is present on the webpage.

Ex :- //option[1]/ancestor::html

Note :-

There is no shortcuts for ancestor, following sibling and preceding sibling.



Following sibling :-

* It represents the elements which are present below the specified element under the same parent.

Ex :- //option[3]/following-sibling::option — A, M

//option[3]/following-sibling::option[1] — A

//option[3]/following-sibling::option[2] — M

Preceding Sibling :-

* The elements which are present above the specified element under the same parent.

Scanned by CamScanner

Ex:- //option[3]/preceding-sibling::option -- T, F

//option[3]/preceding-sibling::option[1] -- F

//option[3]/preceding-sibling::option[2] -- T

Sample web-page :-

```
<table border = "1">
<tr>
    <td> Padmavati </td>
    <td> 20 </td>
</tr>
<tr>
    <td> Rajakumara </td>
    <td> 50 </td>
</tr>
</table>
```

Independent - Dependent xPath :-

*. If the specified element is completely dynamic, then we can identify the element by using its nearest static elements which is called as independent - dependent xPath

Steps to Derive independent - dependent xPath :-

Step-1 :- Identify the nearest static elements and note down the xPath.

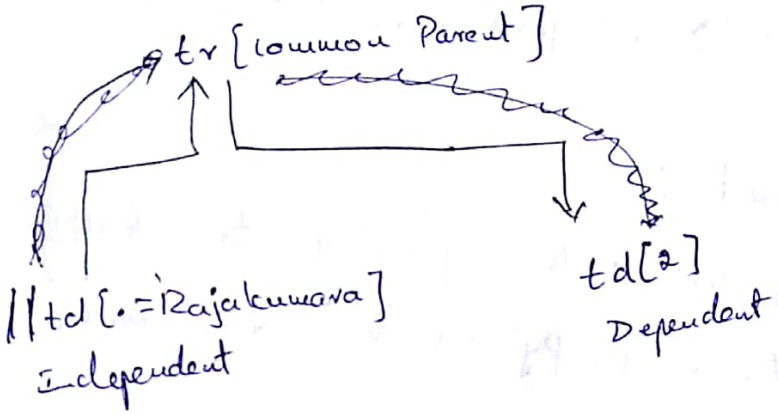
*. Place the cursor on source code of independent element and move it in upward direction, where it meets both independent & dependent elements.

Scanned by CamScanner

*. The place where both independent & dependent elements meets it called as common parent

Step-2 :- *. press the down arrow key until it highlights dependent element.

** ~~//td[.='Rajakumara']~~



:. From the above created ~~tree~~ tree structure derive an xPath which contains Independent element, common parent & dependent element.

Ex:- //td[text()='Rajakumara']/../td[2]

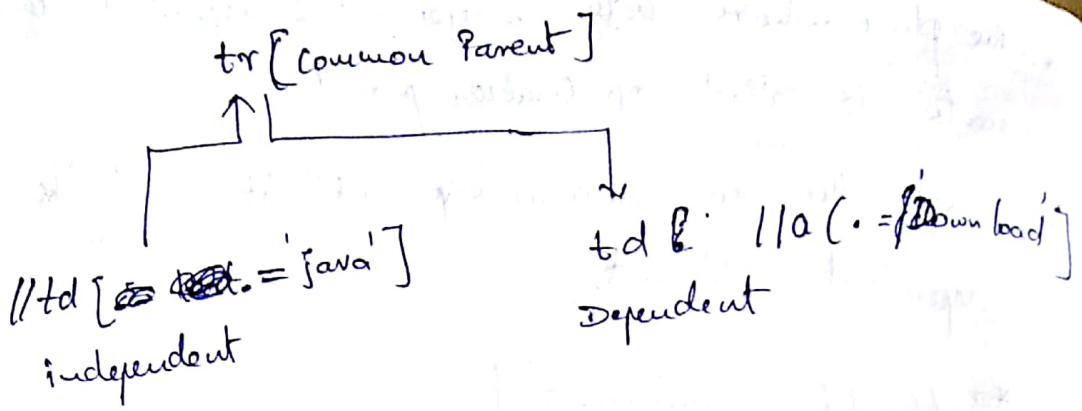
//td[text()='Rajakumar']/following-sibling::td

Handling Duplicate Element :-

*. If the specified element is duplicate, then we can identify by using independent-dependent xPath.

Ex:- Derive an XPath expression to identify the download button of Java language.

Scanned by CamScanner



Ex:- //td [. = 'Java'] /.. // a [. = 'Download']

Assignment :-

① Derive the xPath expression to identify download link of C#, Ruby, Python & Javascript(Node).

C# → //td [. = 'C#'] /.. // a [. = 'Download']

Ruby → //td [. = 'Ruby'] /.. // a [. = 'Download']

Python → //td [. = 'Python'] /.. // a [. = 'Download']

Javascript(Node) → ~~//td [. = 'Javascript'] /.. // a [. = 'Download']~~

//td [. = 'Javascript (Node)'] /.. // a [. = 'Download']

02-02-18

XPath by Group Index :-

*. If we can't identify the elements even after using independent - dependent xPath, then we can identify that element by using xpath by Group Index.

*. In X-path by group index, first it will execute the expression which is present in the paranthesis and stores all the matching elements in virtual xpath array, and index will be applied.

Scanned by CamScanner

* The index value starts from 1.

Ex:-

(//input)[1] → VXA

A	1
B	2
C	3
D	4

* //a → all the links
* //a[1] → all 1st matching links
* (//a)[1] → 1st matching element
* (//a)[5] → 5th matching element
* (//a)(last()) → last matching element

sample web page: (checkbox)

```
<table border="1">
<tr>
    <td>A:</td>
    <td><input type="checkbox"></td>
</tr>
<tr>
    <td>B:</td>
    <td><input type="checkbox"></td>
</tr>
<tr>
    <td>C:</td>
    <td><input type="checkbox"></td>
</tr>
<tr>
    <td>D:</td>
    <td><input type="checkbox"></td>
</tr>
</table>
```

Scanned by CamScanner

* //input[@type='checkbox'] --- all matching elements
* //input[@type='checkbox'][1] --- all 1st matching element
* (//input[@type='checkbox'])[1] --- 1st matching element
* (//input[@type='checkbox'])[5] --- 5th matching element
* (//input[@type='checkbox'])[last()] --- last matching element
* (//input[@type='checkbox'])[position()=6] --- element at position 6
* (//input[@type='checkbox'])[position() mod 2=0] --- elements at even position
* (//input[@type='checkbox'])[position() mod 2=1] --- elements at odd position
* (//input[@type='checkbox'])[position()=1 or position()=last()] --- First & last element
* (//input[@type='checkbox'])[position()>last()-3] --- last 3 element
* (//input[@type='checkbox'])[position()<4] --- 1st three element

Note:-

* The most prefered locator are
  - → id
  - → name
  - → linkText
  - → Xpath
    - → Absolute
    - → Relative
      - → By Attribute
      - → By text()
      - → contains()
      - → Independent - Dependent Xpath
      - → GroupIndex

Scanned by CamScanner

1. Write a scripts to login to Actitime application.

```
public class Demo
{
    static
    {
        System.setProperty("webdriver.gecko.driver", "./drivers/geckodriver.exe");
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
    }
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter a username:");
        String un = scan.nextLine();

        System.out.println("Enter Password:");
        String pw = scan.nextLine();

        WebDriver driver = new FirefoxDriver();

        driver.get("https://demo.actitime.com/login.do");
        driver.get("https://demo.actitime.");
        driver.findElement(By.id("username")).sendKeys(un);
        driver.findElement(By.name("pwd")).sendKeys(pw);
        String xp = "//div[.='Login']";
        driver.findElement(By.xpath(xp)).click();
    }
}
```

Scanned by CamScanner

Assignment :-

② Write a script to login to

1). Facebook

2). Gmail.

① FaceBook :-

```
package qsp;
import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class FacebookLogin
{
  static
    { System.setProperty("webdriver.gecko.driver", "./drivers/
geckodriver.exe");
    }
public static void main(String[] args)
{
  Scanner sc=new Scanner(System.in);
  System.out.println("Enter User Name:");
  String un = sc.nextLine();
  System.out.println("Enter password:");
  String pw=sc.nextLine();
  WebDriver driver = new FirefoxDriver();
  driver.get("https://www.facebook.com/");
```

Scanned by CamScanner

```
driver.findElement(By.id("email")).sendKeys(un);
driver.findElement(By.id("pass")).sendKeys(pw);
String xp1 = "//input[@value='LogIn']";
driver.findElement(By.xpath(xp1)).click();
}
}
```

② Gmail :-

```
package qsp;
public class GmailLogin
{
    static
    {
        System.setProperty("webdriver.gecko.driver", ".\\drivers\\geckodriver.exe");
    }
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in)
        System.out.println("Enter User name:");
        String un = sc.nextLine();
        System.out.println("Enter password:");
        String pw = sc.nextLine();
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.gmail.com/");
        driver.findElement(By.id("      ")).sendKeys(un);
        driver.findElement(By.·          ).sendKeys(pw);
```

Scanned by CamScanner

05/02/18

Methods of Web Element :-

1. sendKeys();
2. click();
3. clear()
4. findElement()
5. findElements()
6. getAttribute()
7. getCssValue()
8. getLocation()
9. getRect()
10. getSize()
11. getTagName()
12. getText()
13. isDisplayed()
14. isEnabled()
15. isSelected()
16. submit()

Scanned by CamScanner

write a script to copy-paste the text from one text field to another field.

=> public static void main (String[] args)

```
{
    WebDriver driver = new FirefoxDriver();
    driver.get("file:///C:/Users/Admin/Desktop/sample.html");
    WebElement fn = driver.findElement(By.id("uu"));
    fn.sendKeys("rakesh");

    //To select & copy
    fn.sendKeys(Keys.CONTROL + "ac");
    WebElement ln = driver.findElement(By.id("pw"));

    //To paste
    ln.sendKeys(Keys.CONTROL+"v");

}
```

How do you click on a Element ?

→ by using click()

How do you clear the text field ?

→ by using clear()

write a script to clear the text field without using clear() ?

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

Scanned by CamScanner

```
WebDriver driver = new FirefoxDriver();
driver.get("file:///C:/Users/Admin/Desktop/sample.html");
WebElement fn=driver.findElement(By.id("uu"));
fn.sendKeys("rakesh");
Thread.sleep(1000);
fn.sendKeys(Keys.CONTROL+"a");
fn.sendKeys(Keys.Back-Space);
}
```

Write a script to get the value of an attribute ?

*. We can get the value of an attribute by using getAttribute().

Ex:-

```
public static void main(String[] args)
{
    WebDriver driver = new FirefoxDriver();
    driver.get("file:///C:/Users/Admin/Desktop/sample.html");
    WebElement fn = driver.findElement(By.id("uu"));
    String v=fn.getAttribute("value");
    System.out.println(v);
}
```

Scanned by CamScanner

Write a script to get colour and size of the font.

*. we can get the Css values by using getCss the method getCssValue().

Ex:-

```
public static void main(String[] args)
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://demo.actitime.com/login.do");

    String xp = "//div[.='Login']";
    WebElement login = driver.findElement(By.xpath(xp));

    String c = login.getCssValue("color");
    String s = login.getCssValue("font-size");
    String st = login.getCssValue("font-style");
    System.out.println(c);
    System.out.println(s);
    System.out.println(st);
    driver.close();
}
```

How do yo get Location (x-axis & y-axis) of an Element?

→ by using getLocation()

Ex:- public static void main (String[] args)
{

Scanned by CamScanner

```
WebDriver driver = new FirefoxDriver();
driver.get("https://demo.actitime.com/login.do");
WebElement user = driver.findElement(By.id("username"));

Point p = user.getLocation();
int x = p.getX();
int y = p.getY();
S.o.p(x+"\n"+y)

driver.close();
}
```

→ methods present in point class

Write a script to verify whether the username & password fields are aligned properly.

```
⇒ main(-)
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://demo.actitime.com/login.do");
    WebElement user = driver.findElement(By.id("username"));
    WebElement pass = driver.findElement(By.name("pwd"));
    int x1 = user.getLocation().getX();
    int x2 = pass.getLocation().getX();

    if (x1==x2)
    {
        S.o.p("aligned");
    }
    else
```

Scanned by CamScanner

```
{
    s.o.p("not aligned");
}
driver.close();
}
```

How do you get size (Height & width) of an element ?

-> by using getSize()

Ex:-

```
=> main()
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://demo.actitime.com/login.do");
    WebElement user = driver.findElement(By.id("username"));

    Dimension d = user.getSize();
```

Dimension → class

→ methods present in a Dimension class

```
    int w = d.getWidth();
    int h = d.getHeight();

    s.o.p(w+"\n"+h);
    driver.close();
}
```

Write a script to check size of username & password field are same (or) not ?

```
=> main()
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://demo.actitime.com/login.do");
```

Scanned by CamScanner

```
WebElement user = driver.findElement(By.id("username"));
WebElement pass = driver.findElement(By.name("pwd"));

int w1 = user.getSize().getWidth();
int h1 = user.getSize().getHeight();

int w2 = pass.getSize().getWidth();
int h2 = pass.getSize().getHeight();

if (w1==w2 && h1==h2)
{
    s.o.p("size is same");
}
else
{
    s.o.p("size is not same");
}
driver.close();
}
```

How do yo get text of an element ?

→ By using getText().

Ex:-

```
public static void main(String[] args)
{
WebDriver driver = new FirefoxDriver();
driver.get("http://demo.actitime.com/login.do");
String xp = "//nobr[contains(text(),'actiTIME')]";
WebElement version = driver.findElement(By.xpath(xp));
```

Scanned by CamScanner

```
String text = version.getText();
s.o.p(text);
driver.close();
}
```

what is the difference between getTitle() & getText():

→ getTitle() is used to get title of the webpage.

→ getText() is used to get the text of an Element.

06/02/18

How do you check, whether the element is displayed or not?

→ by using isDisplayed()

Ex:-

```
public static void main(String[] args)
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://demo.actitime.com/login.do");
    String xp = "//div[.='Login']";
    WebElement login = driver.findElement(By.xpath(xp));
    boolean v=login.isDisplayed();
    if(v)
    { s.o.p("element is displayed");
    }
    else
    { s.o.p("element is not displayed");
    }
    driver.close();
}
```

Scanned by CamScanner

if the specified element is not locating then it throws Exception, this can be avoided by using try-catch block

```
public static void main(String[] args)
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://demo.actitime.com/login.do");
    try
    {
        String xp = "//div[.='Login123']";
        WebElement login = driver.findElement(By.xpath(xp));
        boolean v = login.isDisplayed();
        if(v)
        {
            s.o.p("element is displayed");
        }
        else
        { s.o.p("element is not displayed");
        }
    }
    catch (Exception e)
    {
        s.o.p("element is not displayed:2");
    }
    driver.close();
}
```

Scanned by CamScanner

How do you check whether the element is enabled or not

→ by using isEnabled().

Ex:-

```
public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/sample.html");
    WebElement user = driver.findElement(By.id("un"));
    if(user.isEnabled())
    {
        s.o.p("enabled");
    }
    else
    {
        s.o.p("disabled");
    }
    driver.close();
}
```

*** Write a script for the following scenario.

There are two text fields First name & Last name, where in the value in firstname is ~~Rahul~~ rakesh & it is disabled write a script to get the value from firstname and stored it in last name.

⇒

```
public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/sample.html");
```

Scanned by CamScanner

```
WebElement user = driver.findElement(By.id("un"));
String v = user.getAttribute("value");
driver.findElement(By.id("pw")).sendKeys(v);
}
```

How do you check whether the element is selected or n

→ by using isSelected()

Ex:-

```
public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    WebElement cb = driver.findElement(By.name("remember"));
    if(cb.isSelected());
    {
        s.o.p("selected");
    }
    else
    {
        s.o.p("Not Selected");
    }
    driver.close();
}
```

Assignment:-

Write a script to check whether male (or) female radio button, in facebook login page is selected (or) not.

Scanned by CamScanner

```
public static void main(String[] args)
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://www.facebook.com/");
    WebElement radio1 = driver.findElement(By.xpath("//input[@value='1']"));
    WebElement radio2 = driver.findElement(By.xpath("//input[@value='2']"));

    if(radio1.isSelected() || radio2.isSelected())
    {
        S.o.p("selected");
    }
    else
    {
        S.o.p("not selected");
    }
    driver.close();
}
```

07/02/18

What where do we use submit() ~~method~~ ?

→ We use submit() to click on an element, if the type of the element is submit.

What are possible ways to click on an element ?

→ click()
→ submit()
→ sendKeys()
→ JavaScript

Scanned by CamScanner

The most important methods are

1) sendKeys()
2) click()
3) getAttribute()
4) getText()
5) submit()

:- Collections :-

* Collection is an interface, where we can store any type of objects.
* Collection interface is inherited by other interfaces, such as List, Set & Queue.

List :-

* List is an interface, which inherits collection interface.
* List allows duplicates and also Null values.
* List follows the insertion order and it is index based.
* List is implemented by ArrayList, LinkedList and Vector.
* In List we can add the data using add(), we can get the size using ~~getSize~~ size(), we can get the value using get().

Ex:-

```
public class JavaDemo
{
    public static void main(String[] args)
    {
        List<String> al = new ArrayList<String>();
```

Scanned by CamScanner

```
//To add
al.add("a");
al.add("b");
al.add("c");
//To get the size
int count = al.size();
System.out.println(count);
//To get the data
for(int i=0; i<count; i++)
{
    String s = al.get(i);
    System.out.println(s);
}
}
```

Handling Multiple Elements :-

* We can handle the multiple elements by using findElements()
* The return type of findElements() is List of WebElement.
* In findElements(), if the specified ~~method~~ locator is not matching with any element, then it will return empty list(0).
* In findElements(), if the specified locator is matching with multiple element, then it will return address of all the matching elements.

Scanned by CamScanner

Sample webpage :-

```
<a href="http://www.qspiders.com"> QSpiders </a>
<a href="http://www.jspiders.com"> Jspiders </a>
```

Write a Script to count and print all the links present on the webpage ?

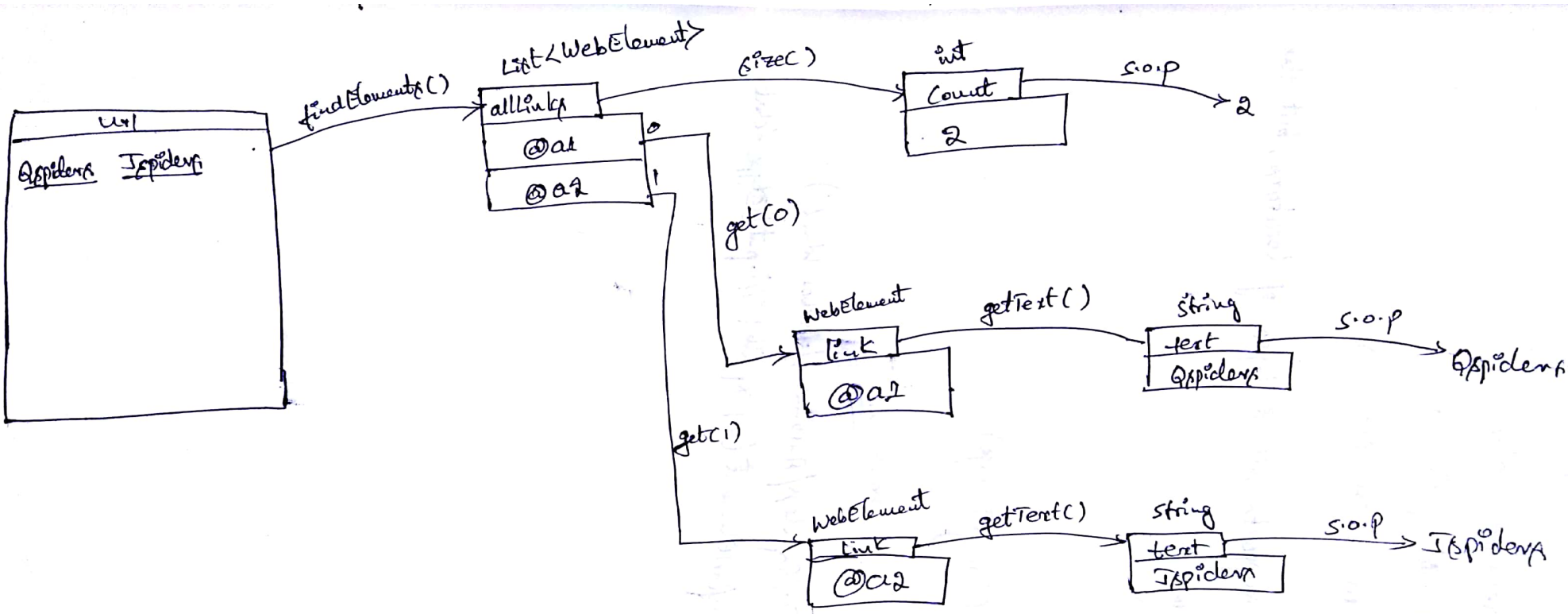
```
public static void main (String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get ("file:///c:/Users/Admin/Desktop/Links.html");
    List<WebElement> allLinks = driver.findElements(By.xpath(
                                "//a"));

    int count = allLinks.size();
    System.out.println(count);

    for (int i=0; i<count ; i++)
    {
        WebElement link = allLinks.get(i);
        String text = link.getText();
        System.out.println(text);
    }
    driver.close();
}
```

(or)

For each loop

```
for(WebElement link : allLinks)
{
    String text = link.getText();
    System.out.println (text);
}
```

Scanned by CamScanner



Scanned by CamScanner

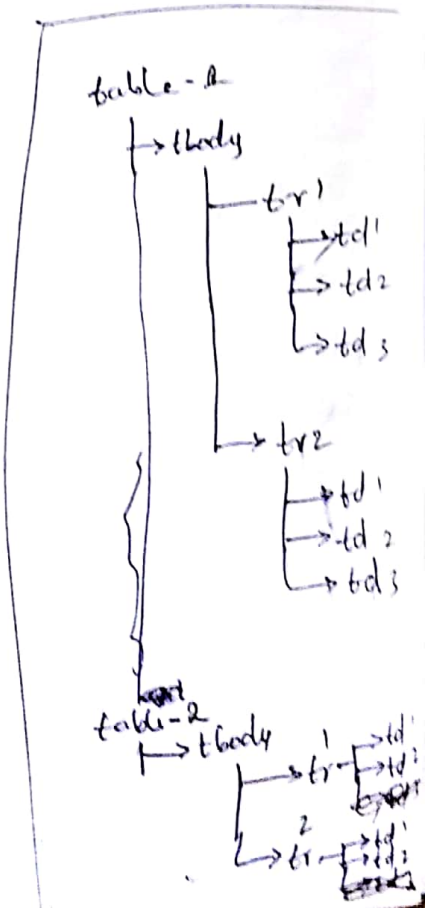
1) What are the difference between findElement() & findElements()

findElement()	findElements()
1) It is used to handle single element	1) It is used to handle multiple elements
2) Return type is WebElement	2) Return type is List<WebElement>
3) If the specified locator is not matching with any element, then it will throw NoSuchElementException	3) If the specified locator is not matching with any element, then it will throw an Empty list
4) If the specified locator is matching with multiple elements then it returns the address of first matching element	4) If the specified locator is matching with multiple elements, then it returns address of all matching elements

Handling Table :-

Simple web page :-

```
<table border="1" id="t1">
    <tr>
        <td>1</td>
        <td>java</td>
        <td>200</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Selenium</td>
        <td>250</td>
    </tr>
</table>
<table id="t2" border="1">
```



Scanned by CamScanner

```
<tr>
    <td> qsp </td>
    <td> 20 </td>
</tr>
<tr>
    <td> jsp </td>
    <td> 25 </td>
</tr>
</table>
```

Write a script to count number of tables ?

```
public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("file:///C:/Users/Admin/Desktop/Links.html");
    List<WebElement> allTables = driver.findElements(By.xpath("//table"));
    int count = allTables.size();
    System.out.println(count);
    driver.close();
}
```

write a script to count number of rows in a table ?

```
public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("file:///C:/Users/Admin/Desktop/Links.html");
```

Scanned by CamScanner

```
String xp = "//table[@id='t1']//tr";
List<WebElement> allRows = driver.findElements(By.xpath(xp));
int count = allRows.size();
System.out.println(count);
driver.close();
}
```

(3) Write a script to count number of cells present in a table?

⇒ "//table[@id='t1']//td"

(4) Write a script to count number of columns present in a table?

⇒ //table[@id='t1']//tr[1]/td

(5) Write a script to print contents of a table?

```
⇒ public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("file:///C:/Users/Admin/Desktop/links.html");
    List<WebElement> allCells = driver.findElements(By.xpath("//t
                                [@id='t1']//td"))

    int count = allCells.size();
    for(WebElement cell : allCells)
    {   String text = cell.getText();
        S.O.P(text);
    }
} driver.close()
```

Scanned by CamScanner

6) Write a script to count number of tables without using xpath?

→ by using tag name

⇒ public static void main(String[] args)

{

WebDriver driver = new ChromeDriver();

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

driver.get("file:///c:/Users/Admin/Desktop/Links.html");

link<WebElement> allTables = driver.findElements(By.tagName("table"));

int count = allTables.size();

S.o.p(count);

driver.close();

}

7) Count number of rows ~~without caps~~ in a table without using xpath ?

8) What is the difference b/w driver.findElement() & element.findElement() ?-

⇒ *. driver.findElement() will search for the element within the webpage.

*. element.findElement() will search for the element within the specified element.

Scanned by CamScanner

Ex:-

```
main(-)
{
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("file:///C:/Users/Admin/Desktop/links.html");

    // Storing address of table 2 in element
    WebElement element = driver.findElement(By.id("t2"));

    // search for the element within HTML page
    WebElement a = driver.findElement(By.tagName("td"));
    s.o.p(a.getText());

    // search for the element within specified element (element
    WebElement b = element.findElement(By.tagName("td"));
    s.o.p(b.getText());
    driver.close();
}
```

(7) =>

```
main(-)
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/links.ht
    WebElement element = driver.findElement(By.id("t
    WebElement element = driver.findElement(By.id("t
    List<WebElement> allRows = element.findElements(By.tagNam
```

Scanned by CamScanner

```
int count = allRows.size();
System.out.println(count);
}
```

(9) Write a script to count number of cells present in a table without using xpath ?

(10) Count number of columns of a table without using xpath?

(11) print contents of a table without using xpath ?

(12) print only numbers present on the web page ?

⇒

```
main(-)
{
  int c=0, sum=0;
  WebDriver driver = new ChromeDriver();
  driver.get("file:///c:/Users/Admin/Desktop/Links.html");
  List<WebElement> allCells = driver.findElements(By.xpath("//td"));
  for (WebElement cell : allCells)
  {
    String text = cell.getText();
    try
    {
      int n = Integer.parseInt(text);
      s.o.p(n);
      c++;
      sum = sum + n;
    }
```

Scanned by CamScanner

```
        catch(Exception e)
        {

        }
    }
    S.o.p("total Number of integer contents " + c);
    S.o.p("Sum of digits: " + sum);
}
```

(9) ⇒

```
p.s.v.m (String[] args)
{
    WebDriver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/links.html");
    WebElement element = driver.findElement(By.id("t1"));
    List<WebElement> allCells = element.findElements(By.tagName("td"
    int count = allCells.size();
    S.o.p(count);
    driver.close();
}
```

(10) ⇒ public static void main(String[] args)

```
{ {
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/Links/html");
    WebElement element = driver.findElement(By.id("t1"));
    List<WebElement> allCells = element.findElements(By.tagNa
    List<WebElement> allColumns = allCells.get(0).
                findElements(By.tagName("td"));
```

Scanned by CamScanner

```
int count = allColumns.size();
s.o.p(count);
driver.close();
}
```

(14)

```
public static void main(String[] args)
{
WebDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/Admin/Desktop/links.html");
WebElement element = driver.findElement(By.id("t1"));
List<WebElement> contents = element.findElements(By.
                    tagName("tbody"));
for (WebElement allContents : contents)
{
  String ch = allContents.getText();
  s.o.p(ch);
}
driver.close();
}
```

Scanned by CamScanner

Synchronization :

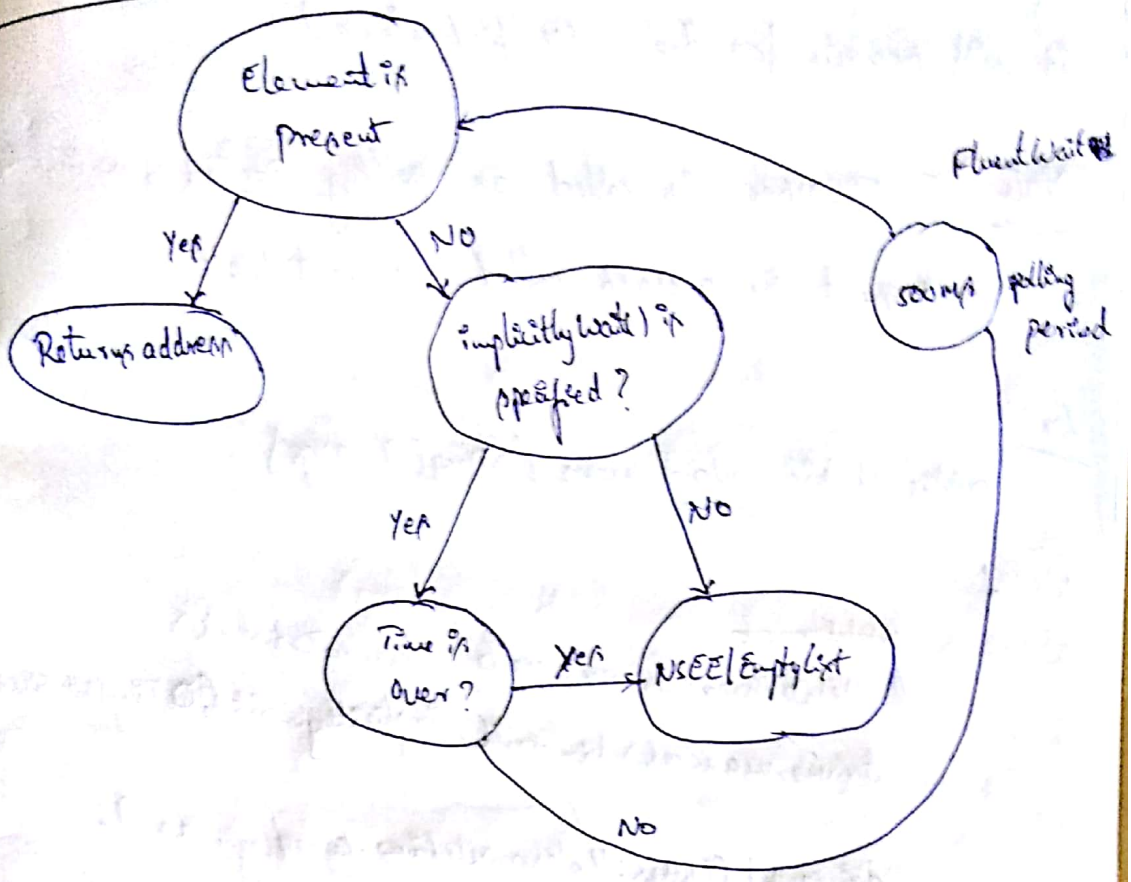
* Matching the speed of selenium with the speed of application is called as Synchronization
* We can handle synchronization by using
  1) implicitlyWait ( )
  2) explicitWait ( )
  3) Thread.sleep ( )

ImplicitlyWait() :-

* implicitlyWait ( ) will take two arguments, which are of type long & timeunit.
* In the first argument, we have to specify the waiting time and in the second argument, we have to specify the time unit.
* The different time units are
  - → DAYS
  - → HOURS
  - → MINUTES
  - → SECONDS
  - → MILLISECONDS
  - → MICROSECONDS
  - → NANOSECONDS

Scanned by CamScanner

Work Flow :-



*. When the control comes to findElement() or findElements(), it will check whether the specified element is present or not.

*. If the specified element is present, then it will returns address of matching elements.

*. If the specified element is not present, then it will check whether the implicitlyWait() is specified or Not.

*. If the implicitlyWait() is not specified, then it will throw NoSuchElementException or EmptyList.

*. If the implicitlyWait() is specified, then it will check whether the specified time is over or not.

*. If the specified time is over, then it will throw NoSuch Element Exception or EmptyList.

Scanned by CamScanner

* If the specified time is not over, then for every 500milliseconds it will search for the specified element.

Note :- 500msec is called as polling period, which is present in a class called Fluent Wait.

Ex :-

```
public static void main (String[] args)
{
    WebDriver driver = new ChromeDriver()
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get ("https://demo.actitime.com/login.do");
    driver.findElement(By.id("username")).sendKeys("admin");
    driver.findElement(By.name("pwd")).sendKeys("manager");
    driver.findElement(By.xpath("//div[.='Login ']")).click();
    driver.findElement(By.id("logoutLink123")).click();
}
```

Note :-

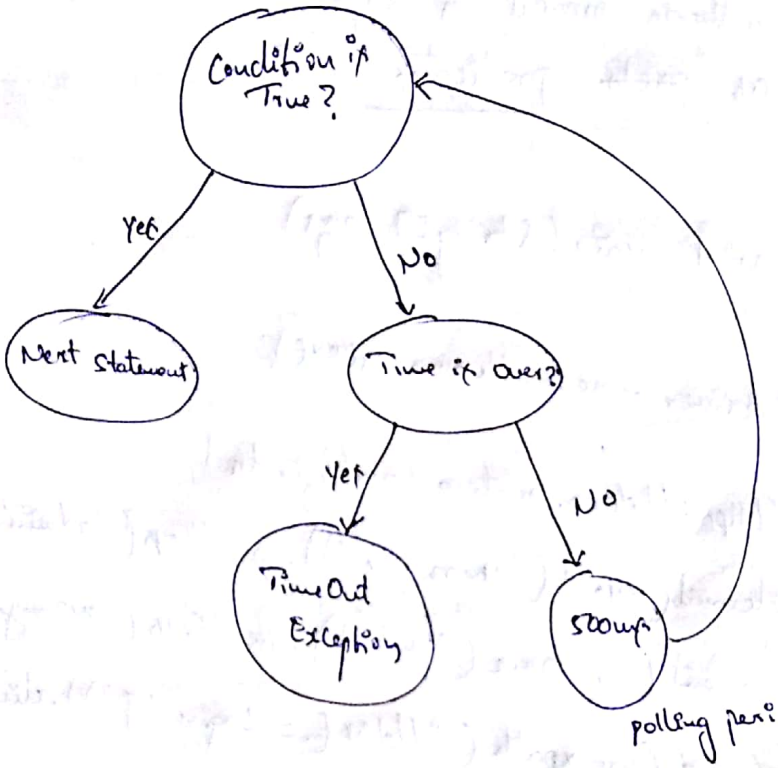
implicitlyWait() will handle the Synchronization of findElement() & findElements() only.

Scanned by CamScanner

Explicitly wait() :-

* using explicitlyWait() we can handle the synchronization of any methods, including findElement() & findElements().

Work Flow :-



* WebDriverWait class is called as explicitWait, which takes two arguments of type WebDriver and long.
* During the run time when the control comes to wait statements it will check, whether the specified condition is true (or) false.
* If the condition is true, then it will execute the next statement.
* If the condition is false, then it will check whether the specified Time is over (or) not.
* If the specified time is over, Then it will throw TimeOut Exception.

Scanned by CamScanner

* If the specified time is not over, then for every 500ms it will check, whether the specified condition is true (or) false.

Note: ① Here also, 500msec is called as polling period, which is present in a class called fluent Wait.

② All the methods present in expected conditions class are called as ~~preds~~ predicates.

Ex:-

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

    driver.get("https://demo.actitime.com/login.do");
    driver.findElement(By.id("username")).sendKeys("admin");
    driver.findElement(By.name("pwd")).sendKeys("manager");
    driver.findElement(By.xpath("//div[.='Login ']")).click();

    try
    {
        WebDriverWait wait = new WebDriverWait(driver, 10);
        wait.until(ExpectedConditions.titleIs("actiTime - Enter Time
                                               Track"));

        s.o.p("pass");
    }
    catch(Exception e)
    {
        s.o.p("fail");
    }
    driver.close();
}
```

Scanned by CamScanner

What are the difference between implicitWait() & explicitWait()

implicitWait	explicitWait
1) It handles the synchronization of findElement() & findElements()	1) It handles the synchronization of any method including findElement() & findElements().
2) Here, we do not specify any condition.	2) Here, we have to specify the condition explicitly
3) Even after the specified time duration, if the elements are not found, then it will throw NoSuchElementException (or) EmptyList.	3) Even after the specified time duration, if the condition is not true, then it will throws TimeOutException
4) Here the time units are DAYS, HOURS, MINUTES, SECONDS, MILLI-SECONDS, MICROSECONDS, NANO-SECONDS	4) Here the time unit is only SECONDS

Interview questions :-

1. What is synchronization ?? How do you handle it ?
2. What are the different possible ways to handle synchronization ??
3. Explain work flow of implicitlyWait() ??
4. Explain work flow of explicitWait() ??
5. What are predicates ?
6. How do you handle synchronization of findElement() & findElements() ??
7. How do you handle synchronization of getTitle() & any other ??

Scanned by CamScanner

Handling AutoSuggestions :-

12/02/18

→ Auto Suggestions can be handled by using findElements()

Ex:- ① Write a script for the following scenario.

Step-1 : Navigate to Google website

Step-2 : Enter Appidera in SearchBox

Step-3 : Count Number of AutoSuggestions.

Step-4 : print all the autoSuggestions.

Step-5 :- click on last AutoSuggestions.

⇒ main()

{

```
WebDriver driver = new ChromeDriver();
driver.manage().timeouts().implicitlyWait(10, TIMEUNIT.
                                          SECONDS);

driver.get("https://www.google.co.in/");
driver.findElement(By.id("lst-ib")).sendKeys("appide
Thread.sleep(1000);
// String address of all autoSuggestions
List<WebElement> allSuggestions = driver.findElements(By.
                                  className("sbqs_c"));

// To count number of autoSuggestions.
int count = allSuggestions.size();
S.o.p(count);
// To print all autoSuggestions.
```

Scanned by CamScanner

```
for(WebElement Suggestions : allSuggestions)
{
    String text = Suggestions.getText();
    Sop(text);
}
// To click on Last Suggestion.
allSuggestions.get(count-1).click();

driver.close();
}
```

② Write a Script for the following Scenario.

Step-1: Navigate Google

Step-2: Search for Gspiders

Step-3: Click on Gspidersbanawadi

```
=> main()
{
WebDriver driver = new ChromeDriver();
driver.manage().timeouts().implicitlyWait(10, TIMEUNIT.SECONDS);
driver.get("https://www.google.co.in");
driver.findElement(By.id("lst-ib")).sendKeys("gspiders");
Thread.Sleep(1000);

// Storing address of all auto Suggestions.
List<WebElement> allSuggestions = driver.findElements
                    (By.className("sbqs_c"));
```

Scanned by CamScanner

```
// To click on appdera banawadi
for(WebElement suggestions : AllSuggestions)
{
    String text = suggestions.getText();
    if(Text.equals("appdera banawadi"))
    {
        suggestions.click();
        break;
    }
}
}
```

Take Screen Shot :-

(*) Write a script to take the screenshot of a webpage

```
=> main()
{
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10,
                    TIMEUNIT.SECONDS);
    driver.get("https://www.google.co.in");

Interface → TakeScreenshot t = (TakeScreenshot) driver;

    // To take the Screenshot
    File src = t.getScreenshotAs(OutputType.FILE)

    // Destination Folder
class → File dest = new File("./screenshot.png");
```

Scanned by CamScanner

//copy files from "src to dest"

FileUtils.copyFile(src, dest);

driver.close();

}

② Write to take the screenshot of a webpage without using TakesScreenshot(I)

=> EventFiringWebDriver e = new EventFiringWebDriver(driver);

14-02-18

Handling List Box (Dropdown List Box) :-

*. List box can be handled by using "Select class" only if it is developed by using "Select Tag".

*. List box contains a Constructor which takes an argument of type "WebElement"

*. List box should be imported from "org.openqa.selenium.support.ui.Select" package.

*. List box contains "11" different types of methods, they are

1) selectByIndex()
2) selectByValue()
3) selectByVisibleText()

selectOption

4) deselectByIndex()
5) deselectByValue()
6) deselectByVisibleText()
7) deselectAll()

deselect option

8) getAllSelectedOptions()
9) getFirstSelectedOption()
10) getOptions()
11) isMultiple().

Scanned by CamScanner

Sample Web page :-

```
<Select id = "slv" >
        <Option value = "i" > Idly </option>
        <option value = "v" > Voda </option>
        <option value = "d" > Dosa </option>
        <option value = "p" > Pongal </option>
        <option value = "p" > Pongal </option>
</Select >
```

Ex :-

```
public static void main(String[] args) throws InterruptedException
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///c:/Users/Admin/Desktop/slv.html");
    WebElement hotel = driver.findElement(By.id("slv"));

    Select sel = new Select(hotel);
    Thread.sleep(1000);                          default Idly
    // By using index
    sel.selectByIndex(1);                        // voda
    Thread.sleep(1000);
    // By using values
    sel.selectByValue("d");                      // Dosa
    Thread.sleep(1000);
    // By using visible text
    sel.selectByVisibleText("Pongal");           // pongal (1st ow)
}
```

Scanned by CamScanner

Note :-

* If we specify invalid index (or) value (or) visibletext it will throw NoSuchElementException.

* Value & visibletext are case sensitive.

* In single Select Listbox, If the specified option is matching with duplicate, then it will select first matching option.

* we can avoid the duplicates by using index values.

* In single select Listbox, we can't deselect the options.

* If we try to deselect, then it will throw "Unsupported operation Exception".

Handling Multiselect Listbox :-

Sample Web page :-

```
<select id = "slv" multiple>
    <option value = "i" > Idly </option>
    <option value = "v" > Vada </option>
    <option value = "d" > Dosa </option>
    <option value = "p" > Pongal </option>
    <option value = "p" > Pongal </option>
</select>
```

① Write a script to select & deselect the Options?

```
=> main ( )
{
    WebDriver driver = new ChromeDriver();
```

Scanned by CamScanner

```
driver.get("file:///C:/Users/Admin/Desktop/plv.html");
WebElement hotel = driver.findElement(By.id("plv"));
Select sel = new Select(hotel);
Thread.sleep(1000);
//select by using index
sel.selectByIndex(1);
Thread.sleep(1000);
//select By using value
sel.selectValue("d");
Thread.sleep(1000);
//select By using visible Text
sel.selectVisibleText("Pongal");
Thread.sleep(1000);
//de-select By using visible text
sel.deselectByVisibleText("Pongal");
Thread.sleep(1000);
//de-select By using value
sel.deselectByValue("d");
Thread.sleep(1000);
//de-select By using index
sel.deselectByIndex(1);
}
```

sel.deselectAll(); // will deselect all the options.

Scanned by CamScanner

Note :-

1. In multi select list box, if the specified option is duplicate, then it will select all the matching options.

2. We can de-select all the selected Options by using deselectAll().

* Write a script to check whether the listbox is Single select (or) Multiselect ?

⟹ We can check whether the listbox is single Select (or) multiselect by using the method "isMultiple()".

Ex :-

```
public main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/Adv.html");
    WebElement hotel = driver.findElement(By.id("Adv"));
    Select sel = new Select(hotel);
    if(sel.isMultiple())
    {
        S.o.p("Multi-select");
    }
    else
    {
        S.o.p("Single-Select");
    }
    driver.close();
}
```

// (or)

```
boolean v = sel.isMultiple();
if(v)
{
}
```

Scanned by CamScanner

*. Write a script to count and print the selected options

⟹ we can get all the selected options by using the method "getAllSelectedOption()".

Ex:-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///c:/Users/Admin/Desktop/slv.html");
    WebElement hotel = driver.findElement(By.id("slv"));
    Select sel = new Select(hotel);

    sel.selectByIndex(0);
    sel.selectByIndex(1);
    sel.selectByIndex(2);
    // Returns address of all selected options
    List<WebElement> allOptions = sel.getAllSelectedOptions();
    // Count number of selected options
    int count = allOptions.size();
    S.O.p(count);
    // print all selected options
    for(WebElement option:allOptions)
    {
        String text = option.getText();
        System.out.println(text);
    }
    driver.close();
}
```

Scanned by CamScanner

Assignment :-

① Write a script to select your date of birth in facebook login page.

```
=> public class FacebookDemo1
{
    static
    {
        System.setProperty("webdriver.gecko.driver", "./drivers/
                            geckodriver.exe");
    }
    main()
    {
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
        WebElement day = driver.findElement(By.id("day"));
        WebElement month = driver.findElement(By.id("month"));
        WebElement year = driver.findElement(By.id("year"))

        Select day1 = new Select(day);
        Select month1 = new Select(month);
        Select year1 = new Select(year);

        Thread.sleep(1000);
        day1.selectByVisibleText("15");
        Thread.sleep(1000);
        month1.selectByVisibleText("May");
        Thread.sleep(1000);
        year1.selectByVisibleText("1993");
        driver.close();
    }
}
```

Scanned by CamScanner

13/02/18

Write a script to count number of options present in the list box and print all the options?

⇒

```
Main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/plv.html");
    WebElement hotel = driver.findElement(By.id("plv"));

    Select sel = new Select(hotel);

    List<WebElement> allOptions = sel.getOptions();

    int count = allOptions.size();

    S.o.p("Number of options" + count);

    for(int i=0; i<count; i++)
    {
        WebElement option = allOptions.get(i);

        String text = option.getText();

        S.o.p(text);

    }
    driver.close();
}
```

Write a script to print all the options in ~~shortcut~~ pointed on

Java logic :-

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

Scanned by CamScanner

```
ArrayList<String> al = new ArrayList<String>();
al.add("c");
al.add("d");
al.add("a");
al.add("b");
Collections.sort(al);
for(String a:al)
{
    s.o.p(a);
}
}
}
```

Selenium :-

```
main()
{
WebDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/Admin/Desktop/Alv.html");
WebElement hotel = driver.findElement(By.id("alv"));
Select sel = new Select(hotel);
// Creating object of Array List
ArrayList<String> allText = new ArrayList<String>();
// Creating get address of all options
List<WebElement> allOptions = sel.getOptions();
// get text of all options & store it in allText
for(WebElement option:allOptions)
```

Scanned by CamScanner

```
{
    String text = option.getText();
    allText.add(text);
}
// sort allText (which contains all the options)
Collections.sort(allText);
// print all Text
for(String text:allText)
{
    s.o.p(text);
}
driver.close();
}
```

(or) // Collection.reverse(allT...
↓
reverse sorted order

Write a script to check whether the options are in sorted (or) not.

Java:-

```
=> main()
{
    ArrayList<String> al = new ArrayList<String>();
    al.add("c");
    al.add("a");
    al.add("b");
    ArrayList<String> alCopy = new ArrayList<String>(al);
    Collections.sort(alCopy);
```

Scanned by CamScanner

```
if (al.equals(allcopy))
{
    S.o.p ("Sorted");
}
else
{
    S.o.p ("Not-sorted");
}
}
```

Selenium :-

=> main()

```
{
    WebDriver
    driver.get
    WebElement
    Select sel = new Select(hotel);
    // Create object of ArrayList (allText);
    ArrayList <String> allText = new ArrayList <String> ();
    // get address of all options
    List <WebElement> allOptions = sel.getOptions();
    // get text of all options and store it in allText
    for (WebElement options : allOptions)
    {
        String ~~all~~text = Option.getText();
        ~~all.text~~
        allText.add (text);
    }
```

Scanned by CamScanner

```
// create object of another ArrayList(allTextCopy)
ArrayList<String> allTextCopy = new ArrayList<String>(allText);

// sort allTextCopy
Collections.sort(allTextCopy);

// compare allText with allTextCopy

if(allText.equals(allTextCopy))
{
    s.o.p("sorted");
}
else
{
    s.o.p("not-sorted");
}
driver.close();
}
}
```

* Write a script to print the options without duplicate

Java program :

```
main()
{
    HashSet<String> hs = new HashSet<String>();
    hs.add("a");
    hs.add("b");
    hs.add("c");
    hs.add("c");
```

Scanned by CamScanner

```
for (String h : hs)
{
    s.o.p(h);
}
}
```

o/p → a
b
c

Selenium ~~program~~ :

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TIMEUNIT.SECONDS);
    driver.manage().window().maximize();
    driver.get("file:///C:/Users/Admin/Desktop/slv.html");
    WebElement hotel = driver.findElement(By.id("slv"));
    Select sel = new Select(hotel);
// create object of HashSet (allText)
    HashSet<String> allText = new HashSet<String>();
// get address of all options.
    List<WebElement> allOptions = sel.getOptions();
// get text of all Option and store it in allText
    for (String ~~allText~~ option : allOptions)
    {
        String text = option.getText();
```

Scanned by CamScanner

```
        allText.add(text);
    }
    for (String text : allText)
    {
        s.o.p(text);
    }
    driver.close();
}
```

Print all the options in sorted order without duplicates

Java:

```
main()
{
    TreeSet<String> ts = new TreeSet<String>();
    ts.add("c");
    ts.add("b");
    ts.add("a");
    ts.add("c");

    for (String t : ts)
    {
        s.o.p(t);
    }
}
```

Scanned by CamScanner

Selenium :-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.manage().window().maximize();
    driver.get("file:///C:/Users/Admin/Desktop/slv.html");
    WebElement hotel = driver.findElement(By.id("slv"));
    Select sel = new Select(hotel);
    // create object of TreeSet(allText)
    TreeSet<String> allText = new TreeSet<String>();
    // get address of all options
    List<WebElement> allOptions = sel.getOptions();
    // get text of all options & store it in allText
    for (String option : allOptions)
    {
        String text = option.getText();
        allText.add(text);
    }
    // print data of allText
    for (String text : allText)
    {
        S.o.p (text);
    }
    driver.close();
}
```

Scanned by CamScanner

Write a script to check whether the list box contains specific option or not :-

Java :-

```
main()
{
    ArrayList<String> al = new ArrayList<String>();
    al.add("a");
    al.add("b");
    al.add("c");
    al.add("d");
    if(al.contains("a")){
    {
        s.o.p("present");
    }
    else
    {
        s.o.p("Not-present");
    }
}
```

Selenium :-

```
main()
{
```

Select sel = new Select(hotel);

Scanned by CamScanner

```
// Create Object of ArrayList (allText)
~~ArrayList al = new ~~
ArrayList<String> allText=new ArrayList<String>();
// get address of all options
List<WebElement> allOptions = sel.getOptions();
// get text of all options and store it in allText
for(String option : allOptions)
{
   String text = option.getText();
   ~~s.o.p(text);~~ allText.add(Text)
}
// check the condition
~~if(al.contains~~
if(allText.contains("Idly"))
{
    s.o.p("present");
}
else
{
    s.o.p("not present");
}
driver.close();
}
```

Scanned by CamScanner

Print the duplicate functions present in the list box

```
main ( )
{
    WebDriver driver = new ChromeDriver();
    driver.get ("URL");
    WebElement hotel = driver.findElement (By.id("SLV"));

    Select sel = new Select(hotel);

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

    List<WebElement> allOptions = sel.getOptions();

    for ( WebElement option : allOptions)
    {
        String text = option.getText();
        if (!allText.add(text))
        {
            s.o.p(text);
        }
    }
    driver.close()

}
```

Scanned by CamScanner

Check whether the specified option is duplicate or not

```
main()
{
    int count=0;
    WebDriver driver = new ChromeDriver();
    driver.get("url");
    WebElement hotel = driver.findElement(By.id("slv"));
    Select sel = new Select(hotel);
    List<WebElement> allOptions = sel.getOptions();
    for(WebElement options : allOptions)
    {
        String text = option.getText();
        if(text.equals("Idly"))
        {
          count++;
        }
    }
    if(count>1)
    {
      S.o.p("Duplicate");
    }
    else
    {
        s.o.p("Not-duplicate");
    }
    driver.close()
}
```

Scanned by CamScanner

Map :-

*. In Map the values are stored based on keys.

*. While creating an object of map we should specify data type of key and values.

*. Key can not be duplicate.

*. To add the values into map, we use put() method and to get the values we use get() method.

*. In both the methods we specified the keys.

*. To check whether the map contains specified key - we use a method containsKey which returns boolean value

*. In order to get all the keys present in "map" we use the method keySet().

Ex:-

```
public class DemoMap
{
    P.S.V.M(String[] args)
    {
        Map<String, String> m1 = new LinkedHashMap<String,
                                                    ();
        // To add values into map
        m1.put("Name", "Rakesh");
        m1.put("Technology", "Selenium");
        // To check the specified key
        S.O.P(m1.containsKey("Name")); // true
        S.O.P(m1.containsKey("name")); // false (case sensitive)
```

Scanned by CamScanner

```
// To get the value of key
String v = m1.get("Name");
S.o.p(v);

Map<String, Integer> m2 = new LinkedHashMap<String, Integer>();
m2.put("pen", 4);
m2.put("Duster", 1);
m2.put("Pen", 3); //it will get the latest key
Integer v1 = m2.get("Pen");
So.p(v1);
// To get all keys
Set<String> allKeys = m1.keySet();
for(String key:allKeys)
{
    String m = m1.get(key);
    S.o.p(key + "---->" + m);
}
}
}
```

O/P :-

True

False

Rakesh

3

Name -> Rakesh

Technology -> Selenium.

Scanned by CamScanner

1) Write a code to count occurrence of each character in a given string.

⇒

```
main()
{
    String s = "qspiders";
    Map<Character, Integer> m = new LinkedHashMap<Character, Integer>
    Char[] ch = s.toCharArray();
    for (char c : ch)
    {
        if (m.containsKey(c))
        {
            Integer v = m.get(c);
            v++;
            m.put(c, v);
        }
        else
        {
            m.put(1, 1);
        }
    }
    Set<Character> allKeys = m.keySet();
    for (Character key : allKeys)
    {
        Integer value = m.get(key);
        S.O.P(key + "------>" + value);
    }
}
```

o/p →

q → 2	d → 1
s → 2	e → 1
p → 1	r → 1
i → 1	

Scanned by CamScanner

6. Write a script to print all the options along with the occurrence.

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("url");
    WebElement hotel = driver.findElement(By.id("s1v"));
    Select sel = new Select(hotel);
    Map<String, Integer> m = new LinkedHashMap<String, Integer>();
    List<WebElement> allOptions = sel.getOptions();
    for(WebElement option : allOptions)
    {
        String text = option.getText();
        if(m.containsKey(text))
        {
            Integer v = m.get(text);
            v++;
            m.put(text, v);
        }
        else
        { m.put(text, 1)
        }
    }
    Set<String> allKeys = m.keySet();
    for(String key: allKeys)
    {
        Integer value = m.get(key);
        s.o.p(key+"------>"+value);
    }
    driver.close();
}
```

Scanned by CamScanner

19/02/18

Write a script to check, whether the specified option is present or not.

```
=>
    if (m.containsKey("chicken biriyani"))
    {
        s.o.p("Present");
    }
    else
    {
        s.o.p("Not present");
    }
```

o/p → Not present

Write a script to check whether the specified option is duplicate or not.

```
=> if (m.get("Pongal") > 1)
   {
       s.o.p("Duplicate");
   }
   else
   {
       s.o.p("Not-Duplicate");
   }
```

o/p → Duplicate

Print only unique values.

```
Set<String> allKeys = m.keySet();
for (String key : allKeys)
{
    Integer v = m.get(key);
    if (v == 1)
```

o/p → Idly
Vada
Dosa

Scanned by CamScanner

```
        {
            s.o.p(key);
        }
    }
```

Print duplicate options.

```
set<String> allkeys = m.keySet();
for(String key : allkeys)
{
    Integer v = m.get(key);
    if(v>1)
    {
        s.o.p(key);
    }
}
```

o/p → Pongal

Print the options without duplicate

```
set<String> allkeys = m.keySet();
for(String key: allkeys)
{
    s.o.p(key);
}
```

o/p → Idly
Vada
Dosa
Pongal

Interview questions :-

1. How do you handle listbox (or) Dropdown list ?
2. Can you deselect the options in single select listbox ?
3. What are the methods present in listbox ?
4. How do you check, whether the list box is single select (or) multiselect list box ?

Scanned by CamScanner

(5) Have you implemented map in your automation frame w

Actions :-

*. Actions is a class which implements Action Interface.

*. Actions class is used to perform mouse & keyboard acti

*. Actions class contains a constructor, which takes an argument of type WebDriver.

*. Actions class should be imported from org.openqa.selenium interactions package.

How do you handle dropdown menu ?

*. When we place cursor on perticular element, it will displa list of menu's, which is called as dropdown menu.

*. We can handle dropdown menu by using ~~the mouse~~ moveToElement() method of Actions class

*. moveToElement() takes an argument of type WebElement

Ex :-

```
public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.actitime.com/");

    String xpA = "//a[.='Features']";
    WebElement features = driver.findElement(By.xpath(
```

Scanned by CamScanner

```
String xp2 = "//a[text()='Key features']";
WebElement keyfeatures = driver.findElement(By.xpath(xp2));

Actions act = new Actions(driver);
act.moveToElement(features).perform();
Thread.sleep(1000);
keyfeatures.click();

}
```

Write a script to count number of ~~auto~~ submenus. and print all the submenus.

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.actitime.com/");
    String xp1 = "//a[text()='Features']";
    WebElement features = driver.findElement(By.xpath(xp1));

    String
    Actions act = new Actions(driver);
    act.moveToElement(features).perform();
    String xp2 = "//a[.='Features']/../li";
    List<WebElement> allMenus = driver.findElements(By.xpath(xp2));
    int count = allMenus.size();
    S.o.p(count);
```

Scanned by CamScanner

```
for(WebElement menu : allMenus)
{
    String text = menu.getText();
    s.o.p(text);
}
driver.close();
}
```

Assignment :-

① Write a Script for the following Scenario:

step-1 :- Navigate to ISTQB.in webpage

step-2 :- Move to foundation → Enrollment → corporate Enrollment - click on online Enrollment.

step-3 :- verify Indian Testing Board registration page is displ
(or) not

Note :- Whenever we are calling any methods of Actions class, then it is mandatory that, we have to call perform() Otherwise Actions class will not perform the Action.

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.istqb.in");
    WebElement feature = driver.findElement(By.xpath("//
                        span[.='FOUNDATION']");
    String xp = "//a[.='FOUNDATION']/../span
                [.='ENROLLMENT']";
```

(or)

```
"//span[.='ENROLLMENT'][1]"
```

Scanned by CamScanner

```
WebElement feature1 = driver.findElement(By.xpath(xp));
WebElement feature2 = driver.findElement(By.xpath("//span
                      [.='CORPORATE ENROLLMENT']"));
WebElement feature3 = driver.findElement(By.xpath("//span
                      [.='ONLINE ENROLLMENT']"));

Actions act = new Actions(driver);
act.moveToElement(feature1).perform();
Thread.sleep(1000);
act.moveToElement(feature2).perform();
Thread.sleep(1000);
act.moveToElement(feature3).perform();
Thread.sleep(1000);
act.moveToElement(feature3).perform();
feature3.click();
String text = driver.getTitle();
if(text.equals("Indian Testing Board Registration/Signin"))
{
    S.o.p("pressed Displayed");
}
else
{
    S.o.p("Not-Displayed");
}
driver.close();
```

Scanned by CamScanner

How do you handle Drag & Drop Actions?

* Drag & Drop Action can be handle by using dragAndDrop() method of Actions class.
* dragAndDrop() takes two arguments of type WebElement.
They are i) source
ii) Target

Ex:

```
public class main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://jqueryui.com/resources/demos/droppable/
                default.html");
    WebElement src = driver.findElement(By.id("draggable"));
    WebElement target = driver.findElement(By.id("droppable"));
    Actions act = new Actions(driver);
    Thread.sleep(1000);
    act.dragAndDrop(src, target).perform();
}
```

Write a script to perform Drag & Drop actions without using dragAndDrop() :

```
Actions act = new Actions(driver);
Thread.sleep(5000);
act.clickAndHold(src).moveToElement(target).click().perform();
```

Scanned by CamScanner

How do you perform double click Action

* doubleclick Action can be handle by using doubleClick() method of Actions class.
* doubleClick() takes an argument, which is of type WebElement.

Sample webpage:-

```
<script>
function doubleClick()
{
    document.getElementById.innerHTML = "Welcome to Qspiders";
}
</script>

<p id="p"></p>
<input type="submit" id="qsp" value="Qspiders ondblclick=
                                    "doubleClick()">
```

Ex:-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///c:/Users/Admin/Desktop/dd.html");
    WebElement q=driver.findElement(By.id("qsp"));
    Actions actions=new Actions(driver);
    actions.doubleClick(q).perform();
}
```

Scanned by CamScanner

Context Click :-

* Right clicking on any element is called as context click.
* After right clicking on an element, the options which are displayed are called as context menu.
* context click can be handled by using contextClick() method of Actions class.
* contextClick() takes an argument which is of type WebElement.
* Context menus can be handled by using Robot class.

Ex :-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    WebElement link = driver.findElement(By.xpath("//a[.='actiTIME. Inc.']"));

    Actions actions = new Actions(driver);

    // To right click on an element
    actions.contextClick(link).perform();

    Robot r = new Robot();

    Thread.sleep(1000);

    r.keyPress(KeyEvent.VK_T);

    r.keyRelease(KeyEvent.VK_T);
}
```

Scanned by CamScanner

Composite actions :-

01/02/18

* performing multiple actions on the same element is called as composite action.
* Composite action can be perform by using build().perform() method.
* While performing multiple actions, if we call only perform() method, it will execute the multiple actions one by one.

Ex :-

```
actions.sendKeys(Keys.CONTROL).click(link).perform()
```

* In the above example first it will press the control key and then it will click on the link.

Ex :-

```
actions.sendKeys(Keys.CONTROL).click(link).build().perform()
```

* In the above example it will press control key and click on the element at the same time.
* The above story is applicable for very old version of Selenium
* In the latest version of Selenium no need to call the build() method explicitly. Because In the latest version of Selenium build() method is already integrated with perform() method.

Ex :-

```
public void perform()
{
    build().perform();
}
```

Scanned by CamScanner

Ex :-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    WebElement link = driver.findElement(By.xpath("//a[.='actiTIME Inc.']"));
    Actions actions = new Actions(driver);
    actions.sendKeys(Keys.CONTROL).click(link).perform();
}
```

What are the advantages of Actions class ?

Using Actions class we can handle

1). Dropdown menu

2). Drag and Drop

3). Double click

4). Context click

5). Composite Action

Frame :-

*. Webpage within another webpage is called as Embeddad webpage.

*. Embedded webpage can be developed using iframe tag

sample webpage :-

Page 1 :-

```
Me:<input type = "text" id = "me">
<iframe src = "page2.html" id = "frm" name = "frame"></iframe>
QF2 : <input type = "text" id = "qq">
```

Scanned by CamScanner

Page - 2 :-

GFI : <input type = "text" id = "g1" >

* When we right click on an element, if we get the option this Frame. Then that means the element is present inside the frame.

* Frames can be handled by using the statement ~~driver. switch~~

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

* frame() method is overloaded. They are

→ frame(int)

→ frame(String)

→ frame(WebElement)

frame(int) :-

* Here, we have to pass index value of the frame as argument

* The index value of frame starts from 0.

Ex :-

int :- index value of the frame - 0

```
driver.switchTo().frame(0);
```

frame(String) :-

* Here, we have to pass id (or) name of the frame.

Ex :-

```
driver.switchTo().frame("frm");  // using id
driver.switchTo().frame("frame"); // using name
```

Scanned by CamScanner

frame (WebElement) :-

* Here, we have to pass address of the frame.

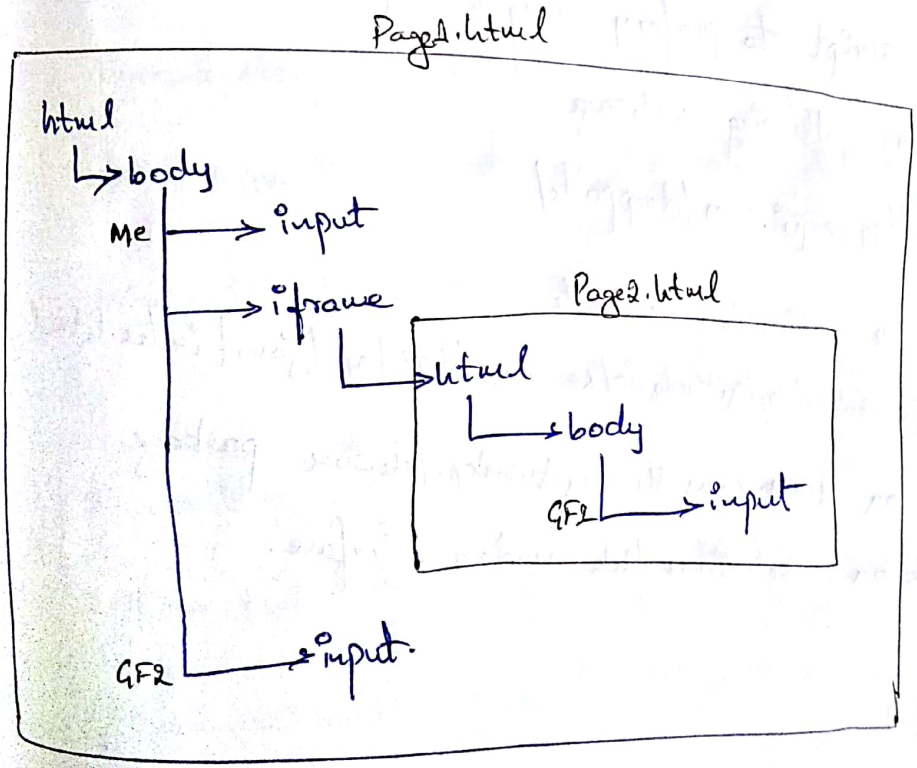
Ex:- WebElement f = driver.findElement(By.id("frm"));
driver.switchTo().frame(f);

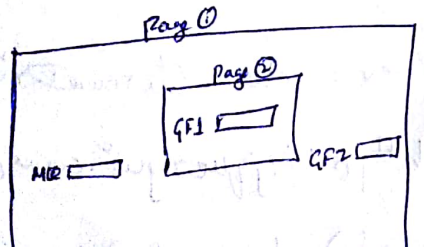
* To switch from frame to the default webpage, we use the statement

```
driver.switchTo().defaultContent();
```

* If the specified frame is not present, then it will throw NoSuchFrameException.

Tree structure :-





Scanned by CamScanner

Ex:-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/page.html");
    driver.findElement(By.id("me")).sendKeys("akash");
    driver.switchTo().frame(0);
    driver.findElement(By.id("gl")).sendKeys("dingi");
    driver.switchTo().defaultContent();
    driver.findElement(By.id("gg")).sendKeys("pinky");
}
```

Assignment :-

① Write a script to perform drag & drop action on the elem present in following webpage.

https://jqueryui.com/droppable/

② ⇒ Navigate to following website

https://seleniumhq.github.io/selenium/docs/api/java/index.html

i) click on ~~form~~ com.thoughtworks.selenium package

ii) click on selenium link under interface.

① ⇒

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://jqueryui.com/droppable/");
    driver.switchTo().frame(0);
```

Scanned by CamScanner

```
WebElement src = driver.findElement(By.id("draggable"));
WebElement target = driver.findElement(By.id("droppable"));
Actions action = new Actions(driver);
Thread.sleep(1000);
action.dragAndDrop(src, target).perform();
}
```

(2) =>

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://seleniumhq.github.io/selenium/docs/api/
                java/index.html");
    Thread.sleep(1000);
    driver.switchTo().frame("packageListFrame");
    driver.findElement(By.xpath("//a[.='com.thoughtworks.
                selenium']"));
    Thread.sleep(1000);
    driver.switchTo().defaultContent();
    driver.switchTo().frame("packageFrame");
    driver.findElement(By.xpath("//span[.='selenium']")).
                click();
}
```

Scanned by CamScanner

* To switch from one frame to its parent frame, we use statement

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

sample webpage :- Page 1 same (previous one)

Page 1 :-

GF1 : <input type="text" id="gf1" >

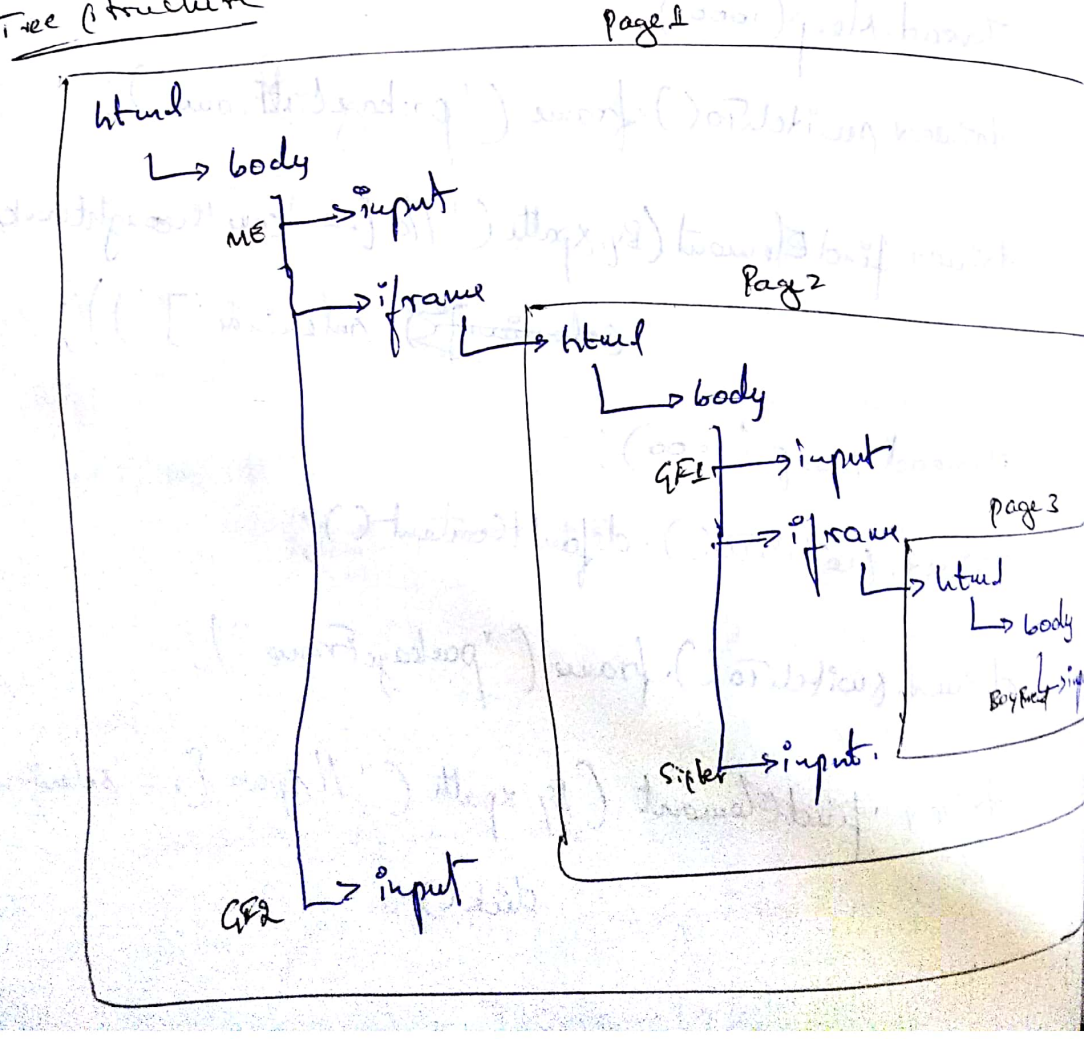
<iframe src="page3.html" id="frm1" name="frame1"></iframe>

Sister : <input type="text" id="s1" >

page 3 :-

Boy Friend : <input type="text" id="bf1" >

Tree structure



Scanned by CamScanner

Ex:-

```
main()
{ WebDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/Admin/Desktop/page1.html");
driver.findElement(By.id("me")).sendKeys("abark");
// To switch into page2.html
driver.switchTo().frame(0);
driver.findElement(By.id("g1")).sendKeys("dinga");
// To switch from page2.html to page3.html
driver.switchTo().frame(0);
driver.findElement(By.id("b1")).sendKeys("pavan");
// To switch from page3.html to page2.html
driver.switchTo().parentFrame();
driver.findElement(By.id("s1")).sendKeys("manga");
// To switch page1.html (default webpage)
driver.switchTo().defaultContent();
driver.findElement(By.id("gg")).sendKeys("pinky");
}
```

Scanned by CamScanner

Java Script Executor :-

* Java Script Executor is an interface, which contains two methods i.e.,
  i) executeAsyncScript
  ii) executeScript

* Java script is a powerfull scripting language, which will perform action on the element, even if it is disabled.

Sample web page :-

```
UserName : <input type="text" id="u4" disabled>
<a href="http://www.facebook.com" id="fb">Facebook</a>
```

Write a script to ~~perform action~~ Enter the Username in the above created webpage ?

=>

```
{
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/s.html");
    Thread.sleep(1000);
    JavaScriptExecutor jse = (JavaScriptExecutor) driver;
    jse.executeScript("document.getElementById('u4').value='xtek'");
}
```

document → current webpage
getElementById → method
value = 'xtek' → value

Scanned by CamScanner

* Here document ref. represents the current webpage and getElementById is a method.

* To execute the java script statement in browser:

→ press F12 and go to console tag

Write the java script statement in the text field and click on Enter ?

Ex:- document.getElementById('un').value = 'rakesh'

Write a script to clear the text field ?

=> main()

{

WebDriver driver = new ChromeDriver();

driver.get("file:///C:/Users/Admin/Desktop/a.html");

Thread.sleep(2000);

JavaScriptExecutor jse = (JavaScriptExecutor) driver

jse.executeScript("document.getElementById("un").

value = "");

}

Write a script to click on Facebook link ?

main()

{

WebDriver driver = new ChromeDriver();

driver.get("file:///C:/Users/Admin/Desktop/a.html");

Thread.sleep(2000);

Scanned by CamScanner

```
JavaScriptExecutor jse = (JavaScriptExecutor) driver;
jse.executeScript ("document.getElementById('fb').click()");
}
```

23-09-18

Write a script to scroll the webpage ?

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.seleniumhq.org");
    Thread.sleep(2000);
    JavaScriptExecutor jse = (JavaScriptExecutor) driver;
    jse.executeScript("window.scrollBy(0,500)");
    Thread.sleep(2000);
    jse.executeScript("window.scrollBy(0,-500)");
}
```

+ve -> downward scroll

-ve -> upward scroll

Write a script to scroll the webpage upto 50 pixels for 10 times ?

```
=> main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.selenium.org");
    Thread.sleep(2000);
    JavaScriptExecutor jse = (JavaScriptExecutor) driver
```

Scanned by CamScanner

```
for (int i=0; i<10 ; i++)
{
    Thread.sleep(2000);
    jse.executeScript("window.scrollBy(0,50)");
}
}
```

Write a script to scroll the webpage upto specified element?

=>

```
main()
{
    WebDriver driver = new ChromeDriver()
    driver.get("https://www.seleniumhq.org");
    String xp = "//a[.='Sponsor the Selenium project']";
    WebElement Link = driver.findElement(By.xpath(xp));
    int y = Link.getLocation().getY();
    Thread.sleep(2000);
    JavascriptExecutor jse = (JavascriptExecutor) driver;
    jse.executeScript("window.scrollBy(0,'"+y+"')");
}
```

(0,y) → It will take y as a integer but not as a variable.

Here 'y' is a variable

Scanned by CamScanner

-: Pop Ups :-

With respect to selenium Pop ups are categorized into 6 different types. such as

1) Javascript Pop up
2) Hidden division Pop up
3) File upload Popup
4) File download Pop up
5) child-browser Pop up
6) window Pop up

26-02-18

Write a script to take the screenshot of window using Robot class. we can take the screenshot of entire window.

```
⇒ main()
  {
    WebDriver driver = new FirefoxDriver();
    driver.get("https://www.google.com");
    Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
    Rectangle rect = new Rectangle(d);
    Robot r = new Robot();
    BufferedImage img = r.createScreenCapture(rect);
    ImageIO.write(img, "png", new File("./windowshot.png"));
    driver.close();
  }
```

Scanned by CamScanner

How do you get the system current date (date) By using Date class which is present in java.util package.

```
=> main()
{
    Date date = new Date();
    s.o.p(date);
}
```

```
main()
{
    Date date = new Date();
    s.o.p(date);

    SimpleDateFormat s1 = new SimpleDateFormat("d");
    String day = s1.format(date);
    s.o.p(day);

    SimpleDateFormat s2 = new SimpleDateFormat("MMMM");
    String month = s2.format(date);
    s.o.p(month);

    SimpleDateFormat s3 = new SimpleDateFormat("YYYY");
    String year = s3.format(date);
    s.o.p(year);

    SimpleDateFormat s4 = new SimpleDateFormat("hh");
    String hour = s4.format(date);

    s.o.p(hour);
}
```

O/p =>

Mon Feb 26 11:34:40 IST 2018

26
Feb
2018
11

Scanned by CamScanner

```
public class JavaDemo
{
    main()
    {
        Date date = new Date();
        s.o.p (date);   // mon Feb 26 11:52:10 IST 2018

        String s = date.toString();
        String[] str = s.split(" ");
        s.o.p (str[0]);   // mon
    }
}
```

*.

```
public class JavaDemo1
{
    main()
    {
        LocalDate date = LocalDate.new();
        s.o.p(date);   // 2018-02-26
        LocalDate day = date.plusDays(1);
        s.o.p (day);   // 2018-02-27
        LocalDate week = date.plusWeeks(1);
        s.o.p (week);   // 2018-03-05
        LocalDate month = date.plusMonths(1);
        s.o.p (month);   // 2018-03-26
        LocalDate year = date.plusYears(1);
        s.o.p (year);   // 2019-02-26
    }
}
```

Scanned by CamScanner

07-02-18

① JavaScript PopUps :-

Characteristics :-

Javascript popups are categorised into three types. They are

i). alert pop-up
ii). Conformation pop-up.
iii). prompt.

Characteristics :-

*. We cannot move the pop-up.
*. We cannot Inspect the pop-up
*. The color is black & white
*. If the pop-up contains only OK button, then we call it as alert pop-up.
*. If the pop-up contains OK & Cancel button, then we will call it as conformation pop-up.
*. We can handle this pop-up by using the statement

```
driver.switchTo().alert()
```

*. In alert interface, we have different methods, they are

i). accept()
ii). dismiss()
iii). getText()
iv). sendKeys()

*. accept() is used to click on OK button.
*. dismiss() is used to click on Cancel button.
*. getText() is used to get the text of which is present on the pop-up

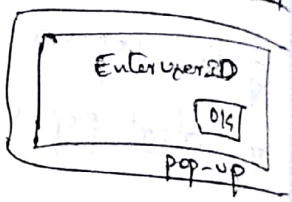
Scanned by CamScanner

* If the pop-up is not present and still if we try to switch into the pop-up, it will throw NoAlertPresentExp

Ex:- alert pop-up:-

```
Main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://irctc.co.in/");
    driver.findElement(By.id("loginbutton")).click();
    Alert a = driver.switchTo().alert();
    // To get the text present on pop-up
    String text = a.getText();
    s.o.p(text);
    // To click on OK
    a.accept();
    driver.close();
}
```

If we are not using this, still try to switch into the pop-up → Exp



o/p:-

Enter UserID

Confirmation Pop-up:-

Sample web page:-

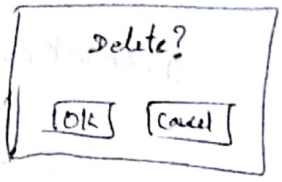
```
<script>
    function c()
    {
        confirm("Delete?");
    }
</script>
<input type="submit" onclick="c()" id="dd"/>
```

Scanned by CamScanner

Ex:-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/confirm.html");
    driver.findElement(By.id("dd")).click();

    Alert a = driver.switchTo().alert();
    // To get the text present on pop-up
    String text = a.getText();

    s.o.p(text);
    // To click on cancel
    a.dismiss();

    driver.close();
}
```

Pop-up



o/p :- Delete?

Prompt :-

* It can be handled by using sendKeys() of alert interface.

Simple web page :-

```
<Script>
    function p()
    {
        var name
        name = prompt("name");
        document.getElementById('r').innerHTML = "Welcome" + name;
    }
</Script>
<p id="r"></p>
<input type="submit" onclick="p()" id="dd"/>
```

Scanned by CamScanner

Ex:-

```
main()
{
    WebDriver driver=new ChromeDriver();
    driver.get("file:///C:/Users/Admin/Desktop/prompt.html");
    driver.findElement(By.id("dd")).click();
    Alert a = driver.switchTo().alert();
    // To enter the text in prompt
    a.sendKeys("rakesh");
    // To click on ok
    a.accept();
}
```

Assignment:-

① Write a script for the following scenario's
   i). Open the browser & enter the url
   ii). click on submit button.
   iii). Enter the text in text field & click on ok.
   iv). verify the welcome message.

⇒

```
main()
{
    Scanner sc=new Scanner(System.in);
    S.O.P("Enter the name:");
    String name=sc.nextLine();
    WebDriver driver=new FirefoxDriver();
```

Scanned by CamScanner

```
driver.get("file:///C:/Users/Admin/Desktop/prompt.html");
driver.findElement(By.id("old")).click();
Alert a = driver.switchTo().alert();
a.sendKeys(name);
a.accept();
String actualText = driver.findElement(By.id("x")).getText();
String expectedText = "welcome" + name;
if (actualText.equals(expectedText))
{
  s.o.p("pass");
}
else
{
  s.o.p("fail");
}
driver.close();
}
```

Scanned by CamScanner

③ Hidden Division Pop-ups :-

Characteristics:-

* We cannot move the pop-up.
* We can inspect the pop-up.
* It is colorful.
* We can Handle Hidden division pop-up by using findElement() ~~& findElements()~~
* calendar is one of the example for hidden division pop

Ex:-

```
main()
{
    Date date = new Date();
    SimpleDateFormat s1 = new SimpleDateFormat("d");
    String day = s1.format(date);

    SimpleDateFormat s2 = new SimpleDateFormat("MMMM");
    String month = s2.format(date);

    WebDriver driver = new FirefoxDriver();
    driver.get("https://www.cleartrip.com");

    driver.findElement(By.id("DepartDate")).click();

    String xp = "//span[.="+month+"]/../../../a[.=
                                    '"+day+"']";

    driver.findElement(By.xpath(xp)).click();

}
```

Scanned by CamScanner

Assignment :-

i. login to altatime application
ii. click on task module
iii. select the task
iv. click on delete delete
v. click on cancel

```
// main()
{
WebDriver driver = new FirefoxDriver();
driver.get("http://demo.actitime.com/login.do");
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.findElement(By.name("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.id("loginButton")).click();
driver.findElement(By.xpath("//span[@class='itemRow'
[1]")).click();
String xp = "//input[@name='taskSelected[1]']";
driver.findElement(By.xpath(xp)).click();
Thread.sleep(1000);
String xp2 = "//input[@value='Delete Selected Tasks']";
driver.findElement(By.xpath(xp2)).click();
Thread.sleep(1000);
String xp3 = "//input[@value='cancel'][3]";
driver.findElement(By.xpath(xp3)).click();
driver.close();
}
```

Scanned by CamScanner

28/02/18

③ File Upload pop-up :-

Characteristics :-

* When we click on browse button, a pop-up will be displayed with a name file upload, which is called as File Upload pop-up
* We can move the pop-up.
* We cannot inspect the pop-up.
* It is colorful.
* It contains two buttons i.e., open & cancel.
* We can handle file upload pop-up by using sendKeys() only if the type of the element is "file"
* If the type of the element is other than file, like attachment button of gmail then we can handle that element by using Robot class (or) a thirdparty tool which is called as AutoIT.

Ex :-

```
main()
{
```

Note :- In sendKeys() we have to specify the absolute path of the file using "\\" (double back word slash).

Ex :-

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.actitime.com/support.php");
    Thread.sleep(2000);
    String path = "C:\\Users\\Admin\\Desktop\\avg.docx";
    driver.findElement(By.name("screenshot")).sendKeys(path);
}
```

Scanned by CamScanner

Handling file upload pop-up using Robot class :-

```
main()
{
WebDriver driver = new FirefoxDriver();
driver.get("https://www.actitime.com/support.php");
String path = "C:\\Users\\Admin\\Desktop\\avg.docx";
StringSelection s = new StringSelection(path);

driver.findElement(By.name("screenshot")).click();

Toolkit.getDefaultToolkit().getSystemClipboard().setContents(s, null);

Thread.sleep(2000);

Robot r = new Robot();

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

Thread.sleep(2000);

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

Thread.sleep(2000);
r.keyPress(KeyEvent.VK_ENTER);
r.keyRelease(KeyEvent.VK_ENTER);
}
```

Scanned by CamScanner

Assignment :-

① → Navigate to naukri.com
→ click on upload cv
→ browse & select the file
→ click on open
→ fill all the mandatory fields
→ click on register.

② → login to facebook, click on photo & video
→ browse & select the photo
→ click on open
→ specify the caption
→ click on post.

③ → login to actitime
→ click on settings
→ click on logo & color scheme
→ select use custom logo button
→ click on browse & select the file
→ click on open
→ click on save settings.

④ ⇒ main()
{
WebDriver driver = new FirefoxDriver();
driver.get("http://127.0.0.1/login.do");
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.findElement(By.name("username")).sendKeys("admin");
driver.findElement(By.name("pwd")).sendKeys("manager");
driver.findElement(By.id("loginButton")).click();

Scanned by CamScanner

```
Thread.sleep(2000);
driver.findElement(By.xpath("//span[@class='itembuttonver']")[5]
                                              )).click();  }

Thread.sleep(2000);
driver.findElement(By.xpath("(//a[@class='hoverLink'])[3]")).
                                              click();

driver.findElement(By.xpath("//input[@id='uploadNew
                                     LogoOption']")).click();

Thread.sleep(2000);

String path = "c:\\Users\\Desktop\\ logos.png";

driver.findElement(By.xpath("//input[@name='formCustom
  InterfaceLogo.logo]")).sendKeys(path);

Thread.sleep(2000);
driver.findElement(By.xpath("//input[@value='Save Settings']
                                              .click();

Thread.sleep(2000);
driver.close();
}
```

Scanned by CamScanner

01/03/18

⑥ File download pop-up :-

Characteristics :-

* We can move the pop-up.
* We cannot inspect the pop-up.
* It is colorful.
* It contains two radio buttons i.e., Save File and Open With.
* We can handle this pop-up by using Robot class.

Note :- In chrome browser we will not get the File Download pop-up. It will be available only in Firefox browser.

Ex :-

```
main()
{
    WebDriver driver = new FirefoxDriver();
    driver.get("http://www.seleniumhq.org/download/");

    String xp = "//td[.='Java']/../../a[.='Download']";
    driver.findElement(By.xpath(xp)).click();

    Thread.sleep(2000);

    Robot r = new Robot();
    r.keyPress(KeyEvent.VK_ALT);
    r.keyPress(KeyEvent.VK_S);

    Thread.sleep(2000);
    r.keyRelease(KeyEvent.VK_S);
    r.keyRelease(KeyEvent.VK_ALT);
    Thread.sleep(2000);
    r.keyPress(KeyEvent.VK_ENTER);
```

Scanned by CamScanner

⑤ Child window browser Pop-up :-

characteristics :-

*. We can move the pop-up

*. We can inspect the pop-up.

*. It is colorful.

*. It contains minimize, maximize & close button.

*. We can handle this pop-up by using getWindowHandles().

*. To switch from one window to another window, we use the method

```
driver.switchTo().window()
```

*. After switching to child windows, we can handle them by using findElement() or findElements()

what is window Handle ?

*. It is the unique address of the windows.

*. The address of the windows are different from browser to browser.

Scanned by CamScanner

What are the difference between getWindowHandle() and getWindowHandles()?

getWindowHandle()	getWindowHandles()
① It returns address of parent window.	① It returns the addresses of all the windows opened by selenium.
② Return type is String	② Return type is Set<String>

Ex:- getWindowHandle():

```
main()
{
    WebDriver driver=new ChromeDriver();
    driver.get("https://www.naukri.com");
    String winHand=driver.getWindowHandle();
    s.o.p(winHand)
}
```

o/p → CDWindow-(AFA6756B1F7CADR22BEE2A5EFA5A9)

getWindowHandles():

```
main
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.naukri.com");
    Set<String> winHand=driver.getWindowHandles();
    for(String wh:winHand)
    {
    } s.o.p(wh);
}
```

Scanned by CamScanner

CDwindow - (5998B5F05B984H4572245S9F4FAF2E09)

CDwindow - (62BFBE7DLF7F65EDE4F24E21763AFCF8)

CDwindow - (93ADC179F624F9372038F5CC61F6ED9B)

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

⇒
```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.naukri.com/");
    Set<String> winHand = driver.getWindowHandles();
    for(String wh : winHand)
    {
        Thread.sleep(2000);
        driver.switchTo().window(wh);
        driver.close();
    }
}
```

02/03/18

Write a script to print title of all the windows ?

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.manage().window().maximize();
    driver.get("https://www.naukri.com/");
    // To get the address of all the windows.
    Set<String> winHand = driver.getWindowHandles();
    // Iterate through all windows
```

Scanned by CamScanner

```
for(String wh : winHand);
{   driver.switchTo().window(wh);
    String title = driver.getTittle();
    s.o.p(title);
}
}
```

Write a script to close the specified browser ?

Same code (above)

```
for (String wh : winHand)
{
    driver.switchTo().window(wh);
    // gettitle
    String title = driver.getTitle();
    if (title.equals ("Amazon"))
    {
        driver.close();
    }
}
```

Write a script to close the child browser ?

```
main()
{
    WebDriver driver = new ChromeDriver()
    driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
    driver.manage.window().maximize();
    driver.get("https://www.naukri.com");
    // To get the address of all windows
    Set<String> winHand = driver.getWindowHandles();
    // To get the address of parent window
    String pw = driver.getWindowHandle();
```

Scanned by CamScanner

```
//Iterate through all windows
for (String wh: winHand)
{
  //Switch from one - another windows
  driver.switchTo().window(wh);
  if (!wh.equals(PW))
  {
    driver.close();
  }
}
}
```

Write a script to break the control in a specific window?

```
{
// get the address of all windows
Set<String> winHand = driver.getWindowHandles();
for (String wh : winHand){
{
  driver.switchTo().window(wh);
  String title = driver.getTitle();
  if (title.equals("Amazon")).
  {
    break;
  }
}
Thread.sleep(2000);
//To maximize Amazon
driver.manage().window().maximize();
```

Scanned by CamScanner

Generic method for above programs / privious code :-

```
public static void winHandle(WebDriver driver, String
{
    Set<String> winHand = driver.getWindowHandles();
    Iterator<String> itr = winHand.iterator();
    while (itr
    {
        String wh = itr.next();
        driver.switchTo().window(wh);
        if (driver.getTitle().equals(e
        {
            break;
        }
    }
}
```

Create the Generic ~~to~~ method to

① close the parent window
② child windows

Handling Tabs :-

multiple table can be handle by using getWindowHandles()

Ex:-

① count no of tabs

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    WebElement link = driver.findElement(By.xpath("//a[text()
                                = 'actiTime. Inc.']"));
    Actions act = new Actions(driver);
    act.sendKeys(Keys.CONTROL).click(link).perform();
```

Scanned by CamScanner

```
Set<String> allTabs = driver.getWindowHandles();
int count = allTabs.size();
S.o.p(count);
}
```

Assignment :-

Write a script to

① → print title of all Tabs

② → To close all Tabs

③ → To switch the control to specified Tab.

⇒ ① public static void closeParent(WebDriver driver)

```
{
    String driver.close();
}
public static void closeChild(WebDriver driver, String childTitle)
{
    Set<String> windHand = driver.getWindowHandles();
    for (String wh : windHand)
    {
        driver.switchTo().window(wh);
        if (driver.getTitle().equals(childTitle)
        {
            driver.close();
        }
    }
}
```

② ⇒ public static void main(String[] args)

```
{
    WebDriver driver = new FirefoxDriver();
    driver.get("url");
    Set<String> allTabs = driver.getWindowHandles();
    for (String tab : allTabs)
```

Scanned by CamScanner

```
{
    driver.switchTo().window(tab);
    s.o.p(driver.getTitle());          //To print title
    driver.close();                    // To close the titles
}
```

(b) ⇒

```
for(String tab : allTabs)
{
    driver.switchTo().window(tab);
    String title = driver.getTitle();
    if(title.equals("expected Title"));
    {
        break;
    }
}
```

07/03/18

(6) Window Pop-up :-

* The pop-up which is displaying not a javascript, hidden div's, file upload, file download & child browser pop-up then we call that pop-up as window pop-up.

characteristics :

* we can move the pop-up, but we cannot inspect the pop-up.
* It is colorful.
* Using selenium we cannot handle window pop-ups
* To handle the window pop we use third party tool which is called as AutoIT.
* AutoIT can be downloaded from following website:

Scanned by CamScanner

Url --> https://www.autoitscript.com/site/autoit/downloads/

Filename --> autoit-v3-setup.exe

To install AutoIT:

Double click on the .exe file and follow the default instructions till finish.

AutoIT:

*. AutoIT is a freely available automation tool which is used to automate the window based applications.

*. To handle the window pop-ups we can take any automation tools which has following features.

i). It should be free

ii). It should be able to handle window pop-ups

iii). It should be easy to learn and write the script

iv). It should be able to integrate with Selenium.

Steps to inspect GUI Object:

*. Go to start → all programs and click on AutoIT

*. click on AutoIT window.

*. Drag & drop the finder tool on the GUI objects where will give the properties such as class, instances, name, title, id etc.

Steps to write AutoIT Script:

① Go to Start → all programs and click on AutoIT

② Select write script editor

③ Write the autoIT script

④ Save the file in a preferred location (extension in .au3)

Scanned by CamScanner

⑤ compile the AutoIT script (Tools → compile ~~to~~ which will generate the .exe file)

⑥ Double click on .exe file which will execute the AutoIT script

Ex:- Run("c:\windows\system32\code.exe")
~~Windo~~ WinWaitActive("calculator")
Sleep(1000)
send("10")
sleep(1000)
Send("-")
Sleep(1000)
Send("5")
sleep(1000)
Send("{ENTER}")

public static void main(String[] args) throws IOException

{
Runtime r = Runtime.getRuntime();
r.exec("D:\\calc.exe");
//(or) Runtime.getRuntime().exec("D:\\calc.exe");

}

WinWaitActive("opening Selenium - server - standard - ~~5.0~~ 3.1.0.0.jar")
sleep(1000)
Send("{LEFT}")
sleep(1000)
Send("{ENTER}")

Scanned by CamScanner

```
public static void main(String[] args) throws InterruptedException, IOException
{
    WebDriver driver = new FirefoxDriver();
    driver.get("https://www.seleniumhq.org/download/");
    String xp = "//a[contains(.,'3.1.0.0')]";
    driver.findElement(By.xpath(xp)).click();
    Thread.sleep(2000);
    Runtime.getRuntime().exec("D:\\MSM19\\win.exe");
}
```

Scanned by CamScanner

8/3/18

Encapsulation

*. Binding the data member or restricting the direct access to the variable is called as Encapsulation.

*. We can achieve encapsulation by using Private variables and public methods.

Ex:-

```
public class Account
{
    //Declaration
    private int amt;
    //Initialization
    public Account(int n)
    {
        amt=n;
    }
    // Utilizing
    public int getAmount()
    {
        return amt;
    }
}
```

```
public class JavaDemo
{
    public static void main(String[] args)
    {
        Account anil = new Account(1000);
        S.o.p(anil.getAmount());
        Account Srikanth = new Account(5000);
        S.o.p(Srikanth.getAmount());
        Account Suresh = new Account(10000);
        S.o.p(Suresh.getAmount());
    }
}
```

⇒ 1000
5000
10000

Scanned by CamScanner

Actitime Login:

```
public class LoginPage
{
    //Declaration
    private WebElement unTB;
    private WebElement pwTB;
    private WebElement loginBTN;

    //Initialization
    public LoginPage(WebDriver driver)
    {
        unTB = driver.findElement(By.id("username"));
        pwTB = driver.findElement(By.name("pwd"));
        loginBTN = driver.findElement(By.xpath("//div[.='Login']"));
    }
    //Utilization
    public void enterUserName(String un)
    {
        unTB.sendKeys(un);
    }
    public void enterPassWord(String pw)
    {
        pwTB.sendKeys(pw);
    }
    public void clickOnLogin()
    {
        loginBTN.click();
    }
}
```

Scanned by CamScanner

```
public class Demo
{
public static void main(String[] args)
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    LoginPage lp = new LoginPage(driver);
    lp.enterUserName("admin");
    lp.enterPassword("manager");
    lp.clickOnLogin();
}
}
```

```
main()
{
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    LoginPage lp = new LoginPage(driver);
    lp.enterUserName("admin");
    lp.enterPassword("manager");
    lp.clickOnLogin();
    Thread.sleep(2000);
    lp.enterUserName("admin");
    lp.enterPassword("manager");
    lp.clickOnLogin();
}
```

Exception #9

Scanned by CamScanner

*. When we execute the above code, it will throw "StaleElement Reference Exception."

What is StaleElement Reference Exception?

*. Reference of the element is too old.

When do we get StaleElement Reference Exception?

*. After identifing the element, before performing the action if the page is reloaded (or) refreshed, the address of the element will be changed. In such cases if we try to perform the action on the same element, it will throw StaleElement Reference Exception.

How do you handle StaleElement Reference Exception?

* By using POM

What is POM?

*. POM stands for Page Object Model.

*. POM is a java design pattern which is used to Design, Develope and Test the application.

How do you declare the elements in POM class?

By using @FindBy.

FindBy Annotation

syntax:-

@FindBy (locator = 'Locator Value');

Scanned by CamScanner

Ex:- @FindBy(id = "Username")
private WebElement unTB.

How do you initialize the element in POM class?

*. By using initElements() method of PageFactory class

*. initElements() is a static method, which takes two arguments of type :- i). WebDriver
ii). Object.

Ex:- PageFactory.initElements(driver, this)

*. Here this represents the current class objects.

Ex:-
```
public class LoginPage
{
    //Declaration
    @FindBy (id = "Username")
    private WebElement unTB;

    @FindBy (name = "pwd")
    private WebElement pwTB;

    @FindBy (xpath = "//div[. = 'Login']")
    private WebElement loginBTN;

    //Initialization
    public LoginPage (WebDriver driver)
```

Scanned by CamScanner

```
{
    PageFactory.initElements(driver, this);
}
//Utilization
public void enterUserName(String un)
{
    unTB.sendKeys(un);
}
public void enterPassword(String pw)
{
    pwTB.sendKeys(pw);
}
public void clickOnLogin()
{
    loginBTN.click();
}
}
```

9/3/18

Can you create a POM class without initializing the elements?

*. Yes, but we cannot execute it.

*. If we try to execute, it will through "NullPointerException".

Can you initialize the elements in test class?

*. Yes

Scanned by CamScanner

```
LoginPage lp = new LoginPage(driver);
PageFactory.initElements(driver, lp);
```

How do you handle multiple elements in POM class ?

By using "List<WebElement>".

Ex:-

```
public class LoginPage
{
    // Declaration
    @FindBy(xpath = "//a")
    private List<WebElement> allLinks;

    // Initialization
    public LoginPage(WebDriver driver)
    {
        PageFactory.initElements(driver, this);
    }
    // Utilization
    public void getLinks()
    {
        int count = allLinks.size();
        s.o.p(count);
        for(WebElement links : allLinks)
        {
            s.o.p(links.getText());
        }
    }
}
```

Scanned by CamScanner

Where do you store all the elements in your framework ?

* In POM class
* POM class is also called as element repository or object repository class.

Note :-

1) In Selenium we will develope two types of classes
   i) POM class
   ii) Test class

2) The number of POM class should be same as number of webpages
   * If we have hundred webpages, we have to create 100 POM class

* The name of the POM class should be same as title of the webpage and it should end with a word Page.

Ex :- LoginPage, EnterTimeTrackPage, TaskListPage ... etc.

3) The number of Test class should be same as number of "Manual Test cases".
   i.e; if we have 1000 manual test cases, we have to create 1000 test classes.

Advantages of POM class :-

1) We can handle StaleElementReferenceException.
2) class name & method name will be realistics.
3) If any changes happens to the element, easily we can identified that element.

Scanned by CamScanner

Test NG :-

* Test NG stands for "Test Next Generation".
* Test NG is a Unit Testing Framework which is basically used by Developers to perform white box testing and also used by Automation team to execute the Framework.

Advantages of Test NG :-

1) Batch Execution :- We can execute multiple test classes at ~~a time~~ ~~case~~ in a single shot.

2) Parallel Execution :- We can execute the framework ~~par~~ in multiple browser parallely

3) Group Execution :- We can execute only set of test classes.

4) Report Generation :- Using Test NG automatically we can generate the report.

* Test NG is available as plugin in eclipse.

Step to install Test NG :-

1) In eclipse go to Help and click on Eclipse Market Place
2) Search for TestNG and click on install button of TestNG for eclipse.
3) Follow the default instructions still finish
4) Restart the eclipse

Scanned by CamScanner

To use the TestNG in the framework we have to add TestNG Libraries.

*. To add ~~test~~ NG libraries → right click on the project
→ go to build path click on add libraries
→ select TestNG & click on next & finish

Do's & Don't in TestNG :-

Don't	Do's
1) Default package is not allowed	1) Use user defined package
2) Main method is not allowed	2) Use Test method
3) S.O.P statement is not allowed	3) Use Reporter.log statement

Note :-

To print the message on both console & report, we use the statement

Reporter.log with the ~~test~~ boolean value true.

Test class :-

*. Any java class which contains atleast one test method is called as Test class.

*. Any method which is developed by using @Test is called as test method.

Scanned by CamScanner

Assignment :-

① create POM class for facebook login page.

Ex :-

```
package POM;
import org.testing.Reporter;
import org.testing.annotations.Test;
public class Demo
{
    @Test
    public void testAC()
    {
        Reporter.log("hiiii", false);
    }
}
```

Assignment :-

⇒

```
public class FacebookLoginPage
{
    //Declaration
    @FindBy(id = "email")
    private WebElement unTB;
    @FindBy(id = "pass")
    private WebElement pwTB;
```

Scanned by CamScanner

```
@FindBy (xpath = "//input[@value='Login In']");
private WebElement loginBTN;
//Initialization
public FacebookLoginPage (WebDriver driver)
{
    PageFactory.initElements (driver, this);
}
//utilization
public void enterUserName (String un)
{
    unTB.sendKeys (un);
}
public void enterPassword (String pw)
{
    pwTB.sendKeys (pw);
}
public void clickOnLogin().
{
    loginBTN.click();
}
}
```

Scanned by CamScanner

10/08-18

*. To run the test class → right click on the test class
→ Go to Run As & click on TestNG test

*. In order to get the report, refresh the project, which will generate a folder called Test output

*. Expand Test output folder, right click on emailable report.html, go to open with, click on web browser.

*. To export the report to excel format, right click on the report click on export to microsoft excel. and follow the default instructions.

Note :-

To export the report to excel format ms-office should have been installed.

Test NG Suite :-

*. It is an xml file, which contains all the test classes present in the framework.

*. To create the TestNG Suite right click on the project, go to test NG, click on convert to testng & follow the default instructions which will create a file called testng.xml

Scanned by CamScanner

Ex:-

```
<suite name = "Suite">
  <test name = "Test">
    <classes>
      <class name = "pkg.DemoA" />
      <class name = "pkg.DemoB" />
    </classes>
  </test>
</suite>
```

*. To run the testng suite, right click on testng.xml file, go to Run As & click on Testng suite.

Can we have multiple test methods in a single test class ?

Yes

*. If a test class of multiple test methods then what will be the order of execution ?

→ Alphabetical order.

How do you execute Test methods in required order ?

by using priority.

Note :-

1). When we use priority, then the test method will be executed in the ascending order of priority.

Scanned by CamScanner

2). Default priority value is 0.

3) If the multiple test methods are the same priority, then testng executed in alphabetical order.

4). We can use +ve no, -ve no & 0 as priority values.

5). If we want to use variables, then it must be declared as final.

6) Ex:-

```
public class DemoB
{
    @Test(priority = 1)
    public void registerUser()
    {
        Reporter.log("register...", true);
    }
    @Test(priority = 2)
    public void deleteUser()
    {
        Reporter.log("delete...", true);
    }
}
```

How do execute a test method on success of another test method

By using dependsOnMethods.

Ex:-

Scanned by CamScanner

* When we use dependsOnMethods, if the independent method failed, then dependent method will be skipped.

Ex:-

```
public class DemoB
{
    @Test
    public void registerUser()
    {
        Reporter.log("register...", true);
    }
    @Test(dependsOnMethods = "registerUser")
    public void deleteUser()
    {
        Reporter.log("delete...", true)
    }
}
```

Note :-

We can specify the dependancy on multiple methods

Ex:- `@Test(dependsOnMethods = {"registerUser", "editUser"})`

Scanned by CamScanner

TestNG Annotation [@Testng] :-

12/03/18

* @Testng gives instructions to testng compiler.
* These annotations decides the execution flow.
* Types of TestNG Annotation :-

1) @BeforMethod
2) @AfterMethod
3) @BeforeClass
4) @AfterClass
5) @BeforTest
6) @AfterTest
7) @BeforeSuite
8) @AfterSuite
9) @DataProvider

@BeforeMethod :-

* @BeforeMethod will be executed before execution of each & every test method.

@AfterMethod :-

@AfterMethod will be executed after execution of each & every test method.

Scanned by CamScanner

@BeforeClass :-

Before execution of each & every test class @BeforeClass will be executed.

@AfterClass :-

After execution of each & every test class @AfterClass will be executed.

@BeforeTest :-

@BeforeTest will be executed before execution of each & every test block.

@AfterTest :-

@AfterTest will be executed after execution of each & every test block.

@BeforSuite :-

@BeforSuite will be executed before execution of entire suite file.

@AfterSuite :-

@AfterSuite will be executed after execution of entire suite file.

Ex :- Base class :-

```
public class BaseTest
{
    @BeforMethod
    public void beforeMethod()
    {
        Reporter.log("before method", true);
    }
```

Scanned by CamScanner

```
@AfterMethod
public void afterMethod()
{
    Reporter.log("after method", true);
}
@BeforeClass
public void beforClass()
{
    Reporter.log("before class", true);
}
@AfterClass
public void afterClass()
{
    Reporter.log("after class", true);
}
@BeforeSuite
public void beforeSuite()
{
    Reporter.log("before suite", true);
}
@AfterSuite
public void afterSuite()
{
    Reporter.log("after suite", true);
}
```

Scanned by CamScanner

```
@BeforeTest()
public void beforeTest()
{
    Reporter.log("beforeTest", true);
}
@AfterTest()
public void afterTest()
{
    Reporter.log("afterTest", true);
}
}
```

class-A:-

```
public class DemoA extends BaseTest
{
    @Test
    public void testA()
    { Reporter.log("testA()", true);
    }
}
```

class-B:-

```
public class DemoB extends BaseTest
{
    @Test
    public void testB()
    { Reporter.log("testB()", true);
    }
```

Scanned by CamScanner

```
@Test
public void test C()
{
    Reporter.log("test C()", true);
}
}
```

o/p :-

before suite
    before test
        before class
            before method
                testA()
            after method
        after class
        before class
            before method
                test B()
            after method
        after class
        before class
            before method
                test C()
            after method
        after class
    after test
after suite

Scanned by CamScanner

How do you execute same test method multiple times?

By using invocationCount

Note :-

*. Default invocation count value is 1.

*. If we use 0 & -ve values, those test method will not be executed.

*.

Ex:-

```
public class DemoA
{
    @Test(invocationCount=5)
    public void createUser()
    {
        Reporter.log("user1", true);
    }
}
```

② DataProvider :-

*. To run the test method with multiple datas, we use @DataProvider.

*. @DataProvider returns the value which is of type two dimensional ~~array~~ object array.

Scanned by CamScanner

Ex:-

```
public class DemoA
{
    @DataProvider
    public Object[][] getData()
    {
        Object[][] data = new Object[2][2];
        data[0][0] = "UserA";
        data[0][1] = "pass1";
        data[1][0] = "UserB";
        data[1][1] = 8675485;

        return data;
    }
    @Test (dataProvider = "getData")
    public void createUser(Object un, Object pw)
    {
        Reporter.log(un+" "+pw, true);
    }
}
```

Note:-

*. Number of iteration of the test method depends on the number of rows.

*. Number of arguments to the test method depends on number of columns.

Scanned by CamScanner

13/03/18

TestNG Groups :-

* In order to perform Regional Regression Testing, we use Testing Groups.

Ex:-

```
public class DemoA
{
  @BeforeMethod(alwaysRun=true)
  public void loginToApplication()
  {
     Reporter.log("Login...", true);
  }
  @AfterMethod(alwaysRun = true)
  public void logoutFromApplication()
  {
     Reporter.log("Logout...", true);
  }
  @Test(priority=1, groups={"user", "smoke"})
  public void createUser()
  {
     Reporter.log("create user", true);
  }
  @Test(priority=2, groups="user")
  public void editUser()
```

Scanned by CamScanner

```
{
    Reporter.log("edit User...", true);
}
@Test(priority=3, groups="user")
public void deleteUser()
{
    Reporter.log("delete user...", true);
}
@Test(priority=4, groups={"task", "smoke"})
public void createTask()
{
    Reporter.log("create task...", true);
}
@Test(priority=5, group="task")
public void editTask()
{
    Reporter.log("edit task...", true);
}
@Test(priority=6, group="task")
public void deleteTask()
{
    Reporter.log("delete user...", true);
}
}
```

Scanned by CamScanner

xml file :-

```
<suite name = "Suite" >
  <test thread-count = "5" name = "Test" >
    <groups>
      <run>
        <include name = "smoke" />
        <!-- <exclude name = "user" /> -->
      </run>
    </groups>
    <classes>
      <class name = "app.DemoA" />
    </classes>
  </test>
</suite>
```

@exclude

Command

How do you execute the test method irrespective of specified groups :-

by using alwaysRun = true

How do you disable the test method ?

by using enabled = false

Ex :-

```
public class DemoA
{
```

Scanned by CamScanner

```
@Test (priority = 1)
public void createUser()
{
    Reporter.log ("create user...", true);
}
@Test (priority = 2, enabled = false)
public void deleteUser()
{
    Reporter.log ("delete user...", true);
}
}
```

Note :-

In the above example delete user method will not be executed.

Verification in TestNG :-

*. In testng we will use "Assert class" to perform the verification.

*. Assert class contains lot of methods, all the methods are static and overloaded.

Scanned by CamScanner

Methods of Assert class :-

1. assertEquals()
2. assertNotEquals()
3. assertSame()
4. assertNotSame()
5. assertNull()
6. assertNotNull()
7. assertTrue()
8. assertFalse()
9. fail()

* - Assert class is also called as "hard Assert".

Ex :-

```
@Test
public void testA()
{
    System.setProperty("webdriver.chrome.driver", "./drivers/
                                                  chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.get("https://demo.actitime.com/login.do");
    String aTitle = driver.getTitle();
    String eTitle = "actiTIME - Login";
    Assert.assertEquals(aTitle, eTitle);
}
```

Scanned by CamScanner

Write a script to verify login button is displayed or not ?

① @Test
public void testAl()
{ System.setProperty("webdriver.chrome.driver", "./driver/chromedriver.exe");
WebDriver driver = new ChromeDriver()
driver.get("https://demo.actitime.com/login.do");
WebElement login = driver.findElement(By.xpath("//div[.='Login']"));
boolean v = login.isDisplayed();
Assert.assertTrue(v);
}

② driver.get("https://demo.actitime.com/login.do");

try
{ WebElement login = driver.findElement(By.xpath("//div[.='Login']"));
boolean v = login.isDisplayed();
Assert.assertTrue(v);
Reporter.log("Present", true);
}
catch(Exception e)
{ Reporter.log("not present", true);
Assert.fail()
}

If the element is not matching with the specified locator

Scanned by CamScanner

Note :-

*. In Assert class, if the comparison fails, it will not execute the remaining statements of current test method, but it will execute remaining test methods.

*. To overcome the above problem, we use another type of class which is called as "SoftAssert".

SoftAssert :-

*. All the methods which are present in Assert class are also present in SoftAssert class, but here it is Non-Static method.

*. In SoftAssert class, even if the comparison fails, it will execute the remaining statements of the test method, but it will not update the result to the status.

*. In order to update the result to the status, we have to use the method "AssertAll()" & it should be the last statement of the test method.

Ex :-

Ex :-

```
@Test
public void testAC()
{
```

Scanned by CamScanner

```
System.setProperty("webdriver.chrome.driver", "./driver/chrome
                                    -driver.exe");
WebDriver driver = new ChromeDriver();
driver.get("https://demo.actitime.com/login.do");
SoftAssert sa = new SoftAssert();
sa.assertEquals(driver.getTitle(), "actiTIME - Logi");
driver.close();
sa.assertAll();
}
```

What are the difference between Assert & SoftAssert ?

Assert	SoftAssert
① All the methods are static methods	① All the methods are Non-static methods
② If the comparison fails it will not execute the remaining statements of the current test methods	② Even if the comparison fails it will execute the remaining statements of the current test methods
③ It will do not call assertAll() method	③ We have to call assertAll() & it should be the last statement

Scanned by CamScanner

Frame Works

14/02/18

* Frame work is standard set of rules, guidelines which should be followed by all the automation engineers while automating the applications.
* Framework is also called as semi-developed application.
* We have to use the framework to have the consistancy in the project.
* There are three stages in the Frame work
  1) Automation FrameWork Design
  2) Automation FrameWork Implementation
  3) Automation FrameWork execution.

Automation frameWork Design:-

* Automation framework is mostly designed by lead (or) project manager. based on the project experience.
* While designing the automation framework, they will set standard rules, guidelines and also they will develope some generic libraries.
* Initially, they will list out all the required softwares and files.

Ex:- ① Jdk
② Eclipse with testng
③ Browsers
④ AutoIT
⑤ MS office
⑥ Application under testing

Scanned by CamScanner

All the above softwares should have been installed in the local workstations.

(7) Jar files (selenium & poi)

(8) driver executable files

All the above files should have been downloaded.

(9) Excel file.

Standard folder Structure :-

*. Go to prefered location and create a workspace.

*. The name of the workspace should be same as domain of the application.

Ex:- Project Management

*. Open the eclipse with the above created workspace.

*. Create a new Java project

The name of the project should be same as name of the application

Ex:- ActiTime

*. Associate testng libraries.

*. Under src folder, create three packages and name it as

1) com.actitime.generic

2) com.actitime.pages

3) com.actitime.testfs

*. Under java project create four folders and name it as

1) data

2) drivers

Scanned by CamScanner

3) jars
4) screenshots

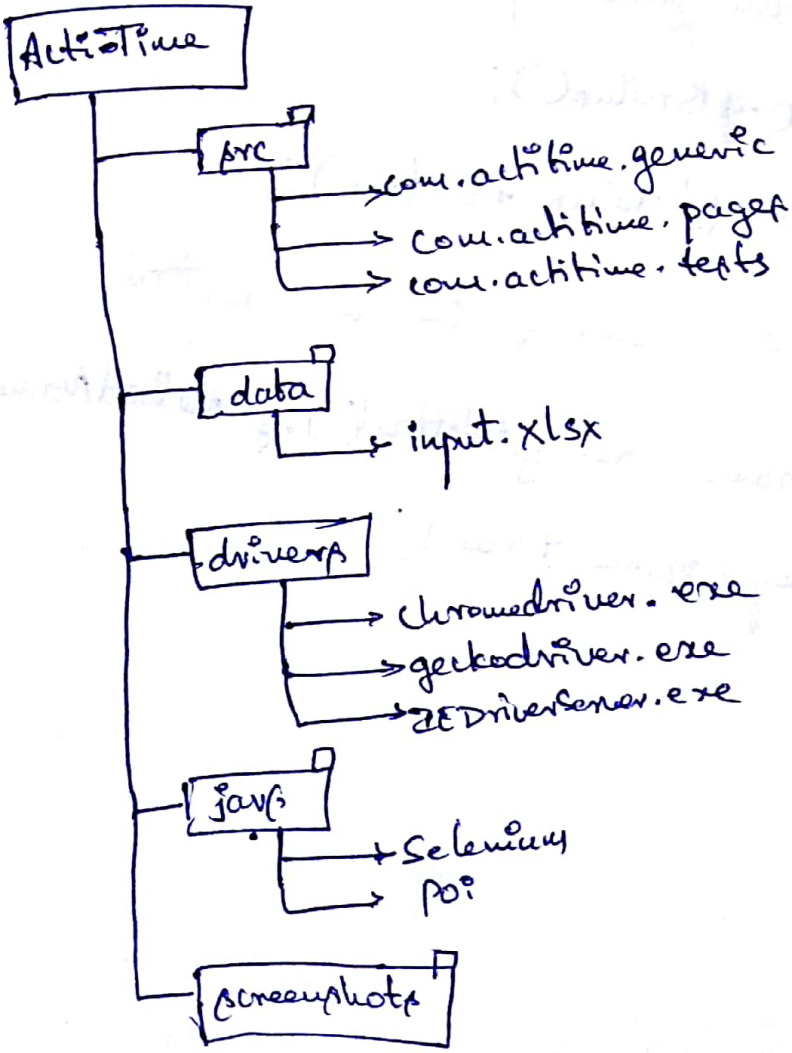
* Go to the data folder location and create an excel file and name it as input.xlsx
* Store all the driver executable files under drivers folder
* Download poi jar files from the following website

http://poi.apache.org/download → Url

~~poi-bin-3.~~ poi-bin-3.17-20170915.zip → file name

* Extract the poi jar files and copy all the 13 jar files and store it in jars folder.
* Associate all the jar files with current java project.

Ex:-



Anil Kumar H

Scanned by CamScanner

Script to get status & name of the test method

```
public class Demo
{
    @Test
    public void testA()
    {
        Assert.fail();
        Reporter.log("hii", true);
    }
    @AfterMethod
    public void postcondition(ITestResult res)
    {
        // to get the status, if s=1 then pass, if s=2 then fail.
        int s = res.getStatus();
        Reporter.log("status:"+s, true);
        // to get the name of the test method
        String name = res.getMethod().getMethodName();
        Reporter.log(name, true);
    }
}
```

Scanned by CamScanner

15/07/19

* - Interface & abstract class in ~~our~~ Framework :-

* In our automation framework some of the values such as key & values of the driver executable file, path of the excel files will never change.
* These values should be declared as constant.
* According to java standard, interface is the best place to store the constant variables.
* In our automation framework, we have created an interface called ~~A~~ AutoConstant which contains all the constant variables.

Ex :-

```
public interface AutoConstant
{
    String chrome_key = "webdriver.chrome.driver";
    String chrome_value = "./drivers/chromedriver.exe";
    String gecko_gkey = "webdriver.gecko.driver";
    String gecko_value = "./drivers/geckodriver.exe";
    String file_path = "./data/input.xlsx";
}
```

Scanned by CamScanner

* In our automation framework for every test case we have created separate test class.

* In all the test classes there are some common repeatable steps such as open the browser, enter the url and close the browser.

* Instead of writing same statements in all the test classes, we have created a class called BaseTest which contains the common statements of test classes.

* In base test class we have created two methods, i.e., precondition & post condition.

* In precondition method, we have written the code to open the browser & enter the url and in post condition method we have written code to close the browser.

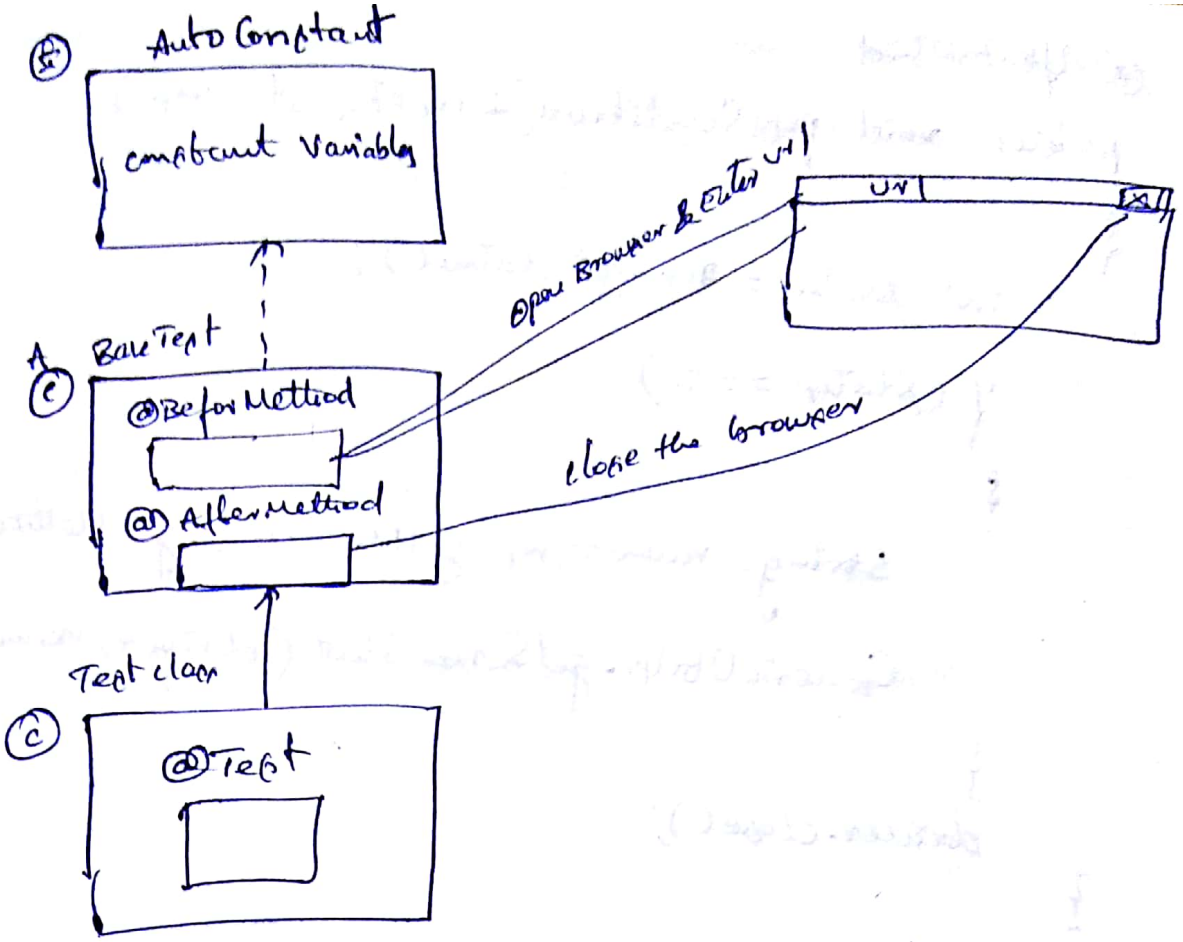
* To avoid the object creation, we have declared pre-condition method with @BeforeMethod & post condition method with @AfterMethod.

* BaseTest class implements AutoConstant interface.

* In our automation framework all the test classes extends from BaseTest class.

* BaseTest class doesn't contains any test method and it is an incomplete class & hence we declared it as Abstract.

Scanned by CamScanner




```
public abstract class BaseTest implements AutoConstant
{
    public WebDriver driver;
    @BeforeMethod
    public void preCondition()
    {
        System.setProperty(chrome-key, chrome-value);
        System.setProperty(gecko-key, gecko-value);

        driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit_SECONDS);
        driver.get("https://demo.actitime.com/login.do");
    }
```

Scanned by CamScanner

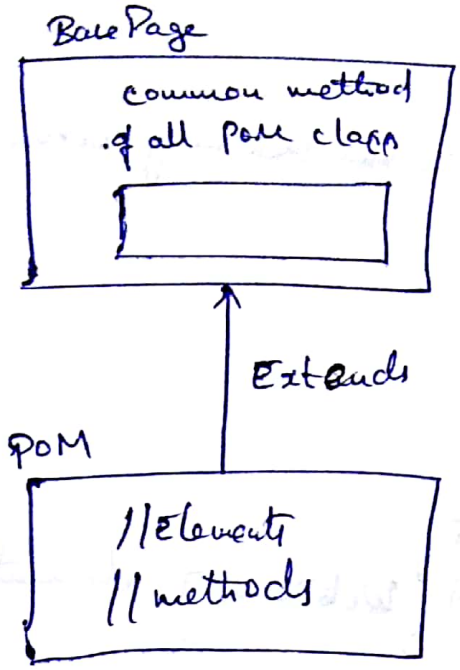
```
@AfterMethod
public void postCondition(ITestResult res)
{
    int status = res.getStatus();
    if(status==2)
    {
        String name = res.getMethod().getMethodName();
        GenericUtils.getScreenShot(driver, name);
    }
    driver.close();
}
}
```

Inheritance in Automation Framework :-

* In our automation framework for every webpage we have created separate POM class.
* In all the POM class, there are some common repeatable methods such as verify the title, verify the element etc.
* Instead of writing the same methods in all the POM classes we have created a class called as BasePage.
* BasePage class contains the common ~~Repeated~~ methods of all the POM class.
* In our automation framework all the POM classes extends from BasePage class.

Scanned by CamScanner

Ex:-




```
public class BasePage
{
    public WebDriver driver;
    public BasePage ( WebDriver driver)
    {
        this.driver = driver;
    }
    // To verify the title
    public void verifyTitle (String eTitle)
    {
        WebDriverWait wait = new WebDriverWait (driver, 10);

        try
        {
            wait.until ( ExpectedConditions.titles(eTitle));
            Reporter.log( "Title is matching "+eTitle, true);
        }
```

Scanned by CamScanner

```
catch (Exception e)
{
    Reporter.log ("Title is not matching: Actual title is: "+driver.getTitle(),
                                                        true);
    Assert.fail();
}
}
// To verify the element
public void verifyElement (WebElement element)
{
    WebDriverWait wait = new WebDriverWait (driver, 10);
    try
    {
        wait.until (ExpectedConditions.visibilityOf (element));
        Reporter.log ("element is present: ", true);
    }
    catch (Exception e)
    {
        Reporter.log ("Element is not present: ", true);
        Assert.fail();
    }
}
}
```

Scanned by CamScanner

Interview Questions :-

① Have you implemented alphabetical order in your Automation framework?

Yes.

② Have you implemented Palindrome in your Automation framework?

Yes.

Generic Utils :-

* In our automation framework we have created a class called Generic Utils, which contains the generic methods to handle Listbox, popups, window handles etc.

Ex:-

```
public class GenericUtils
{
    // To take screenshot
    public static void getScreenshot(WebDriver driver, String name)
    {
        try
        {
            TakeScreenshot t = (TakeScreenshot) driver;
            File src = t.getScreenshotAs(OutputType.FILE);
            File dest = getFile("./screenshots"+name+".png");
            FileUtils.copyFile(src, dest);
        }
        catch (IOException e)
        {
        }
    }
```

Scanned by CamScanner

```
// To select by index
public static void selectByIndex(WebElement element, int index)
{
    Select select = new Select(element);
    select.selectByIndex(index);
}
// To select by value
public static void selectByValue(WebElement element, String value)
{
    Select select = new Select(element);
    select.selectByValue(value);
}
// To select by visible text
public static void selectByVisibleText(WebElement element, String text)
{
    Select select = new Select(element);
    select.selectByVisibleText(text);
}
}
```

Scanned by CamScanner

16/03/18

Data Driven testing :-

* Testing the application with multiple datas is called as Data driven testing (or) parameterization.
* Manual testing team will derive the test datas using Test case Design techniques such as Error guessing, equivalence partition & Boundary value analysis.
* These test datas usually stored in an Excel files.
* Automation team will take the test datas from manual testing team and they will organized in a proper manner.
* To take the data from the Excel file, we use an API which is provided by Apache. which is called as POI (Poor Obfuscation.
* POI is stands for poor Obfuscation (abstraction) Implementation

Steps to get data from Excel files :-

1) Create and read the file
2) create workbook
3) get sheet
4) get Row
5) get cell
6) get data

Scanned by CamScanner

Ex:
Write a script to get the data from excel file ?

```
public static void main(String[] args) throws Exception
{
    // 1. create and read the file
    File file = new File("./data/input.xlsx");
    FileInputStream fis = new FileInputStream(file);

    // 2. create workbook
    Workbook wb = WorkbookFactory.create(fis);

    // 3. get sheet
    Sheet sh = wb.getSheet("sheet1");

    // 4. get Row
    Row r = sh.getRow(0);

    // 5. get cell
    Cell c = r.getCell(0);

    // 6. get data
    String data = c.getStringCellValue();

    S.o.p(data);
}
```

Scanned by CamScanner

Write a script to count number of rows present in a sheet ?

```
public static void main(String[] args) throws Exception
{
    // 1. create and read the file
    File file = new File("./data/input.xlsx");
    FileInputStream fis = new FileInputStream(file);

    // 2. create workbook
    Workbook wb = WorkbookFactory.create(fis);

    // 3. get sheet
    Sheet sh = wb.getSheet("Sheet1");

    // 4. get number of rows
    int rn = sh.getLastRowNum();
    s.o.p(rn);
}
```

Write a script to count number of cells present in a row?

```
public static void main(String[] args) throws Exception
{
    // 1. create and read the file
    File file = new File("./data/input.xlsx");
    FileInputStream fis = new FileInputStream(file);
```

Scanned by CamScanner

```
// 2. Create workbook
Workbook wb = WorkbookFactory.create(fis);

// 3. get Sheet
Sheet sh = wb.getSheet("Sheet1");

// 4. get Rows
Row r = sh.getRow(0);

// 5. to count number of cells in a row
int cn = r.getLastCellNum();

S.O.P(cn);
}
```

Note :-

* getLastRowNum() returns index value of last row.
* getLastCellNum() returns ~~last~~ number of cells present in a row.

A1	B1	C1
A2	B2	C2
A3	B3	C3

Write a script to print all the datas in the table format.

```
public static void main(String[] args) throws Exception
{
// 1. create & read the file
```

Scanned by CamScanner

```
File file = new File("./data/input.xlsx");
FileInputStream fis = new FileInputStream(file);
// To 2. create workbook
Workbook wb = WorkbookFactory.create(fis);
// 3. get sheet
Sheet sh = wb.getSheet("Sheet1");
int rn = sh.getLastRowNum();
for (int i=0; i<=rn; i++)
{
    int cc = sh.getRow(i).getLastCellNum();
    for (int j=0; j<cc; j++)
    {
        system.out.print(sh.getRow(i).getCell(j).toString()+" ");
    }
    system.out.println();
}
}
```

getStringCellValue()

Scanned by CamScanner

```
public class ExcelData
{
    //To get the data
    public static String getData(String filePath, String sheetName, int rn, int cn)
    {
        try
        {
            FileInputStream fis = new FileInputStream(new File(filePath));
            Workbook wb = WorkbookFactory.create(fis);
            String data = wb.getSheet(sheetName).getRow(rn).getCell(cn).
                                                        toString();
            return data;
        }
        catch(Exception e)
        {
            return "";
        }
    }

    // To get row count
    public static int getRowCount(String filePath, String sheetName)
    {
        try
        {
            FileInputStream fis = new FileInputStream(new File(filePath));
            Workbook wb = WorkbookFactory.create(fis);
            int rc = wb.getSheet(sheetName).getLastRowNum();
            return rc;
        }
```

Scanned by CamScanner

```
    catch(Exception e)
    {
        return 0;
    }
}
//To get cell count
public static int getCellCount(String filePath, String sheetName, int rn)
{
    try
    {
        FileInputStream fis = new FileInputStream(new File(filePath));
        Workbook wb = WorkbookFactory.create(fis);
        int cc = wb.getSheet(sheetName).getRow(rn).getLastCellNum();
        return cc;
    }
    catch(Exception e)
    {
        return 0;
    }
}
}
```

Scanned by CamScanner

Test NG Suite :-

*. In our automation framework we have created Testng Suite to execute the framework.

*. In the Testng.xml file instead of specifying individual class names, we have specified the package which contains all the test classes.

```
<suite name="Suite">
  <test thread-count="5" name="Test">
    <packages>
      <package name="com.actitime.tests"/>
    </packages>
  </test>
</suite>
```

Scanned by CamScanner

Automation framework implementation :-

*. It is mostly done by the senior automation Engineers.

*. In this stage we will convert manual test cases into automation test script.

① On what basis do you select the test case for automation?

*. It should be a part of regression test case

*. It should not contain any manual Interventions.

② Do you automate Smoke test case?

Yes, if it is part of regression Suite

③ can you achieve 100% automation.

NO, because

→ Technology will not support

→ cost will be very high

Note :-

In automation we call the smoke testing as Dry Run

④ Give some examples for Manual Interventions.

→ ~~Captha~~ Capcha

→ OTP

→ Barcode reader

→ Multimedia

→ Game Testing

→ Embedded System

→ ~~Beuter~~ Setter box testing & etc.

→ Swipe machine

Scanned by CamScanner

(5) what is your approach to handle OTP & captcha

→ Talk to the development team & ask him to make it static (or) remove the feature until completion of automation

Test case :-

1. valid Login Logout:

Step-1 :- open the browser

2 :- Enter the url

verify actiTime - login page is displayed or not

3 :- Enter valid un

4 :- Enter valid pw

5 :- click on login

~~verify login displayed or not~~

verify Enter Time Track page is displayed or not

Test Case : 2 : Invalid Login :

Step-1 : open the browser

2 : Enter the url

verify actiTime - Login page is displayed or not

3 : Enter invalid un

4 : Enter invalid pw

5 : click on Login

verify error message is displayed or not

6 : Close browser

Scanned by CamScanner

Test case-3 :

Verify version

Step-1 : Open the browser

2 : Enter the url

→ Verify actiTime - login page is displayed or not

→ verify the version of actitime.

3 : close the browser

Test case-4 : Verify build number

Step-1 : open the browser

2 : Enter the url

→ verify actiTime login page is displayed or not

3 : Login to actiTime

4 : click on help

5 : click on about actiTime

→ verify the build number

Execute test cases manually

Note :-

Scanned by CamScanner

#1 Rules to develop POM class :-

*. Number of POM class should be same as number of webpage.

*. The name of the POM class should be same as Title of webpage and it should end with word Page.

*. For every webpage we have to create separate POM under POM.actiTime.page.package.

*. All the POM class should executed from basePage class

*. In every POM class we should declared element using @FindBy, initialize the elements by using public.

Scanned by CamScanner

Note :-

If the super class contains parameterized constructor then we have to call that constructor in sub class

#1

```
public class LoginPage extends BasePage
{
    //Declaration
    @FindBy(id = "username")
    private WebElement userNameTB;
```

Scanned by CamScanner

```
@FindBy (name = "pwd")
private WebElement passwordTB;
@FindBy (xpath = "//div[. = 'Login']")
private WebElement LoginBTN;
@FindBy (xpath = "//span[contains(. , 'invalid')]")
private WebElement errorMsg;
@FindBy (xpath = "//nobr[contains(text(), 'active')]")
private WebElement version;
// Initialization
public LoginPage (WebDriver driver)
{
    super(driver);
    PageFactory.initElement(driver, this);
}
// utilization
public void enterUserName(String un)
{ userNameTB.sendKeys(un);
}
public void enterPassword(String un)
{   passwordTB.sendKeys(pw);
}
```

Scanned by CamScanner

```
public void clickOnLogin()
{   loginBTN.click();

}
public void verifyTheTitle(String eTitle)
{ VerifyTitle(eTitle);

}
public void verifyVersion(String eVersion)
{   String aVersion = version.getText();
    Assert.assertEquals(aVersion, eVersion);

  }
}
public class EnterTimeTrackPage extends BasePage
{   //declaration
    @FindBy(id="logoutLink")
    private WebElement LogoutBTN;

    //initialization
    public EnterTimeTrackPage(WebDriver driver)
    {
        super(driver);
        PageFactory.initElement(driver, this)
    }
```

Scanned by CamScanner

```
//utilization
public void clickOnLogout()
{
    LogoutBTN.click();
}
}
```

* The name of the test class should be same as Test case name or test case id & word test.
* Inside the test method call the methods of POM class to perform action on the browser
* Before calling the POM class methods specify the test case steps in line comment
* If any methods of test class requires any test data

Scanned by CamScanner

#8 Rules to develope test class :-

*. For every manual test cases, we have to create separate Test class.

*. For every test case we have to create seperate testclass under com.actitime.TestPackage.

*. The name of Testclass should be same as test case id and it should end with the word test.

*. The name of testmethod should start with the word test.

*. In the test method call pom class method to perform the action

*. Before calling the POM class method specify the test case steps InLine Comment.

Scanned by CamScanner

* If any method of POM class need any test data then those data should be taken from Excel file.

Steps to store the data in Excel File:-

* For every webpage, we have to create separate sheet and name of that sheet should be same as title of webpage
* In every sheet the first row should be header and the data should be insert from 2nd row.

Ex:-

username	password	Title	version
admin	manager	actiTime - login	actiTime 2017.4
QSP	JSP		
1	2		
e	&		

Title
actiTime - Enter Time - Track

Scanned by CamScanner

Valid Login Logout() Test :-

```
@Listeners(com.activitime.generic.ListenersTest.class)
public class ValidLoginLogoutTest extends BaseTest
{
    @Test
    public void testValidLoginLogout()
    {
        String loginTitle = ExcelData.getData(file_Path, "Login", 1, 2);
        String user = ExcelData.getData(file_path, "Login", 1, 0);
        String pass = ExcelData.getData(file_Path, "Login", 1, 1);
```

Scanned by CamScanner

```
String enterTimeTrackTitle = ExcelData.getData(file_Path, "EnterTimeTrack",
                                                1, 0);
LoginPage lp = new LoginPage(driver);
EnterTimeTrackPage ep = new EnterTimeTrackPage(driver);
//verify the title of login Page
lp.verifyTheTitle(loginTitle);
//Enter valid username
 lp.enterUserName(user);
// verify the password
 lp.enterPassword(pass);
// click on login
 lp.clickOnLogin();
// verify EnterTimeTrackPage
lp.verifyTheTitle(enterTimeTrackTitle)
// click on logout
 ep.clickOnLogout();
// verify the title of LoginPage
 lp.verifyThetitle(loginTitle)
}
}
```

Scanned by CamScanner

Invalid Login Test :-

```
@Listener(com.actitime.generic.ListenerTest.class)
public class InvalidLoginTest extends BaseTest
{
    @Test
    public void testInvalidLogin()
    {
        String loginTitle = ExcelData.getData(file_path, "login", 1, 2);
        String user = ExcelData.getData(file_path, "login", 2, 0);
        String pass = ExcelData.getData(file_path, "login", 2, 1);
        LoginPage lp = new LoginPage(driver);
    // verify the title of login Page
        lp.verifyTheTitle(loginTitle);
    // verify the invalid username
        lp.enterUserName(user);
    // verify Invalid password
        lp.enterPassword(pass);
    // click on login
        lp.clickOnLogin();
```

Scanned by CamScanner

```
//verify the error message
    lp.verifyErrorMessage();
}

{
```

VersionTest :-

```
@Listeners(com.actitime.generics.ListenersTest.class)
public class verifyVersionTest extends BaseTest
{
    @Test
    public void testVersion()
    {
        String loginTitle = ExcelData.getData(file_path, "login", 1, 2);
        String version = ExcelData.getData(file_path, "login", 1, 3);
        LoginPage lp = new LoginPage(driver);
        //verify the title of loginPage
        lp.verifyTheTitle(loginTitle);
        //verify the version
        lp.verifyVersion(version);
    }
}
```

Scanned by CamScanner

Automation framework Execution :-

It is mostly done by the freshers

(1) How do you execute your automation framework?

By using testng.xml file

(2) Can any manual Test engineers execute your framework

Yes

(3) Can you execute your framework without using Eclipse?

Yes, using command Prompt we can execute the framework.

Steps :-

→ Go to the location where testng.xml file is present

→ create a file with the following command

Java -cp bin; jars* org.testng.TestNG testng.xml

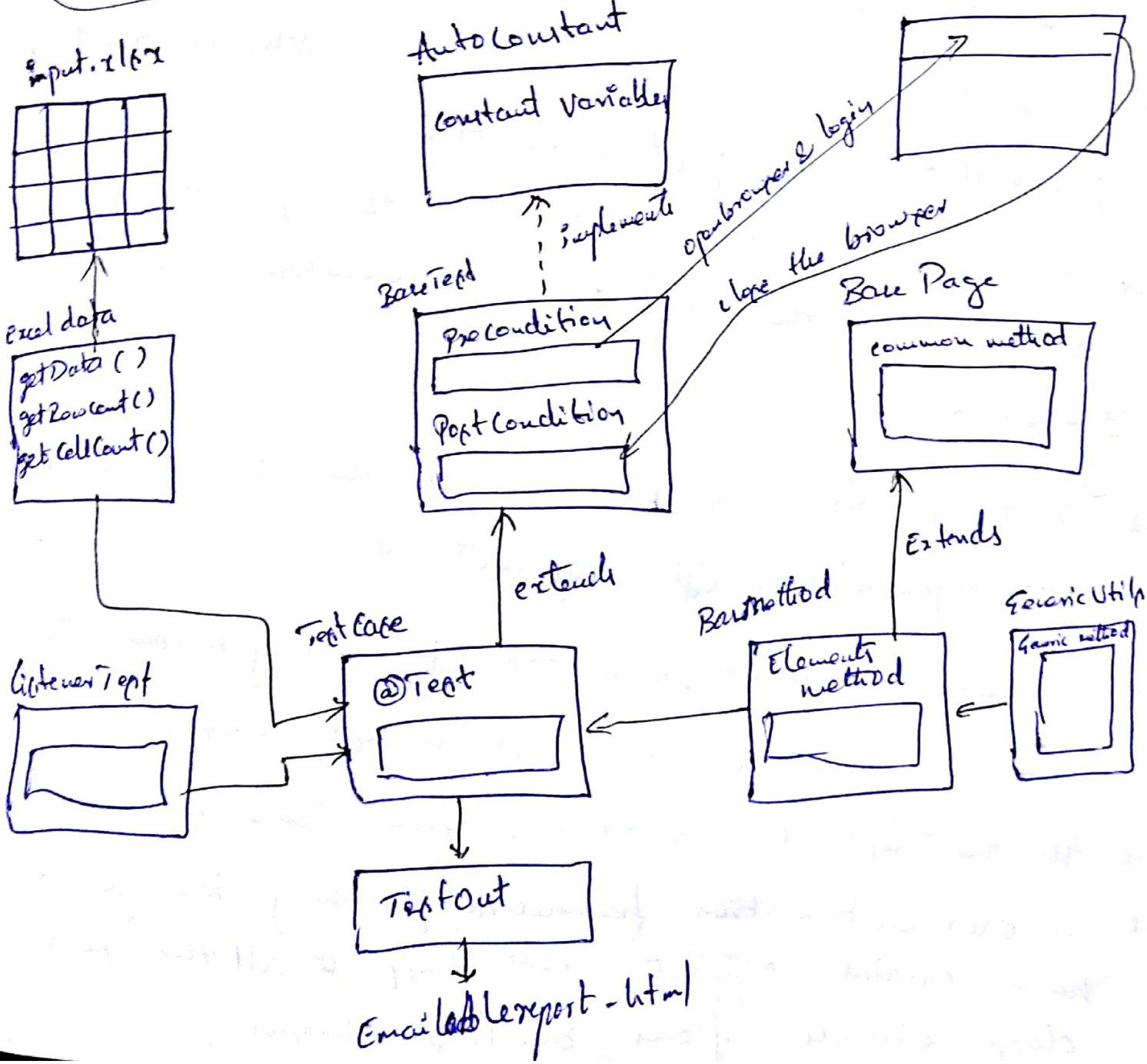
→ Save the above file with .Bat extension

→ double click on the Batch file which we execute the automation framework.

Scanned by CamScanner

Explain your automation framework

* If any scripts is failed then testng will generate an xml file which is called testng-failed.xml which contains only with fail & skipped test classes.
* we can use that testng file to execute failed & skipped test scripts.



Scanned by CamScanner

* It is a hybrid framework which is combination of data driven & method driven
* It is developed using testng.
* In our automation framework we have created an interface called autoConstant which contains all the Constant Variables.
* In our automation framework we have a class called BaseTest which implements autoConstant interface.
* BaseTest class contains two methods that is precondition & post condition.
* Precondition method is used to open the browser and enter the url & it is developed by using @Beforemethod.
* Post condition is used to close the browser & is developed by using @Aftermethod.
* In our automation framework for every manual Test case we have created seperate test class.
* All the test classes extends from BaseTest class
* In our automation framework for every webpage, we have created seperate POM class & all the POM class extends from basePage Classes

Scanned by CamScanner

* Base Page class contains the common methods which are present in POM classes.

* To take the data from the excel file we have created a class called excelData which contains generic methods.

* To get the data, to count no of rows, to count the no of cells.

* To take the screenshot of pagescript we have implemented an interface called ~~[illegible]~~ ITestListener from the class Listenersutest.

* In our automation framework we have created a class ~~[illegible]~~ called generic ~~[illegible]~~ utils which contains Generic methods under Listbox, popups etc.

* Once after execution of framework the result will be stored in emailable report.html format under test output folder.

Scanned by CamScanner

Selenium Grid :-

→ testing the application in multiple platform it called as compatability Testing

→ In selenium we can achieve compatability testing by using selenium Grid.

→ Here we will take two types of systems.

1) hubs / server
2) Node / client

→ Hub it a system where the automation framework it present

→ Technically we call it has server.

→ Node it a system where we will execute the automation framework, technically we call it has client

Note :- for one hub we can connect one (A) more node

→ configuring node

Configuring Node :-

Tools Required :

* JDK
* Driver executable files
* Selenium JRE files
* Browsers
* Application under testing :

Scanned by CamScanner

*. In the node System create a folder with the name node and store all the driver executable file and selenium Jar file.

*. Rename the driver executable file and jar file c.exe, g.exe ~~FF~~ s.JRE.

*. In the node folder create a file with the following command and save it pad (batch file).

Java -D WebDriver.Chrome.driver.exe -D WebDriver.gecko.driver

g.exe -Jar -jar.

here "-" it stands for drivers.

double click on batch file it will execute the command and display the following messages

Selenium Server is up and Running

Scanned by CamScanner

#3) Configuring Hub:-

* To execute the framework in the remote machine we used a class called remote webDriver.

remoteWebDriver class takes 2 arguments of type

i) Url

ii) DesiredCapability.

URL:-

here we have to specify the URL of the remote machine by using the class called URL

Ex:-

```
public void precondition()
{
URL url = new URL("http://localhost:4444/wd/hub/");
}
```

DesiredCapability:-

it is used to set the name of the browser.

Ex:-

update precondition method as shown

```
@BeforeMethod
public void precondition()
{
URL url = new URL("http://localhost:4444/wd/hub/");
DesiredCapability dc = new DesiredCapability();
dc.setBrowserName("chrome");
```

Scanned by CamScanner

```
driver = new RemoteWebDriver(url, dc);
driver.get("https://demo.actiTime.com/login.do");
}
```

* After execution of framework the result will be stored in the testoutput folder of hub system.

Scanned by CamScanner

TestNG Parameter (or) Parameterization :-

passing the value from testng.xml to testNG class is called as testng Parameter (or) parameterization.

Ex: testng.xml

```
<Suite name = "Suite" >
    <test name = "Test" >
        <parameter name = "city" value "USA" />
        <parameter name = "area" value = "kalaripalya" />
            <classes>
            <class name = "pkg.Demo" />
            </classes>
    </test>
</Suite>
```

Test class :-

```
public class Demo
{
    @ Parameters ({"city", "area"})
    @ Test
    public void testA (String city, String area)
    {
        Reporter.log (city, true);
        Reporter.log (area, true);
    }
}
```

Scanned by CamScanner

Note :-

After Execution of framework in the node system the result will be in test output folder of hub system. update testng.xml file & precondition method as shown below.

testng.xml

```
<suite name = "suite" parallel = "tests" >
    <test name = "Chrome Test" >
        <parameter name = "nodeUrl" value = "http://192.168.05.4444/web/hub/" />
        <parameter name = "browser" value = "Chrome" />
        <parameter name = "appUrl" value = "http://demo.actitime.com/login.do" />

        <packages>
            <package name = "com.actitime.test" />
        </packages>
    </test>
    <test name = "Firefox Test" >
        <parameter name = "nodeUrl" value = "http://192.168.05:4444/wb/hub" />
        <parameter name = "browser" value = "Firefox" />
```

Scanned by CamScanner

```
<parameter name="appUrl" value="https://demo.actitime.
        com/login.do" />

<packages>
    <package name="com.actitime.tests" />
</packages>

</test>
</suite>
```

test class :-

@parameters ({"nodeUrl", "browser", "appUrl"})

@BeforeMethod

```
public void preCondition(String nodeUrl, String browser,
                         String appUrl) throws Malformed
                                               UrlException
{
    URL Url = new URL(nodeUrl);
    DesiredCapabilities dc = new DesiredCapabilities();
    dc.setBrowserName(browser);
    driver = new RemoteWebDriver(url, dc);
    driver.manage().window().maximize();
```

Scanned by CamScanner

#4 Maven :-

* Maven is dependency tool
* In Selenium we use the maven project to update the jar file automatically.
* implementation of maven project is a part of automation framework design.
* There are 3 stages in Maven
  1. create a maven project
  2. Specify the dependency
  3. Execute the framework

Create a Maven Project :-

step-1 : Right click on java project

Goto configure and click on convert to maven project follow the default instructions and click finish. which will create file called * pom.xml.

Scanned by CamScanner

Step-2: Specify dependency

Here dependency refers to jar files

① To add the dependency double click on pom.xml file which will open the pom editor.

② Go to dependency tab and click on add

③ Specify the Group id. Artifact id and version

④ click on ok and save

⑤ we can get the dependency from the following website

URL :- https://~~mave~~ mvnrepository.com/

Scanned by CamScanner

Specify the dependency :-

→ Here dependency refers to jar files

step ① :- Double click on the pom.xml file which will open POM editor.

step ② :- go to dependencies (tab) & click on add.

step ③ :- Specify the Group id, Artifact id & version form the following website.

https://mvnrepository.com/

Selenium :-

```
<dependency>
<groupId> org.seleniumhq.selenium </groupId>
<artifactId> selenium-~~server~~ java </artifactId>
<version> 3.11.0 </version>
</dependency>
```

poi :-

```
<dependency>
<groupId> org.apache.poi </groupId>
<artifactId> poi </artifactId>
<version> 3.17 </version>
</dependency>
```

poi-ooxml :-

```
<dependency>
<groupId> org.apache.poi </groupId>
<artifactId> poi-ooxml </artifactId>
<version> 3.17 </version>
</dependency>
```

Scanned by CamScanner

Testng :-

```
<dependency>
    <groupId> org.testng </groupId>
    <artifactId> testng </artifactId>
    <version> LATEST </version>
</dependency>
```

* Here groupId refers to authorization name.
artifactId refers to productname
* To download the latest version of surefile we have to specify the version as LATEST.

Execute Maven FrameWork :-

* Here to execute the maven framework by POM.xml file.
* during the run time it will download all the specified dependency stored the jars file in maven local Repository (.m2\repository).
* To compile the maven project we will have all maven compiler.plugin., and if execute the maven project we use maven surefire plugin.

Scanned by CamScanner

* To execute the maven project we have to use another plugin which is called Maven ~~Surefire~~ Surefire Plugin which is available in the below website.

http: maven.apache.org/surefire/maven-surefire-plugin/examples/testng.html

* Copy the maven surefire plugin & paste it after maven compiler plugin
* To run the pom.xml file, right click on pom.xml file
* go to run as & click on maven test
* pom.xml file will trigger the testng.xml file & testng.xml file will execute all the test class present with the framework.

what are the difference b/w Java & Maven Project

Java	Maven
→ all the jar files should be updated manually.	→ jar file will be updated automatically.
→ all the jar files should be present under jars folder	→ all the jar files are stored in maven local repository (.m2/repository)
→ all the associated jar files are present under reference libraries	→ all the associated jar files are stored under maven dependencies
→ Here we use testng.xml file to execute the framework	→ Here we use pom.xml file to execute the framework
→ Here the reports will be stored under test output folder	→ The report will be stored in surefire report of target folder

Scanned by CamScanner

GIT Hub :-

* Git Hub is a version control tool / online repository tool. which is used to store the automation framework and it will manage the automation scripts.
* Once the frame work is design, lead will create repository in the github server and he will upload the automation framework in git hub.
* all the automation engineer should download the frame work from github to there local system.
* Once after writing the script the automation engineer should upload the script into the github server.
* This process is called as push.
* To download the files from the github, we use an option called poll.

Uploading framework to github :-

It is done by lead (or) the project manger

step-1 :- login to following website
https://github.com/login

step-2 :- click on start a project

step-3 :- Specify the repository name, repository name should be same as project name

Scanned by CamScanner

step-4 :- public & private buttons

step-5 :- click on create repository and it will generate URL

Ex:- https://github.com/sBRakesh/mavenDemo1.git

step-6 :- copy the URL

step-7 :- In eclipse right click on the project team and click on share project.

select the checkbox and select the location and click on create repository and finish.

which will create repository in the local system.

step-8 :- Right click on the project, ~~and specify the commit message~~ Go to team and click on commit

step-9 :- select all the files and specify the commit message.

step-10 :- click on commit and push

step-11 :- specify the URL, username, password and click on next

step-12 :- follow the default instruction until OK.

Scanned by CamScanner

Download framework to github :-

*. It should be downloaded by all the automation engineers

*. Once the framework is design, lead will store the framework in github server and he will share URL to all the automation Engineers.

Step-1 :- In eclipse go to file menu and click on import.

Step-2 :- Expand git folder and select project from git and click on next

Step-3 :- Select Chrome URL click on next

Step-4 :- Specify the URL and username and password.

Step-5 :- follow the default instruction until finish.
Which will download frame work from github to local work station

Steps to upload modified file (or) new file to github :-

→ It should be done by all the automation engineer, once if modify the existing file (>) or create a new file (?)

Step-1 :- Right click modified file (or) new file and click on commit.

Step-2 :- Select the files and drag and drop into staged changes and specify the commit msg.

Scanned by CamScanner

Step-3:- Click on commit & push and follow the default instructions till finish.

steps to Download framework from github :-

Step-1 :- Right click on project

Step-2 :- select team

Step-3 :- click on pull and follow the default instructions which will download the modified files from github servers to local workstation.

Jenkins :-

* Jenkins is ~~a~~ continuous integration tool which basically used by developers. to create and manage the build.
* We can integrate with Jenkins so that the framework executed automatically when the new build is created.
* To integrate the framework with jenkins we need the following informations
  1. URL of jenkins
  2. Project Name
  3. Username & password of Jenkins.

Note :- * Installation and configuration is done by developers
* Build is created by developers.
* we have to integrate our framework with jenkins.

Scanned by CamScanner

Steps to integrate with framework with jenkins:

Step-1: Login to jenkins
Step-2: click on project
Step-3: click on configure
Step-4: Select git under source code management
Step-5: Specify the URL of the github repository
Step-6: click on add select jenkins
Step-7: - Enter the jenkins username & password
Step-8: click on ok button
Step-9: click on Apply & Save.
Step-10: To create builds developer click on option build now
by developer click build now option jenkins perform the following steps.

* Download source code from developer repository.
* Compile the source code,
* longpress the source code
* Sends email notification.
* Download and framework.
* Execute pom.xml
* pom.xml will execute the TESTNG.xml file.

After execution of frame work the result will be stored in jenkins server.
To check the result click on project and click on link present under build history and click on console output.

Scanned by CamScanner