



MR. P.NAGESWARA RAO SIR

GUIDE TO

Software Development

**Engineer in
Testing**

FINAL EDITION



CONTENTS

I.	SDET [Software Development Engineer in Testing]	1-3
a)	Software Testing.....	1
b)	Quality Software.....	1
c)	Manual Testing vs Test Automation	1
d)	Need for Test Automation	2
e)	Types of software's	3
II.	SWD [Selenium Web Driver].....	4-74
a)	Configure Selenium Web Driver in tester computer	4
b)	Launch website under testing	11
c)	Locating elements in webpages.....	12
d)	xpath and types of xpath.....	13
e)	Methods in “WebDriver”	20
f)	Methods in “select” class	43
g)	“Actions” class	47
h)	“JavascriptExecutor” class (Interface)	55
i)	“WebDriverWait” & “FluentWait” classes in SWD	59
j)	Handling AJAX calls.....	61
k)	Working with Angular JS related pages	62
l)	Handling SSL (Secure Sockets Layer) Certificates.....	63
m)	Handling Browser Notifications.....	64
n)	Voice Automation	65
o)	cssSelector	69
III.	JAVA ROBOT	75-82
IV.	SIKULI X.....	83-90
a)	Configure Sikulix in tester computer	83
b)	Locating Elements	84
c)	Methods in “Screen” class.....	85
V.	EXTENT REPORTS	91-93
a)	Configure Extent Reports in tester computer	91
b)	Classes to be used.....	91

VI.	STAF [S/W TEST AUTOMATION FRAMEWORK]	94-96
a)	Waterfall vs Agile Scrum	94
b)	Agile Scrum Process	94
c)	Framework.....	96
d)	Prerequisites to follow a framework.....	96
VII.	JAVA BASICS	97-153
a)	Variables in Java	98
b)	Constants in Java	98
c)	Data types in Java.....	99
d)	Primitive Data Types.....	99
e)	Derived Data Types.....	99
f)	User Defined Data Type.....	100
g)	Types of classes.....	104
h)	Instance classes.....	104
i)	Static classes.....	105
j)	Interfaces	106
k)	Abstract class.....	108
l)	Singleton class.....	109
m)	Wrapper classes	110
n)	Inheritance (Relation in between instance classes)	111
o)	Polymorphism	115
p)	“super” & “this” keywords.....	117
q)	Operators in Java	117
r)	Control statements in Java	118
s)	“Scanner” class in JDK	128
t)	Type casting, type conversion and java generics	128
u)	Loops in java	129
v)	“String” class methods	141
w)	Exception Handling	146
x)	Run java code from command prompt	150
y)	Pattern Matching using Regular Expressions.....	151
z)	OOPS Concepts.....	152

VIII.	DDT [Data Driven Testing Framework].....	154-175
a)	Using Arrays.....	154
b)	DDT using HashMaps	161
c)	Using “.txt” file	164
d)	Using “.xls” file.....	169
IX.	KWD [Key Word Driven Framework]	176-185
X.	POM [Page Object Model Framework].....	186-194
XI.	Module Driven Framework using “TestNG”	195-217
XII.	BDD [Behaviour Driven Development Framework]	218-243
a)	BDD Framework execution flow	218
b)	BDD Framework development.....	218

I. SDET

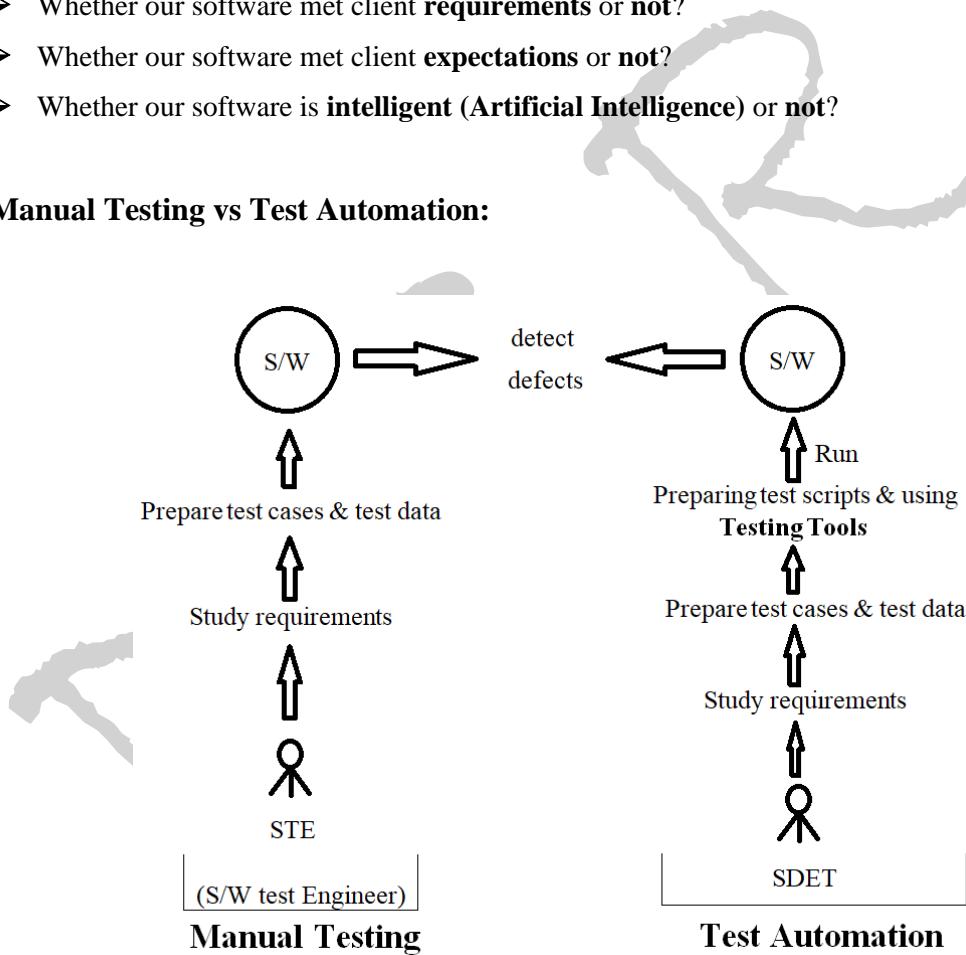
[Software Development Engineer in Testing]

- a) **Software Testing:** Test software with respect to client requirements and expectations.
(Ex: Usability, Performance, Security, Compatibility, Multilanguity ...etc.)

b) **Quality Software:**

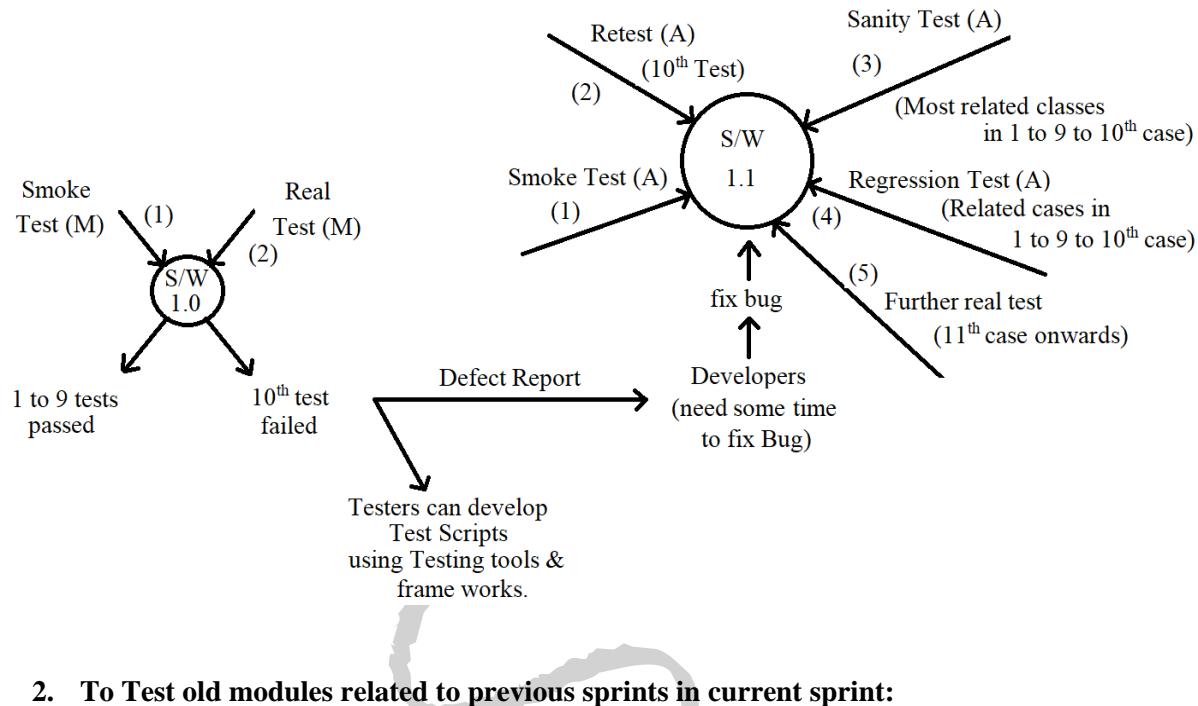
- Whether our software met client **requirements or not?**
- Whether our software met client **expectations or not?**
- Whether our software is **intelligent (Artificial Intelligence) or not?**

c) **Manual Testing vs Test Automation:**

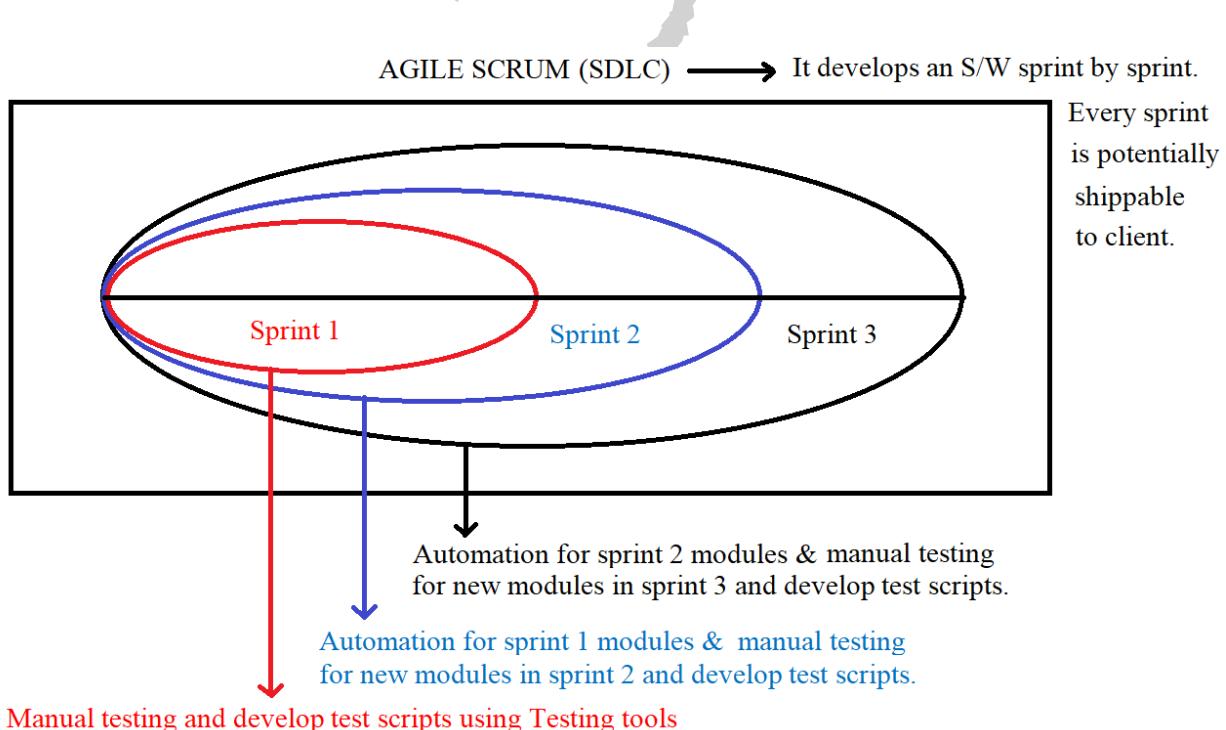


d) Need for Test Automation:

1. For Re and Regression Testing:

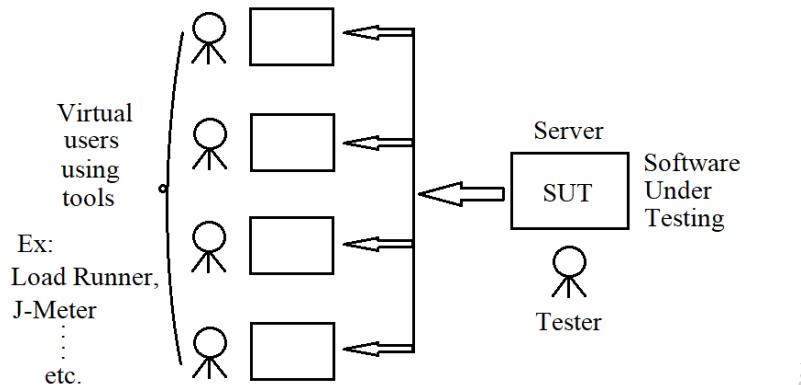


2. To Test old modules related to previous sprints in current sprint:



3. Complex tests using Automation only:

1) Ex: Performance Test



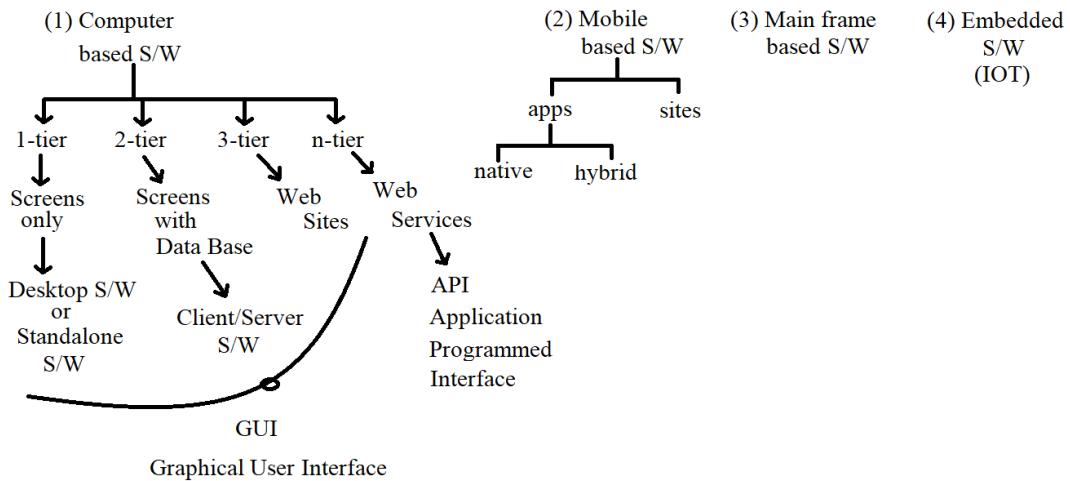
2) Ex:

Mobile apps test in various android/ios versions
Use emulators/simulators.

Conclusion:

- ❖ For Re and Regression Test → Time Save.
- ❖ For complex testing → Decrease cost.

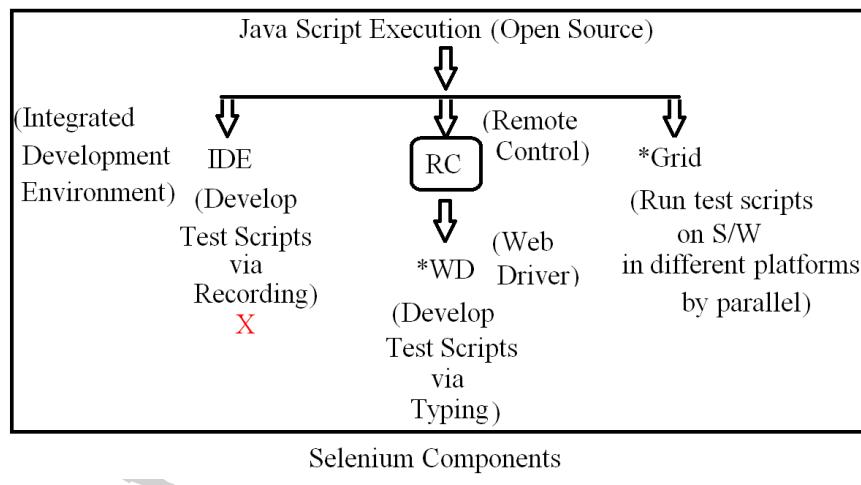
e) Types of software's:



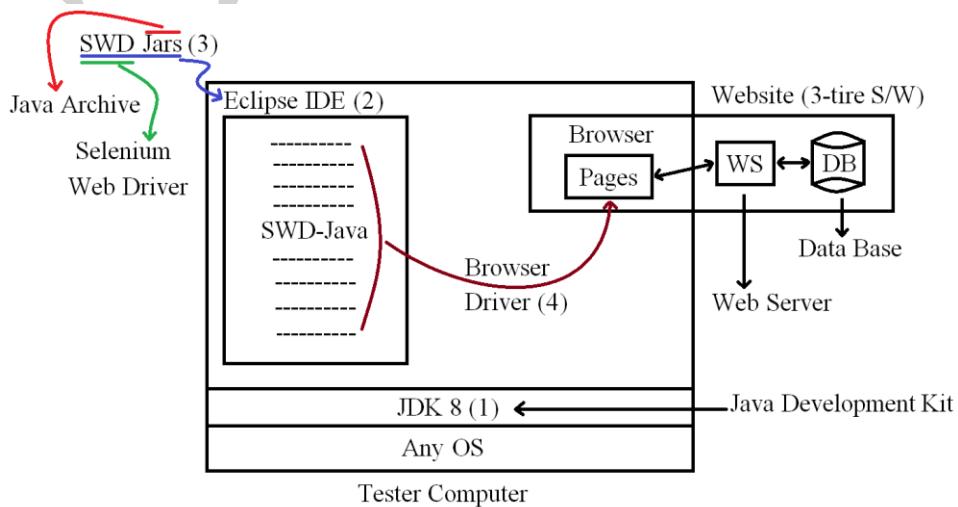
II. SWD

[Selenium Web Driver]

- ❖ Developed by Simon Stewart.
- ❖ Useful to automate websites in computers and useful to automate apps and websites in mobiles by integrating with appium.
- ❖ Open source.
- ❖ Available in various flavors. (Java, c#.net, ruby, perl, php, python and Java script).
- ❖ Due to platform-independent, java based SWD is famous.
- ❖ We are able to automate websites by opening those sites using any browser.
- ❖ Jason Huggins



a) Configure Selenium Web Driver in tester computer:



OS → Operating System

JDK → Java Development Kit

IDE → Integrated Development Environment

Jar → Java Archive

SWD → Selenium Web Driver

WS → Web Server

DB → Data Base

Step 1: Download and install JDK8:

- Create a new folder (folder name is a single word).
- Right click on “computer”.
- Go to “properties”.
- Identify bit-size. (Ex: 32-bit/64-bit).
- Close computer properties window.
- Go to “Google” site.
- Enter “JDK8 download”.
- Go to “oracle.com” site.
- Scroll down and click “accept license agreement”.
- Click on JDK download link with respect to operating system and bit-size.
(Ex: 32-bit → winx86, 64-bit → winx64)
- Paste that download in personal folder (Previously created folder).
- Double click on that downloaded file.
- Click “next” until getting “finish”.

Note 1: After completion of JDK installation we can get folder path shown in below.

C:\Program Files\Java\jdk1.8.0_181

Note 2: After completion of JDK installation, we need to follow below navigation to configure java in computer.

(Windows):

- Copy path of JDK folder [Ex: C:\Program Files\Java\jdk1.8.0_181].
- Right click on “Computer”/“This PC”.
- Go to “Properties”.
- Go to “Advanced system settings”.

- Go to “Environment variables”.
- Go to “system variables”.
- Click on “New”.
- Enter variable name as JAVA_HOME
- Paste path of JDK folder as value.

Ex:

Name	JAVA_HOME
Value	C:\Program Files\Java\jdk1.8.0_181

- Select “path” variable in “system variables”.
- Click on “edit” [click on “edit text” again in windows 10].
- Paste path of JDK\bin; at end of existing path value.

Ex:

Name	Path
Value	-----;Path of JDK\bin;

- Click on “ok” again and again [3 times].
- Close computer properties window.

Note 3: To ensure the correctness of JDK installation and configure, we need to run below commands at command prompt.

- javac
- java

Step 2: Download and launch eclipse IDE.

To write selenium-java based code for websites automation, we need an IDE (Integrated Development Environment) like eclipse IDE, IntelliJ and Net beans.

Here eclipse IDE is more familiar to get eclipse IDE we need to follow below navigation.

- Go to “Google” site.
- Enter “eclipse IDE oxygen download”.
- Click “search”.
- Go to open “eclipse.org” site.
- Go to “more downloads”.
- Select and click on “eclipse oxygen”.

- Go to “eclipse IDE for Java developers”.
- Click on bit-size with respect to OS [Ex: 32-bit/64-bit].
- Click on “download”.
- Paste that download in personal folder.
- Extract that download.
- Open that extracted folder.
- Create desktop shortcut for eclipse software.

Note 1: When we launch eclipse IDE first time we need to provide a local folder as workspace.

- Double click on “eclipse IDE” icon on desktop.
- Browse a personal folder as workspace.
- Select “don’t ask again check box”.
- Click on “launch”.
- Close welcome screen.

Note 2: After completion of eclipse IDE launching we need to follow below navigation to create a java project.

- File menu
- New
- Java project
- Enter a name to project as single word in lower case.
- Click “finish”.
- Right click on that created project.
- Select “properties”.
- Select “java compiler”.
- Set to latest [Ex: 1.8].
- Java build path
- Go to “libraries”.
- Observe JRE version and set to latest. [Ex: 1.8]
- Click “apply” and close button.

Note 3: When java project was created successfully, we need to create a package under that project.

- Right click on “java project”.
- Select “new”.

- Click on “package”.
- Enter a name as single word in lower case.
- Click on “finish”.

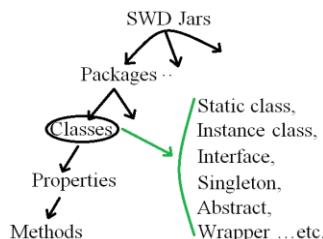
Note 4: After completion of package creation we need to create a class with **main()** method.

- Right click on “package” and select “new”.
- Select “class”.
- Enter a name to class as single word with init cap [initially capital letter].
- Select **main()** method option
- Click “finish”.

Step 3: Download and associate selenium web driver jars.

- Go to “seleniumhq.org” site.
- Click on “download”.
- Go to Web Driver language bindings.
- Click on “download for Java”.
- Paste that download in personal folder.
- Extract that download in folder.
- Go to “eclipse IDE”.
- Right click on “project”.
- Properties.
- Java build path.
- Libraries.
- Add external JARs.
- Browse the path of JARs inside of libs folder [10] and outside of libs folder [2].
- Click “apply” and close button.

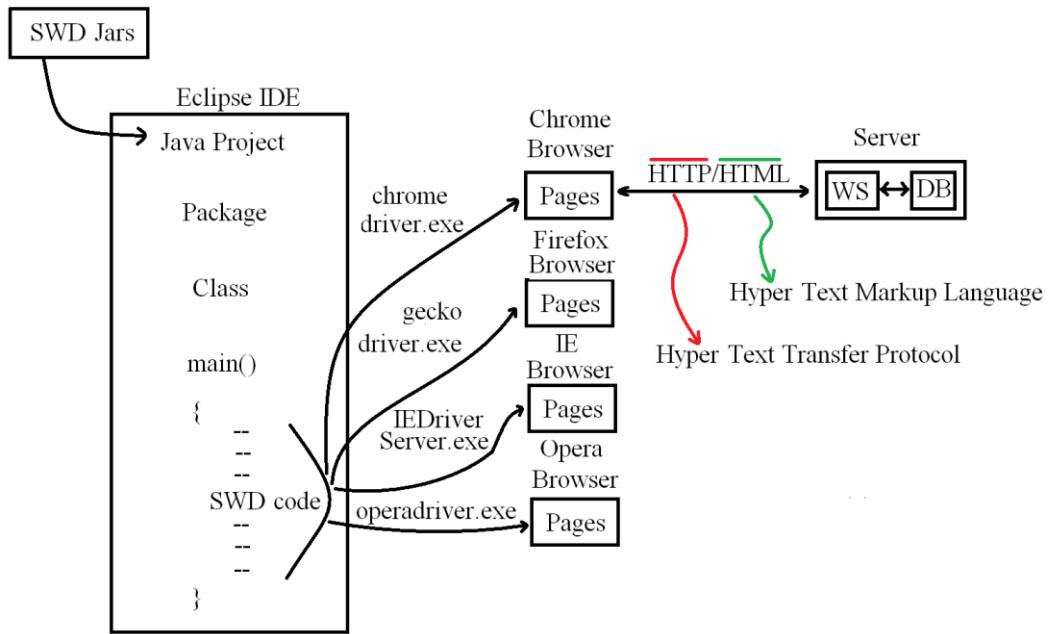
Note 1: Every JAR file in Selenium web driver JARs can provide packages. Every package is having classes. Every class is having properties and methods. Every method can perform an operation.



Ex:

- Click on a button.
- Fill a text box with data.
- Double click on element.
- Right click on element...etc.

Step 4: Download and use browser drivers



✓ To work with "chrome" browser:

- Go to "seleniumhq.org" site.
- Click on "download".
- Go to "third party drivers".
- Click on "latest for chrome driver".
- Download "chrome driver" with respect to OS and bit-size.
- Paste that download in personal folder.
- Extract that download to get **chromedriver.exe**
- Write below like code in **main()** method of a class in eclipse IDE.

```
System.setProperty("webdriver.chrome.driver", "D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
```

Note: Here "System" is a static class in JDK, "ChromeDriver" is an instance class in SWD JARs and "driver" is an object-instance.

✓ **To work with “Firefox” browser:**

- Go to “seleniumhq.org” site.
- Click on “download”.
- Go to “third party drivers”.
- Click on “latest for gecko driver”.
- Download “gecko driver” with respect to OS and bit-size.
- Paste that download in personal folder.
- Extract that download to get **geckodriver.exe**
- Write below like code in **main()** method of a class in eclipse IDE.

```
System.setProperty("webdriver.gecko.driver", "D:\\DineshReddy\\geckodriver.exe");
FirefoxDriver driver=new FirefoxDriver();
```

Note: Here “System” is a static class in JDK, “FirefoxDriver” is an instance class in SWD JARs and “driver” is an object-instance.

✓ **To work with “IE” browser:**

- Go to “seleniumhq.org” site.
- Click on “download”.
- Go to “IE Driver Server”.
- Download “IE driver” with respect to OS and bit-size.
- Paste that download in personal folder.
- Extract that download to get **IEDriverServer.exe**
- Write below like code in **main()** method of a class in eclipse IDE.

```
System.setProperty("webdriver.ie.driver", "D:\\DineshReddy\\IEDriverServer.exe");
InternetExplorerDriver driver=new InternetExplorerDriver();
```

Note: Here “System” is a static class in JDK, “InternetExplorerDriver” is an instance class in SWD JARs and “driver” is an object-instance.

✓ **To work with “Opera” browser:**

- Go to “seleniumhq.org” site.
- Click on “download”.
- Go to “third party drivers”.
- Click on “latest for opera driver”.
- Download “opera driver” with respect to OS and bit-size.
- Paste that download in personal folder.
- Extract that download to get **operadriver.exe**

- Write below like code in **main()** method of a class in eclipse IDE.

```
System.setProperty("webdriver.opera.driver", "D:\\DineshReddy\\operadriver.exe"
);
OperaDriver driver=new OperaDriver();
```

- If error gets then write below like code

```
OperaOptions oo=new OperaOptions();
oo.setBinary("C:\\Program Files\\opera\\53.0.2907.68\\opera.exe");
System.setProperty("webdriver.opera.driver", "D:\\DineshReddy\\operadriver.exe"
);
OperaDriver driver=new OperaDriver();
```

Note : Here “System” is a static class in JDK, “OperaDriver” and “OperaOptions” is an instance class in SWD JARs and “driver” is an object-instance.

“53.0.2907.68” You have to select this version according to your latest browser version.

Note 1:

Browser Name	Required Browser Driver	Using Platforms(OS)
Chrome (latest)	chromedriver (latest)	Windows Linux Mac : etc.
Firefox (latest)	geckodriver (latest)	Windows Linux Mac : etc.
Opera (latest)	operadriver (latest)	Windows Linux Mac : etc.
IE	IEDriverServer	Windows
Edge	edgedriver	Windows
Safari	safaridriver	Mac
:	:	:

Note 2: We responsible to maintain compatibility between browsers and corresponding browser drivers.

b) Launch website under testing:

We can use below code to launch a website in corresponding browser.

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

URL →Uniform Resource Locator

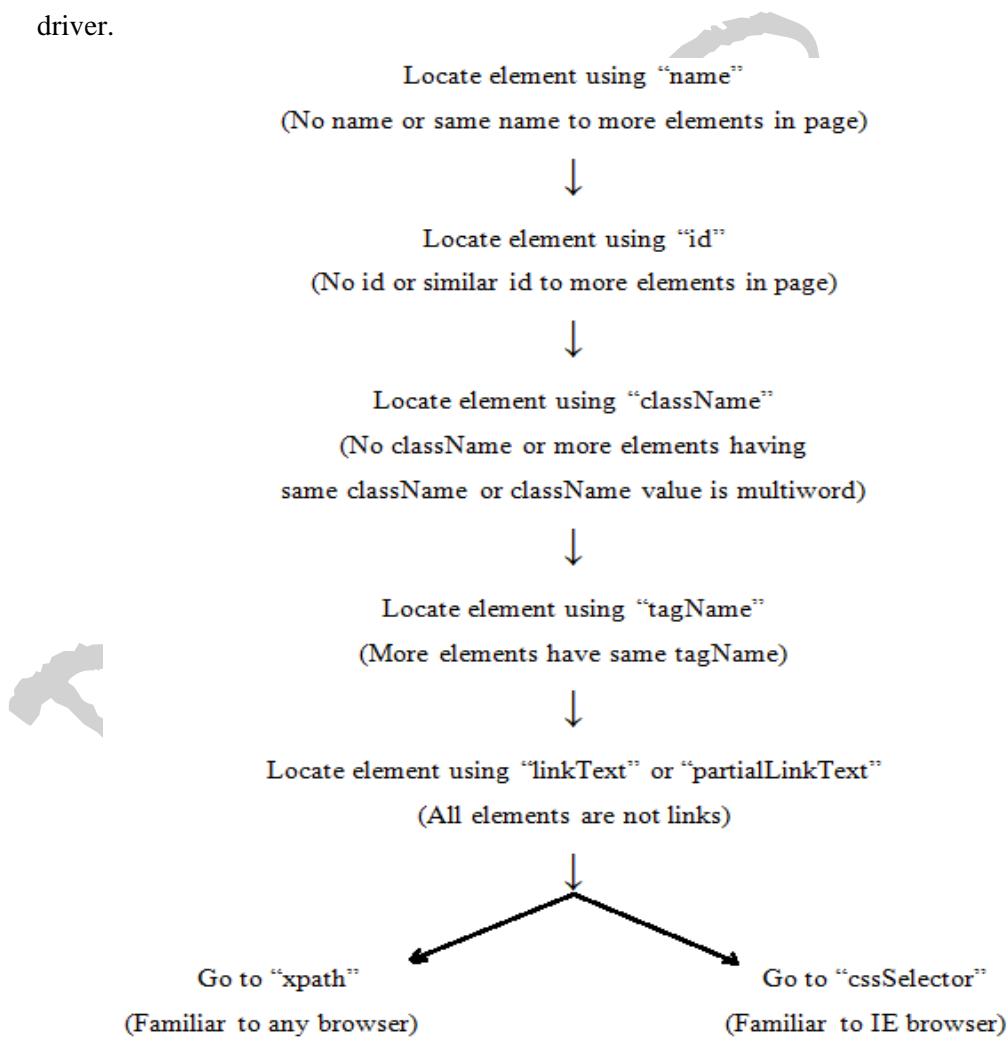
Ex: <http://www.google.co.in>

c) Locating elements in webpages:

In general, every test automation code need to below activities.

- Launch Site Under Testing (SUT)
- Locate elements
- Operate elements
- Observe elements functionality to confirm test passed/failed.
- Close SUT

While locating elements in webpages we need to follow below algorithm related to selenium web driver.



From the above algorithm, selenium web driver can allow as to locate elements using 8 ways.

- | | | |
|----------------|--------------------|------------------------|
| → By.name() | → By.id() | → By.className() |
| → By.tagName() | → By.linkText() | → By.partialLinkText() |
| → By.xpath() | → By.cssSelector() | |

Example 1:

- Launch mercury tour site using chrome browser.
- Click register link.
- Fill fields and select INDIA as country.
- Click register button.
- Close site.

Automation code [in main() of a class]:

```
//Launch site using chrome
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://newtours.demoaut.com/");
Thread.sleep(5000);
//Click register link
driver.findElement(By.linkText("REGISTER")).click();
Thread.sleep(5000);
driver.findElement(By.name("firstName")).sendKeys("xxxx");
driver.findElement(By.name("lastName")).sendKeys("xxxx");
driver.findElement(By.name("phone")).sendKeys("xxxx");
driver.findElement(By.name("userName")).sendKeys("xxxx");
driver.findElement(By.name("address1")).sendKeys("xxxx");
driver.findElement(By.name("address2")).sendKeys("xxxx");
driver.findElement(By.name("city")).sendKeys("xxxx");
driver.findElement(By.name("state")).sendKeys("xxxx");
driver.findElement(By.name("postalCode")).sendKeys("xxxx");
//Drop-down Automation
WebElement we=driver.findElement(By.name("xxxx"));
Select s=new Select(we);
s.selectByVisibleText("INDIA");
//Automate further elements
driver.findElement(By.id("email")).sendKeys("xxxx");
driver.findElement(By.name("password")).sendKeys("xxxx");
driver.findElement(By.name("confirmPassword")).sendKeys("xxxx");
driver.findElement(By.name("register")).click();
Thread.sleep(5000);
//Close site
driver.close();
```

d) xpath and types of xpath:

“xpath” is an address of an element. In general, xpath’s are two types such as **absolute xpath** and **relative xpath**.

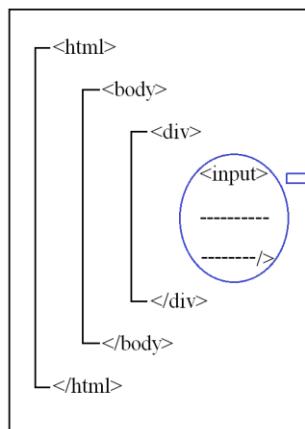
- **Absolute xpath** is a list of all tags from **/html** to target element tag.

In general SDET’s cannot prefer in **absolute xpath** to locate elements because:

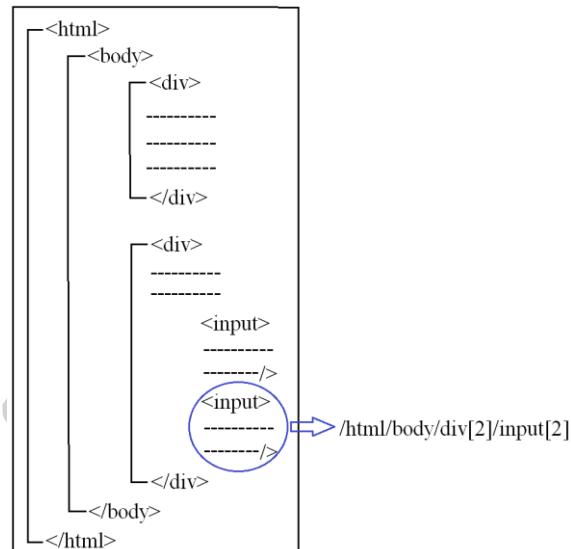
- When our webpage source is lengthy **absolute xpath** preparation for an element is complex.

- When our webpages are dynamic (continuously changing) **absolute xpath** is not correct to locate an element.

Examples are shown in the below figures:



Example 1



Example 2

- Due to above reasons, SDET's can go to **relative xpath** to locate an element. **Relative xpath** is a syntax based like shown below.

Syntax 1:

`//tagName[@attribute='value']`

Ex: `<input type="email" class="whsOnd zHQkBf" jsname="YPqjbf" autocomplete="username" spellcheck="false">`

`//input[@spellcheck='false']`

Here in example

- input is tagName.
- type, class, jsname, autocomplete, spellcheck are attributes.
- email, whsOnd zHQkBf, YPqjbf, username, false are values of attributes.

Syntax 2:

`//tagName[@attribute='value'][@attribute='value']`

Syntax 3:

`//*[@attribute='value']`

Here * indicates any tag in html.

Syntax 4:

`(//tagName[@attribute='value'])[index]`

or

`(//*[@attribute='value'])[index]`

Here index starts with number **1**.

Syntax 5:

`//tagName[@attribute='value' or @attribute='value']`

or

`//*[@attribute='value' or @attribute='value']`

Ex: If such of toggle element present this type of syntax will be used.

 OK/CANCEL

`//*[@name='OK' or @name='CANCEL']`

Syntax 6:

`(//tagName[@attribute='value'])|((//tagName[@attribute='value'])`

or

`(//*[@attribute='value'])|((//*[@attribute='value'])`

Ex: If only anyone is visible in web page at that time this type of syntax is used.

 OK CANCEL

`(//*[@name='OK'])|((//*[@name='CANCEL'])`

Syntax 7:

`//*[text()='text value']`

or

`//tagName[text()='text value']`

Here `text()` is the method

Ex: `Next`

Here **Next** is the text value.

`//*[text()='Next']`

Syntax 8:

`//*[starts-with(@attribute,'starting value of attribute')]`

or

`//*[starts-with(text(),'starting value of text value')]`

Ex: <div jsname="YRMmle" class="AxOyFc snByac" aria-hidden="true">Enter your password</div>

Here **AxOy** is the starting value of class attribute, **Enter** is the starting value of text value.

//*[starts-with(@class,'AxOy')]

or

//*[starts-with(text(),'Enter')]

Syntax 9:

//*[contains(@attribute,'substring value of attribute')]

or

//*[contains(text(),'substring value of text value')]

Ex: <div jsname="YRMmle" class="AxOyFc snByac" aria-hidden="true">Enter your password</div>

Here **yFc snB** is the substring value of class attribute, **your pass** is the substring value of text.

//*[contains(@class,'yFc snB')]

or

//*[contains(text(),'your pass')]

Syntax 10:

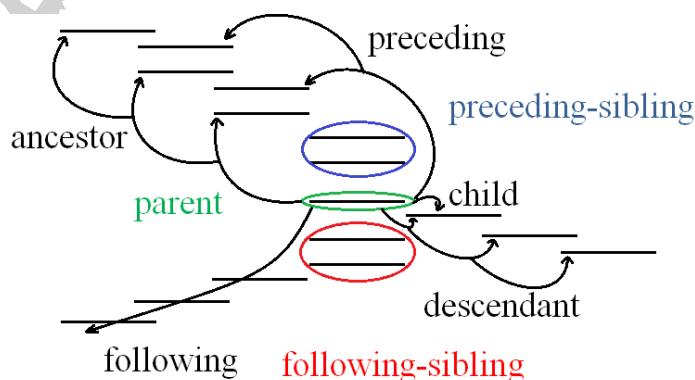
(//*[@attribute='value'])[last()]

This indicates that last element in matched elements.

(//*[@attribute='value'])[last()-1]

This indicates that last but one element in matched elements.

Relations:



Syntax 11:

`//*[@attribute='value']/parent::*`

or

`//*[@attribute='value']/parent::tagName`

Syntax 12:

`//*[@attribute='value']/child::*[index]`

or

`//*[@attribute='value']/child::tagName[index]`

Syntax 13:

`//*[@attribute='value']/ancestor::*[index]`

or

`//*[@attribute='value']/ancestor::tagName[index]`

Syntax 14:

`//*[@attribute='value']/descendant::*[index]`

or

`//*[@attribute='value']/descendant::tagName[index]`

Syntax 15:

`//*[@attribute='value']/preceding::*[index]`

or

`//*[@attribute='value']/preceding::tagName[index]`

Syntax 16:

`//*[@attribute='value']/following::*[index]`

or

`//*[@attribute='value']/following::tagName[index]`

Syntax 17:

`//*[@attribute='value']/preceding-sibling::*[index]`

or

`//*[@attribute='value']/preceding-sibling::tagName[index]`

Syntax 18:

`//*[@attribute='value']/following-sibling::*[index]`

or

`//*[@attribute='value']/following-sibling::tagName[index]`

Note 1:

Absolute xpath can start with / whereas relative xpath can start with //. Here “/” represents relation between parent and child whereas “//” represents locating element anywhere in page.

Note 2:

In general relative xpath syntaxes are formed with symbols, methods and axes.

Symbols	Methods	Axes
//	text()	Parent
*	starts-with()	child
@	contains()	ancestor
=	last()	descendant
,	:	preceding
/	etc.	following
::		preceding-sibling
,		following-sibling
[]		
()		
or		
-		

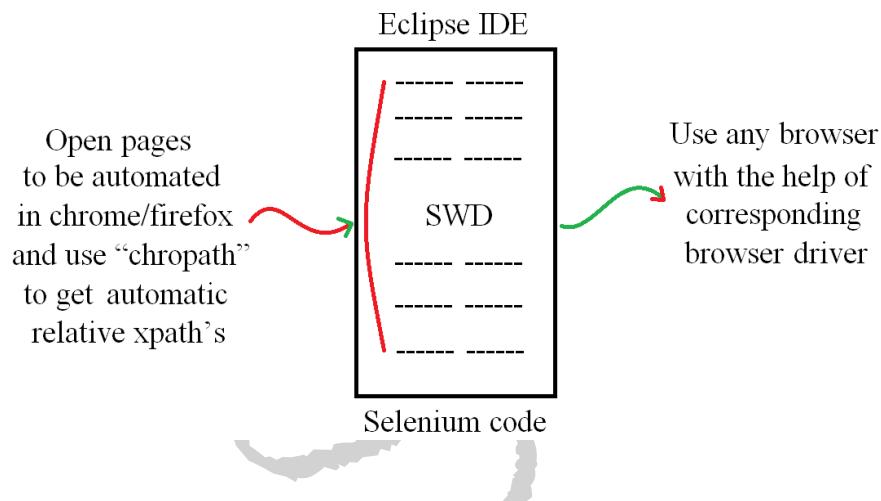
Note 3:

Axes	Meaning
Parent	Select the parent of the current node.
Child	Selects all children of the current node.
Ancestor	Selects all ancestors (parent, grandparent...etc.) of the current node.
Descendant	Selects all descendants (children, grandchildren...etc.) of the current node.
Preceding	Selects all nodes that appear before the current node in the document, except ancestors.

Following	Selects everything in the document after the closing tag of the current node.
preceding-sibling	Selects all the siblings before the current node.
following-sibling	Selects all the siblings after the current node.

Note 4:

While automating any website pages elements, SDET can open that website pages in Chrome/Firefox browser because those browsers can provide inspectors to get proper details of elements including **xpath**.



Browser Name	Inspector Name	Need to install
Chrome	Inspect	In-built with chrome browser
	Chropath	Need to install: (open browser, search for ‘chropath’, click ‘add to chrome’ button).
Firefox	Inspect element	In-built with firefox browser
	Chropath	Need to install: (open browser, search for ‘chropath’, click ‘add to firefox’ button).
IE browser	Developer tools (in tools menu)	In-built with IE browser
	DOM inspector (Document Object Model)	Need to install

Example 2:

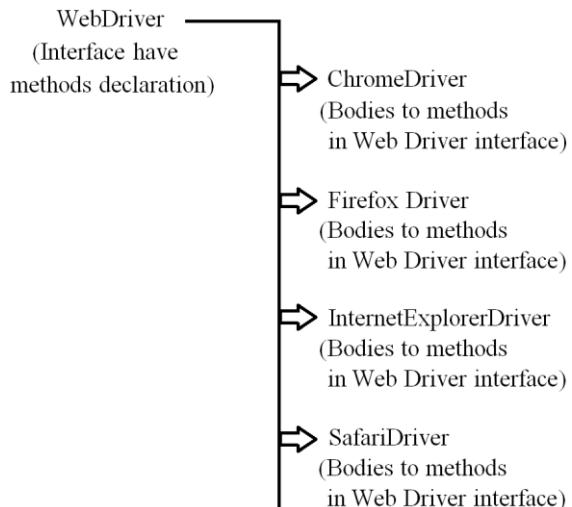
- Open Gmail site.
- Login to the account.
- Click on compose and send mail.
- Do Logout.
- Close the site.

Automate Gmail Compose:

```
//Launch Gmail site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.gmail.com");
Thread.sleep(5000);
//Do login
driver.findElement(By.name("identifier")).sendKeys("xxxx");
driver.findElement(By.xpath("//*[text()='Next']")).click();
Thread.sleep(5000);
driver.findElement(By.name("password")).sendKeys("xxxx");
driver.findElement(By.xpath("//*[text()='Next']")).click();
driver.manage().timeouts().implicitlyWait(50, TimeUnit.SECONDS);
//Click compose
driver.findElement(By.xpath("//div[@class='T-I J-J5-Ji T-I-KE L3']")).click();
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
//Fill details
driver.findElement(By.name("to")).sendKeys("xxxx");
driver.findElement(By.name("subjectbox")).sendKeys("xxxx");
driver.findElement(By.xpath("//*[@aria-label='Message Body'][2]")).sendKeys("xxxx");
driver.findElement(By.xpath("//*[@class='T-I J-J5-Ji a00 T-I-at1 L3']")).click();
Thread.sleep(5000);
//Sign Out
driver.findElement(By.xpath("//*[@class='gb_9a gbii']")).click();
Thread.sleep(5000);
driver.findElement(By.xpath("//*[text()='Sign out']")).click();
Thread.sleep(5000);
//Close site
driver.close();
```

e) Methods in “WebDriver” :

“WebDriver” is an interface in Selenium WebDriver JAR’s. It consists of methods declarations. These methods are getting bodies in “ChromeDriver”, “Firefox”, “InternetExplorerDriver”, “OperaDriver”...etc. classes. These classes are also called as concrete classes.



Program to Interface for all browsers:

```
Scanner sc=new Scanner(System.in);
System.out.println("Enter Browser name:");
String a=sc.nextLine();
WebDriver driver;
if(a.equals("chrome"))
{
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
driver=new ChromeDriver();
driver.manage().window().maximize();
}
else if(a.equals("firefox"))
{
System.setProperty("webdriver.gecko.driver","D:\\DineshReddy\\geckodriver.exe");
driver=new FirefoxDriver();
}
else if(a.equals("ie"))
{
System.setProperty("webdriver.ie.driver","D:\\DineshReddy\\IEDriverServer.exe");
driver=new InternetExplorerDriver();
}
else if(a.equals("edge"))
{
System.setProperty("webdriver.edge.driver","D:\\DineshReddy\\MicrosoftWebDriver.exe");
driver=new EdgeDriver();
}
else
{
System.out.println("Unknown Browser");
System.exit(0);
}
```

From the above example code driver object was declared for WebDriver interface. Depends on browser name we were changed driver object definition with the help of concrete classes with respect to browser.

Here

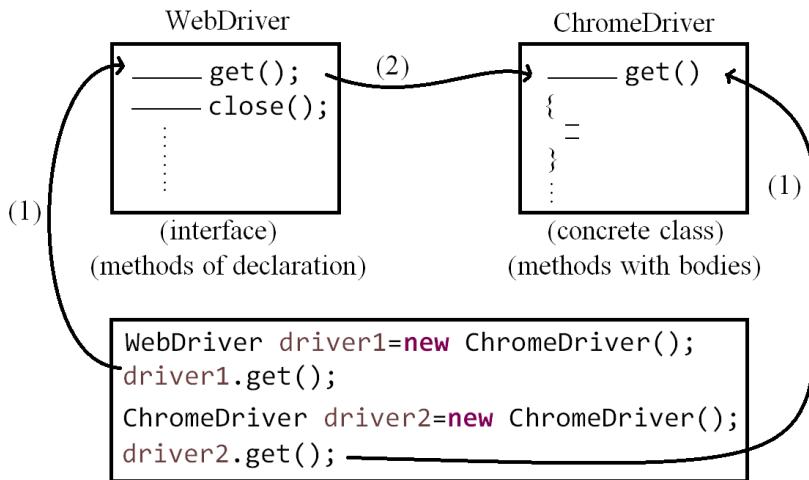
```
WebDriver driver;
```

- 1) 'WebDriver driver;' is an object declaration.
- 2) 'WebDriver' is an interface.

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

Here

- 1) 'driver=new ChromeDriver();' is an object definition.
- 2) 'ChromeDriver' is a concrete class.



From the above explanation and examples, the methods declared in WebDriver interface were defined/implemented in concrete classes like “Chromedriver”, “Firefox”...etc.

These methods are:

get():

We can use this method to launch a site in corresponding browser by giving URL.

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

close():

We can use this method to close active browser window.

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

quit():

We can use this method to close active browser window **along with related browser windows or tabs.**

```
driver.quit();
```

getTitle():

We can use this method to get title of active browser window.

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

getPageSource():

We can use this method to get source code of active browser window page.

```
String x=driver.getPageSource();
```

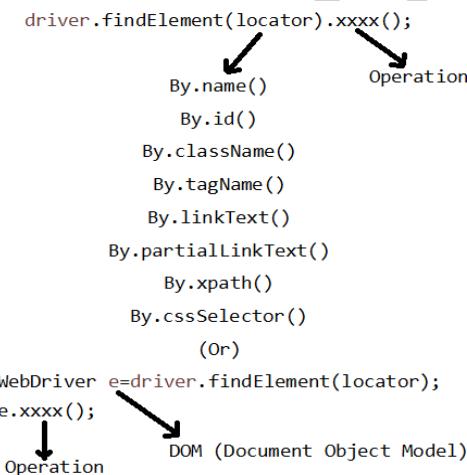
getCurrentUrl():

We can use this method to get URL of current active browser window page.

```
String x=driver.getCurrentUrl();
if(x.contains("https"))
{
System.out.println("Site is Secured");
}
else
{
System.out.println("Site is Not Secured");
}
```

findElement():

We can use this method to locate an element in active browser window page source by using any 1 of 8 locators.



findElement().click():

We can use this element to locate an element and to perform click on that element.

```
driver.findElement(locator).click();
(Or)
WebElement e=driver.findElement(locator);
e.click();
```

findElement().sendKeys():

We can use this method to locate an element and to fill that element with given data.

Syntax 1:

```
driver.findElement(locator).sendKeys("data");
(Or)
WebElement e=driver.findElement(locator);
e.sendKeys();
```

This syntax method can support polymorphism means method with different number of arguments or different types of arguments.

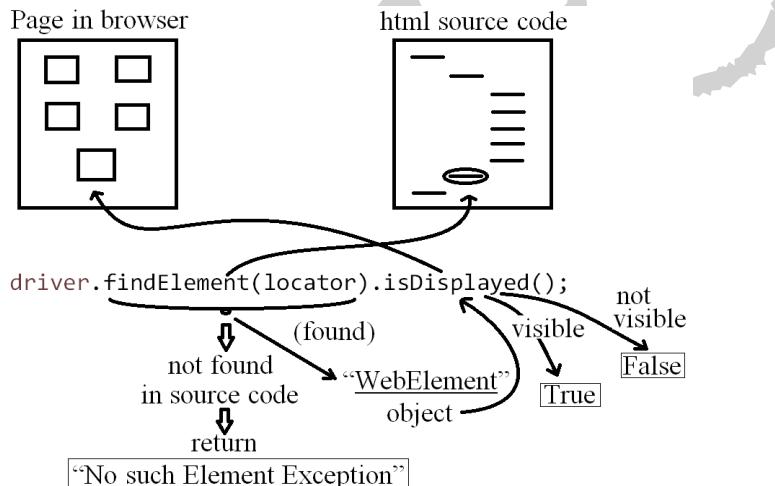
```
driver.findElement(locator).sendKeys("xxxx");
driver.findElement(locator).sendKeys(Keys.TAB);
driver.findElement(locator).sendKeys("xxxx", Keys.TAB);
```

While automating keyboard keys using **sendKeys()** method, we are able to automate combinational Keys like shown below

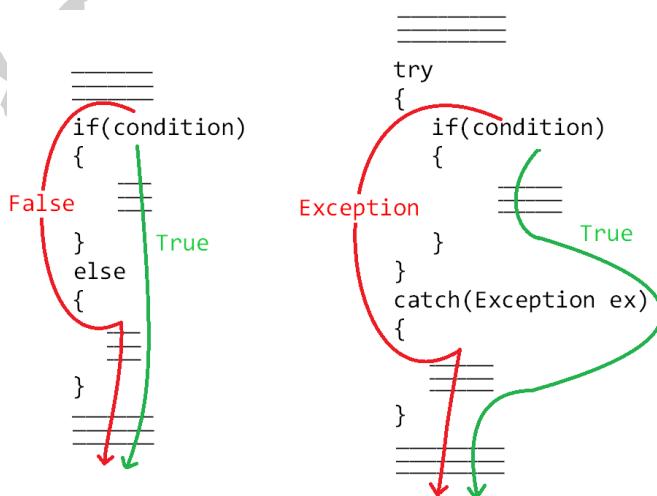
```
driver.findElement(locator).sendKeys(Keys.chord(Keys.CONTROL, "a"));
driver.findElement(locator).sendKeys(Keys.chord(Keys.CONTROL, Keys.SHIFT, Keys.DELETE));
```

findElement().isDisplayed():

We can use this method to check the usability of located element in webpage.



While using above like code in test automation we need to add try & catch block also because conditional statements can return true or false or exceptions.

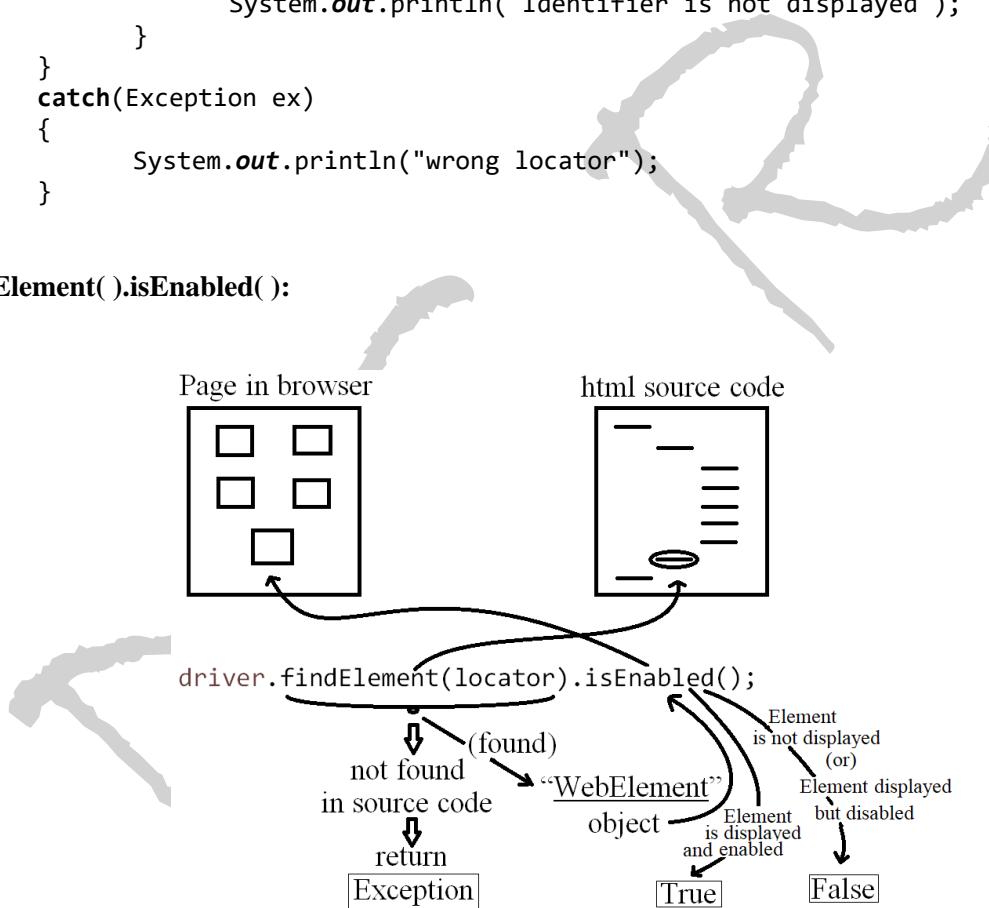


```

System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://gmail.com");
try
{
    if(driver.findElement(By.name("identifier")).isDisplayed())
    {
        System.out.println("Identifier is displayed");
    }
    else
    {
        System.out.println("Identifier is not displayed");
    }
}
catch(Exception ex)
{
    System.out.println("wrong locator");
}

```

findElement().isEnabled():

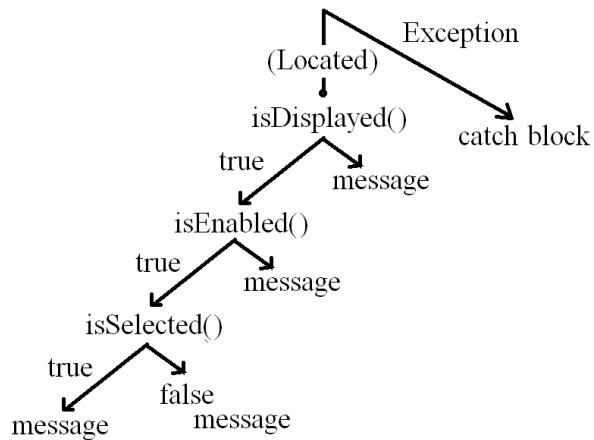


findElement().isSelected():

We can use this method to get the status of elements like radio buttons, checkboxes, dropdowns...etc.

In general “`isDisplayed()`” and “`isEnabled()`” are applicable for any type of elements whereas “`isSelected()`” can be applicable for radio buttons, checkboxes, dropdowns...etc.

Example:



```
System.setProperty("webdriver.chrome.driver","D:\\\\DineshReddy\\\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://makemytrip.woohoo.in/egiftcard");
Thread.sleep(5000);
try
{
    WebElement e=driver.findElement(By.id("dateNow"));
    if(e.isDisplayed())
    {
        System.out.println("Is Displayed");
        if(e.isEnabled())
        {
            System.out.println("Is Enabled");
            if(e.isSelected())
            {
                System.out.println("Is Selected");
            }
            else
            {
                System.out.println("Is Not Selected");
            }
        }
        else
        {
            System.out.println("Is Not Enabled");
        }
    }
    else
    {
        System.out.println("Is Not Displayed");
    }
}
catch(Exception ex)
{
    System.out.println("Wrong Locator");
}
driver.close();
```

findElement().clear():

We can use this method to delete existing data in element.

```
driver.findElement(locator).clear();  
        (Or)  
WebElement e=driver.findElement(locator);  
e.clear();
```

findElement().getAttribute():

We can use this method to get value of an attribute of located element.

```
String x=driver.findElement(locator).getAttribute("attribute name");  
        (Or)  
WebElement e=driver.findElement(locator);  
String x=e.getAttribute("attribute name");
```

findElement().getText():

We can use this method to get text value of an element.

```
<span class="RVeJvd snByac">Forgot email?</span>
```

Here **Forgot email?** is text.

```
String x=driver.findElement(locator).getText();  
        (Or)  
String x=driver.findElement(locator).getAttribute("value");  
Here value is written as value only.
```

findElements().getCssValue():

We can use this method to get style details of an element.

```
String x=driver.findElement(locator).getCssValue("Style Property Name");
```

Here Style Property Name is example like color, font-size, font-family.

In general browser level inspections can provide HTML source of an element and style properties for corresponding element. HTML source of an element is useful to get attribute value and text value. Style properties are useful to get **CssValue**.

```
//Launch Site  
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");  
ChromeDriver driver=new ChromeDriver();  
driver.manage().window().maximize();  
driver.get("http://www.way2sms.com");  
Thread.sleep(5000);  
//Get Attribute Value  
WebElement e1=driver.findElement(By.name("mobileNo"));  
String x=e1.getAttribute("id");
```

```

System.out.println(x);
//Get Text Value and Css Value
WebElement e2=driver.findElement(By.xpath("//*[@class='forgot'][2]"));
String y=e2.getText();
System.out.println(y);
String z=e2.getCssValue("color");
System.out.println(z);
//Close Site
driver.close();

```

switchTo().frame():

We can use this method to change driver object focus from page level to frame level.

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

Here index of frame starts with **0**.

(or)

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

Here frame name is found in HTML source code. Example like **iframe**.

(or)

```
WebElement e= driver.findElement(By.xpath("___"));
```

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

Example 1:

The diagram shows a rectangular box labeled "Page". Inside this box, there is another rectangular box representing a frame. Within the frame, there are three elements: a text input field labeled "User ID" with a blue border, a text input field labeled "Password" with a red border, and a button labeled "OK" at the bottom right.

```

//Switch to Frame
driver.switchTo().frame(0);
driver.findElement(By.name("user id")).sendKeys("xxxx");
//Back to Page
driver.switchTo().defaultContent();
driver.findElement(By.name("password")).sendKeys("xxxx");
driver.findElement(By.name("ok")).click();

```

Example 2:

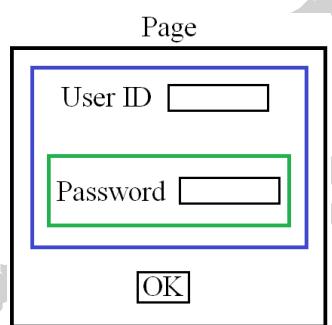
The diagram shows a rectangular box labeled "Page". Inside this box, there is another rectangular box representing a frame. Within the frame, there are three elements: a text input field labeled "User ID" with a blue border, a text input field labeled "Password" with a red border, and a button labeled "OK" at the bottom right.

```

//Switch to Frame 1
driver.switchTo().frame(0);
driver.findElement(By.name("user id")).sendKeys("xxxx");
//Back to Page
driver.switchTo().defaultContent();
//Switch to Frame 2
driver.switchTo().frame(1);
driver.findElement(By.name("password")).sendKeys("xxxx");
driver.switchTo().defaultContent();
//Back to Page
driver.findElement(By.name("ok")).click();

```

Example 3:

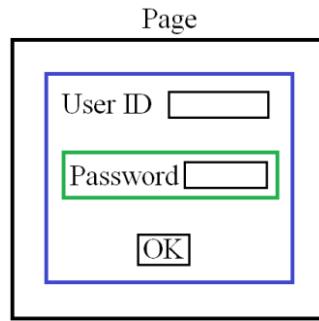


```

//Switch to Frame 1 in page
driver.switchTo().frame(0);
driver.findElement(By.name("user id")).sendKeys("xxxx");
//Switch to Frame in Frame 1
driver.switchTo().frame(0);
driver.findElement(By.name("password")).sendKeys("xxxx");
//Back to Page
driver.switchTo().defaultContent();
driver.findElement(By.name("ok")).click();

```

Example 4:



```

//Switch to Frame 1
driver.switchTo().frame(0);
driver.findElement(By.name("user id")).sendKeys("xxxx");
//Switch to Frame in Frame 1
driver.switchTo().frame(0);
driver.findElement(By.name("password")).sendKeys("xxxx");

```

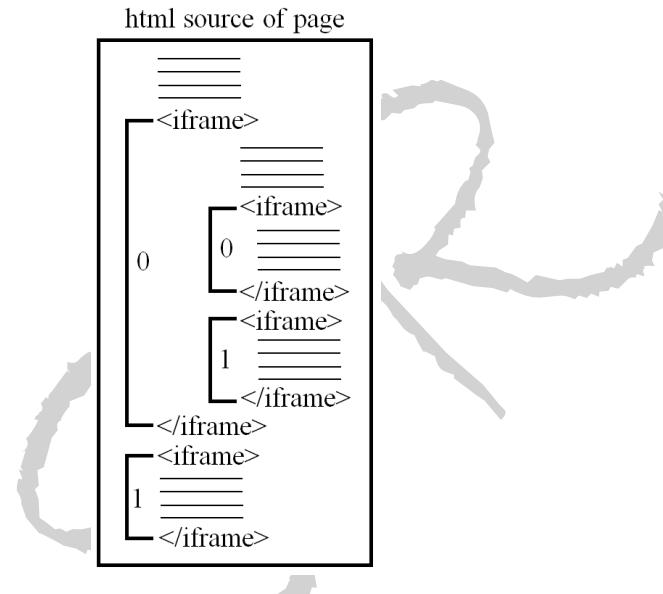
```

//Back to Frame 1 in page
driver.switchTo().parentFrame();
driver.findElement(By.name("ok")).click();
//Back to Page
driver.switchTo().defaultContent();

```

Note:

While using “**switchTo().frame()**”, we can use index of frame as argument. We can follow below way to identify index for required frame.



switchTo().parentFrame():

We can use this method to change focus of driver object from current frame to parent frame of current frame.

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

switchTo().defaultContent():

We can use this method to change focus of driver object from current frame to page.

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

Note 1:

To locate frame, use “name”

↓(no name)

Count and use index

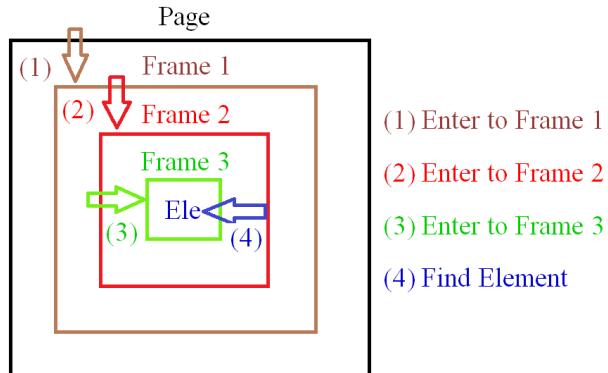
↓(counting is complex)

Locate using xpath

Note 2:

While using `switchTo()` for frames, we need to follow proper hierarchy.

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



- (1) Enter to Frame 1
- (2) Enter to Frame 2
- (3) Enter to Frame 3
- (4) Find Element

Note 3:

To know availability of frames in webpage we need to search “`iframe`” tag in source code.

Example:

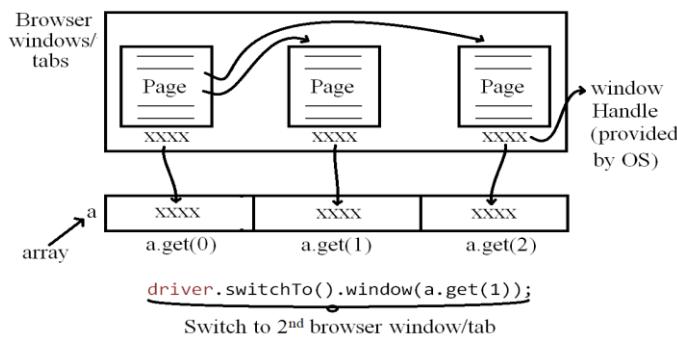
- Launch `jqueryui.com` site
- Switch to frame
- Select a radio button and check box
- Back to page
- Close site

Selection of Radio button and Check box in Frames:

```
//Launch site  
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");  
ChromeDriver driver=new ChromeDriver();  
driver.manage().window().maximize();  
driver.get("https://jqueryui.com/checkboxradio");  
Thread.sleep(5000);  
//Switch to Frame  
driver.switchTo().frame(0);  
//Operate Elements  
driver.findElement(By.xpath("//span[contains(@class,'checkboxradio')])[5]")).click();  
Thread.sleep(5000);  
driver.findElement(By.xpath("//span[contains(@class,'checkboxradio')])[13]")).click();  
Thread.sleep(5000);  
//Back to page  
driver.switchTo().defaultContent();  
//Close site  
driver.close();
```

switchTo.window():

We can use this method to change focus of driver object from current browser window (tab) to other browser window/tab.



Example:

- Launch **sentia.in** site
- Click on pay online button
- Click on transport fee & course fee buttons to get new browser window/tabs
- Switch to each browser window and close

Changing driver object focus to other Window/Tabs:

```
//Launch site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.sentia.in");
Thread.sleep(5000);
driver.findElement(By.xpath("//img[@class='img-responsive'][3]")).click();
Thread.sleep(5000);
driver.findElement(By.xpath("//div[@class='col-md-12'][1]/a")).click();
Thread.sleep(5000);
driver.findElement(By.xpath("//div[@class='col-md-12'][3]/a")).click();
Thread.sleep(5000);
//Get browser windows/tabs handles
ArrayList<String> a=new ArrayList<String>(driver.getWindowHandles());
for(int i=0;i<a.size();i++)
{
    System.out.println(a.get(i));
}
//Switch to 3rd window/tab and close
driver.switchTo().window(a.get(2));
driver.close();
Thread.sleep(5000);
//Switch to 2nd window/tab and close
driver.switchTo().window(a.get(1));
driver.close();
Thread.sleep(5000);
//Switch to 1st window/tab and close
driver.switchTo().window(a.get(0));
driver.close();
```

Note:

“ArrayList” class is related to **JDK**.

switchTo().activeElement():

We can use this method to work with active element in page by default.

```
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.way2sms.com");
Thread.sleep(5000);
driver.switchTo().activeElement().sendKeys("9491947838");
```

switchTo().alert():

We can use this method to handle web alerts related to pages.

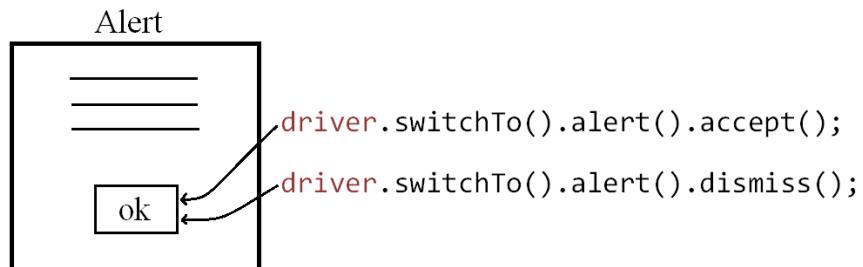
Example 1:

- Go to “w3schools.com”
- Switch to frame
- Click on “try it” button
- Close alert
- Close site

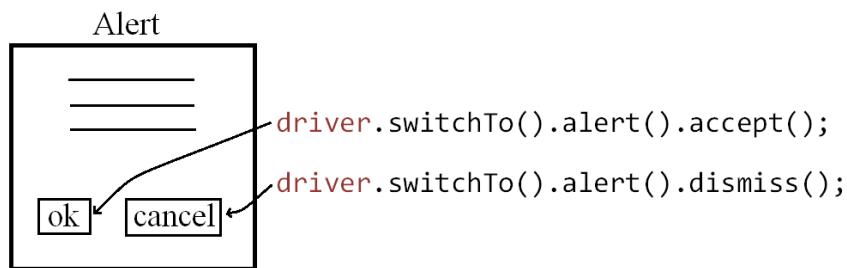
Operating on alerts:

```
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_alert");
Thread.sleep(5000);
WebElement b=driver.findElement(By.name("iframeResult"));
driver.switchTo().frame(b);
driver.findElement(By.xpath("//button[text()='Try it']")).click();
Thread.sleep(5000);
driver.switchTo().alert().dismiss();
Thread.sleep(5000);
driver.close();
```

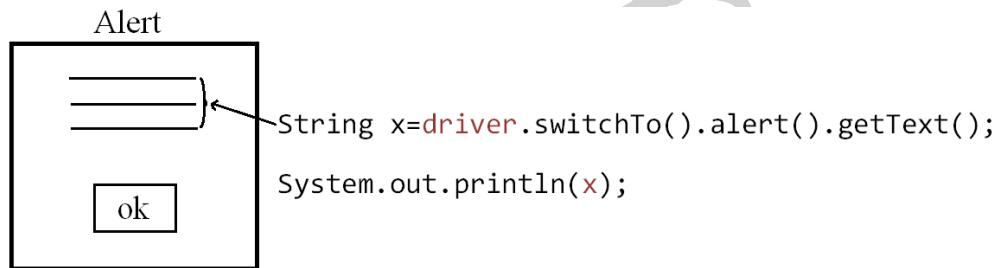
Example 2:



Example 3:



Example 4:



Example 5:



Note 1:

From **w3c** rules, alerts are useful to provide messages to users. We are not able to expect different type of elements in alerts.

Note 2:

Type of Pop-Up	How to automate
Web alerts	Using selenium web driver switchTo().alert()
Banners in page	Using selenium web driver findElement()
Browser notifications	Using selenium web driver " DesiredCapabilities " class
Pop-up windows	Selenium cannot automate; we can use Java Robot/Auto IT.

findElements():

We can use this method to collect multiple similar elements from current page or frame.

```
List<WebElement> l=driver.findElements(locator);
```

Here: `List` is a class in JDK (`java.util`)

`WebElement` is a class in selenium (WebDriver JAR's)

Example 1:

Get count of links in given page.

```
List<WebElement> l=driver.findElements(By.tagName("a"));  
int count=l.size();  
System.out.println(count);
```

Example 2:

Get count of dropdowns in given page.

```
List<WebElement> l=driver.findElements(By.tagName("select"));  
int count=l.size();  
System.out.println(count);
```

Example 3:

Get count of rows in 2nd table in page.

```
List<WebElement> tl=driver.findElements(By.tagName("table"));  
WebElement tb=tl.get(1).findElement(By.tagName("tbody"));  
List<WebElement> rl=tb.findElements(By.tagName("tr"));  
System.out.println(rl.size());  
(or)  
List<WebElement> l=driver.findElements(By.xpath("//table)[2]/tbody/tr"));  
System.out.println(l.size());
```

Example 4:

Get count of columns in 3rd row in 4th table of page.

```
List<WebElement> l=driver.findElements(By.xpath("//table)[4]/tbody/tr[3]/td"));  
System.out.println(l.size());
```

Example 5:

Click on second link in 4th column in 3rd row in 2nd table in page

```
driver.findElement(By.xpath("//table)[2]/tbody/tr[3]/td[4]/a[2]")).click();
```

Example 6:

Get count of images in 2nd column in 3rd row in 4th table in pages.

```
List<WebElement> l=driver.findElements(By.xpath("//table)[4]/tbody/tr[3]/td[2]/img"));  
System.out.println(l.size());
```

Note 1:

While using **findElements()** method we can collect required elements by using **tagName()** or **xpath()** in most of times.

Note 2:

In **html** language we can get different type of tags related to different type of elements.

HTML tag	Purpose
a	Link
b	Bold
img	Image
table	Table
tbody	body of table
tr	row in table
td	column in row
select	Dropdown
iframe	Frames
button	push button
span	plain text
div	division (bootstrap elements)
input	To develop 22 types of elements like: button, checkbox, color, date, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week

Example 7:

Get count of push buttons in a page.

```
List<WebElement> l=driver.findElements(By.xpath("//table)[4]/tbody/tr[3]/td[2]/img"));
System.out.println(l.size());
```

Example 8:

Fill 2nd text box with Abdul Kalam in 4th column in 3rd row in 2nd table in page.

Example 9:

- Launch Gmail site
- Do login
- Count no. of mails in mail box table
- Do logout
- Close site

Counting number mails in mail box:

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.gmail.com");
Thread.sleep(5000);
//Do Login
driver.findElement(By.name("identifier")).sendKeys("****");
driver.findElement(By.xpath("//*[text()='Next']")).click();
Thread.sleep(5000);
driver.findElement(By.name("password")).sendKeys("****");
driver.findElement(By.xpath("//*[text()='Next']")).click();
Thread.sleep(5000);
//Get count of mails in mail-box table
List<WebElement> l=driver.findElements(By.xpath("(//table)[3]/tbody/tr"));
int x=l.size();
System.out.println(x);
List<WebElement> m=l.get(0).findElements(By.tagName("td"));
int y=m.size();
System.out.println(y);
//Do Logout
driver.findElement(By.xpath("//*[contains(@aria-label,'Google Account')]")).click();
Thread.sleep(5000);
driver.findElement(By.LinkText("Sign out")).click();
Thread.sleep(5000);
//Close Site
driver.close();
```

Example 10:

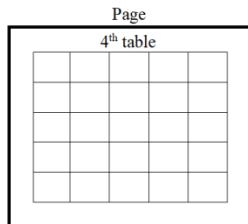
Select India in 2nd dropdown in first frame in page.

```
//Goto first Frame
List<WebElement> fl=driver.findElements(By.tagName("iframe"));
driver.switchTo().frame(fl.get(0));
//Select 2nd Dropdown
List<WebElement> ddl=driver.findElements(By.tagName("select"));
Select s=new Select(ddl.get(1));
s.selectByVisibleText("India");
//Back to Page
driver.switchTo().defaultContent();
```

Note 1:

```
WebElement e=driver.findElement(locator);
List<WebElement> l=driver.findElements(locator);
```

Note 2:



```
List<WebElement> cl=driver.findElements(By.xpath("//table)[4]/tbody/tr[1]/td"));
(or)
List<WebElement> rl=driver.findElements(By.xpath("//table)[4]/tbody/tr"));
List<WebElement> cl=rl.get(0).findElements(By.tagName("td"));
```

getScreenshotAs():

We can use this method to get screenshot of a active browser window page.

```
File f=driver.getScreenshotAs(OutputType.FILE);
File dest=new File("path of file\\name.png");
FileUtils.copyFile(f, dest);
```

Here: `File` is a class in JDK, `OutputType` is a class in SWD Jar's (Static class), `FileUtils` is a class in apache.org.common.io Jar (Download from “java2s.com” site and associate to project in Eclipse IDE).

findElement().getLocation() and findElement().getSize():

We can use these methods to get **x-coordinate**, **y-coordinate**, **width** and **height** of specific element in page.

```
WebElement e=driver.findElement(locator);
int x=e.getLocation().getX();
int y=e.getLocation().getY();
int w=e.getSize().getWidth();
int h=e.getSize().getHeight();
```

Example 1: Get screenshot of an element in page.

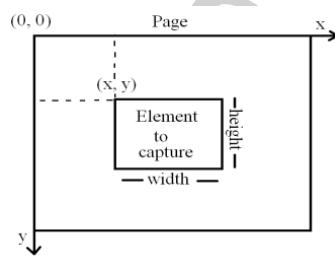
```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
Thread.sleep(5000);
//Find element and get x,y,width and height
WebElement e=driver.findElement(By.xpath("//*[@id='hplogo']/descendant::*[2]"));
```

```

int x=e.getLocation().getX();
int y=e.getLocation().getY();
int w=e.getSize().getWidth();
int h=e.getSize().getHeight();
//Get page Screenshot
File f=driver.getScreenshotAs(OutputType.FILE);
BufferedImage bi=ImageIO.read(f);
BufferedImage ci=bi.getSubimage(x,y,w,h);
ImageIO.write(ci,"png",f);
//Save element Screenshot
File s=new File("D:\\DineshReddy\\Screenshots\\Image1.png");
FileUtils.copyFile(f,s);
//Close Site
driver.close();

```

Example 2:



```

//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.mercurytravels.co.in");
Thread.sleep(5000);
//Find elements and get x,y,width and height
List<WebElement> e=driver.findElements(By.xpath("//input[@type='text']"));
for(int i=0;i<e.size();i++)
{
    if(e.get(i).isDisplayed())
    {
        int x=e.get(i).getLocation().getX();
        int y=e.get(i).getLocation().getY();
        int w=e.get(i).getSize().getWidth();
        int h=e.get(i).getSize().getHeight();
        //Get page Screenshot
        File f=driver.getScreenshotAs(OutputType.FILE);
        BufferedImage bi=ImageIO.read(f);
        BufferedImage ci=bi.getSubimage(x,y,w,h);
        ImageIO.write(ci,"png",f);
        //Save element Screenshot
        File s=new File("D:\\DineshReddy\\Screenshots\\Image"+i+".png");
        FileUtils.copyFile(f,s);
    }
}
//Close Site
driver.close();

```

Note 1:

From the above Script:

JDK	Commons.io	SWD Jars
List (java.util)		
File (java.io)		ChromeDriver
ImageIO	FileUtils	WebElement
BufferedImage		OutputType
System}java.language		By
Thread}java.language		

Note 2:

While getting screenshot for pages and elements, we need to save those screenshots with **.png** (Portable Network Graphics).

In general screenshot is saving with:

- .gif
- .jpg
- .bmp
- .psd (Photoshop Data)...etc.
- .png
- .tif
- .jpeg

manage().window():

We can use this method to maximize browser window, change size of browser window and change position of browser window on desktop.

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.get("https://www.mercurytravels.co.in");
driver.manage().window().maximize();
Thread.sleep(5000);
//Get sizes of window
int w=driver.manage().window().getSize().getWidth();
int h=driver.manage().window().getSize().getHeight();
System.out.println(w+ " "+h);
//Change size of browser window
Dimension d=new Dimension(500,500);
driver.manage().window().setSize(d);
Thread.sleep(5000);
//Get positions of window
int x=driver.manage().window().getPosition().getX();
int y=driver.manage().window().getPosition().getY();
System.out.println(x+ " "+y);
```

```

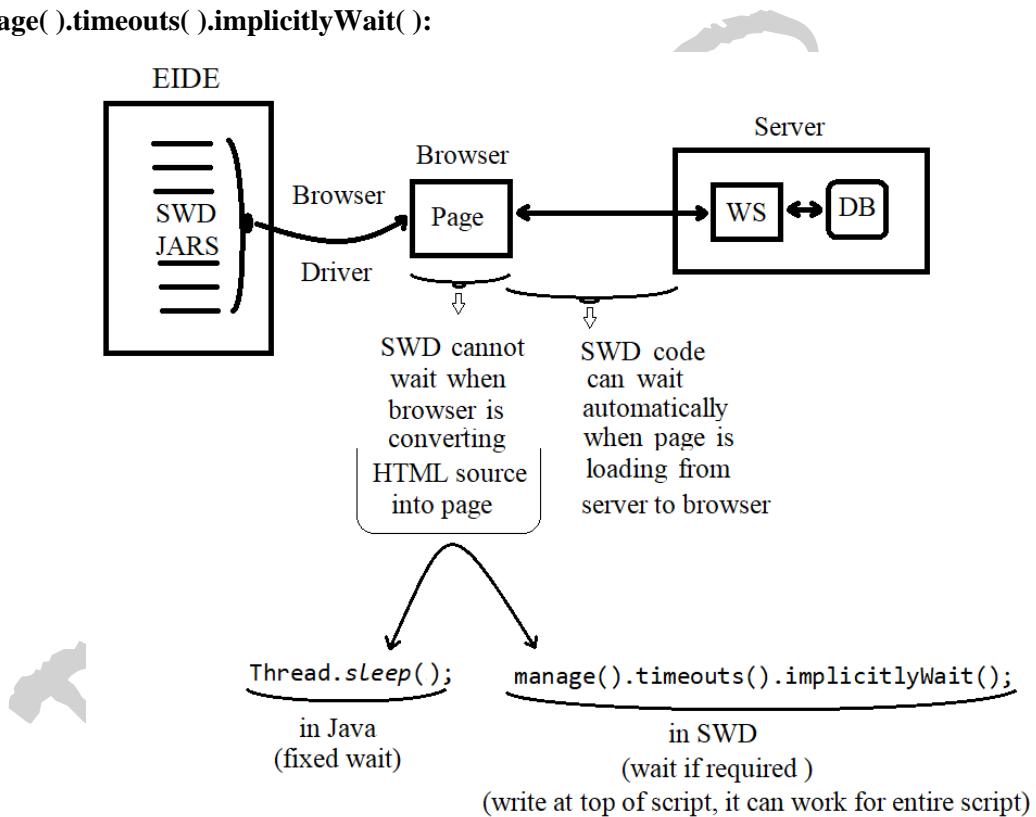
//Change position of browser window
Point p=new Point(873,0);
driver.manage().window().setPosition(p);
Thread.sleep(5000);
//Close site
driver.close();

```

Note:

“Point” & “Dimension” classes are related to selenium web driver Jars.

manage().timeouts().implicitlyWait():



```

driver.manage().timeouts().implicitlyWait(xxxx, TimeUnit.SECONDS);

```

in JDK

manage().deleteAllCookies() & manage().getCookies():

- Cookie is a server side program, it can run in client system (Browser).
- In general cookies are useful to transfer some details regarding client system to server.
- To test cookies availability for a site, we can write below like code in selenium.

```

//Open Browser
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();

```

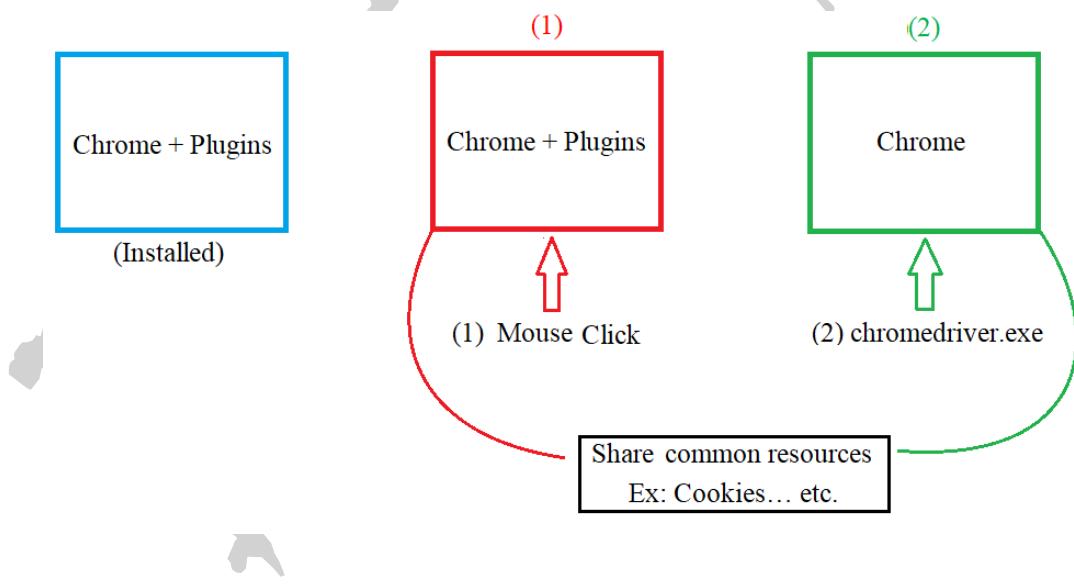
```

driver.manage().window().maximize();
//Delete all existing cookies
driver.manage().deleteAllCookies();
//Launch Site
driver.get("http://www.way2sms.com");
Thread.sleep(5000);
//Cookies Testing
if(driver.manage().getCookies().size()!=0)
{
    System.out.println("Test Passed");
}
else
{
    System.out.println("Test Failed");
}
//Close Site
driver.close();

```

Note 1:

Selenium is session based tool it can work on currently opened browser window by using corresponding browser driver.

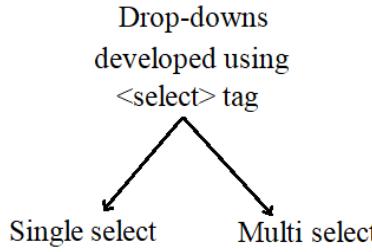


Note 2:

- While automating webpages using selenium we need to work with cookies and cache.
- Cookies and cache are two ways to store data on client's machine, but there are differences between cache and cookies.
- Cookie is used to store information to track different characteristics related to user, while cache is used to make the loading of web pages faster.

f) Methods in “select” class:

“Select” class in SWD Jars is used to automate “dropdowns” developed using “select” tag in HTML.



isMultiple():

We can use this method to check the given dropdown is “single select” or “multi select”.

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_select_multiple");
Thread.sleep(5000);
driver.switchTo().frame("iframeResult");
WebElement e=driver.findElement(By.name("cars"));
Select s=new Select(e);
if(s.isMultiple())
{
    System.out.println("Multi Select");
}
else
{
    System.out.println("Single Select");
}
//Close Site
driver.switchTo().defaultContent();
driver.close();
```

getOptions():

We can use this method to get all items from corresponding dropdown.

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_select_multiple");
Thread.sleep(5000);
driver.switchTo().frame("iframeResult");
WebElement e=driver.findElement(By.name("cars"));
Select s=new Select(e);
List<WebElement> le=s.getOptions();
System.out.println(le.size());
```

```

for(int i=0;i<le.size();i++)
{
    System.out.println(le.get(i).getText());
}
//Close Site
driver.switchTo().defaultContent();
driver.close();

```

To select items in “**multi select**” dropdown we can use below like logic:

```

//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_select_multiple");
Thread.sleep(3000);
driver.switchTo().frame("iframeResult");
WebElement e=driver.findElement(By.name("cars"));
Select s=new Select(e);
//Select more than one item
Actions a=new Actions(driver);
a.keyDown(Keys.CONTROL).click(s.getOptions().get(3))
.click(s.getOptions().get(1)).keyUp(Keys.CONTROL).build().perform();
Thread.sleep(3000);
//Close Site
driver.switchTo().defaultContent();
driver.close();

```

deselectAll():

We can use this method to deselect all selected items in dropdowns.

```

//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_select_multiple");
Thread.sleep(3000);
driver.switchTo().frame("iframeResult");
WebElement e=driver.findElement(By.name("cars"));
Select s=new Select(e);
//Select more than one item
Actions a=new Actions(driver);
a.keyDown(Keys.CONTROL)
.click(s.getOptions().get(3))
.click(s.getOptions().get(1))
.keyUp(Keys.CONTROL).build().perform();
Thread.sleep(3000);
//Deselect all
s.deselectAll();
//Close Site
driver.switchTo().defaultContent();
driver.close();

```

deselectByIndex()
deselectByValue()
deselectByVisibleText():

We can use these methods to deselect selected items in multi select dropdown by giving “**Index**” or “**VisibleText**” or “**Value**”.

We can get value of an item from source code.

A	C
A 0	▼
B 1	
C 2	
D 3	

```
s.deselectByIndex(2); //deselect C  
s.deselectByVisibleText("C"); //deselect C  
s.deselectByValue("xxxx"); //deselect C
```

Here: xxxx is value for “C” in source code.

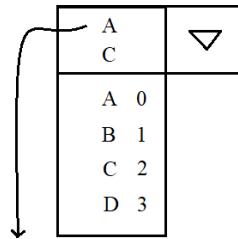
getAllSelectedOptions():

We can use this method to get all selected items from multi selected dropdown.

```
//Launch Site  
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");  
ChromeDriver driver=new ChromeDriver();  
driver.manage().window().maximize();  
driver.get("https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_select_multiple");  
Thread.sleep(3000);  
driver.switchTo().frame("iframeResult");  
WebElement e=driver.findElement(By.name("cars"));  
Select s=new Select(e);  
//Select more than one item  
Actions a=new Actions(driver);  
a.keyDown(Keys.CONTROL).click(s.getOptions().get(3))  
.click(s.getOptions().get(1)).keyUp(Keys.CONTROL).build().perform();  
Thread.sleep(3000);  
//Get all selected options  
List<WebElement> l=s.getAllSelectedOptions();  
System.out.println(l.size());  
for(int i=0;i<l.size();i++)  
{  
    System.out.println(l.get(i).getText());  
}  
//Close Site  
driver.switchTo().defaultContent();  
driver.close();
```

getFirstSelectedOption():

We can use this method to get first item in selected list of items in dropdown.



```
String x=s.getFirstSelectedOption().getText();
System.out.println("First Selected Option is "+x);
```

selectByValue():

selectByIndex():

selectByVisibleText():

We can use these methods to select an item in single select dropdown.

```
s.deselectByIndex(5);
//index of 6th item in single select dropdown
s.deselectByVisibleText("6Nights+7Days");
//visible text of an item in single select
s.selectByValue("39Nights/40Days");
//value of an item with respect to source code.
```

Note 1:

Select class methods can work for <select> tag based dropdown only.

Note 2:

isMultiple()
getOptions()

Single select/ Multi select

deselectAll()
deselectByValue()
deselectByIndex()
deselectByVisibleText()
getAllSelectedOptions()
getFirstSelectedOptions()

Multi-select

selectByValue()
selectByIndex()
selectByVisibleText()

Single-select

g) “Actions” class:

To automate some special situations / scenarios in webpages we can use “Actions” class methods.

✓ Situation 1: Click on an element

```
driver.findElement(locator).click();  
(or)
```

```
Actions a=new Actions(driver);  
WebElement e=driver.findElement(locator);  
a.click(e).build().perform();
```

✓ Situation 2: Fill an element

```
driver.findElement(locator).sendKeys("xxxx");  
(or)
```

```
Actions a=new Actions(driver);  
WebElement e=driver.findElement(locator);  
a.sendKeys(e,"xxxx").build().perform();
```

✓ Situation 3: Move mouse pointer to an element

```
//Launch Site  
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");  
ChromeDriver driver=new ChromeDriver();  
driver.manage().window().maximize();  
driver.get("https://www.flipkart.com");  
Thread.sleep(3000);  
//Close login banner  
driver.findElement(By.xpath("//*[@class=' _2AkmmA _29YdH8 ']")).click();  
Thread.sleep(3000);  
//Move mouse pointer to an element  
WebElement  
e=driver.findElement(By.xpath("//*[@class=' _2f5Jjv ']/descendant::span[1]"));  
Actions a=new Actions(driver);  
a.moveToElement(e).build().perform();  
Thread.sleep(5000);  
//Close site  
driver.close();
```

(or)

```
WebElement  
e=driver.findElement(By.xpath("//*[@class=' _2f5Jjv ']/descendant::span[1]"));  
Actions a=new Actions(driver);  
int x=e.getLocation().getX();  
int y=e.getLocation().getY();  
a.moveByOffset(x,y).build().perform();
```

✓ **Situation 4: Double click on an element**

```
Actions a=new Actions(driver);
WebElement e=driver.findElement(locator);
a.click(e).build().perform();
```

✓ **Situation 5: Right click on an element**

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.seleniumhq.org");
Thread.sleep(3000);
//Right Click on Download
WebElement e=driver.findElement(By.LinkText("Download"));
Actions a=new Actions(driver);
a.contextClick(e).build().perform();
//Close Site
driver.close();
```

✓ **Situation 6: Drag and Drop**

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://jqueryui.com/droppable");
Thread.sleep(3000);
//Drag and Drop
driver.switchTo().frame(0);
WebElement e1=driver.findElement(By.id("draggable"));
WebElement e2=driver.findElement(By.id("droppable"));
Actions a=new Actions(driver);
a.dragAndDrop(e1,e2).build().perform();
Thread.sleep(3000);
//Close site
driver.switchTo().defaultContent();
driver.close();
```

✓ **Situation 7: Automate Sliders**

In general, webpages can provide two types of sliders such as **Horizontal Slider** and **Vertical Slider**.

Example:

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://jqueryui.com/slider/");
Thread.sleep(3000);
```

```

//Horizontal Slider
driver.switchTo().frame(0);
WebElement e=driver.findElement(By.xpath("//div[@id='slider']/span"));
Actions a=new Actions(driver);
a.dragAndDropBy(e,300,0).build().perform();
Thread.sleep(3000);
a.dragAndDropBy(e,-200,0).build().perform();
Thread.sleep(3000);
//Vertical Slider
driver.switchTo().defaultContent();
driver.findElement(By.linkText("Vertical slider")).click();
Thread.sleep(3000);
driver.switchTo().frame(0);
WebElement e1=driver.findElement(By.xpath("//div[@id='slider-vertical']/descendant::span"));
a.dragAndDropBy(e1,0,100).build().perform();
Thread.sleep(3000);
a.dragAndDropBy(e1,0,-200).build().perform();
Thread.sleep(3000);
//Close site
driver.switchTo().defaultContent();
driver.close();

```

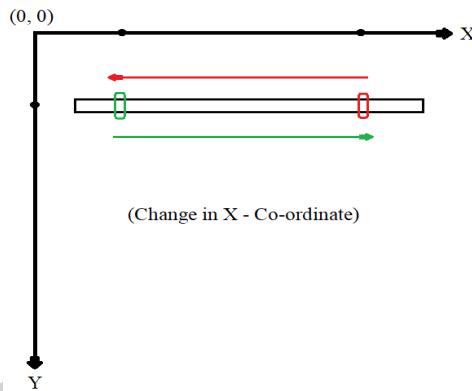


Fig: Horizontal Sliding

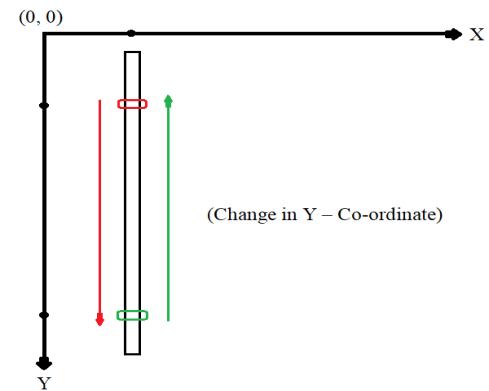
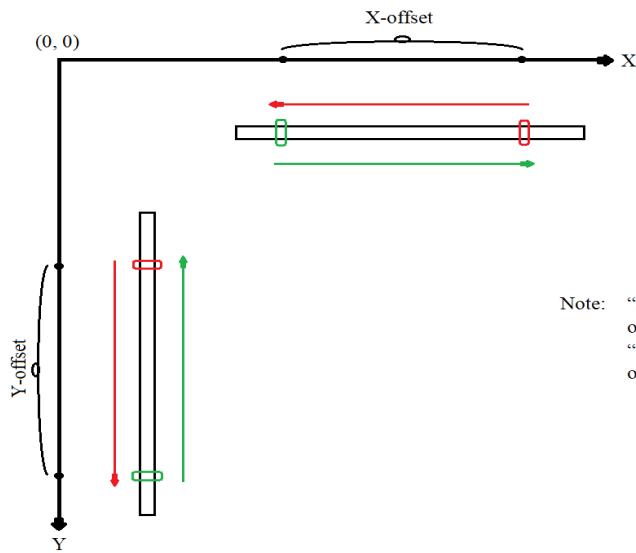


Fig: Vertical Slider



Note: ‘X-offset’ means change only on X-axis (or) Co-ordinate.
 ‘Y-offset’ means change only on Y-axis (or) Co-ordinate.

✓ **Situation 8: Automate Cache (Auto Complete)**

Example 1: Select 4th item in cache

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
Thread.sleep(5000);
//Fill text box with data to get cache/auto complete
driver.findElement(By.name("q")).sendKeys("Steve Jobs");
Thread.sleep(5000);
//Select 4th item in cache
Actions a=new Actions(driver);
for (int i=1;i<=4;i++)
{
    a.sendKeys(Keys.DOWN).build().perform();
    Thread.sleep(1000);
}
a.sendKeys(Keys.ENTER).build().perform();
//Close Site
driver.close();
```

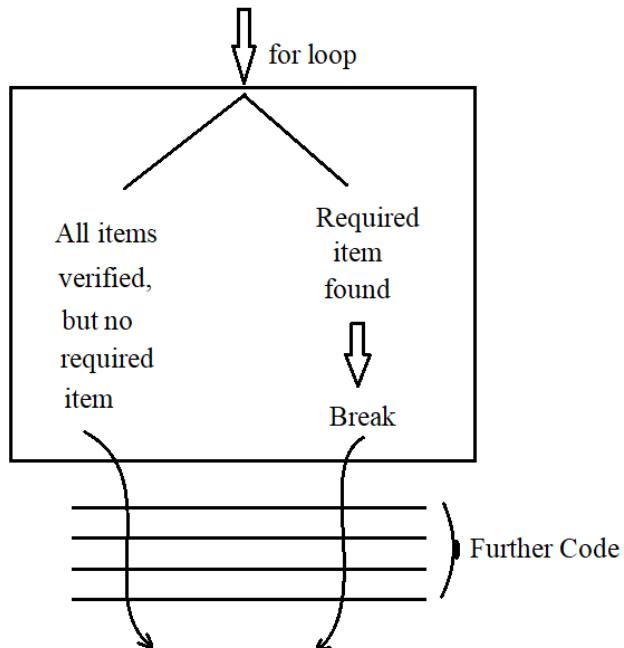
Example 2: Select required item in cache

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
Thread.sleep(5000);
//Fill text box with data to get cache/auto complete
driver.findElement(By.name("q")).sendKeys("Steve Jobs");
Thread.sleep(5000);
//Get count items in cache
List<WebElement> l=driver.findElements(By.xpath("//*[@role='listbox']/li"));
int cs=l.size()-2;//Last 2 items are buttons
//Search for required item in cache
Actions a=new Actions(driver);
int flag=0;
for (int i=1;i<=cs;i++)
{
    a.sendKeys(Keys.DOWN).build().perform();
    Thread.sleep(1000);
    String x=driver.findElement(By.name("q")).getAttribute("value");
    if(x.equalsIgnoreCase("Steve jobs book"))
    {
        a.sendKeys(Keys.ENTER).build().perform();
        flag=1;
        break;
    }
}
```

```

if(flag==0)
{
    System.out.println("No item Matched");
}
else
{
    System.out.println("Item found and selected");
}
//Close Site
driver.close();

```



✓ Situation 9: Handling tooltips

In webpages tooltips are two types such as

- 1) Static tooltip
- 2) Dynamic tooltip

In general developers can maintain tooltips in two ways in pages.

Example 1:

```

<a class="w3-bar-item w3-button" href="/js/default.asp"
title="JavaScript Tutorial">JAVASCRIPT</a>

```

↓ ↓
Attribute for tooltip Tooltip Message

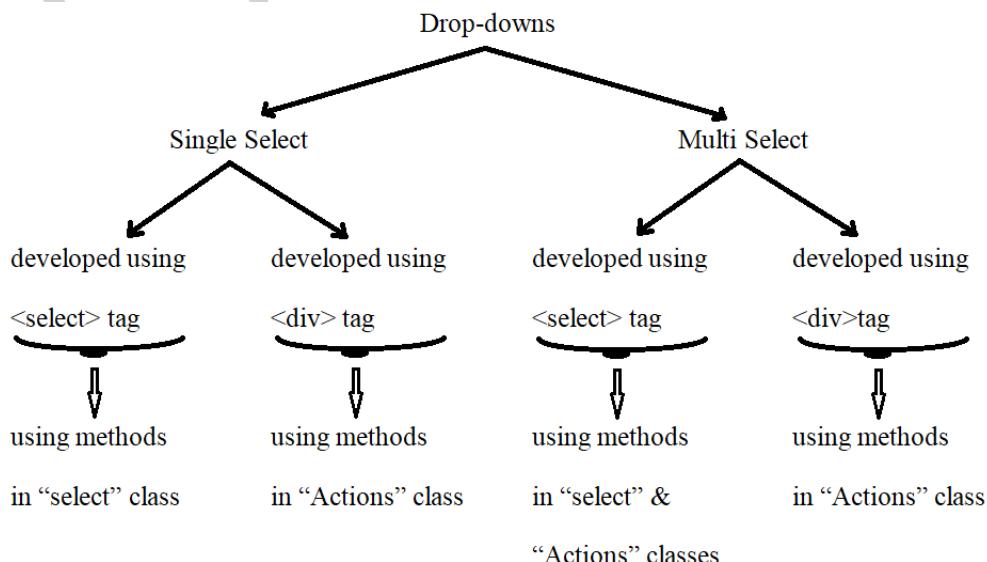
Example 2:

```
<div class="tooltip">Top  
<span class="tooltiptext tooltip-top">Tooltip text</span></div>  
↓  
Sub/child for tooltip element  
↓  
Tooltip message
```

Practical Example:

```
//Launch Site  
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");  
ChromeDriver driver=new ChromeDriver();  
driver.manage().window().maximize();  
driver.get("https://www.w3schools.com/css/css_tooltip.asp");  
Thread.sleep(5000);  
//Get tooltip via title attribute  
WebElement e1=driver.findElement(By.linkText("JAVASCRIPT"));  
String x=e1.getAttribute("title");  
System.out.println(x);  
Thread.sleep(5000);  
//Move mouse pointer to an element for tooltip  
WebElement e2=driver.findElement(By.xpath("//*[@class='tooltip'][1]"));  
Actions a=new Actions(driver);  
a.moveToElement(e2).clickAndHold().build().perform();  
Thread.sleep(5000);  
WebElement e3=driver.findElement(By.xpath("//*[@class='tooltip'][1]/span"));  
String y=e3.getText();  
System.out.println(y);  
a.release().build().perform();  
//Close Site  
driver.close();
```

✓ Situation 10: Automate <div> tag (Bootstrap) dropdowns



Example 1: Automate single select drop-down developed using <div> tag

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://semantic-ui.com/modules/dropdown.html");
Thread.sleep(5000);
//Open drop-down (developed using <div> tag)
WebElement e=driver.findElement(By.xpath("//*[contains(@class,'ui selection dropdown')])[1]"));
Actions a=new Actions(driver);
a.click(e).build().perform();
Thread.sleep(5000);
//Get items and display
List<WebElement> l=driver.findElements(By.xpath("//*[contains(@class,'ui selection dropdown')])[1]/child::*/div"));
for (int i=0;i<l.size();i++)
{
    System.out.println(l.get(i).getText());
}
//Select 2nd item
a.click(l.get(1)).build().perform();
Thread.sleep(5000);
//Close Site
driver.close();
```

Example 2: Automate Multi select drop-down developed using <div> tag

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://semantic-ui.com/modules/dropdown.html");
Thread.sleep(5000);
//Open Drop-Down (Developed using <div> tag)
WebElement e=driver.findElement(By.xpath("//div[contains(@class,'ui fluid dropdown selection multiple'))"));
driver.executeScript("arguments[0].scrollIntoView()",e);
Actions a=new Actions(driver);
a.click(e).build().perform();
//Get and Display items
List<WebElement> le=driver.findElements(By.xpath("//div[contains(@class,'ui fluid dropdown selection multiple'))/div[2]/div"));
System.out.println("Number of items is "+le.size());
for (int i=0;i<le.size();i++)
{
    System.out.println(le.get(i).getText());
}
//Select Multiple items(1,7,11,15,18)
le.get(0).click();
Thread.sleep(3000);
le.get(6).click();
Thread.sleep(3000);
le.get(10).click();
```

```

Thread.sleep(3000);
le.get(14).click();
Thread.sleep(3000);
le.get(17).click();
Thread.sleep(3000);
//Get selected items
List<WebElement> il=driver.findElements(By.xpath("//div[contains(@class,'ui fluid dropdown selection multiple')]/a"));
System.out.println("Number of items Selected is "+il.size());
for (int i=0;i<il.size();i++)
{
    System.out.println(il.get(i).getText());
}
//Deselect 2nd item in Selected items
driver.findElement(By.xpath("//div[contains(@class,'ui fluid dropdown selection multiple')]/a[2]/i")).click();
Thread.sleep(3000);
//Close Site
driver.close();

```

Note 1:

From the above situations “Actions” class methods useful to automate some situations in webpages

→ click()	→ contextClick()	→ keyUp()
→ sendKeys()	→ dragAndDrop()	→ build()
→ moveToElement()	→ clickAndHold()	→ perform()
→ doubleClick()	→ release()	→ tick()
→ moveByOffset()	→ keyDown()	→ pause()

tick(): This method is similar to click.

pause(): This method is similar to Thread.sleep();

Example 1:

```

Actions a=new Actions(driver);
a.pause(xxxx).build().perform();

```

Here: “xxxx” is in milliseconds.

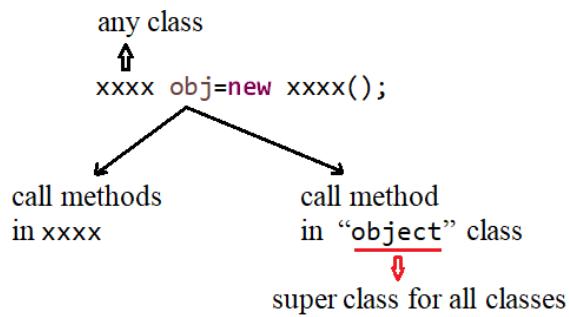
Example 2:

```

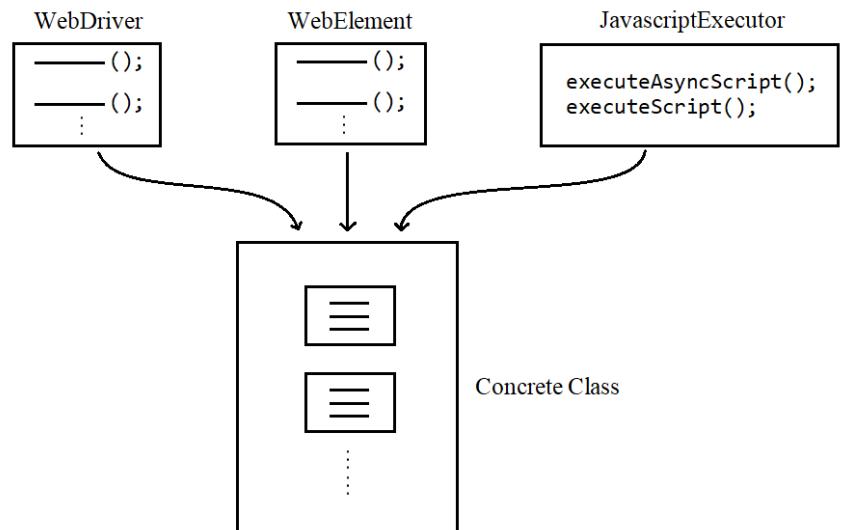
Duration d=Duration.of(10, ChronoUnit.SECONDS);
Actions a=new Actions(driver);
a.pause(d).build().perform();

```

Note 2:



h) “JavascriptExecutor” class (Interface):



ChromeDriver/FirefoxDriver/InternetExplorerDriver/OperaDriver...

From the above diagram **JavascriptExecutor** is an interface in selenium WebDriver Jars. It has 2 methods with declarations. Bodies to those methods are available in concrete classes like ChromeDriver, FirefoxDriver... etc. in SWD Jars. These methods declared in “JavascriptExecutor” and defined in concrete classes are useful to automate some situations in webpages.

✓ **Situation 1: Click on an element**

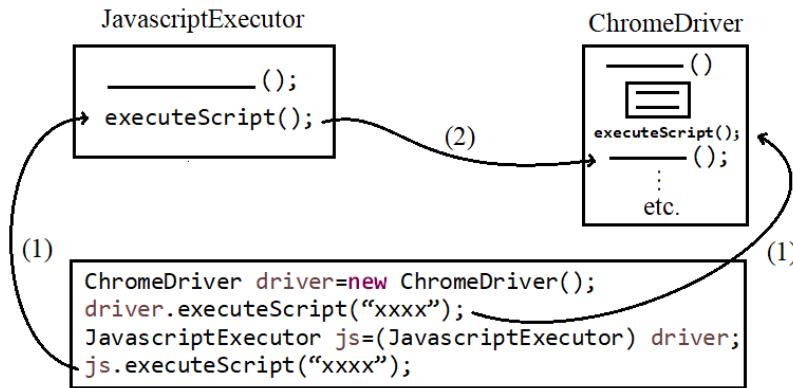
```
JavascriptExecutor js=(JavascriptExecutor) driver;
js.executeScript("document.getElementById('xxxx').click();");
```

Here: `document.getElementById('xxxx').click();` is Java script.

(or)

```
JavascriptExecutor js=(JavascriptExecutor) driver;
WebElement e=driver.findElement(Locator);
js.executeScript("arguments[0].click();",e);
```

Here: `arguments[0].click();` is Java script.



✓ **Situation 2: Fill an element with data**

```
JavascriptExecutor js=(JavascriptExecutor) driver;
js.executeScript("document.getElementById('xxxx').value='xxxx';");
```

Here: `document.getElementById('xxxx').value='xxxx';` is Java script.

(or)

```
JavascriptExecutor js=(JavascriptExecutor) driver;
WebElement e=driver.findElement(Locator);
js.executeScript("arguments[0].value='xxxx';",e);
```

Here: `arguments[0].value='xxxx';` is Java script.

✓ **Situation 3: Highlight an element in webpage**

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.google.co.in");
Thread.sleep(5000);
//Highlight an element
JavascriptExecutor js=(JavascriptExecutor) driver;
WebElement e=driver.findElement(By.xpath("//div[@class='content']/descendant::*[4]"));
js.executeScript("arguments[0].style.border='5px brown solid';",e);
Thread.sleep(5000);
//Close Site
driver.close();
```

Here: 5px indicates border size in pixels.

brown indicates name of the color in correct spelling.

solid indicates style of the border.

(or)

```
JavascriptExecutor js=(JavascriptExecutor) driver;
js.executeScript("document.getElementById('xxxx').style.border='5px brown solid';");
```

Style of the border examples:

dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden.

✓ **Situation 4: Disable an element**

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.google.co.in");
Thread.sleep(5000);
//Disable an element
WebElement e=driver.findElement(By.name("q"));
JavascriptExecutor js=(JavascriptExecutor) driver;
js.executeScript("arguments[0].setAttribute('disabled','true')",e);
Thread.sleep(5000);
js.executeScript("arguments[0].removeAttribute('disabled')",e);
Thread.sleep(5000);
//Close Site
driver.close();
```

✓ **Situation 5: Page Scrolling**

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.seleniumhq.org");
Thread.sleep(5000);
//Scroll to Bottom
driver.executeScript("window.scrollTo(0,document.body.scrollHeight);");
Thread.sleep(5000);
//Scroll to Top
driver.executeScript("window.scrollTo(document.body.scrollHeight,0);");
Thread.sleep(5000);
//Scroll to specific Element
WebElement e=driver.findElement(By.xpath("//div[@class='selenium-sponsors']"));
driver.executeScript("arguments[0].scrollIntoView()",e);
Thread.sleep(5000);
//Close Site
driver.close();
```

✓ **Situation 6: Creating alert**

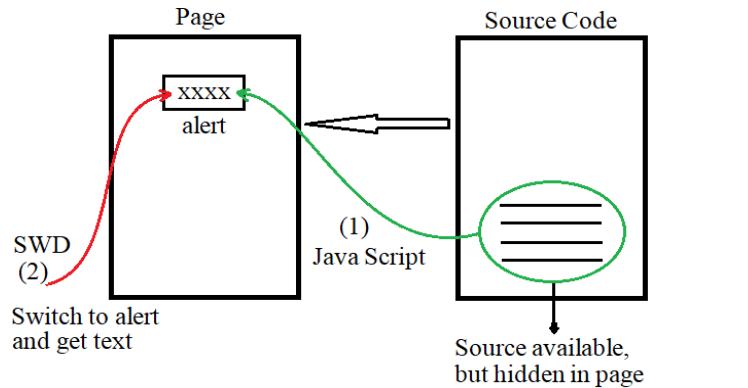
```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://bhugol1.weebly.com/about.html");
Thread.sleep(5000);
//Create an Alert
driver.executeScript("alert('This website is created by the student of Dr.a.p.j.Abdul Kalam Sir \\nThanks for visiting');");
Thread.sleep(5000);
//Get text of an Alert
String x=driver.switchTo().alert().getText();
System.out.println(x);
```

```

driver.switchTo().alert().dismiss();
driver.close();

```

✓ Situation 7: Get text or value of invisible elements



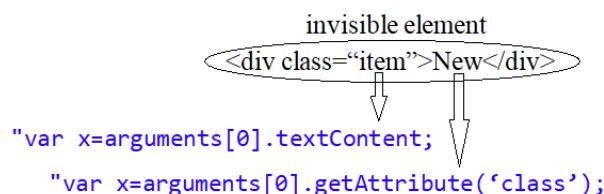
```

//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://semantic-ui.com/modules/dropdown.html");
Thread.sleep(5000);
//Scroll to Selected DropDown
WebElement e=driver.findElement(By.xpath("//div[@class='ui fluid dropdown selection multiple']"));
driver.executeScript("arguments[0].scrollIntoView()",e);
//Get items(invisible) of a dropdown without open
List<WebElement> le=driver.findElements(By.xpath("//div[@class='ui fluid dropdown selection multiple']/select/option"));
System.out.println("No: of items in DropDown is "+le.size());
for (int i=0;i<le.size();i++)
{
    driver.executeScript("var x=arguments[0].textContent; alert(x)",le.get(i));
    String y=driver.switchTo().alert().getText();
    System.out.println(y);
    Thread.sleep(2000);
    driver.switchTo().alert().dismiss();
}
driver.close();

```

Note 1:

While working with invisible elements java script is useful to get text or attribute values of those elements.



Note 2:

Java script can allow as locating and operating element, but java script can support 4 ways to locate element whereas selenium can support 9 ways to locate.

Java Script	Selenium
	By.name()
	By.id()
getElementById()	By.className()
getElementsByName()[index]	By.tagName()
getElementsByTagName()[index]	By.linkText()
getElementsByClassName()[index]	By.partialLinkText()
	By.xpath
	By.cssSelector()
	DOM

OOPL	OBPL
<ul style="list-style-type: none"> ✓ Java → Classes (properties & Methods) <ul style="list-style-type: none"> Instance classes create object & call properties & methods 	<ul style="list-style-type: none"> ✓ Java Script → Having built in objects. Ex: document arguments[0] window → default object : etc.

i) “WebDriverWait” & “FluentWait” classes in SWD:

✓ implicitlyWait:

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
✓ It can work for findElements() & findElements() only.
```

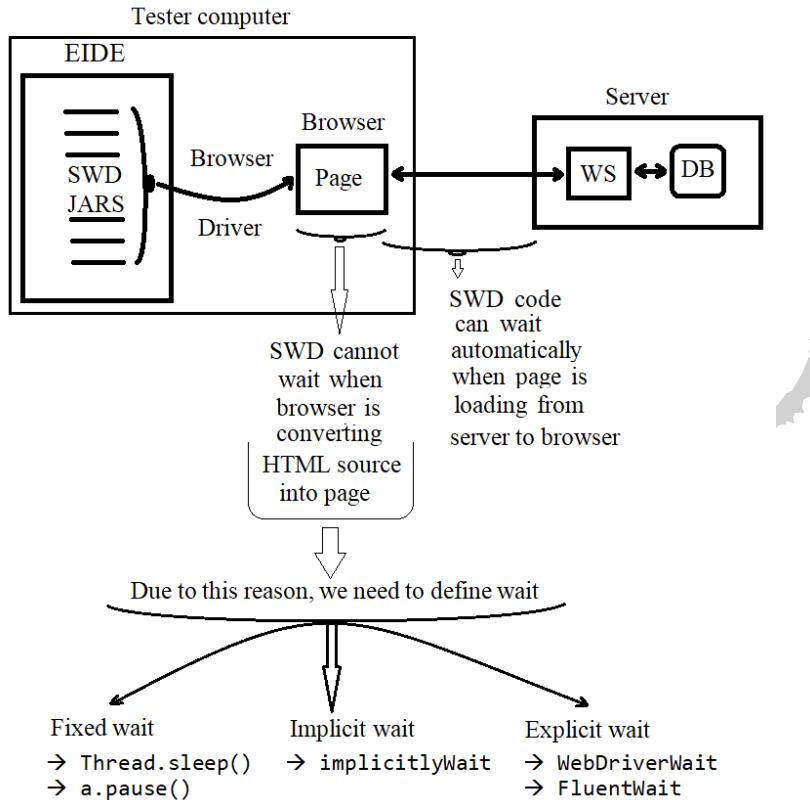
✓ WebDriverWait:

```
WebDriverWait w=new WebDriverWait(driver,10);
✓ It checks condition continuously.
```

✓ **FluentWait:**

```
FluentWait f=new FluentWait(driver).pollingEvery(2, TimeUnit.SECONDS)
.withTimeout(10, TimeUnit.SECONDS);
```

- ✓ It checks condition interval by interval.



Example 1:

```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://demos.telerik.com/aspnet-
ajax/ajaxloadingpanel/functionality/explicit-show-hide/defaultcs.aspx");
//Wait for page ready
WebDriverWait w=new WebDriverWait(driver,10);
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[@class='
optanon-alert-box-wrapper ']")));
//Click on accept cookies
driver.findElement(By.xpath("//a[@class='optanon-allow-all']")).click();
//Wait for visible & Click next 'next' icon for getting next month
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[@class=
'rcTitle']")));
driver.findElement(By.xpath("//*[@class='t-button rcNext']/span")).click();
//Wait for load completion
w.until(ExpectedConditions.invisibilityOfElementLocated(By.xpath("//*[@class='
raDiv']")));
```

```

//Wait for visibility of Date
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[@class='rcMain']")));
//Click on date and wait for load completion
driver.findElement(By.linkText("19")).click();
w.until(ExpectedConditions.invisibilityOfElementLocated(By.xpath("//*[@class='raDiv']")));
//Close site
driver.close();

```

Example 2:

Replace **WebDriverWait** method with

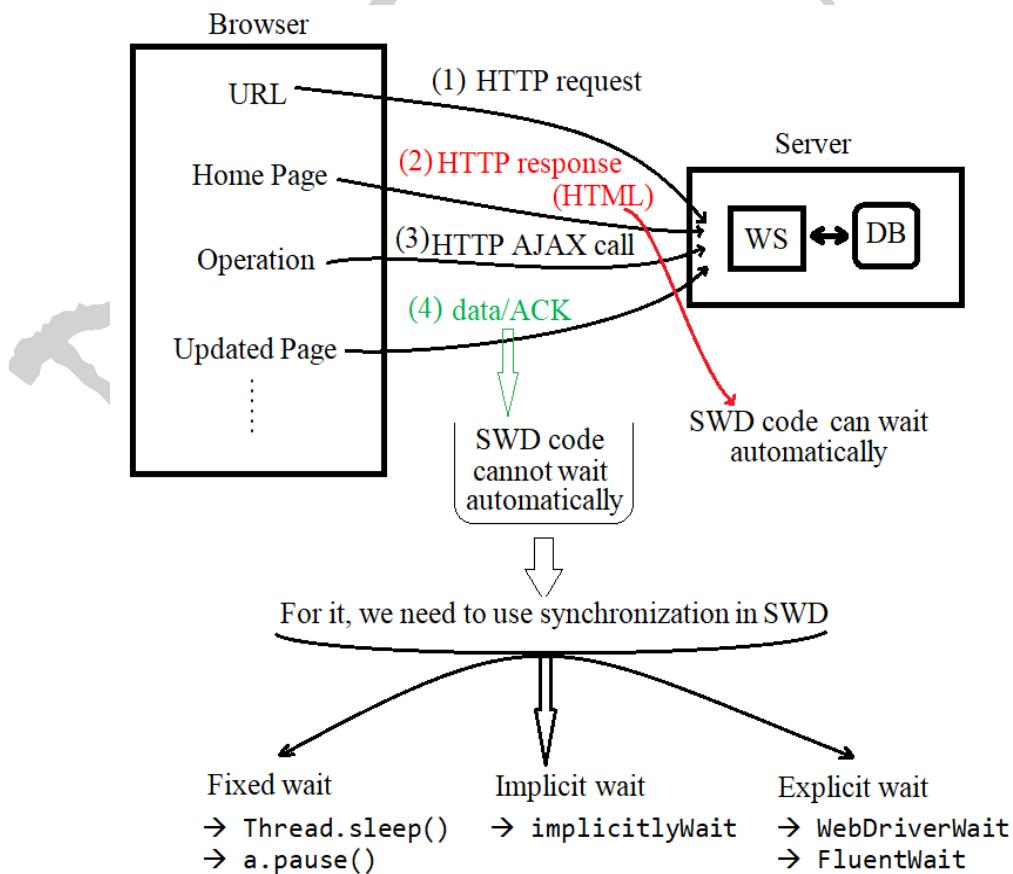
```

FluentWait f=new FluentWait(driver).pollingEvery(2,TimeUnit.SECONDS)
                           .withTimeout(10,TimeUnit.SECONDS);
f.until(ExpectedConditions.visibilityOfElementLocated(By.Locator));

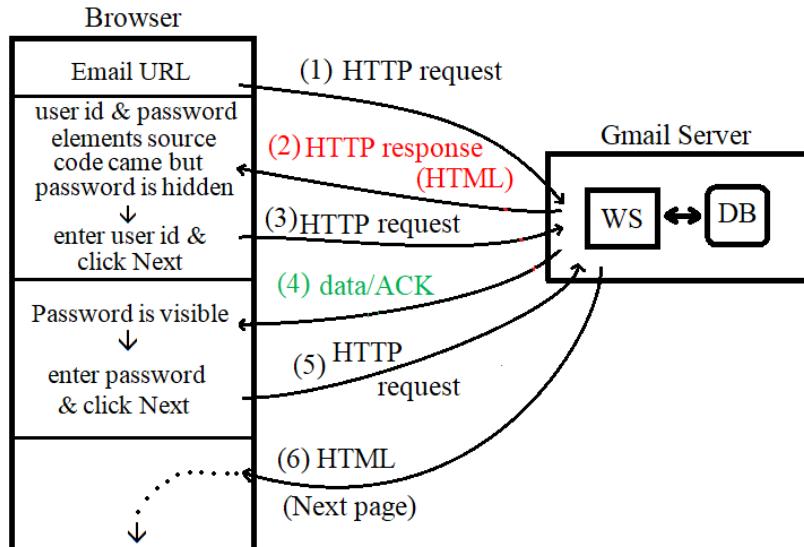
```

j) Handling AJAX calls:

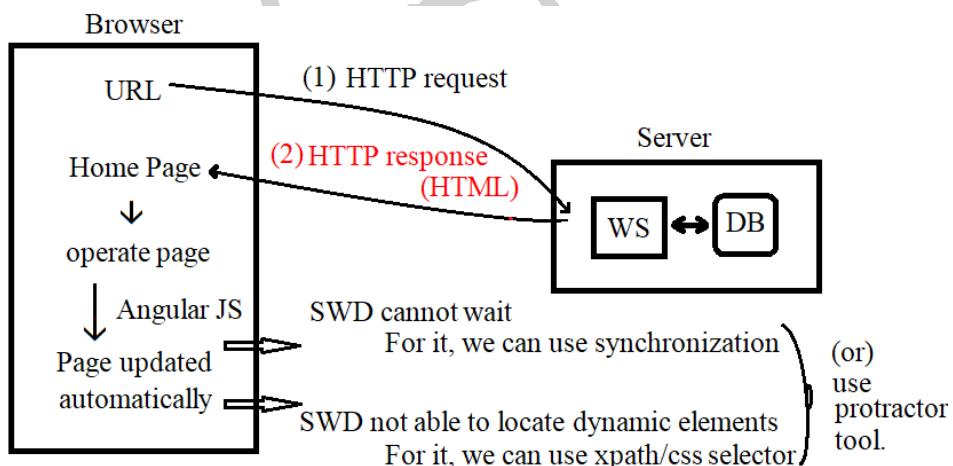
AJAX → Asynchronous Javascript And XML.



From the above diagram we need to maintain wait state in selenium code to handle AJAX calls to identify the availability of AJAX in current site pages, we need to observe source code for **jsname**, **jsid**, **jsclass** ...etc. like attributes for elements.



k) Working with Angular JS related pages:



```

Scanner sc=new Scanner(System.in);
System.out.println("Enter to fill text box");
String x=sc.nextLine();
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://angularjs.org");
//wait for page ready
WebDriverWait w=new WebDriverWait(driver,10);
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//[@ng-
model='yourName']")));
    
```

```

//Scroll to view and fill text box with data
WebElement
e=driver.findElement(By.xpath("//span[@plnkr='hello.html']/ancestor::*[1]"));
driver.executeScript("arguments[0].scrollIntoView();",e);
driver.findElement(By.xpath("//*[@ng-model='yourName']")).sendKeys(x);
w.until(ExpectedConditions.textToBePresentInElement(By.xpath("//h1[contains(te
xt(),'Hello')]")), "Hello "+x+"!");
//Close Site
driver.close();

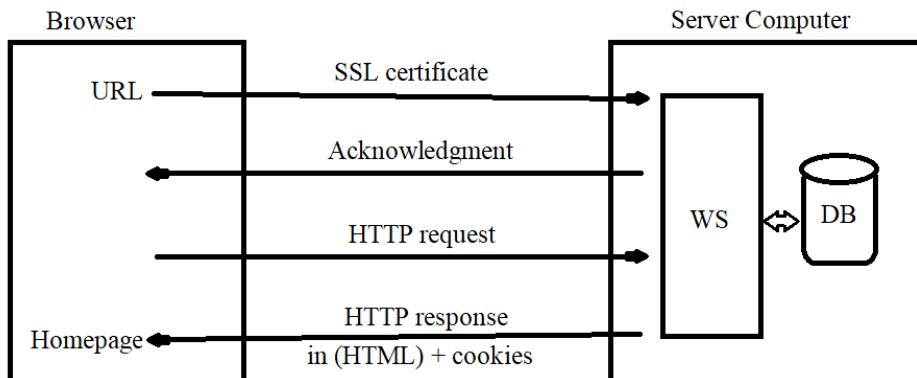
```

Note:

When webpages are developed using **html** with **AJAX**, we can get “**js**” in attributes of elements in source code.

When webpages are developed using **html** with **Angular JS**, we can get “**ng**” in attributes of elements in source code.

I) Handling SSL (Secure Sockets Layer) Certificates:



The process is going same like above figure by default in every browser (Chrome, Firefox, IE, Opera ...etc.) but sometimes, server is not able to respond for browser level SSL certificates, when there is no Acknowledgement from server browser is not able to show homepage of site. To handle this type of situations we can use selenium code to prevent browser level SSL Certificates.

Ex: 1) Your connection is not private.

2) Your clock is behind.

✓ **Stop certificate errors in Chrome browser:**

```

//Stop Certificate errors in Chrome
DesiredCapabilities c=DesiredCapabilities.chrome();
c.setCapability(CapabilityType.ACCEPT_SSL_CERTS,true);
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver cd=new ChromeDriver(c);
cd.manage().window().maximize();
cd.get("https://cacert.org");

```

✓ **Stop certificate errors in IE browser:**

```
//Stop Certificate errors in IE
DesiredCapabilities i=DesiredCapabilities.internetExplorer();
i.setCapability(CapabilityType.ACCEPT_SSL_CERTS,true);
System.setProperty("webdriver.ie.driver","D:\\DineshReddy\\IEDriverServer.exe");
InternetExplorerDriver ied=new InternetExplorerDriver(i);
ied.get("https://cacert.org");
```

✓ **Stop certificate errors in Firefox browser:**

```
//Stop Certificate errors in firefox
DesiredCapabilities f=DesiredCapabilities.firefox();
f.setCapability(CapabilityType.ACCEPT_SSL_CERTS,true);
System.setProperty("webdriver.gecko.driver","D:\\DineshReddy\\geckodriver.exe");
FirefoxDriver fd=new FirefoxDriver(f);
fd.get("https://cacert.org");
```

✓ **Stop certificate errors in Opera browser:**

```
//Stop Certificate errors in opera browser
OperaOptions oo=new OperaOptions();
oo.setBinary("path of the opera browser file in local disc c");
DesiredCapabilities o=DesiredCapabilities.opera();
o.setCapability(CapabilityType.ACCEPT_SSL_CERTS,true);
o.setCapability(OperaOptions.CAPABILITY,oo);
System.setProperty("webdriver.opera.driver","D:\\DineshReddy\\operadriver.exe");
OperaDriver od=new OperaDriver(o);
od.get("https://cacert.org");
```

While writing selenium WebDriver code by using browser options and desired capabilities, we need to combine those options and capabilities objects to create driver objects.

```
//Stop Certificate errors in Chrome using Chrome Options
ChromeOptions co=new ChromeOptions();
co.setBinary("path of the chrome browser file in local disc c");
DesiredCapabilities dc=new DesiredCapabilities();
dc.setCapability(CapabilityType.ACCEPT_SSL_CERTS,true);
dc.setCapability(ChromeOptions.CAPABILITY,co);
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver cdo=new ChromeDriver(c);
cdo.get("https://cacert.org");
```

m) Handling Browser Notifications:

While automating some webpages we can get browser notifications. We need to handle or stop those notifications by working extra code in selenium WebDriver.

✓ **Stop notifications in Chrome Browser:**

```
//Disable Notifications in Chrome
ChromeOptions co=new ChromeOptions();
co.addArguments("--disable-notifications");
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver(co);
driver.get("http://www.bookmyshow.com");
```

✓ **Stop notifications in Firefox Browser:**

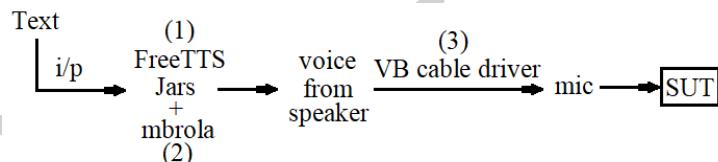
```
//Disable Notifications in Firefox
FirefoxProfile fp=new FirefoxProfile();
fp.setPreference("dom.webnotifications.enabled",false);
//Launch Site
System.setProperty("webdriver.gecko.driver","D:\\DineshReddy\\geckodriver.exe");
FirefoxDriver drive=new FirefoxDriver();
drive.manage().window().maximize();
drive.get("http://www.bookmyshow.com");
```

✓ **Stop notifications in IE Browser:**

- Go to IE browser
- Go to tools menu
- Internet options
- Privacy
- Deselect “Turn on pop up blocker”
- Click apply
- Click ok

n) **Voice Automation:**

While automating some websites we need to give input as voice. To automate the scenario we need to follow below algorithm.



Step 1: Configure free TTS in tester Computer

Free TTS is available as Jars to convert text into voice.

- Go to Google site.
- Enter “freeTTS download” and search.
- Go to <http://freetts.sourceforge.io> site.
- Click on downloading and installing link.
- Click on download.
- Click on download latest version.
- Paste that downloaded file in personal folder.
- Extract download.
- Open extracted folder.
- Go to lib folder.

- Click on **JSAPI** (Java Script Application Programming Interface) and click on i agree.
- Click close after completion.
- Go to eclipse IDE.
- Right click on project.
- Click on properties.
- Click on Java build path.
- Click on libraries and click on add external Jars.
- Browse Jars in lib folder in freeTTS.
- Select all click apply and close.

Example:

```
//Get text
Scanner sc=new Scanner(System.in);
System.out.println("Enter Text Here");
String x=sc.nextLine();
//Convert into voice
VoiceManager vm=VoiceManager.getInstance();
Voice v=vm.getVoice("kevin");
v.allocate();
v.speak(x);
v.deallocate();
```

Step 2: Configure “mbrola” in tester computer

To get various languages with various slangs, frequencies we can use “mbrola” libraries. We need to follow below navigation to configure “mbrola”.

- Go to Google site.
- Enter “mbrola download” and search.
- Select “tets.fpm.ac.be” site.
- Click on MBROLA binary and voices.
- Click on PC/DOS for windows.
- Paste that downloaded file in personal folder.
- Extract that download to a folder (extract to mbr301d).
- Rename extracted folder to “mbrola”.
- Go to same site.
- Click on required languages to download.
- Example US1, US2, US3 ...etc.
- Paste that download in mbrola folder and extract.

Example:

```
Scanner sc=new Scanner(System.in);
System.out.println("Enter Text Here");
String x=sc.nextLine();
VoiceManager vm=VoiceManager.getInstance();
System.setProperty("mbrola.base", "D:\\DineshReddy\\mbrola");
Voice v=vm.getVoice("mbrola_us1");
v.allocate();
v.speak(x);
v.deallocate();
```

Note 1:

In above code “Voice Manager” and “Voice” are instance classes in FreeTTS Jars.

Note 2:

While using mbrola voices we need to add below statement to program.

```
System.setProperty("mbrola.base", "path of mbrola\\mbrola");
```

Note 3:

We need to mention mbrola language folder or voice folder in code like shown below.

```
Voice v=vm.getVoice("mbrola_xxxx");
```

Here: xxxx voice library name in mbrola folder.

Step 3: Download and install VB Cable driver

To get virtual connection in between speaker and mic in our computer, we need to use VB Cable driver.

- Go to Google site.
- Enter “VB Cable driver download” and search.
- Select “vb-audio.com” site.
- Click on “download”.
- Paste that download in personal folder.
- Extract to folder.
- Install “VBCable” by clicking “VBCable” setup.
- Click “next” until finish.

Note 1:

VB Cable driver can create Cable Input and Cable Output in sounds in Hardware and Sound in Control panel.

Note 2:

VB Cable driver can provide two setup files. Such as

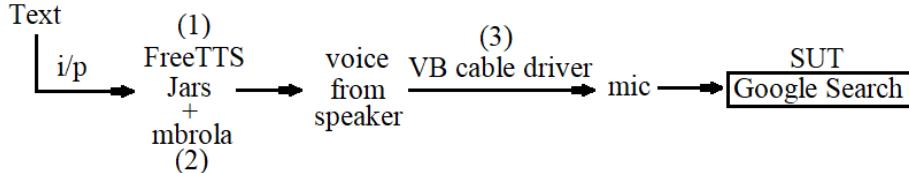
- 1) VBCABLE_setup.
- 2) VBCABLE_setup_x64.

Note 3:

We need to follow below navigation to activate Virtual Cable IN and Virtual Cable OUT.

- Go to Control Panel.
- Click on Hardware and Sound.
- Click on Sound.
- Go to Playback.
- Disable all physical devices & enable CABLE Input.
- Go to Recording.
- Disable all physical devices & enable CABLE Output.
- Click ok & close Control Panel.

Example:



```
Scanner sc=new Scanner(System.in);
System.out.println("Enter Text Here");
String x=sc.nextLine();
ChromeOptions co=new ChromeOptions();
co.addArguments("use-fake-ui-for-media-stream=1");
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver(co);
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
WebDriverWait w=new WebDriverWait(driver,10);
w.until(ExpectedConditions.visibilityOfElementLocated(By.id("gsri_ok0")));
driver.findElement(By.id("gsri_ok0")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.id("spchb")));
VoiceManager vm=VoiceManager.getInstance();
System.setProperty("mbrola.base", "D:\\DineshReddy\\mbrola");
Voice v=vm.getVoice("mbrola_us1");
v.allocate();
v.speak(x);
v.deallocate();
```

Note 4:

```
co.addArguments("use-fake-ui-for-media-stream=1");
```

Here: `use-fake-ui-for-media-stream=1` will allow virtual mic and speaker instead of physical devices (Don't show notification in browser).

Note 5:

We are able to write xpath to specific attribute of an element.

xpath of an element

`//*[@@name='q']/@type`

xpath to "type" attribute of an element in HTML source code (DOM)

Note 6:

When we are not able to write wait condition depends on specific element we can use lambda function concept with Java Script.

```
w.until(temp->driver.executeScript("return document.readyState;").equals("complete"));
```

Here: `temp->` is a lambda function, `return document.readyState;` is Javascript code.

o) cssSelector:

In general we can use anyone of 8 locators related to "By" in SWD to locate element in webpages.

Xpath	cssSelector
<ul style="list-style-type: none">→ Support from all browsers.→ Slow.→ Various models of syntaxes (Parent to child & child to parent).	<ul style="list-style-type: none">→ Support from all browsers, but familiar to IE browser.→ Fast.→ Less no.of syntaxes (parent to child only).

Syntax 1:

tagName#idvalue

Syntax 2:

tagName.classvalue

Syntax 3:

tagName[attribute=value]

Syntax 4:

tagName.classvalue[attribute=value]

Syntax 5:

tagName[attribute^='starting value']

Syntax 6:

tagName[attribute\$='ending value']

Syntax 7:

tagName[attribute*=‘substring value’]

(or)

tagName[attribute:contains(‘substring value’)]

Syntax 8:

tagName#idvalue>tagName

Here ‘>’ indicates relation from parent to child

(or)

tagName#idvalue tagName

Here space between **idvalue & tagName** indicates relation from parent to child/sub child.

Syntax 9:

tagName#idvalue>tagName:nth-of-type(index)

Here it indicates parent to specific child

tagName#idvalue tagName:nth-of-type(index)

Here it indicates parent to specific child/sub child

Syntax 10:

tagName#idvalue>tagName:last-child

Here it indicates parent to last child

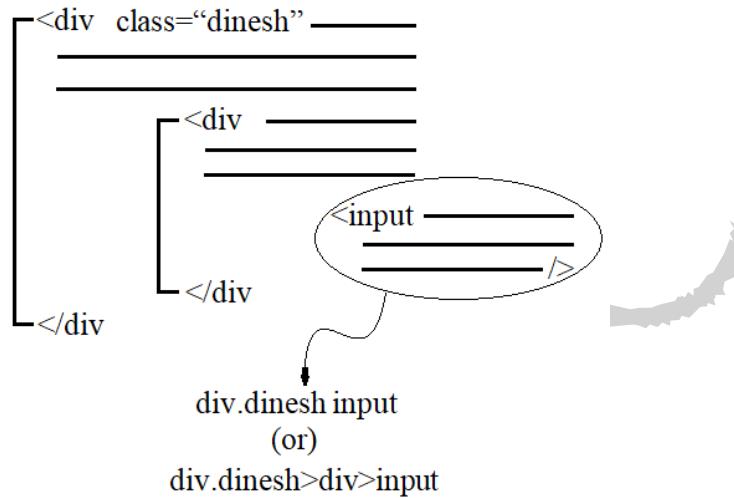
tagName#idvalue tagName:last-child

Here it indicates parent to last child/sub child.

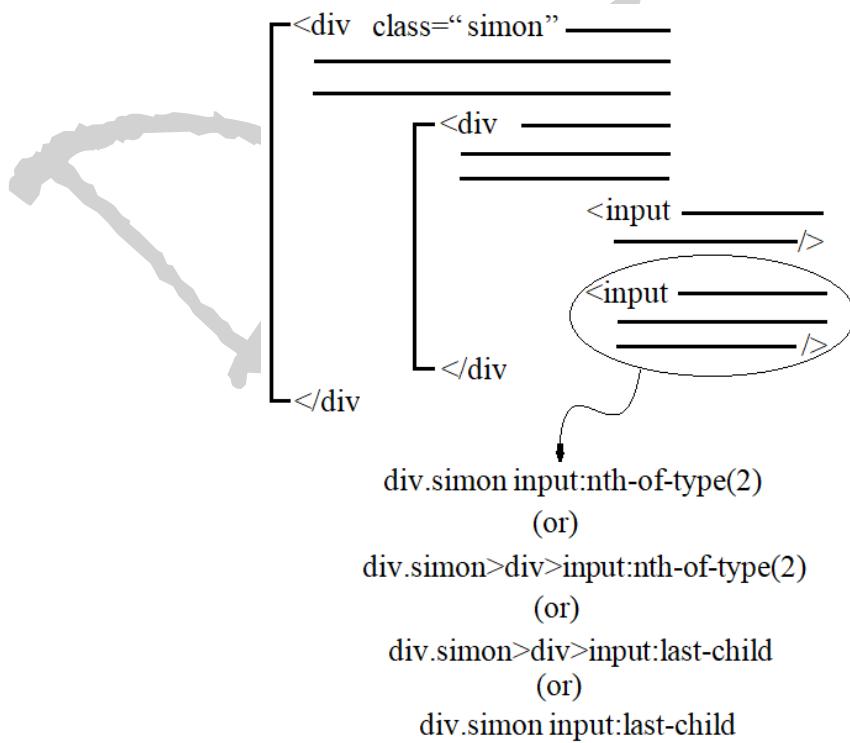
Note:

From the above syntaxes **tagName** indicates name of the tag in source code. **Idvalue** indicates value for the id attribute in source code. Similarly, **classvalue** indicates value for the class attribute in source code.

Example 1:



Example 2:



Example 3:

```
<input type="email" class="inputtext" name="email" id="email" tabindex="1" data-testid="royal_email">
```

```
input#email
```

```
input.inputtext
```

```
input[name=email]
```

```
input.inputtext[name=email]
```

```
input[data-testid^='royal']
```

```
input[data-testid$='email']
```

```
input[data-testid*='yal_email']
```

Note 1:

In xpath and cssSelector index can start with '1'.

Note 2:

We can use cssSelector to locate an element like shown below.

```
driver.findElement(By.cssSelector("input#email")).sendKeys("mindq");
```

Here: email is id value.

Note 3:

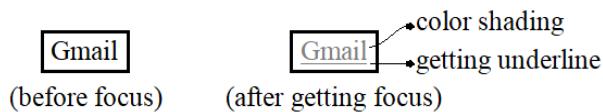
Browser level inspect window can allow us to check the given cssSelector or xpath is right or wrong.

Note 4:

In general in webpages developer can use css attribute to elements to change look dynamically

- color
- background-color
- font-size
- opacity
- text-decoration
- font-family ...etc.,

We need to follow below logic to work with above attributes.



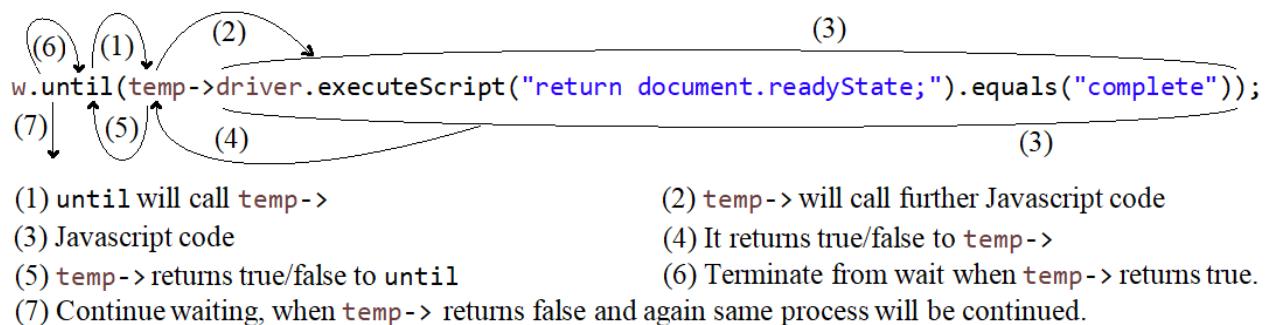
```

//Launch Site
System.setProperty("webdriver.chrome.driver", "D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
WebDriverWait w=new WebDriverWait(driver,10);
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[text()='Gmail'][1]")));
//Get details before focus
String o1=driver.findElement(By.xpath("//*[text()='Gmail'][1]").getCssValue("opacity");
String d1=driver.findElement(By.xpath("//*[text()='Gmail'][1]").getCssValue("text-decoration");
System.out.println("Opacity 1 is: "+o1);
System.out.println("Text-Decoration 1 is: "+d1);
//Focus on element
Actions a=new Actions(driver);
WebElement e=driver.findElement(By.xpath("//*[text()='Gmail'][1]"));
a.moveToElement(e).build().perform();
//Get details once focus
String o2=driver.findElement(By.xpath("//*[text()='Gmail'][1]").getCssValue("opacity");
String d2=driver.findElement(By.xpath("//*[text()='Gmail'][1]").getCssValue("text-decoration");
System.out.println("Opacity 2 is: "+o2);
System.out.println("Text-Decoration 2 is: "+d2);
//Close site
driver.close();

```

Note 5:

To define explicit wait we can use Javascript code by following lambda function concept in java.



Example:

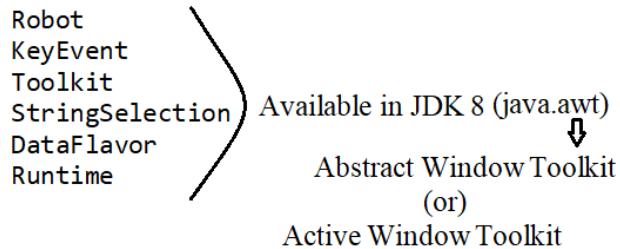
Related to	navigate().to()
	navigate().back()
	navigate().forward()
	navigate().refresh()

```
//Launch Site  
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");  
ChromeDriver driver=new ChromeDriver();  
driver.manage().window().maximize();  
driver.get("https://www.facebook.com");  
Thread.sleep(3000);  
driver.navigate().to("http://www.google.com");  
Thread.sleep(3000);  
driver.navigate().back();//facebook  
Thread.sleep(3000);  
driver.navigate().forward();//Google  
Thread.sleep(3000);  
driver.navigate().refresh();//Google  
Thread.sleep(3000);  
driver.close();
```



III. JAVA ROBOT

- ❖ Inbuilt in JDK 8.
- ❖ To automate “window” based screens.
Ex: windows, file upload, login pop-up...etc.
- ❖ Alternative for it is AutoIT, but **Java Robot** is platform-independent whereas AutoIT can run in “windows computer only”.
- ❖ While testing websites via automation using SWD, we can get:
 - Win menu
 - File upload window
 - File download window
 - Authentication pop-up window...etc.
- ❖ We need to use below classes in “window” based automation.



Example 1: Automate win menu

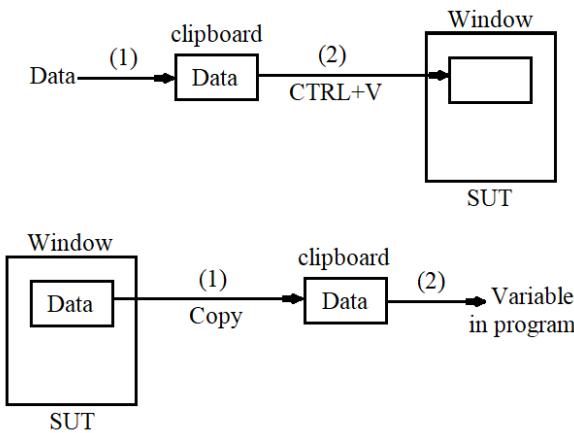
```
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.seleniumhq.org");
WebDriverWait w=new WebDriverWait(driver,10);
w.until(ExpectedConditions.visibilityOfElementLocated(By.LinkText("Download")));
//Right click on link
Actions a=new Actions(driver);
WebElement e=driver.findElement(By.LinkText("Download"));
a.contextClick(e).build().perform();
//Automate win-menu (java robot(in JDK))
Robot r=new Robot();
for(int i=1;i<=2;i++)
{
    r.keyPress(KeyEvent.VK_DOWN);
    r.keyRelease(KeyEvent.VK_DOWN);
}
r.keyPress(KeyEvent.VK_ENTER);
r.keyRelease(KeyEvent.VK_ENTER);
//Close site
driver.close();
```

Example 2: Automate window based screen

```
//Launch Calculator
Runtime.getRuntime().exec("calc.exe");
Thread.sleep(2000);
//Send input 1 to clip-board
StringSelection x=new StringSelection("94919");
Toolkit.getDefaultToolkit().getSystemClipboard().setContents(x, null);
//Send clip-board data to window
Robot r=new Robot();
r.keyPress(KeyEvent.VK_CONTROL);
r.keyPress(KeyEvent.VK_V);
r.keyRelease(KeyEvent.VK_V);
r.keyRelease(KeyEvent.VK_CONTROL);
Thread.sleep(1000);
//Click +
r.keyPress(KeyEvent.VK_ADD);
r.keyRelease(KeyEvent.VK_ADD);
Thread.sleep(1000);
//Send input 1 to clip-board
StringSelection Y=new StringSelection("47838");
Toolkit.getDefaultToolkit().getSystemClipboard().setContents(Y, null);
//Send clip-board data to window
r.keyPress(KeyEvent.VK_CONTROL);
r.keyPress(KeyEvent.VK_V);
r.keyRelease(KeyEvent.VK_V);
r.keyRelease(KeyEvent.VK_CONTROL);
Thread.sleep(1000);
//Click +
r.keyPress(KeyEvent.VK_EQUALS);
r.keyRelease(KeyEvent.VK_EQUALS);
Thread.sleep(1000);
//Get output
r.keyPress(KeyEvent.VK_CONTEXT_MENU);
r.keyRelease(KeyEvent.VK_CONTEXT_MENU);
Thread.sleep(1000);
r.keyPress(KeyEvent.VK_DOWN);
r.keyRelease(KeyEvent.VK_DOWN);
Thread.sleep(1000);
r.keyPress(KeyEvent.VK_ENTER);
r.keyRelease(KeyEvent.VK_ENTER);
Thread.sleep(1000);
String z=(String)
Toolkit.getDefaultToolkit().getSystemClipboard().getData(DataFlavor.stringFlavor);
System.out.println(z);
Thread.sleep(1000);
r.keyPress(KeyEvent.VK_ALT);
r.keyPress(KeyEvent.VK_F4);
r.keyRelease(KeyEvent.VK_F4);
r.keyRelease(KeyEvent.VK_ALT);
```

Note 1:

While automating window based screens we can use below ways to send data and get output.



Note 2:

```
Runtime.getRuntime().exec("Path of application");
```

Here: Runtime is a static class in **java.awt** package.

Path of application is a .exe/audiofile/videofile/cmd...etc.

Note 3: To send data to clipboard

```
StringSelection x=new StringSelection("xxxx");
Toolkit.getDefaultToolkit().getSystemClipboard().setContents(x, null);
```

Here: StringSelection is an instance class in **java.awt** package in JDK

Toolkit is a static class in **java.awt** package in JDK

x is a variable, which have data

null is used when there is no users of computer (no owner).

Note 4: To send clipboard data to application

```
Robot r=new Robot();
r.keyPress(KeyEvent.VK_CONTROL);
r.keyPress(KeyEvent.VK_V);
r.keyRelease(KeyEvent.VK_V);
r.keyRelease(KeyEvent.VK_CONTROL);
```

Here: Robot is an instance class in **java.awt** package in JDK

KeyEvent is a static class in **java.awt** package in JDK

Note 5: Get data from clipboard into a variable

```
String z=(String)
Toolkit.getDefaultToolkit().getSystemClipboard().getData(DataFlavor.stringFlavor);
Here: DataFlavor is a static class in java.awt package in JDK
```

Example 3: Automate file upload in Gmail

```
Scanner sc=new Scanner(System.in);
System.out.println("Enter gmail id");
String x=sc.nextLine();
System.out.println("Enter gmail password");
String y=sc.nextLine();
System.out.println("Enter recipient email id");
String z=sc.nextLine();
System.out.println("Enter subject");
String u=sc.nextLine();
System.out.println("Enter message");
String v=sc.nextLine();
//Launch Gmail Site(SWD)
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.gmail.com");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("identifier")));
//Do login
driver.findElement(By.name("identifier")).sendKeys(x);
w.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[text()='Next']")));
driver.findElement(By.xpath("//span[text()='Next']")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("password")));
driver.findElement(By.name("password")).sendKeys(y);
w.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[text()='Next']")));
driver.findElement(By.xpath("//span[text()='Next']")).click();
//Do compose
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[text()='Compose']")));
driver.findElement(By.xpath("//div[text()='Compose']")).click();
//Fill fields
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("to")));
driver.findElement(By.name("to")).sendKeys(z);
driver.findElement(By.name("subjectbox")).sendKeys(u);
driver.findElement(By.xpath("//div[@aria-label='Message Body']")).sendKeys(v);
driver.findElement(By.xpath("//div[@aria-label='Attach
files']/descendant::div[3]")).click();
Thread.sleep(10000);
//Automate file upload window(Java Robot)
StringSelection ss=new StringSelection("D:\\SEMINARS\\SELENIUM NOTES\\Selenium
Notes.docx");
Toolkit.getDefaultToolkit().getSystemClipboard().setContents(ss,null);
Robot r=new Robot();
if(System.getProperty("os.name").contains("Windows"))
{
    r.keyPress(KeyEvent.VK_CONTROL);
    r.keyPress(KeyEvent.VK_V);
    r.keyRelease(KeyEvent.VK_V);
    r.keyRelease(KeyEvent.VK_CONTROL);
    Thread.sleep(5000);
}
else if(System.getProperty("os.name").contains("Mac"))
{
    r.keyPress(KeyEvent.VK_META);
```

```

        r.keyPress(KeyEvent.VK_V);
        r.keyRelease(KeyEvent.VK_V);
        r.keyRelease(KeyEvent.VK_META);
        Thread.sleep(5000);
    }
    else if(System.getProperty("os.name").contains("Linux"))
    {
        r.keyPress(KeyEvent.VK_CONTROL);
        r.keyPress(KeyEvent.VK_V);
        r.keyRelease(KeyEvent.VK_V);
        r.keyRelease(KeyEvent.VK_CONTROL);
        Thread.sleep(5000);
    }
    else
    {
        System.out.println("Unknown Browser");
        System.exit(0); //stop execution forcibly
    }
    r.keyPress(KeyEvent.VK_ENTER);
    r.keyRelease(KeyEvent.VK_ENTER);
    Thread.sleep(5000);
    //Wait until file upload to start
    w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@class='vq']")));
    //Wait until complete file upload
    w.until(ExpectedConditions.attributeContains(By.xpath("//*[@class='vq']"), "role", "button"));
    driver.findElement(By.xpath("//div[contains(@aria-label, 'Send ')]")).click();
    //Do logout after sending message
    w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[contains(text(), 'Message sent')]")));
    driver.findElement(By.xpath("//*[contains(@class, 'gbii')]")).click();
    w.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Sign out")));
    driver.findElement(By.linkText("Sign out")).click();
    w.until(ExpectedConditions.visibilityOfElementLocated(By.name("password")));
    //Close site
    driver.close();
}

```

Note 6:

While automating window based screens, we need to use different keyboard keys combinations with respect to OS.

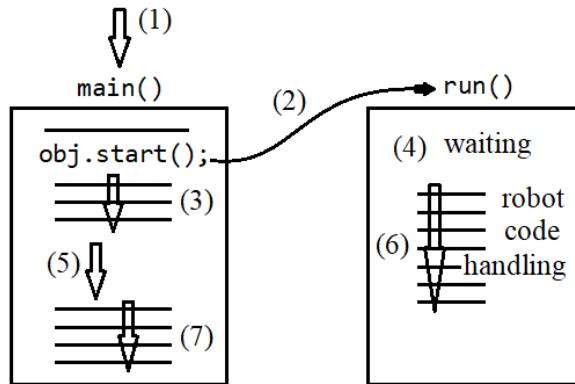
Ex:

OS in tester computer	Short key for copy	Short key for paste
Windows	CTRL+C	CTRL+V
Mac	COMMAND+C	COMMAND+V
Linux	CTRL+ALT+C	CTRL+V

Example 4: Handling login pop-up via multi-threading

```
public class MultiThreading extends Thread
{
    public void run()
    {
        try
        {
            Thread.sleep(10000);
            Robot r=new Robot();
            StringSelection x=new StringSelection("xxxx");
            Toolkit.getDefaultToolkit()
                .getSystemClipboard().setContents(x,null);
            r.keyPress(KeyEvent.VK_CONTROL);
            r.keyPress(KeyEvent.VK_V);
            r.keyRelease(KeyEvent.VK_V);
            r.keyRelease(KeyEvent.VK_CONTROL);
            r.keyPress(KeyEvent.VK_TAB);
            r.keyRelease(KeyEvent.VK_TAB);
            StringSelection y=new StringSelection("xxxx");
            Toolkit.getDefaultToolkit()
                .getSystemClipboard().setContents(y,null);
            r.keyPress(KeyEvent.VK_CONTROL);
            r.keyPress(KeyEvent.VK_V);
            r.keyRelease(KeyEvent.VK_V);
            r.keyRelease(KeyEvent.VK_CONTROL);
            Thread.sleep(5000);
            r.keyPress(KeyEvent.VK_TAB);
            r.keyRelease(KeyEvent.VK_TAB);
            Thread.sleep(5000);
            r.keyPress(KeyEvent.VK_ENTER);
            r.keyRelease(KeyEvent.VK_ENTER);
        }
        catch(Exception e)
        {
        }
    }
    public static void main(String[] args)
    {
        MultiThreading obj=new MultiThreading();
        obj.start(); //Call run method
        //Launch Site
        System.setProperty("webdriver.chrome.driver",
        "D:\\DineshReddy\\chromedriver.exe");
        ChromeDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://eforms.agility.com");
    }
}
```

Further code remains normal



- (1) First when we run it will start running **main()** method.
- (2) When we written **obj.start();** it will call **run()** method to start.
- Here both (3) & (4) running at the same time.
- (3) Driver getting home page URL.
- (4) Here **run()** method is waiting to get push pop-up window.
- (5) Selenium is waiting to get source code from server.
- (6) Java Robot code is handling push pop-up window.
- (7) Since, Java Robot cleared push pop-up window and source code is loaded from server further code is executing at this stage.

Case Study:

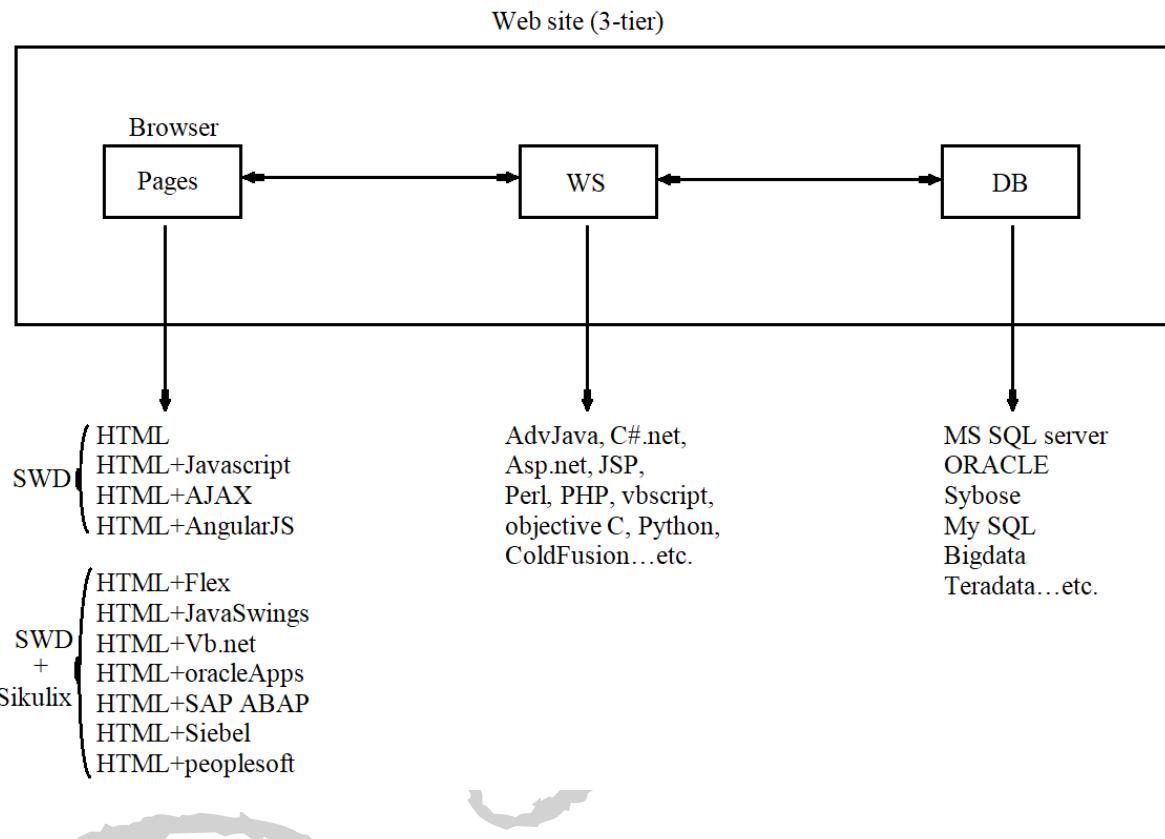
External UI in pages	How to handle in Automation
Banner (part of page, but visible on top)	<ul style="list-style-type: none"> → No need for multi-threading → No need for JAVA Robot → Write SWD code to handle banner elements in page, in same main() method
Alert (part of page, but mandatory to close)	<ul style="list-style-type: none"> → No need for multi-threading → No need for JAVA Robot → Write SWD code to handle alerts, in same main() method
Browser Notifications	<ul style="list-style-type: none"> → No need for multi-threading → No need for JAVA Robot → Write SWD code to handle browser notifications in page, in same main() method

Pop-up window (File upload, file download, win menu...etc.)	<ul style="list-style-type: none"> → No need for multi-threading → Write JAVA Robot code to handle pop-up windows, in same main() method
Push pop-up window (Authentication window) Ex: http://www.eforms.agility.com	<ul style="list-style-type: none"> → Multi-threading required → Write Java Robot code in run() handle method to handle push pop-up window, at the same time main() method is having SWD code to continue further steps.



IV. SIKULIX

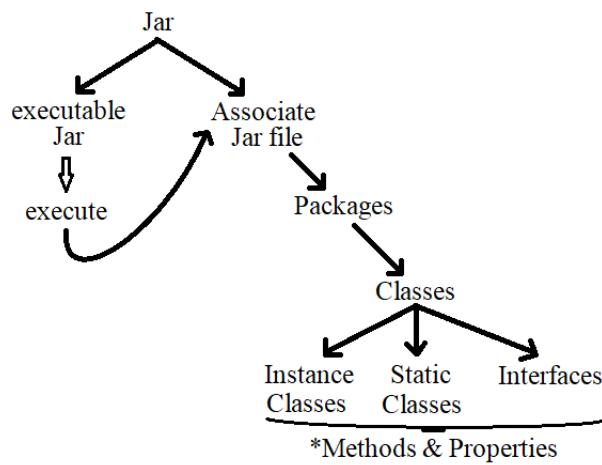
- ❖ Developed by Raiman.
- ❖ Useful to automate anything in screens but to automate non-html elements in web pages.



a) Configure Sikulix in tester computer:

- Go to Google site
- Enter Sikulix download
- Go to <https://launchpad.net/sikulix>
- Click on Sikulix setup jar
- Paste that downloaded file in personal folder.
- Double click on that setup jar
- Select second option “pack 2”
- Click on “setup now”
- Click “yes” until get “fun” message
- Click ok
- Observe the availability of Sikulix api jar
- Associate that jar to project in eclipse

Note 1:



Note 2:

SWD, FreeTTS, commons.io...etc. related jars are associative jars to associate with java project.

“Sikulix setup” jar is an executable jar and it can provide “Sikulix api” associative jar. We can associate “Sikulix api” jar to java project in eclipse.

Note 3:

When “Sikulix” setup jar is not respond to double click, we can follow anyone of below two ways:

- 1) → Right click on “Sikulix setup” jar
→ Run as administrator.
- 2) → Go to command prompt
→ Run below command
→ `java -jar D:\DineshReddy\sikulixsetup-1.1.3.jar`

Here: `D:\DineshReddy\sikulixsetup-1.1.3.jar` is path of personal folder where Sikulix is pasted.

b) Locating Elements:

Sikulix is useful to automate anything but we can use Sikulix to automate non html elements in web pages. Here we need to use snipping tool (inbuilt in windows OS) or paint (inbuilt in windows OS) or qsnap (need to install) or duck capture (need to install)...etc. to take non html element as images. Here we need to save image (elements) as **.png** files. We need to save those images in project folder.

Ex: `D:\DineshReddy\alluarjun`

While running Sikulix based code, sikuli run time can locate corresponding element on “desktop” by matching with given **.png** image.

c) Methods in “Screen” class:

click():

We can use this method to click on element by matching with .png image.

```
Screen s=new Screen();
s.click("path of .png image");
```

doubleClick():

We can use this method to “double click” on element by matching with .png image.

```
Screen s=new Screen();
s.doubleClick("path of .png image");
```

type():

We can use this method to fill an element with data by matching .png images.

```
Screen s=new Screen();
s.type("path of .png image","data");
```

exists():

We can use this method to check the availability of element on desktop by matching with .png image.

```
Screen s=new Screen();
if(s.exists("skipadd.png")!=null)
{
    s.click("skipadd.png");
}
```

mouseMove():

We can use this method to move mouse pointer to specific location on desktop or to specific element matching with .png image.

```
s.mousePosition(xxx,yyy);
(or)
Location l=new Location(xxx,yyy);
s.mousePosition(l);
(or)
s.mousePosition("path of .png image");
```

Here: xxx are X-coordinates & yyy are Y- coordinates of desktop.

Location is a class in Sikulix api Jar.

find():

We can use this method to create a Dom to an element.

```
WebElement e=driver.findElement(locator); //Dom in SWD
```

```
Match e=s.find("path of .png image"); //Dom in Sikulix
```

Here: Match is a class in Sikulix api Jar.

dragDrop():

To perform drag and drop between element to element (or) element to specific locating we can use this method.

Example 1:

```
Match e1=s.find("path of .png image");
Match e2=s.find("path of .png image");
s.dragDrop(e1,e2);
```

Example 2:

```
Match e=s.find("path of .png image");
int x=e.getX();
int y=e.getY();
Location l=new Location(x-50,y);
s.dragDrop(e,l);
```

Example:

- Launch YouTube site [SWD]
- Search for a video [SWD]
- Start a video by clicking link [SWD]
- Automate video options/icons [Sikulix]
- Close site [SWD]

```
//Launch Site(SWD)
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.youtube.com");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.id("search")));
//Search for a video(SWD)
driver.findElement(By.id("search")).sendKeys("abdul kalam sir speeches");
driver.findElement(By.xpath("//input[@id='search']/following::yt-icon[2]")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[contains(@aria-label,'Abdul Kalam in European Parliament')]")));
//Start video(SWD)
driver.findElement(By.xpath("//a[contains(@aria-label,'Abdul Kalam in European Parliament')]")).click();
```

```

//Skip add if exists(Sikulix)
Thread.sleep(5000);
Screen s=new Screen();
if(s.exists("skipadd.png")!=null)
{
    s.click("skipadd.png");
}
//Pause video(Sikulix)
s.mouseMove(300,376);
s.click("pause.png");
//Play video(Sikulix)
Thread.sleep(5000);
s.mouseMove(300,376);
s.click("play.png");
//Decrease volume(Sikulix)
Thread.sleep(5000);
s.mouseMove(300,376);
s.mouseMove("volume.png");
Match e=s.find("bubble.png");
int x=e.getX();
int y=e.getY();
Location l=new Location(x-50,y);
s.dragDrop(e,l);
Thread.sleep(5000);
//Increase volume(Sikulix)
Location l1=new Location(x+50,y);
s.dragDrop(e,l1);
Thread.sleep(5000);
//Close site
driver.close();

```

wheel():

We can use this method to perform “scrolling” on screens.

```

Screen s=new Screen();
s.wheel(Button.MIDDLE,5);
s.wheel(Button.WHEEL_UP,5);

```

Here: Button is a static class in Sikulix

We can also use `Button.WHEEL_DOWN` in the place of `Button.MIDDLE`

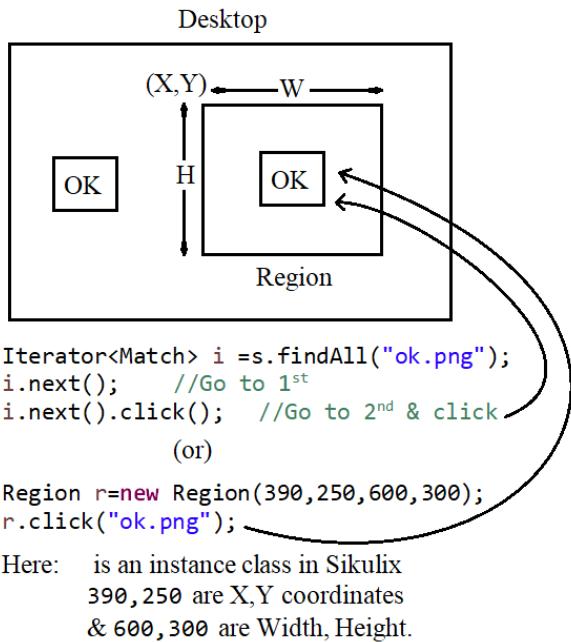
5 indicates number of times

capture():

- We can use this method to capture visible area of desktop as a screenshot.
- This method can support screenshot at a specific region instead of full screen.

findAll():

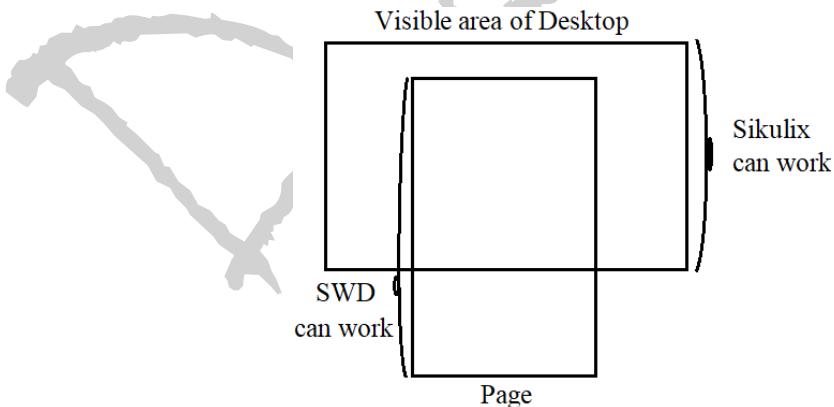
When more than one element is matching with our given .png image, we can follow any one of below 2 ways to operate specific element.



keyDown()

keyUp():

We can use these methods to get keyboard Keys effects on visible area of screen.



```
s.keyDown(Key.ALT);
s.keyDown(Key.F4);
s.keyUp(Key.F4);
s.keyUp(Key.ALT);
```

Here: `Key` is a static class in Sikulix api

Example:

```
//Launch Site(SWD)
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("q")));
//Screen capture (full screen) via sikulix
File f1=new File("google.png");
Screen s=new Screen();
ScreenImage si1=s.capture();
BufferedImage bi1=si1.getImage();
ImageIO.write(bi1,"png",f1);
//Screen capture (required region) via sikulix
File f2=new File("google clip.png");
Region r=new Region(390,250,600,300);
ScreenImage si2=s.capture(r);
BufferedImage bi2=si2.getImage();
ImageIO.write(bi2,"png",f2);
//Close site(Sikulix)
s.keyDown(Key.ALT);
s.keyDown(Key.F4);
s.keyUp(Key.F4);
s.keyUp(Key.ALT);
```

In above code, we used few Java classes (JDK) like: File, BufferedImage, ImageIO.

Note 1:

While automating non html elements in our webpages using Sikulix, we can use below classes related to “Sikulix api” Jar.

→ **Screen**

→ **ScreenImage**

→ **Match**

→ **Key**

→ **Region**

→ **Button**

Note 2:

While using Sikulix, we can face below challenges:

- Sikulix can run on visible area of desktop only. When our automatable elements are not in visible area of Desktop, Sikulix can return “FindFailed” Exception.
- If image appearance varies in pixel size, Sikulix will also return “FindFailed” Exception.
- If two or more similar images are available on the screen, Sikulix will attempt to select the wrong image.
- Required more memory to store 100's of .png images to locate elements for automation.

Note 3:

While collecting multiple elements using selenium WebDriver and Sikulix, we need to use 2 Java collection classes. Such as, List and Iterator respectively.

Operate 3rd element (Random Access in SWD)

```
List<WebElement> l=driver.findElements(locator)
l.get(2).operation();
Screen s=new Screen();
```

Operate 3rd element (Sequential Access in Sikulix)

```
Iterator<Match> i=s.findAll("path of .png image");
i.next();
i.next();
i.next().operation();
```



V. EXTENT REPORTS

- ❖ Developed by “Relevent Codes” organization.
- ❖ Open source Jars.
- ❖ Useful to generate test reports/logs.
- ❖ Support external **Screenshots Injection**.
- ❖ Save reports/logs as “**html**” file only.
- ❖ Provide **pie charts** to analyze test results.
- ❖ Maintain colors, dates and time.
- ❖ Support results **overwriting** and **Appending**.

a) Configure Extent Reports in tester computer:

- Go to <http://extentsreports.com> site
- Go to community edition
- Go to version 2
- Click on Java to start download
- Paste that download in personal folder
- Extract that download
- Go to Eclipse IDE
- Right click on project & go to “properties”
- Go to “Java Build Path”
- Go to “libraries” and click “add external jars”
- Browse for extent reports Jar which are inside of lib folder and outside of lib folder
- Select all those Jars and click “apply & close”

b) Classes to be used:

- ExtentReports
(It is an instance class extentsreports jar)
- ExtentTest
(It is an instance class extentsreports jar)
- LogStatus
(It is a static class extentsreports jar)

Example:

```
ExtentReports er=new ExtentReports("path of the html file to save\\name.html",
true/false);
ExtentTest et=er.startTest("Title for test");
```

Here: er is new object & et is referred object.

If we write **true** then results are over writing, if we write **false** then results are appending.

er.startTest("Title for test"); it can return an object, which can be pointed by "et".

```
//Take current date and time as file name
Date d=new Date();
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
String y=sdf.format(d)+".png";
```

Here: Date, SimpleDateFormat & String are classes in JDK

Date is an instance class in java.util

SimpleDateFormat is an instance class in java.text

```
//Get Screenshot
File src=driver.getScreenshotAs(OutputType.FILE);
//Save screenshot
File dest=new File(y);
FileHandler.copy(src,dest);
//attach screenshot to extent reports
et.log(LogStatus.FAIL,"Title test failed"+et.addScreenCapture(y));
```

Here: File is an instance class in JDK (java.io)

FileHandler is a static class in SWD (instead of "FileUtils" class in commons.io Jar).

LogStatus is a static class in extentreports Jar

Note 1:

LogStatus class can allow us to define test results as:

→ ERROR	→ PASS
→ FAIL	→ SKIP
→ FATAL	→ UNKNOWN
→ INFO	→ WARNING

Note 2:

Extent Reports html file can provide pie charts via "enable dash board" icon.

Example for saving Extent Reports:

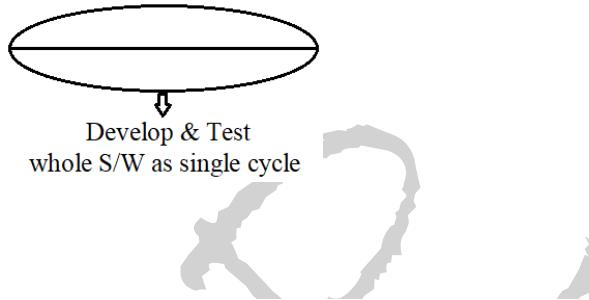
```
ExtentReports er=new ExtentReports("Google.html", false);
ExtentTest et=er.startTest("Tittle for test");
//Launch Site(SWD)
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("q")));
//Get tittle
String x=driver.getTitle();
//Take current date and time as file name
Date d=new Date();
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
String y=sdf.format(d)+".png";
//Get Screenshot
File src=driver.getScreenshotAs(OutputType.FILE);
//Save screenshot
File dest=new File(y);
FileHandler.copy(src,dest);
if(x.equals("Google"))
{
    et.log(LogStatus.PASS,"tittle test passed");
}
else
{
    //attach screenshot to extent reports
    et.log(LogStatus.FAIL,"Tittle test failed"+et.addScreenCapture(y));
}
er.endTest(et);
er.flush();
//Close site
driver.close();
```

VI. STAF

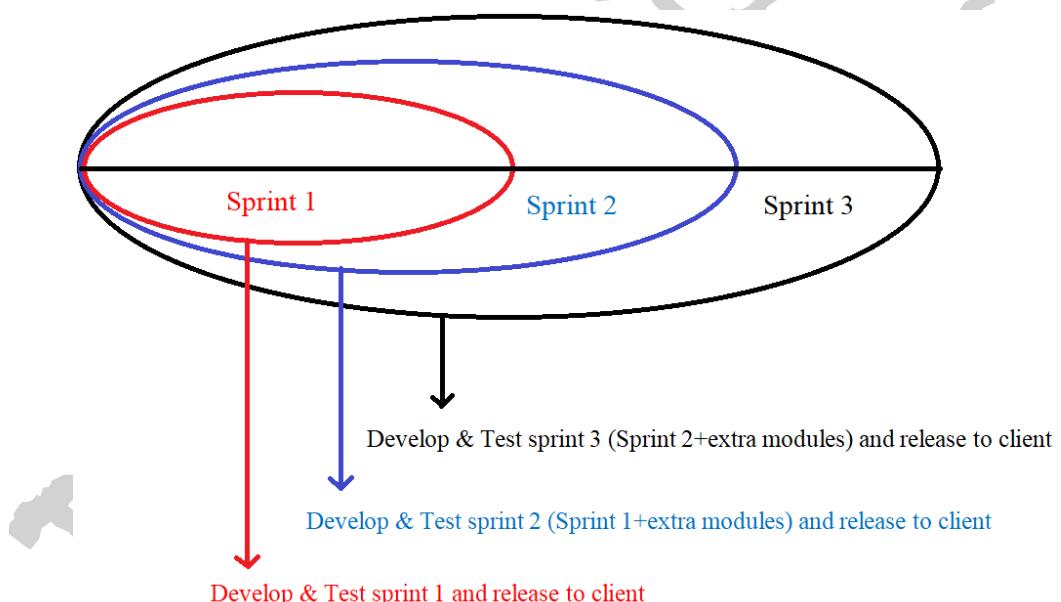
[S/W TEST AUTOMATION FRAMEWORK]

a) Waterfall vs Agile Scrum:

- Water fall



- *Agile Scrum



b) Agile Scrum Process:

(SDLC+STCL+BLC+ALC):

In the below figure:

SM → Scrum Master

PBL → Product Back Log

ST → Scrum Team

PO → Product Owner

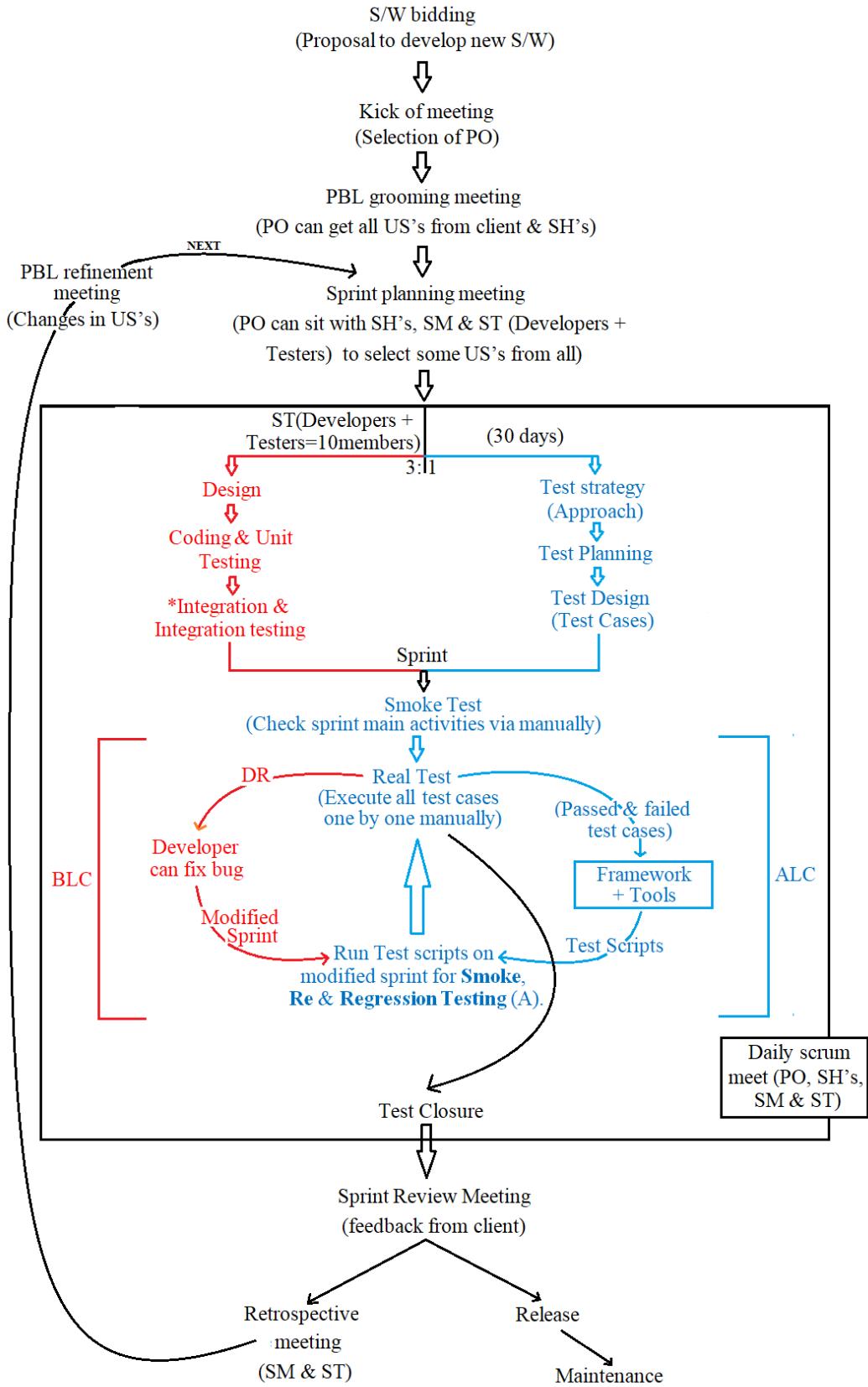
SH's → Stake Holders

US's → User Stories

ALC → Automation Life Cycle

DR → Defect Report

BLC → Bug Life Cycle



c) Framework:

Framework is a process to convert manual test cases into test automation scripts using testing tools.

Ex:

- Page Object Model (POM).
- Data driven Framework using **Jxl/POI**
- Module driven Framework using **TestNg**
- Keyword driven Framework
- BDD (Behaviour Data Driven) Framework using **Cucumber**

From the above 5 Frameworks we can use anyone among them but each Framework have some easy and difficult instructions. So, by combining those 5 Frameworks we can get a Framework called **Customized/Hybrid Framework**.

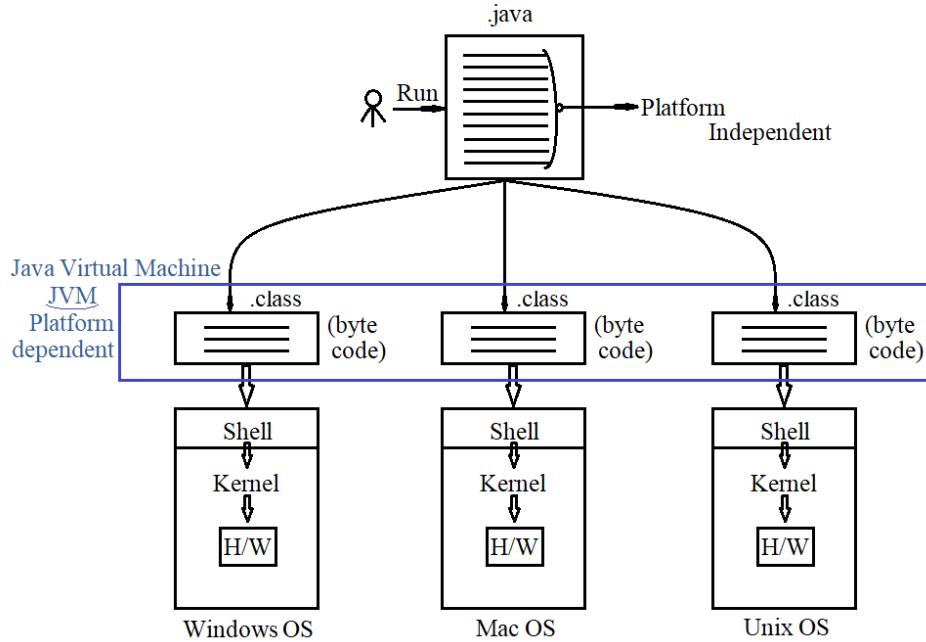
- Customized/Hybrid Framework

d) Prerequisites to follow a framework:

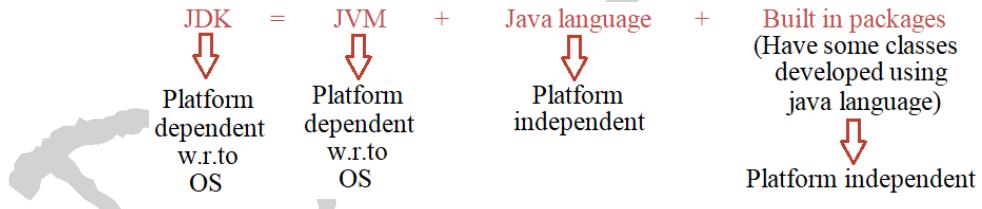
Testing Tools	Test Management Tools	Programming Language
Selenium WD SikuliX Appium : etc	Maven TestNg Jxl/POI Cucumber Junit Gherkin Jenkins : Etc	* Java Javascript HTML : etc

VII. JAVA BASICS

- ❖ Developed by SUN microsystems and take over by ORACLE corporation.
- ❖ Platform independent language.

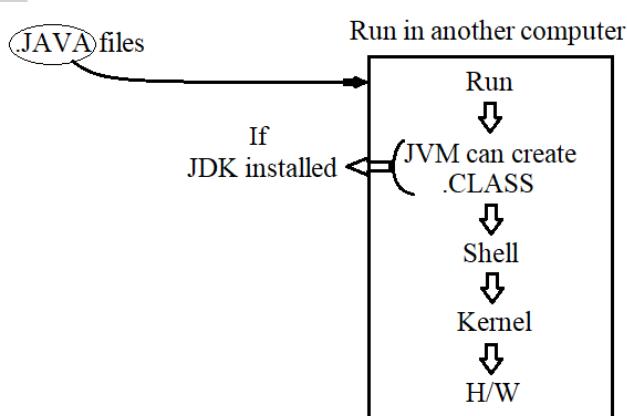


- ❖ Java language is available as JDK (Java Development Kit).

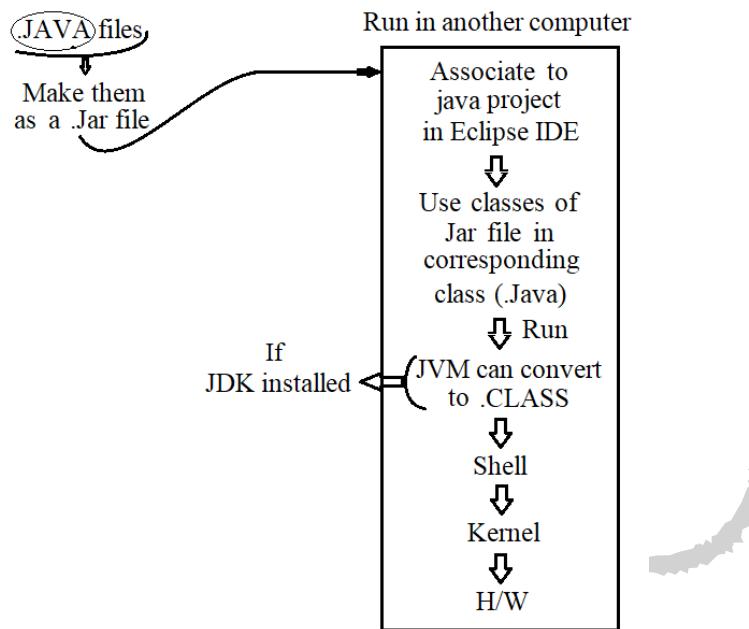


- ❖ We are able to follow any one of two ways to run java programs in another computer.

Way 1:



Way 2:



- ❖ Java language is object oriented program in language (Data abstraction, Data encapsulation, Inheritance & Polymorphism).
- ❖ Java language is case sensitive.

a) Variables in Java:

Variable is a storage to store one value.

`int x=10;`

`x
10`
4 bytes in RAM

b) Constants in Java:

`int x=10;`

=====

x=20;✓
=====

Variable value will be changed

`final int x=10;`

=====

x=20; X
=====

Constant value cannot be changed

c) Data types in Java:

Java language can support 3 data types like Primitive, Derived & User Defined (class).

d) Primitive Data Types:

We can use these data types to create variables and constants. In general one variable or constant can store one value at a time.

In java language, we can get below primitive data types.

Primitive Data Types	Storage (RAM) size in byte (1 byte = 8 bits)	Range of data to store
byte	1 byte	-2^7 to $2^7 - 1$
short	2 bytes	-2^{15} to $2^{15} - 1$
int	4 bytes	-2^{31} to $2^{31} - 1$
long	8 bytes	-2^{63} to $2^{63} - 1$
float	4 bytes (with decimal point)	-2^{31} to $2^{31} - 1$
double	8 bytes (with decimal point)	-2^{63} to $2^{63} - 1$
char	2 bytes	Unicode
boolean	1 bit	0 for false/1 for true

e) Derived Data Types:

Derived data types are useful to create variable which behave like array, object...etc.

Example 1:

```
int x=10;  
int [] y= {10,20,30};
```

Here: x is a variable & y is a variable, behaves like array

Example 2:

```
char x='Q';  
String y="mind q";
```

Here: x is a variable & y is a variable, behaves like object to call some methods.

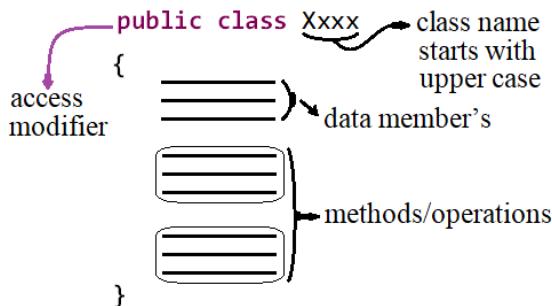
Note:

In java language, we can follow below syntaxes to assign data to variables.

```
int x=10; //no quotes for numeric values  
int [] y= {10,20,30}; //no quotes for numeric values  
char c='Q'; //char value enclosed by single quote  
String s="mind q"; //string value enclosed by double quotes  
boolean b=true; //no quotes for boolean
```

f) User Defined Data Type:

User defined data type is also called as class. Every class is a blue print or template or a structure. Every class consists of data members (same type or different type) and methods to perform operations.



- In every class data members and methods are not mandatory.
- If no one as access modifier, corresponding class can be accessible in current package only by default.
- If access modifier is “**public**”, corresponding class can be accessible in any package in current project.
- To access any “**public**” class in another project, we need to create “**Jar**” file for that class and associate that “**Jar**” file to another project before going to use that class.
- Every class can support different access modifiers for data members and methods.

Access modifier for a class:

public/default (no one specified)

Access modifiers for members and methods:

private	→ call in same class
public	→ call anywhere in current project class
default	→ call anywhere in current package classes
protected	→ call in current class and in child classes of current class

- In general, a public class with public members and methods is famous in Java.

Example:

```
public class Sample1
{
    //Data members
    public int x;
    public float y;
    public char z;
    public String w;
    public boolean b;
```

```

//Methods
public void display()
{
    System.out.println(x);
    System.out.println(y);
    System.out.println(z);
    System.out.println(w);
    System.out.println(b);
}

```

We need to follow below syntax to prepare methods in a class.

```

public returnType method name(arguments)
{
    _____
    _____
    return(z);
}

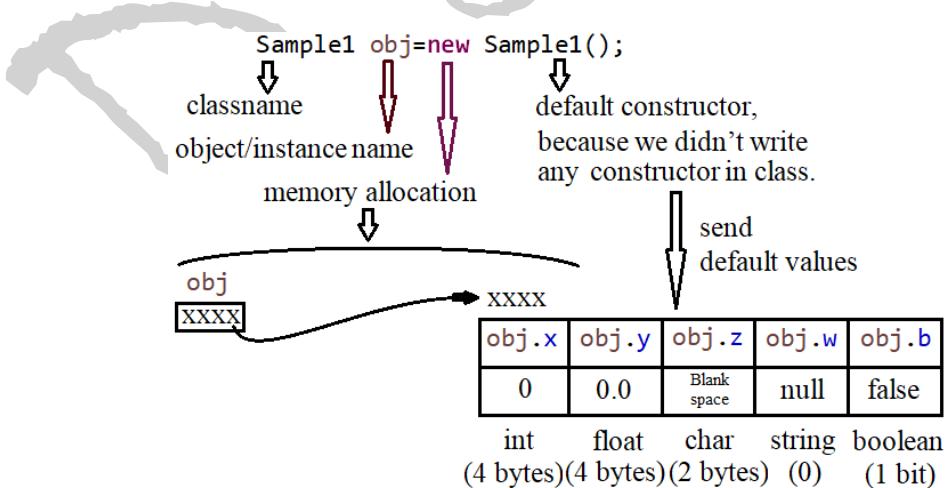
```

Here: If **returntype** is **void** then there will be no return value.

After completion of class creation with data members and methods, we need to create an object for that class.

`ClassName objectname=new ConstructorMethod();`

Example:

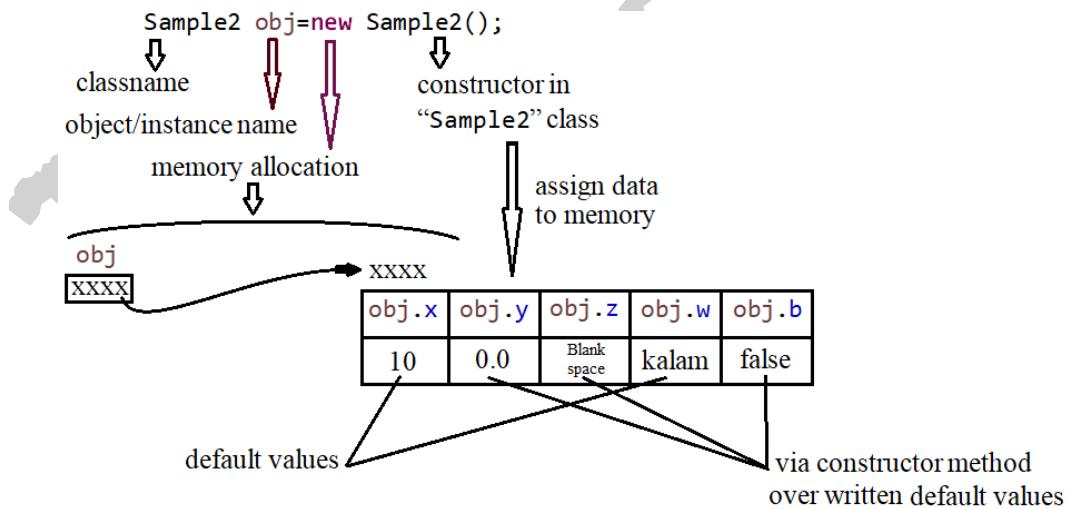


Java language can allow us to create own constructor methods like shown below:

Class:

```
public class Sample2
{
    //Data members
    public int x;
    public float y;
    public char z;
    public String w;
    public boolean b;
    //Methods
    public Sample2()//constructor method
    {
        x=10;
        w="kalam";
    }
    public void display()//operational method
    {
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
        System.out.println(w);
        System.out.println(b);
    }
}
```

We need to create object for above class like shown below:



While writing constructor methods in a class, we need to follow below rules:

- Constructor method name is equal to class name.
- No return type.
- To assign our own data to memory related to corresponding objects.
- Constructor method will be executable automatically when we went to create objects for corresponding class.

- If there is no constructor method in a class, default constructor can run automatically while creating objects to that class.

We are able to provide more than one constructor in a class. Here all constructor method names are equal to class name but those methods are having different number of arguments or different type of arguments called as polymorphism.

Class:

```
public class Sample3
{
    //Data members
    public int x;
    public float y;
    public char z;
    public String w;
    public boolean b;
    //Methods
    public Sample3()//constructor method
    {
        x=10;
        w="kalam";
    }
    public Sample3(int a)//constructor method
    {
        x=a;
        w="kalam";
    }
    public Sample3(float a)//constructor method
    {
        y=a;
        w="kalam";
    }
    public Sample3(int a, String s)//constructor method
    {
        x=a;
        w=s;
    }
    public void display()//operational method
    {
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
        System.out.println(w);
        System.out.println(b);
    }
}
```

In above class, four constructor methods are available. We need to create objects to above class by using any 1 of 4 constructors.

Runner Class:

```
public static void main(String[] args)
{
    Sample3 obj1=new Sample3();
    obj1.display();
    Sample3 obj2=new Sample3(100);
    obj2.display();
    Sample3 obj3=new Sample3((float) 10.0);
    obj3.display();
    Sample3 obj4=new Sample3(100,"steave jobs");
    obj4.display();
}
```

Note 1:

We are able to follow any one of below 2 ways to create an object to a class.

```
Sample3 obj1=new Sample3();
(or)
Sample3 obj1; //object declaration
_____
_____
obj1=new Sample3(); //object creation
```

Note 2:

No need to us single quotes and double quotes for variables and objects.

String x="kalam"; String y=x;	✓	String x="kalam"; String y="x"; System.out.println(y);	✗
----------------------------------	---	--	---

Note 3:

Every statement in java can end with ‘;’

g) Types of classes:

Java language can allow as to create various types of classes like **instance classes, static classes, interfaces, abstract classes, singleton classes & wrapper classes.**

h) Instance classes:

- Instance classes are having data members, constructor methods and operational methods.
- To access members and methods of instance class, we need to create object for that class.

i) Static classes:

Java language can support static members and methods in classes. They are not object oriented because they are class oriented, they can get memory at class level which is common for all the objects of class.

Example 1: (not a static class)

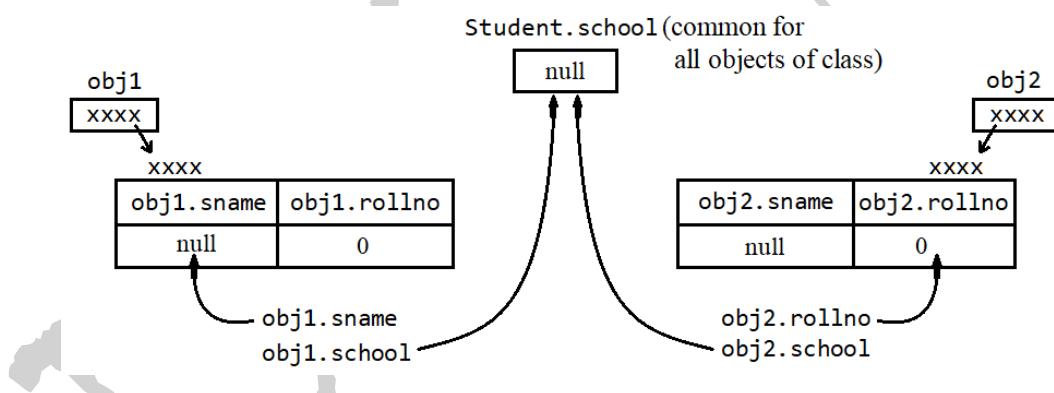
```
public class Student
{
    public static String school;
    public String studentname;
    public int rollnumber;
}
```

Here: Student is instance class because all members and methods are not static.

Runner class:

```
Student obj1=new Student();
Student obj2=new Student();
```

Memory allocation:



When a class have all members and methods as static then that class is called as **static class**.
Here constructor method was not required because all members and methods will be accessible to class name instead of objects.

Example 2:

Static Class:

```
public class Sample4
{
    public static int x=10;
    public static void display()
    {
        System.out.println(x);
    }
}
```

Runner class:

```
public static void main(String[] args)
{
    Sample4.display(); //10
    Sample4.x=20;
    Sample4.display(); //20
}
```

Note:

Static topic is not object oriented because it is class oriented but this concept is useful to save memory by maintaining common memory locations.

j) Interfaces:

Interface is one type of class in java it consists of operational methods declarations without bodies.

Example:**Interface class:**

```
public interface Sample5
{
    public int add(int x, int y);
    public int subtract(int x, int y);
    public int multiply(int x, int y);
    public int divide(int x,int y);
}
```

We need to develop concrete classes like shown below to provide bodies to methods of interface.

Concrete class:

```
public class Sample6 implements Sample5
{
    public int add(int x, int y)
    {
        int z;
        z=x+y;
        return(z);
    }
    public int subtract(int x, int y)
    {
        int z;
        z=x-y;
        return(z);
    }
    public int multiply(int x, int y)
    {
        int z;
        z=x*y;
        return(z);
    }
}
```

```

public int divide(int x, int y)
{
    int z;
    z=x/y;
    return(z);
}
}

```

We are able to use interfaces and corresponding concrete classes in regular programs like shown below

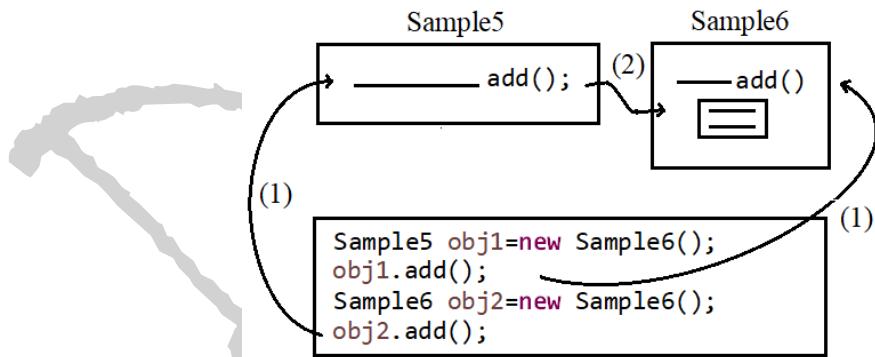
Runner class:

```

public static void main(String[] args)
{
    //create object to interface with help of concrete
    Sample5 obj1=new Sample6();
    int x=obj1.add(30,70);
    System.out.println(x);
    //Create object to concrete class
    Sample6 obj2=new Sample6();
    int y=obj2.add(40,20);
    System.out.println(y);
}

```

Memory allocation:



Note 1:

Constructor concept is not available for interfaces because interface related objects creation depends on corresponding concrete class's constructor methods.

```

WebDriver driver=new ChromeDriver();
↓
Interface
object of interface
↓
memory allocation
↓
constructor method
of corresponding
concrete class

```

Note 2:

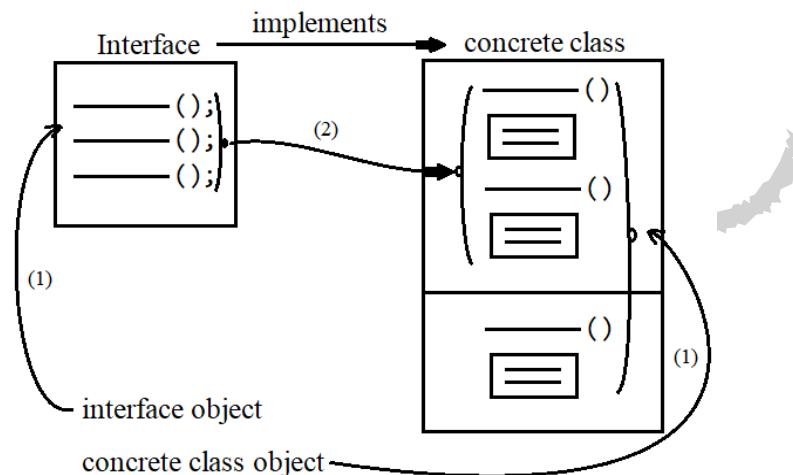
“implements” keyword can create relation in between interface and concrete class.

Note 3:

Any concrete class of an interface need to provide bodies to all methods of interface.

Note 4:

In concrete classes, we are able to provide extra methods if required.



k) Abstract class:

A class consists of methods with bodies to few and without bodies to remaining called as abstract class.

Example:

Abstract class:

```
public abstract class Sample7
{
    public abstract int add(int x, int y);
    public int subtract(int x, int y)
    {
        int z;
        z=x-y;
        return(z);
    }
}
```

In above abstract class add() method is unimplemented method. We need to develop another class to implement unimplemented methods.

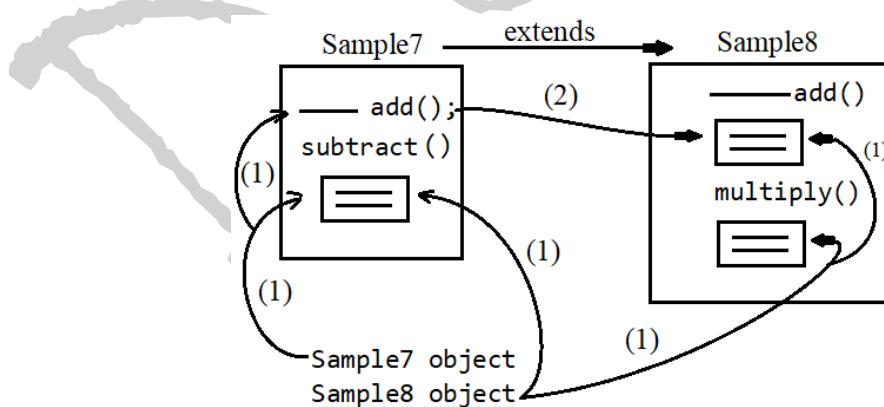
Concrete class:

```
public class Sample8 extends Sample7
{
    public int add(int x, int y)
    {
        int z=x+y;
        return(z);
    }
    public int multiply(int x, int y)
    {
        int z=x*y;
        return(z);
    }
}
```

"Sample8" class provided body to add() method, which is unimplemented in "Sample7". Due to this reason, object of Sample8 can access methods in Sample7 & Sample8 itself.

Runner class:

```
public static void main(String[] args)
{
    Sample7 obj1=new Sample8();
    obj1.add(10,20);
    obj1.subtract(50,36);
    Sample8 obj2=new Sample8();
    obj2.add(30,60);
    obj2.subtract(60,52);
    obj2.multiply(30,6);
}
```



I) Singleton class:

Singleton class is a class that can have only one object at a time. To create a singleton class, we need to make constructor as a private and we need to create a static method. This static method can return object of singleton class.

Example:

Singleton class:

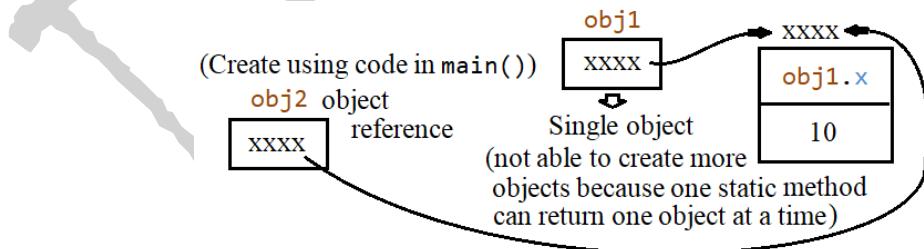
```
public class Sample9
{
    public int x;
    private Sample9()
    {
        x=10;
    }
    public static Sample9 create()
    {
        Sample9 obj1=new Sample9();
        return(obj1);
    }
    public void display()
    {
        System.out.println(x);
    }
}
```

We need to follow below like code to access singleton class members and methods.

Runner class:

```
public static void main(String[] args)
{
    Sample9 obj2=Sample9.create();
    obj2.display();
}
```

Memory allocation:



m) Wrapper classes:

Integer
Float
Long
Double
Character
Boolean
etc.

wrapper classes to initiate primitive data types in java

n) Inheritance (Relation in between instance classes):

- 1) **is a**
- 2) **has a**

1) “is a” relation in between two instance classes is called as inheritance. To create this relation, we need to follow below syntax.

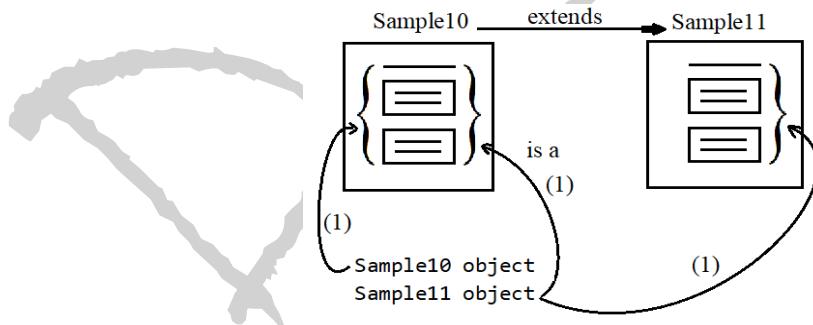
Example: “is a” relation

Instance class 1:

```
public class Sample10 //base or parent
{
    public int x;
    public void display1()
    {
        System.out.println(x);
    }
}
```

Instance class 2:

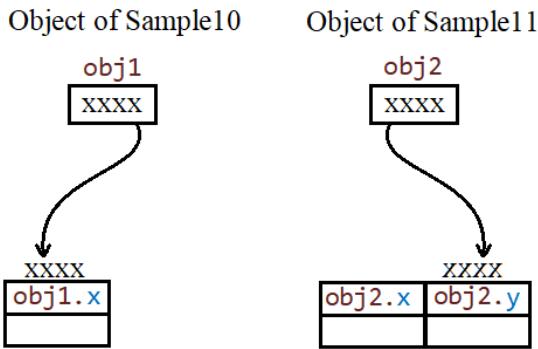
```
public class Sample11 extends Sample10 //child/sub (derived class)
{
    public int y;
    public void display2()
    {
        System.out.println(y);
    }
}
```



Runner class:

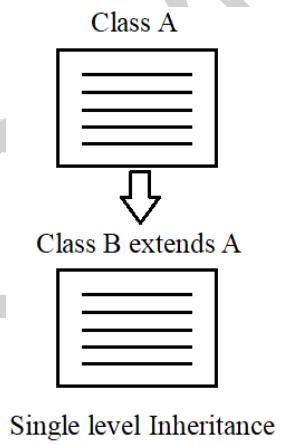
```
public static void main(String[] args)
{
    Sample10 obj1=new Sample10();
    obj1.x=20;
    obj1.display1();
    Sample11 obj2=new Sample11();
    obj2.x=10;
    obj2.y=20;
    obj2.display1();
    obj2.display2();
}
```

Memory allocation:

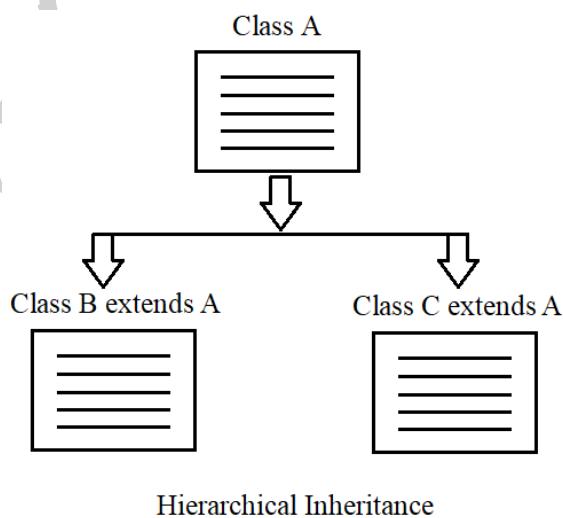


Above like “is a” relation in between instance classes is possible to define in different ways:

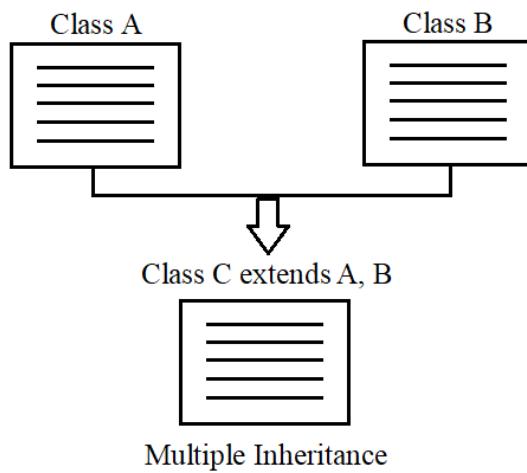
➤ Single level inheritance:



➤ Hierarchical inheritance:



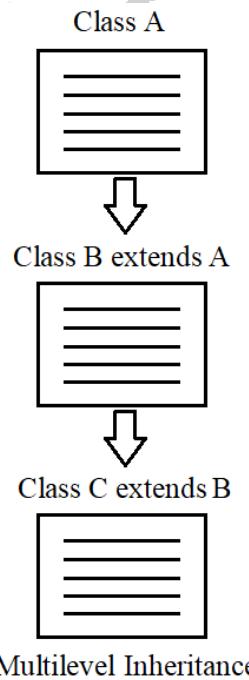
- **Multiple inheritance (Java cannot support):**



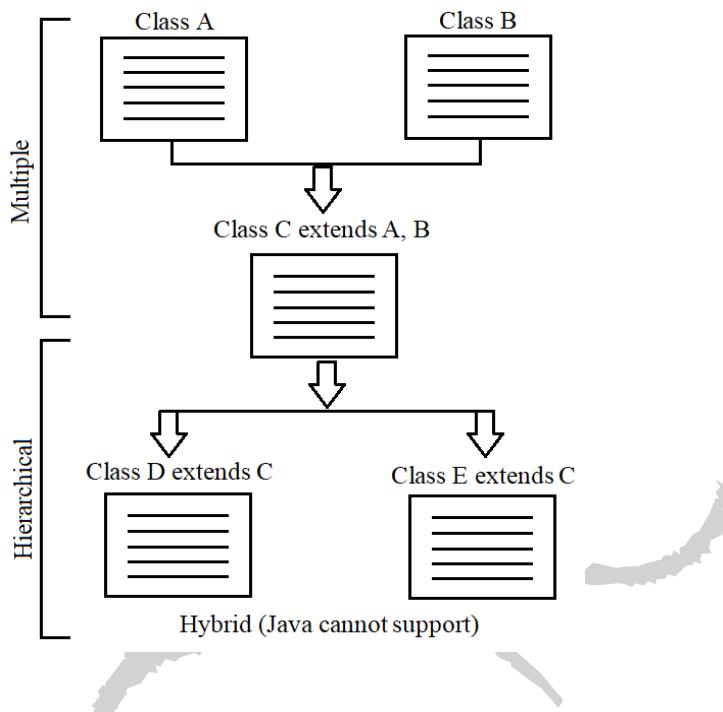
Java **cannot** support in between **instance classes**, but Java **can** support multiple for **interfaces**.

```
public class ChromeDriver implements WebDriver, JavascriptExecutor
{
    _____
}
```

- **Multilevel inheritance:**



- **Hybrid (Java cannot support because multiple is present):**



- 2) Sometimes we can use “has a” relation in between instance classes like shown below.

Example: “has a” relation

Instance class 1	Instance class 2
<pre>public class IOStream { public void println(String x) { _____ } }</pre>	<pre>public class System { public static IOStream out; _____ }</pre>

Runner class:

```
public static void main(String[] args)
{
    //“has a” relation (object of one class is a member in another class)
    System.out.println("xxxx");
}
```

Here:

- **System** has right to call “out” object directly because it is a static member in **System** class. “out” is a object in **IOStream** class so “out” has the right to call members in **IOStream** class.
- **System** doesn’t have the right to call members in **IOStream** class directly so we are going to use “has a” relation.

o) Polymorphism:

- 1) Methods overloading
- 2) Methods overriding

1) Methods overloading:

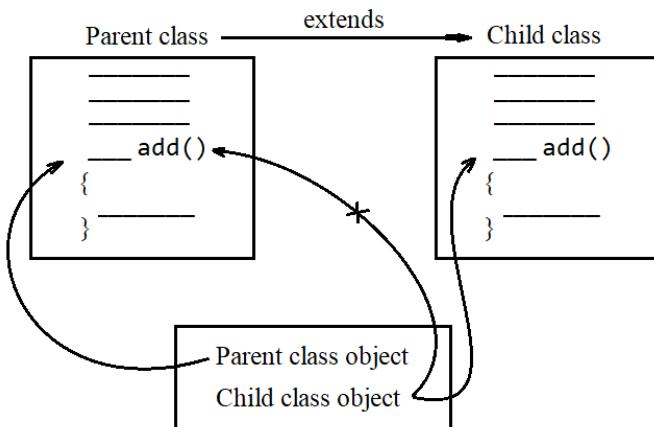
From methods overloading concept, one class is having multiple methods with same name and different number/type of arguments. Here no importance to return type.

Example:

```
public class Sample12
{
    public int add()
    {
        int x,y,z;
        x=10;
        y=20;
        z=x+y;
        return(z);
    }
    public int add(int a)
    {
        int x,y,z;
        x=10;
        y=20;
        z=x+y;
        return(z);
    }
    public int add(int a,int b)
    {
        int x,y,z;
        x=a;
        y=b;
        z=x+y;
        return(z);
    }
    public int add(int a, float b)
    {
        int x,z;
        float y;
        x=a;
        y=b;
        z=(int) (x+y);
        return(z);
    }
}
```

2) Methods overriding:

From method overriding concept parent class and child class are having methods with same name.



Example:

Instance class (parent):

```

public class Sample13
{
    public void display()
    {
        System.out.println("i am in parent");
    }
}
    
```

Instance class (child):

```

public class Sample14 extends Sample13
{
    public void display()
    {
        System.out.println("i am in child");
    }
}
    
```

Runner class:

```

public static void main(String[] args)
{
    Sample13 obj1=new Sample13();
    obj1.display(); //Sample13 display()
    Sample14 obj2=new Sample14();
    //Sample14 display() only accessible because of overriding
    obj2.display();
}
    
```

In above parent and child classes, display() method signature was same due to this reason, parent class object can call display() in parent class, child class object can call display() in child class only.

p) “super” & “this” keywords:

We can use these keywords in child classes to distinguish commonly named members & methods in parent class and child class.

Example:

Instance class (parent):

```
public class Sample15
{
    public int x;
}
```

Instance class (child):

```
public class Sample16 extends Sample15
{
    public int x;
    public void display()
    {
        System.out.println(super.x);
        System.out.println(this.x);
    }
}
```

q) Operators in Java:

Operators in java can classified into 3 categories such as unary operators, binary operators and ternary operators.

➤ Unary operators (work for one operand):

Ex: 1 `int x=10;` x
10 y
-10
`int y=-x; //negation`

Ex: 2 `int x=10;` x
11 y
11
`int y=++x; //pre increment`
by 1

Ex: 3 `int x=10;` x
11 y
10
`int y=x++; //post increment`
by 1

Ex: 4 `int x=10;` x
9 y
9
`int y=--x; //pre decrement`
by 1

Ex: 5 `int x=10;` x
9 y
10
`int y=x--; //post decrement`
by 1

➤ Binary operators (work for two operands):

`+, -, *, /, %` → Used to perform arithmetic operations.

`<, >, <=, >=, !=, ==` → Used to check the condition.

`&&, ||, !` → Used to combine more than one condition.

`=` → Used to assign a value to a variable (assignment operator).

- **Ternary operator (work for 3 operands):**

```
int z=(x<y)?x:y;
```

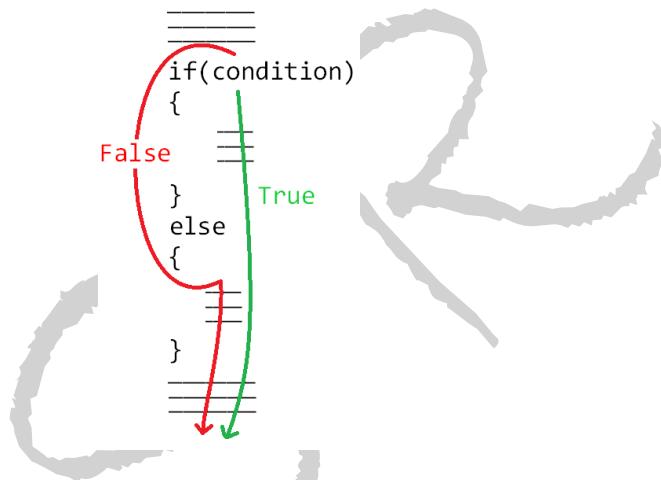
Here: If $x < y$ condition true then x value returns to z

If $x < y$ condition false then y value returns to z

r) Control statements in Java:

- 1) if-else:

We can use this statement to execute a block of code depends on a condition



Example 1:

Take a number and check whether the given number is even or odd.

```
public static void main(String[] args)
{
    //Get data from keyboard
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a word");
    int x=sc.nextInt();
    //Check for even (or) odd
    if(x%2==0)
    {
        System.out.println(x+" is even number");
    }
    else
    {
        System.out.println(x+" is odd number");
    }
}
```

Note:

“+” operator is overloaded operator in java. Any one or both operands are “String”, “+” operator work for concatenation. In remaining situations “+” operator work for addition.

Ex 1: `int x=10;
int y=20;
int z=x+y;`

Here: “+” operator works for addition because both operands are numeric.

Ex 2: `int x=10;
char y='q';
int z=x+y;`

Here also “+” operator works for addition.

Ex 3: `char x='Q';
char y='q';
int z=x+y;`

Here also “+” operator works for addition by taking ASCII values.

Ex 4: `String x="mind";
char y='q';
String z=x+y;`

Here: “+” operator works for concatenation because String is present.

2) *if-else-if statement:

To check one condition, we can use if-else statement. To check multiple conditions, we can use if-else-if statement.

```
if(condition1)  
{  
    _____  
}  
_____  
else if(condition2)  
{  
    _____  
}  
:  
:  
else  
{  
    _____  
}
```

→ else block can run when given all conditions were false.

Example:

- Launch way2sms site
- Do login by filling mobile number & password
- If mobile number is blank // “Enter your mobile number” message will be displayed.

- If mobile number size is <10 digit // “Enter valid mobile number” message will be displayed.
- If password is blank // “Enter password” message will be displayed.
- If mobile number or password is invalid w.r.to DB (Data Base) // “Incorrect number or password! Try Again.” message will be displayed.
- If mobile number and password are valid w.r.to DB (Data Base) // “sendSMS” message will be displayed.
- Close site

```

public static void main(String[] args) throws Exception
{
//Get test data from keyboard
Scanner sc=new Scanner(System.in);
System.out.println("Enter mobile number");
String mbno=sc.nextLine();
System.out.println("Enter mobile number criteria");
String mbnoc=sc.nextLine();
System.out.println("Enter password");
String pswd=sc.nextLine();
System.out.println("Enter password criteria");
String pswdc=sc.nextLine();
//Launch site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.way2sms.com");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("mobileNo")));
//Do login
driver.findElement(By.name("mobileNo")).sendKeys(mbno);
driver.findElement(By.name("password")).sendKeys(pswd);
driver.findElement(By.xpath("//button[contains(text(),'Login')])[1]").click();
w.until(temp->driver.executeScript("return document.readyState;").equals("complete"));
Thread.sleep(10000);
//validations (Observations)
if(mbno.length()==0 && driver.findElement(By.xpath("//b[text()='Enter your
mobile number'])").isDisplayed())
{
    System.out.println("Blank mobile number test passed");
}
else if(mbno.length()<10 && driver.findElement(By.xpath("//b[text()='Enter
valid mobile number'])").isDisplayed())
{
    System.out.println("Wrong size mobile number test passed");
}
else if(pswd.length()==0 && driver.findElement(By.xpath("//b[text()='Enter
password']))[2]").isDisplayed())
{
    System.out.println("Blank password test passed");
}

```

```

else if(mbnoc.equals("invalid") &&
driver.findElement(By.xpath("//b[contains(text(),'Try Again')]")).isDisplayed())
{
    System.out.println("Invalid mobile number test passed");
}
else if(pswdc.equals("invalid") &&
driver.findElement(By.xpath("//b[contains(text(),'Try Again')]")).isDisplayed())
{
    System.out.println("Invalid password test passed");
}
else if(mbnoc.equals("valid") && pswdc.equals("valid") &&
driver.findElement(By.xpath("//div[text()='SendSMS']")).isDisplayed())
{
    System.out.println("Valid data test passed");
}
else
{
    Date d=new Date();
    SimpleDateFormat s=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
    String x=s.format(d)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(x);
    FileHandler.copy(src,dest);
    System.out.println("Login test failed");
}
//close site
driver.close();
}

```

Note 1:

In java every executable statement can end with “;”. Don’t use “;” for conditional statements & loops because they are related to further code.

Ex:

<code>if(condition)</code> <code>{</code> <code>}</code>	<code>=</code>	<code>for(int i=1;i<=10;i++)</code> <code>{</code> <code>}</code>	<code>=</code>
--	----------------	--	----------------

Note 2:

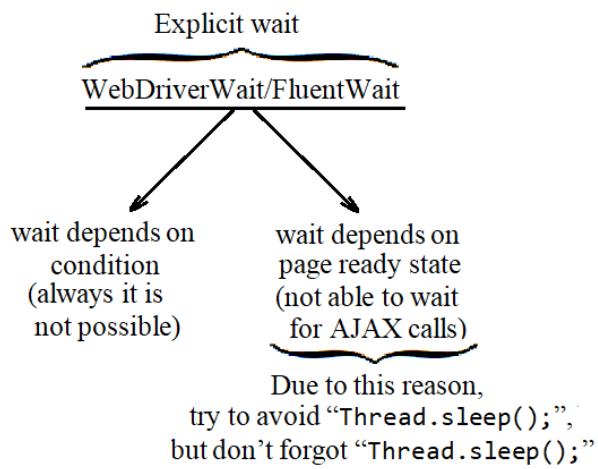
To write single line comment we can use “//” in java.

To write multi line comments, we can use `/*-----*/`

Note 3:

To compare numeric (int, float, long, double, short, byte), characters & Boolean values, we can use “==”. To compare “String” values, we need to use “equals()” method.

Note 4:



3) Nested if:

"if" in "if" is called as nested if.

```
if(condition1)
{
    _____
}
_____
if(condition2)
{
    _____
}
```

- Test cases one after other } if-else-if
- Dependant test cases } nested if

Example:

- Launch Gmail
- Enter user id and click next
- If user id blank // “Enter an email or phone number” message will be displayed.
- If user id is invalid // “Couldn’t find your Google Account” message will be displayed.
- If user id is valid, “password” element will be displayed.
 - Enter password and click next
 - If password is blank // “Enter a password” message will be displayed.
 - If password is invalid // “Wrong password” message will be displayed.
 - If password is valid // “Compose” button will be displayed.
- Close site

```

public class Nestedif
{
    //Static class for screenshot with file name as data and time
    public static String screenshot(ChromeDriver driver) throws Exception
    {
        Date d=new Date();
        SimpleDateFormat s=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
        String x=s.format(d)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(x);
        FileHandler.copy(src,dest);
        return(x);
    }
    public static void main(String[] args) throws Exception
    {
        //Create a html results file
        ExtentReports er=new ExtentReports("GmailLoginResults.html",false);
        ExtentTest et=er.startTest("Gmail Login Testing");
        //Get test data from keyboard
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter user id");
        String uid=sc.nextLine();
        System.out.println("Enter user id criteria");
        String uidc=sc.nextLine();
        String pswd="";
        String pswdc="";
        if(uidc.equalsIgnoreCase("valid"))
        {
            System.out.println("Enter password");
            pswd=sc.nextLine();
            System.out.println("Enter password criteria");
            pswdc=sc.nextLine();
        }
        //Launch site
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        ChromeDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://www.gmail.com");
        WebDriverWait w=new WebDriverWait(driver,30);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                       .name("identifier")));
        //user id testing
        driver.findElement(By.name("identifier")).sendKeys(uid);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                       .xpath("//span[text()='Next']")));
        driver.findElement(By.xpath("//span[text()='Next']")).click();
        if(uid.length()==0)
        {
            try
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                       .xpath("//div[contains(text(),'Enter an email')]")));
                et.log(LogStatus.PASS,"Blank user id test passed");
            }

```

```

        catch(Exception ex)
        {
            String x=Nestedif.screenshot(driver);
            et.log(LogStatus.FAIL,"Blank user id test failed"+ex
                .getMessage()+et.addScreenCapture(x));
        }
    }
    else if(uidc.equalsIgnoreCase("invalid"))
    {
        try
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//div[contains(text(),'Google Account')]")));
            et.log(LogStatus.PASS,"Invalid user id test passed");
        }
        catch(Exception ex)
        {
            String x=Nestedif.screenshot(driver);
            et.log(LogStatus.FAIL,"Invalid user id test failed"+ex
                .getMessage()+et.addScreenCapture(x));
        }
    }
    else
    {
        try
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .name("password")));
            et.log(LogStatus.PASS,"Valid user id test passed");
            //Password testing
            driver.findElement(By.name("password")).sendKeys(pswd);
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//span[text()='Next']")));
            driver.findElement(By.xpath("//span[text()='Next']")).click();
            if(pswd.length()==0)
            {
                try
                {
                    w.until(ExpectedConditions
                        .visibilityOfElementLocated(By
                        .xpath("//div[contains(text(),'Enter a password')]")));
                    et.log(LogStatus.PASS,
                        "Blank password test passed");
                }
                catch(Exception ex)
                {
                    String x=Nestedif.screenshot(driver);
                    et.log(LogStatus.FAIL,
                        "Blank password test failed"+ex
                        .getMessage()+et.addScreenCapture(x));
                }
            }
        }
    }
}

```

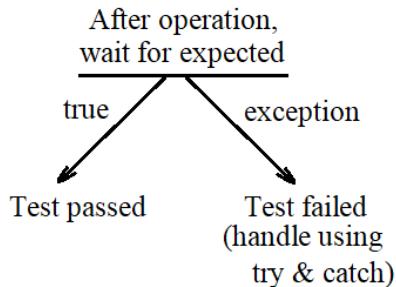
```

        else if(pswdc.equalsIgnoreCase("invalid"))
        {
            try
            {
                w.until(ExpectedConditions
                    .visibilityOfElementLocated(By
                        .xpath("//div[contains(text(),'Wrong password')]")));
                et.log(LogStatus.PASS,
                    "Invalid password test passed");
            }
            catch(Exception ex)
            {
                String x=Nestedif.screenshot(driver);
                et.log(LogStatus.FAIL,
                    "Invalid password test failed"+ex
                        .getMessage()+et.addScreenCapture(x));
            }
        }
        else
        {
            try
            {
                w.until(ExpectedConditions
                    .visibilityOfElementLocated(By
                        .xpath("//div[contains(text(),'Compose')]")));
                String x=Nestedif.screenshot(driver);
                et.log(LogStatus.PASS,
                    "Valid user id test passed"+et
                        .addScreenCapture(x));
            }
            catch(Exception ex)
            {
                String x=Nestedif.screenshot(driver);
                et.log(LogStatus.FAIL,
                    "Valid password test failed"+ex
                        .getMessage()+et.addScreenCapture(x));
            }
        }
        catch(Exception ex)
        {
            String x=Nestedif.screenshot(driver);
            et.log(LogStatus.FAIL,"Valid user id test failed"+ex
                .getMessage()+et.addScreenCapture(x));
        }
    }
    //close site
    driver.close();
    //save results
    er.endTest(et);
    er.flush();
}
}

```

Note 1:

In above test script, we followed a common logic for synchronization and validations.



Note 2:

To improve code reusability, we are able to maintain more than one method along with main() method in test class (runner classes)

4) Switch case statement:

To execute a block of code depends on a value instead of condition.

Example 1:

```
public class Switchcase1
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter day number");
        int x=sc.nextInt();
        switch(x)
        {
            case 1:
                System.out.println("Sunday");
                break;
            case 2:
                System.out.println("Monday");
                break;
            case 3:
                System.out.println("Tuesday");
                break;
            case 4:
                System.out.println("Wednesday");
                break;
            case 5:
                System.out.println("Thursday");
                break;
            case 6:
                System.out.println("Friday");
                break;
            case 7:
                System.out.println("Saturday");
                break;
        }
    }
}
```

```

        default:
            System.out.println("Wrong day number");
            break;
    }
}
}

```

Example 2:

```

public class Switchcase2
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter marital status(y/n)");
        String l=sc.nextLine();
        char x=l.charAt(0); //0 means first character in string
        switch(x)
        {
            case 'y':
                System.out.println("Married");
                break;
            case 'n':
                System.out.println("Unmarried");
                break;
            default:
                System.out.println("Wrong answer");
                break;
        }
    }
}

```

From the above switch case, character values are enclosed by single quotes ' ' & they are case sensitive.

Example 3:

```

public class Switchcase3
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter credit card type");
        String x=sc.nextLine();
        switch(x)
        {
            case "visa":
                System.out.println("valid card");
                break;
            case "rupay":
                System.out.println("valid card");
                break;
        }
    }
}

```

```

        case "master":
            System.out.println("valid card");
            break;
        default:
            System.out.println("Wrong type");
            break;
    }
}

```

From the above switch case, character values are enclosed by single quotes ' ' & they are case sensitive.

s) “Scanner” class in JDK:

We can use this class to read data from keyboard.

```

Scanner sc=new Scanner(System.in);
//get "int" value from keyboard
int x=sc.nextInt();
//get "float" value from keyboard
float y=sc.nextFloat();
//get "string" value from keyboard
String z=sc.nextLine();

```

Note 1:

While using scanner class to get string value from keyboard, previously operated enter key value can store for that string.

```

Scanner sc=new Scanner(System.in);
int x=sc.nextInt();
String y=sc.nextLine();

```

While running above code, if data will be:

1234 ↵
x y
(Enter key as string)

To prevent this, we can use type conversion:

```

Scanner sc=new Scanner(System.in);
int x=Integer.parseInt(sc.nextLine());
String y=sc.nextLine();

```

t) Type casting, type conversion and java generics:

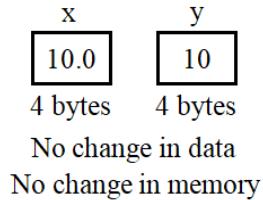
✓ Type casting:

We can use type casting in possible conditions to consider one type of value as another type.

```

float x=10; //x is float
int y=(int) x; //y is type casting
//in further code x is float

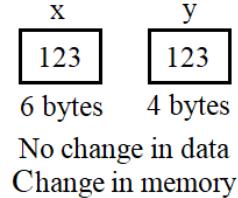
```



✓ **Type conversion:**

We can use type conversion in possible conditions to convert one type of data into another type permanently.

```
String x="123";
int y=Integer.parseInt(x);
```



✓ **Java Generics:**

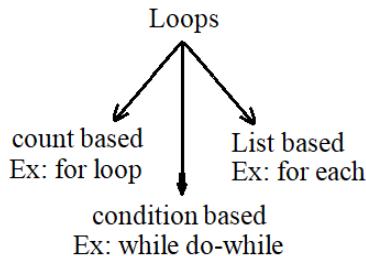
Java language can support generics concept. While creating collections of elements (List/Set/Iterator) and collection of data (ArrayList/HashMap), we can use generics concept like shown below:

```
List<WebElement> l=driver.findElement(By.tagName("a"));
ArrayList<String> a=new ArrayList<String>(driver.getWindowHandles());
```

Here: a indicates anchor tag (link).

u) Loops in java:

To execute a block of code more than one time, we can use loops.



Ex: 1

```
List<WebElement> l=driver.findElement(By.tagName("a"));
int x=l.size(); //get count
for(int i=0;i<x;i++)
{
  _____
}
```

Ex: 2

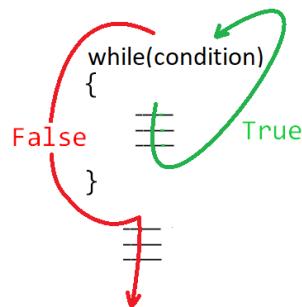
```
while(2>1) //infinite loop
{
    _____
    try
    {
        if(driver.findElement(By.xpath("//span[text()='Next']"))
            .isDisplayed())
        {
            driver.findElement(By.xpath("//span[text()='Next']"))
                .click();
        }
    }
    catch(Exception ex)
    {
        break; //terminate from infinite loop
    }
}
```

Ex: 3

```
List<WebElement> l=driver.findElement(By.tagName("a"));
for(WebElement e:l)
{
    _____
}
```

1) while loop:

To execute a block of code more than one time as long as given condition was true.



Example 1:

Reverse the given number.

```
public static void main(String[] args)
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number");
```

```

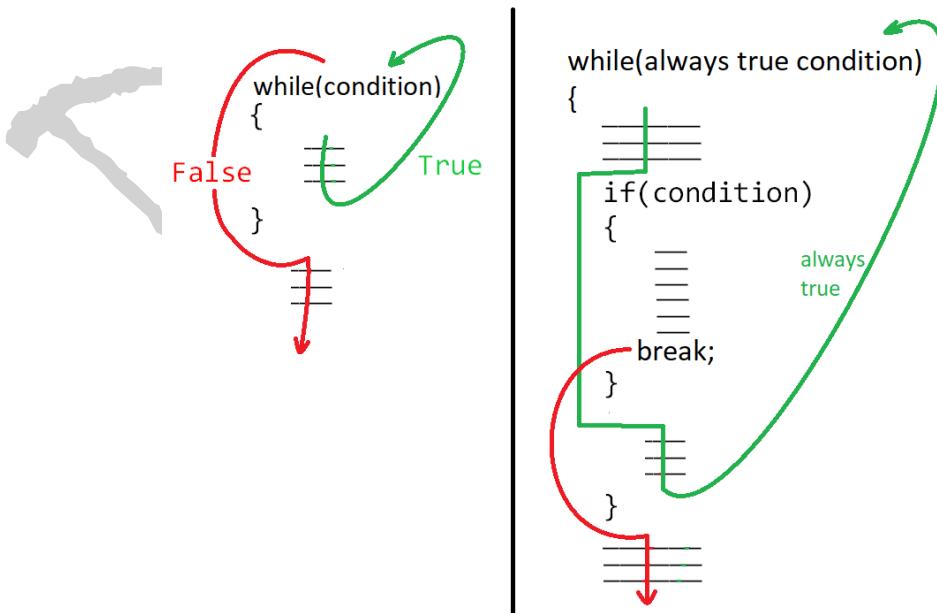
int x=sc.nextInt();
int y=0;
while(x!=0)
{
    int d=x%10;
    y=y*10+d;
    x=x/10;
}
System.out.println("Reverse number is : "+y);
}

```

x=723	x=72	x=7
y=0	y=3	y=32
d=x%10;	d=x%10;	d=x%10;
d=723%10;	d=72%10;	d=7%10;
d=3;	d=2;	d=7;
y=y*10+d;	y=y*10+d;	y=y*10+d;
y=0*10+3;	y=3*10+2;	y=32*10+7;
y=3;	y=32;	y=327;
x=(int)x/10;	x=(int)x/10;	x=(int)x/10;
x=723/10=72.3	x=72/10=7.2	x=7/10=0.7
x=72	x=7	x=0

Note:

In general we can use while loop to execute a block of code depends on condition in below ways.



Example 2:

- Launch Google site
- Enter a word to search
- Validate each result page title, which consists of search word.
- Close site

```
//make results file
ExtentReports er=new ExtentReports("googlenext.html",false);
ExtentTest et=er.startTest("Click until last page");
Scanner sc=new Scanner(System.in);
System.out.println("Enter a word in google");
String x=sc.nextLine();
//Launch Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("http://www.google.co.in");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("q")));
driver.findElement(By.name("q")).sendKeys(x,Keys.ENTER);
while(2>1)
{
    w.until(temp->driver.executeScript("return document.readyState;")
        .equals("complete"));
    if(!x.contains(x))
    {
        //Get Screenshot
        Date d=new Date();
        SimpleDateFormat s=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
        String y=s.format(d)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(y);
        FileHandler.copy(src,dest);
        et.log(LogStatus.FAIL,"google title test failed"+et
            .addScreenCapture(y));
        //terminate from loop
        break;
    }
    //scroll down to last
    driver.executeScript("window.scrollTo(0,document.body.scrollHeight);");
    //go to next page
    try
    {
        if(driver.findElement(By.xpath("//span[text()='Next']"))
            .isDisplayed())
        {
            driver.findElement(By.xpath("//span[text()='Next']"))
                .click();
        }
    }
}
```

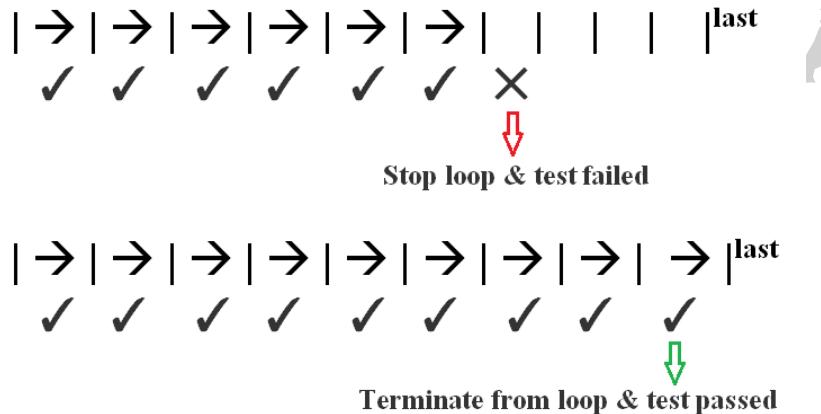
```

        catch(Exception ex)
        {
            et.log(LogStatus.PASS, "Google test passed");
            break; //terminate from infinite loop
        }
    }
//close site
driver.close();
//Save results
er.endTest(et);
er.flush();

```

Note 1:

Infinite loop in above code can give termination in any one of two possibilities.



Note 2:

Validations			
Check one condition one time Ex: if else	Check multiple conditions one after other Ex: if else if	Check one condition multiple times Ex: loops & if else	* Check multiple conditions multiple times Ex: loops & if else if

Note 3:

Parameterization means the replacement of constant value with a variable in code.

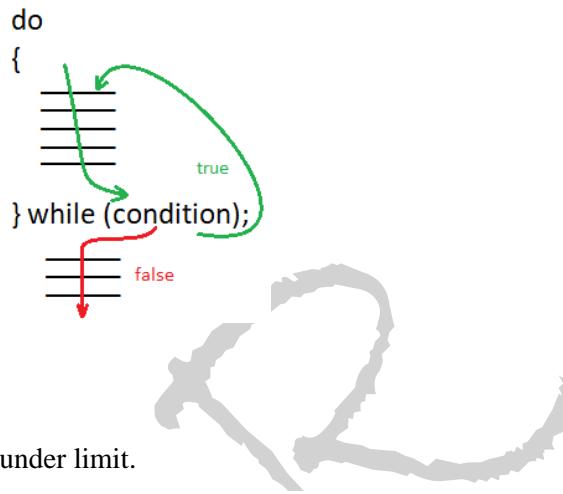
```

driver.findElement(locator).sendKeys("mindq");
driver.findElement(locator).sendKeys(x); //parameterized code

```

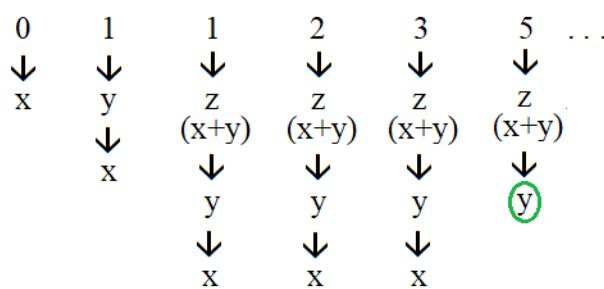
2) do-while loop:

To execute a block of code as long as given condition was true. We can use this loop but in this loop block of code can be executed at least one time when given condition was false.



Example 1:

Display Fibonacci series under limit.



```
public class FibonacciSeries
{
    public static void main(String[] args) throws Exception
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter limit");
        int l=sc.nextInt();
        int x=0;
        int y=1;
        System.out.print(x+" "+y+" ");
        do
        {
            int z=x+y;
            System.out.print(z+" ");
            x=y;
            y=z;
        }while((x+y)<l);
    }
}
```

Example 2:

- Launch Gmail site
- Do login with valid data
- Count number of mails in mail box via pagination and compare with visible count
- Do logout
- Close site

```
//Save results in html using extent reports
ExtentReports er=new ExtentReports("gmails count.html",false);
ExtentTest et=er.startTest("Gmail mails count testing");
//Launch Gmail Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.gmail.com");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("identifier")));
//Do login with valid data
driver.findElement(By.name("identifier")).sendKeys("xxxx");
w.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[text()='Next']")));
driver.findElement(By.xpath("//span[text()='Next']")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("password")));
driver.findElement(By.name("password")).sendKeys("xxxx");
w.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[text()='Next']")));
driver.findElement(By.xpath("//span[text()='Next']")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//table)[4]")));
//close notification dialog
driver.findElement(By.xpath("//div[@class='bBe']")).click();
//get expected count of mails in page
int exmails=0;
do
{
    //get count of mails in current page
    int x=driver.findElement(By.xpath("//table)[4]/tbody"))
                    .findElements(By.tagName("tr")).size();
    //add count of mails in current page to total count
    exmails=exmails+x;
    //goto next page
    try
    {
        if(driver.findElement(By.xpath("//div[@data-tooltip='Older']"))
                .getAttribute("aria-disabled").equals("true"))
        {
            break;
        }
    }
    catch(Exception ex)
    {
        driver.findElement(By.xpath("//div[@data-tooltip='Older']")).click();
        //wait for loading...
        Thread.sleep(2000);
```

```

        //if displayed
        if(driver.findElement(By.xpath("//span[text()='Loading...']"))
                .isDisplayed())
        {
            //wait until invisible
            w.until(ExpectedConditions.invisibilityOfElementLocated(By
                    .xpath("//span[text()='Loading...']")));
        }
    }
}while(2>1); //infinite loop
//get actual count from page
String temp=driver.findElement(By
    .xpath("//div[@aria-label='Newer']/preceding::span[1]").getText();
//if "," character present replace with null (nothing but removing)
String x=temp.replaceAll(",","");
int acmails=Integer.parseInt(x);
System.out.println(exmails+ " "+acmails);
if(acmails==exmails)//validation
{
    et.log(LogStatus.PASS, "Gmail mails counting test passed");
}
else
{
    //Get Screenshot
    Date d=new Date();
    SimpleDateFormat s=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
    String y=s.format(d)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(y);
    FileHandler.copy(src,dest);
    et.log(LogStatus.FAIL,"Gmail mails counting test failed",et
        .addScreenCapture(y));
}
//Do logout
driver.findElement(By.xpath("//*[contains(@class,'gbii')]").click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.LinkText("Sign out")));
driver.findElement(By.LinkText("Sign out")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("password")));
//Close site
driver.close();
//Save results
er.endTest(et);
er.flush();

```

Note 1:

Pagination means navigating (moving) from current page to next page until last page.

Note 2:

“String” is default data type in java.

3) for loop:

We can use this loop to execute a block of code for specified number of times.

Example 1:

```
for(int i=1;i<=10;i++)
{
    System.out.println("Dinesh Reddy");
}
//10 times
```

Example 2:

```
for(int i=0;i<10;i++)
{
    System.out.println("Dinesh Reddy");
}
//10 times
```

Example 3:

```
for(int i=10;i>0;i--)
{
    System.out.println("Dinesh Reddy");
}
//10 times
```

Example 4:

```
for(;;) //no initialization, no increment/decrement && no condition
{
    System.out.println("mom & dad i love you");
}
//infinite loop
```

Example 5:

- Launch Gmail site
- Do login with valid data
- Get count of unread mails via pagination
- Get visible count of unread mails
- Compare both counts
- Do logout
- Close site

```

//Save results in html using extent reports
ExtentReports er=new ExtentReports("gmails count.html",false);
ExtentTest et=er.startTest("Gmail unread mails count testing");
//Launch Gmail Site
System.setProperty("webdriver.chrome.driver","D:\\DineshReddy\\chromedriver.exe");
ChromeDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.gmail.com");
WebDriverWait w=new WebDriverWait(driver,30);
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("identifier")));
//Do login with valid data
driver.findElement(By.name("identifier")).sendKeys("xxxx");
w.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[text()='Next']")));
driver.findElement(By.xpath("//span[text()='Next']")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("password")));
driver.findElement(By.name("password")).sendKeys("xxxx");
w.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[text()='Next']")));
driver.findElement(By.xpath("//span[text()='Next']")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//table)[4]")));
//close notification dialog
driver.findElement(By.xpath("//div[@class='bBe']")).click();
//get expected count of unread mails in page
int exmails=0;
do
{
    //get count of unread mails in current page
    int x=driver.findElement(By.xpath("//table)[4]/tbody"))
                    .findElements(By.tagName("tr")).size();
    for(int i=1;i<=x;i++)
    {
        WebElement e=driver.findElement(By
            .xpath("//table)[4]/tbody/tr["+i+"]/td[5]/div[1]"));
        driver.executeScript("var v=arguments[0].textContent; window.alert(v);", e);
        String z=driver.switchTo().alert().getText();
        if(z.contains("unread,"))
        {
            exmails=exmails+1;
        }
        driver.switchTo().alert().dismiss();
    }
    //goto next page
    try
    {
        if(driver.findElement(By.xpath("//div[@data-tooltip='Older']"))
            .getAttribute("aria-disabled").equals("true"))
        {
            break;
        }
    }
    catch(Exception ex)
    {
        driver.findElement(By.xpath("//div[@data-tooltip='Older']")).click();
        //wait for loading...
        Thread.sleep(2000);
    }
}

```

```

//if displayed
if(driver.findElement(By.xpath("//span[text()='Loading...']"))
    .isDisplayed())
{
    //wait until invisible
    w.until(ExpectedConditions.invisibilityOfElementLocated(By
        .xpath("//span[text()='Loading...']")));
}
}

}while(2>1); //infinite loop
//get actual count from page
String a=driver.findElement(By.xpath("//*[contains(@data-tooltip, 'Inbox')]"))
    .getAttribute("data-tooltip"));

String b=a.substring(7, a.length()-1);
//if "," character present, replace with null (nothing but removing)
String x=b.replaceAll(",","");
int acmails=Integer.parseInt(x);
System.out.println(exmails+ " "+acmails);
if(acmails==exmails)//validation
{
    et.log(LogStatus.PASS, "Gmail unread mails counting test passed");
}
else
{
    //Get Screenshot
    Date d=new Date();
    SimpleDateFormat s=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
    String y=s.format(d)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(y);
    FileHandler.copy(src,dest);
    et.log(LogStatus.FAIL,"Gmail unread mails counting test failed",et
        .addScreenCapture(y));
}
//Do logout
driver.findElement(By.xpath("//*[contains(@class, 'gbii')]").click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.LinkText("Sign out")));
driver.findElement(By.LinkText("Sign out")).click();
w.until(ExpectedConditions.visibilityOfElementLocated(By.name("password")));
//Close site
driver.close();
//Save results
er.endTest(et);
er.flush();

```

Note 1:

While automating few elements we need to parameterize locators.

```

driver.findElement(By.name("user id")).sendKeys(x); //data parameterization
driver.findElement(By.name(x)).sendKeys("dinesh"); //locator parameterization

```

Note 2:

Sometimes required plain text is in source code but not visible in page as per developer logic. To get that required plain text we can use java script language code.

```
driver.executeScript("var x=document.getElementById('xxxx').textContent; alert(x);");  
                      (or)  
WebElement e=driver.findElement(By.id("xxxx"));  
driver.executeScript("var v=arguments[0].textContent; window.alert(v);", e);
```

Note 3: (Selenium Exceptions)

- **WebDriverException:** It can raise when given statements syntax was wrong.
- **NoSuchElementException:** It can raise when given element locator was wrong.
- **ElementNotVisibleException:** It can raise when element is located in source code but not visible (hidden elements) in a page.
- **ElementNotSelectableException:** Although an element is present in the DOM, it may be disabled (cannot be clicked/selected).
- **NoSuchFrameException:** WebDriver is switching to an invalid frame, which is not available.
- **NoAlertPresentException:** WebDriver is switching to an invalid alert, which is not available.
- **NoSuchWindowException:** WebDriver is switching to an invalid window, which is not available.
- **StaleElementReferenceException:** The referenced element is no longer present on the DOM page (reference to an element is now Stale).

Ex: The Element belongs to a different frame than the current one OR the user has navigated away to another page.

- **SessionNotFoundException:** The WebDriver is performing the action immediately after 'quitting' the browser.
- **TimeoutException:** The command did not complete in enough time.

Ex: The element didn't display in the specified time. Encountered when working with waits.

Example 6:

Take a number and check it as prime or not. (x divided by 1 & x, not divided by 2 to x-1).

```
public static void main(String[] args)  
{  
    Scanner sc=new Scanner(System.in);  
    System.out.println("Enter a Number");  
    int x=sc.nextInt();  
    int flag=0;
```

```

for(int i=2;i<x;i++)
{
    if(x%i==0)
    {
        flag=1;
        break;
    }
}
if(flag==0)
{
    System.out.println(x+" is a prime number");
}
else
{
    System.out.println(x+" is not a prime number");
}
}

```

v) “String” class methods:

While automating website pages using Selenium Web Driver Java, we can get output values as string values from webpages. We need to perform alterations on those values using “String” class methods before going to validations. We can get below methods in String class.

length():

We can use this method to get number of characters in given String.

```

String x="Dinesh Reddy";
int y=x.length();
System.out.println(y); //12
char z=x.charAt(7);
System.out.println(z);

```

charAt():

We can use this method to get character from specified position of given String.

```

String x="Dinesh Reddy";
char z=x.charAt(7);
System.out.println(z); //R

```

Example 1:

Write java code for below scenario.

Input → 949194

Output → nine four nine one nine four

```

public static void main(String[] args)
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a number a string");
}

```

```
String x=sc.nextLine();
for(int i=0;i<x.length();i++)
{
    char y=x.charAt(i);
    switch(y)
    {
        case '0':
            System.out.print("zero ");
            break;
        case '1':
            System.out.print("one ");
            break;
        case '2':
            System.out.print("two ");
            break;
        case '3':
            System.out.print("three ");
            break;
        case '4':
            System.out.print("four ");
            break;
        case '5':
            System.out.print("five ");
            break;
        case '6':
            System.out.print("six ");
            break;
        case '7':
            System.out.print("seven ");
            break;
        case '8':
            System.out.print("eight ");
            break;
        case '9':
            System.out.print("nine ");
            break;
        default:
            System.out.print(" ");
            break;
    }
}
```

Example 2:

```
String reverse

public static void main(String[] args)
{
    Scanner sc=new Scanner(System.in);
    System.out.println("enter a string");
    String x=sc.nextLine();
    String z="";
```

```
for(int i=x.length()-1;i>=0;i--)
{
    char y=x.charAt(i);
    z=z+y;
}
System.out.println(z);
if(x.equalsIgnoreCase(z))
{
    System.out.println(z+" is a palindrome");
}
else
{
    System.out.println(z+" is not a palindrome");
}
}
```

equals():

We can use this method to compare two strings for equal (it is a method by considering case sensitive of data).

```
public static void main(String[] args)
{
    String x="Mindq";
    String y="mindq";
    if(x.equals(y))
    {
        System.out.println("Same word");
    }
    else
    {
        System.out.println("Not same word"); //answer
    }
}
```

equalsIgnoreCase():

We can use this method to compare two strings for equal (without considering case).

```
public static void main(String[] args)
{
    String x="Mindq";
    String y="mindq";
    if(x.equalsIgnoreCase(y))
    {
        System.out.println("Same word"); //answer
    }
    else
    {
        System.out.println("Not same word");
    }
}
```

Note:

If data type is numeric like int, float, char or boolean...etc, we can use “==” for comparison.

Substring():

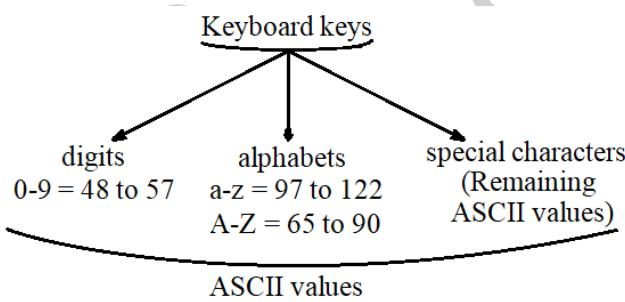
We can use this method to get part of value from String.

Ex 1:

```
String x="Dinesh Reddy";
String y=x.substring(7);
System.out.println(y); //begin index
String z=x.substring(7,12);
System.out.println(z); //begin index, end index
```

Ex 2:

Get count of digits, lower case alphabets, upper case alphabets and special characters in given string.



```
public static void main(String[] args)
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a line of text");
    String x=sc.nextLine();
    int nod=0;
    int nol=0;
    int nou=0;
    int nosc=0;
    for(int i=0;i<x.length();i++)
    {
        char y=x.charAt(i);
        System.out.println(y);
        if(y>=48 && y<=57)
        {
            nod=nod+1;
        }
        else if(y>=97 && y<=122)
        {
            nol=nol+1;
        }
    }
}
```

```

        else if(y>=65 && y<=90)
    {
        nou=nou+1;
    }
    else
    {
        nosc=nosc+1;
    }
}

System.out.println("Number of digits : "+nod);
System.out.println("Number of lower case alphabet : "+nol);
System.out.println("Number of upper case alphabet : "+nou);
System.out.println("Number of special characters : "+nosc);
}

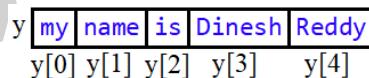
```

split():

We can use this method to divide one string into multiple substrings depends on a separator.

```
String a="my name is Dinesh Reddy";
String[] b=a.split(" "); //blank space
```

Here: In “String a” a is a variable, b is an array and in “a.split” a is an object.



join():

We can use this method to join multiple substrings as one string.

Ex 1:

```
Scanner sc=new Scanner(System.in);
System.out.println("Enter a line of text");
String x=sc.nextLine();
String[] y=x.split(" "); //blank space
for(int i=0;i<y.length;i++)
{
    System.out.println(y[i]);
}
```

Ex 2:

```
String[] c= {"My", "name", "is", "Dinesh", "Reddy"};
String z=""; //null
z=String.join(" ", c); //blank space
System.out.println(z);
```

replace():

We can use this method to replace one character with another character or to remove unwanted character.

Ex:

```
String g="ramanamma";
String h=g.replace("a","k");
System.out.println(h); //rkmknkmmk
String d=g.replace("a","");
System.out.println(d); //rmnmm
```

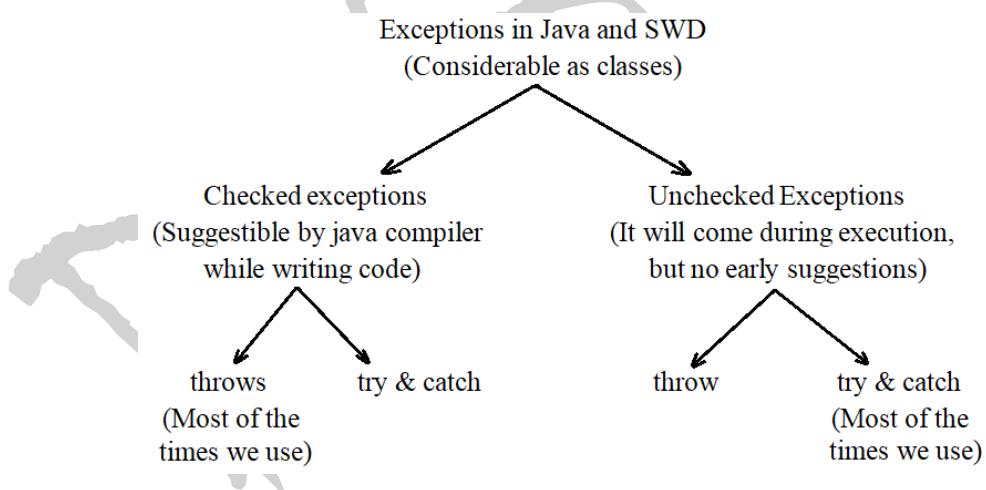
toLowerCase()

toUpperCase():

We can use these methods to convert given string to required case.

w) **Exception Handling:**

Exception is a runtime error. It can stop execution when it was raised. To continue execution we can try to handle those exceptions. In general exceptions are working like classes in java & selenium WebDriver.



Ex 1:

```
public static void main(String[] args) throws InterruptedException
{
    Thread.sleep(5000);
}
```

From the above example “**throws**” keyword can allow 1 or more checked exceptions at method signature as classes.

Ex 2:

```
public static void main1(String[] args)
{
    int x=10;
    int y=0;
    try
    {
        int z=x/y;
        System.out.println(z);
    }
    catch(ArithmeticException ex)
    {
        System.out.println("wrong division");
    }
}
```

To handle unchecked exceptions, we can use try & catch block because those exceptions will come during runtime.

Ex 3:

```
public static void main2(String[] args)
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter Voter age");
    int x=sc.nextInt();
    if(x<18)
    {
        ArithmeticException obj=new ArithmeticException("invalid to vote");
        throw obj;
        (or)
        throw new ArithmeticException();
        (or)
        throw new ArithmeticException("invalid to vote");
    }
    else
    {
        System.out.println("valid to vote");
    }
}
```

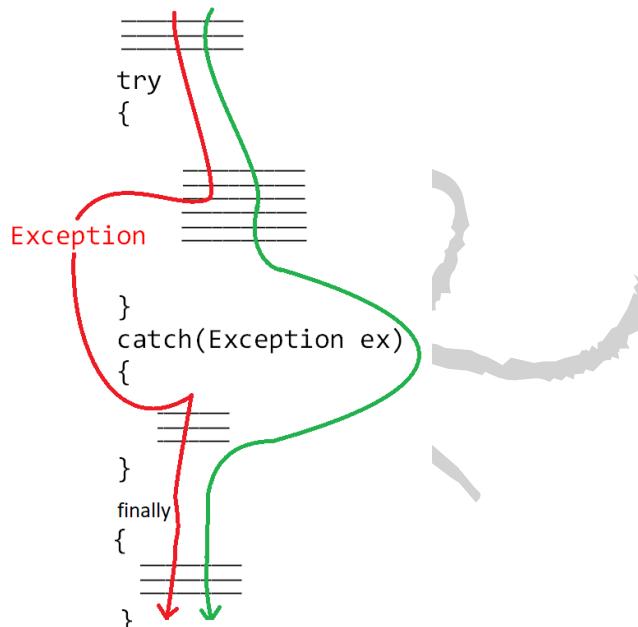
“**throw**” keyword can take object of exception related class to raise exception at runtime if required (Explicitly).

Note 1:

Every exception name can start with uppercase like class because exceptions are consider as classes in java.

finally:

“finally” block is used to place important code, it will be executed whether the exception is handled or not.



Note 2:

“Exception” can represent any exception related to java, selenium...etc.

Note 3:

No.	throw	throws
1)	Java throw keyword is used to explicitly throw an exception.	Java throws keyword is used to declare an exception.
2)	Checked exception cannot be propagated using throw only.	Checked exception can be propagated with throws.
3)	Throw is followed by an instance.	Throws is followed by class.
4)	Throw is used within the method.	Throws is used with the method signature.
5)	You cannot throw multiple exceptions.	You can declare multiple exceptions Ex: public void method()throws IOException, SQLException.

final:

Java language can provide “final” keyword. We can apply this word for class, method and variable.

- “final” class cannot be inherited.
- “final” method cannot be overridden.
- “final” variable value can't be changed.

Ex 1: “final” variable

```
final int x=10; //x value can't be changed
```

Ex 2: “final” method

```
public final void mindq() //Can't be overridden but can overload
{
    -----
    -----
}
```

Ex 3: “final” class

```
public final class Test23 //Can't allow/give access to child classes
{
    -----
    -----
}
```

finalize():

Java can support “finalize()” method to run automatically just before explicit garbage collection of memory allocated for objects, variables...etc, in corresponding program.

Ex:

```
public class Test26
{
    public void finalize()
    {
        System.out.println("Testing completed");
    }
    public static void main(String[] args) throws Exception
    {
        Test26 obj=new Test26();
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        ChromeDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://google.co.in");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .name("q")));
    }
}
```

```

        driver.close();
        obj=null;
        //finalize() can run automatically
        System.gc();
    }
}

```

x) Run java code from command prompt:

- Open note pad
- Write java code like shown below

```

public class Sample20
{
    public static void main(String[] args)
    {
        int x=Integer.parseInt(args[0]);
        int y=Integer.parseInt(args[1]);
        int z=x+y;
        System.out.println(z);
    }
}

```

- Save above code with class name “.java” and change type to all files.
Ex: D:\DineshReddy\Sample20.java

- Close note pad
- Go to CMD (Command Prompt)
- Run below like commands

```

D:\DineshReddy>javac Sample20.java ↵
D:\DineshReddy>java Sample20 xx xx↵

```

Here: xx xx are two values these will go to “args” array of main() method

- If directory is in C:\> then change directory according to your preference like shown below

```
C:\>D:\>
```

```
D:\>cd DineshReddy\
```

```

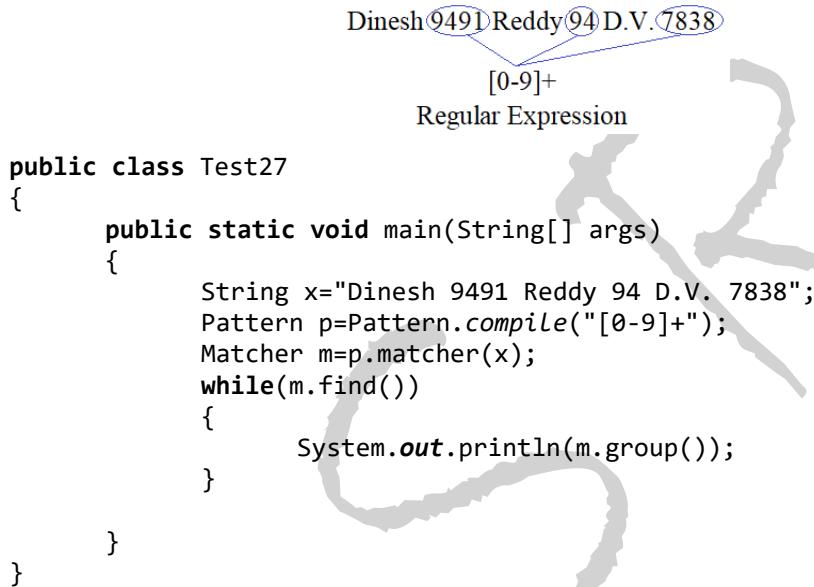
public static void main(String[] args)
{
    no return
    name is mandatory
    as main because
    JVM can search for it
    to start automatically
}

```

y) Pattern Matching using Regular Expressions:

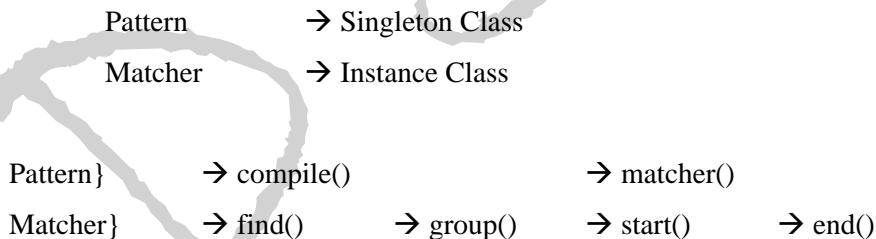
While conducting data driven testing with multiple test data on websites, SDETS can take Test data from manual testers or SH'S (Stack Holders). Sometimes, manual testers or SH'S provided data is having wanted & unwanted content for testing. To separate wanted data from unwanted data, we can use pattern matching with Regular Expressions.

Example 1:



```
public class Test27
{
    public static void main(String[] args)
    {
        String x="Dinesh 9491 Reddy 94 D.V. 7838";
        Pattern p=Pattern.compile("[0-9]+");
        Matcher m=p.matcher(x);
        while(m.find())
        {
            System.out.println(m.group());
        }
    }
}
```

For above operations, we can use below java classes (java.util.regex package in JDK):



Pattern	→ Singleton Class
Matcher	→ Instance Class
Pattern{	→ compile()
Matcher{	→ find() → group()
	→ start() → end()

find(): To point matched values one after another.

group(): To get value of matched.

start(): To get starting position.

end(): To get ending position of matched values in given string.

compile(): To provide regular expression.

matcher(): To apply given regular expression on given string

Example 2:

```
public static void main(String[] args)
{
    String x="Mobile No: 9491947838; Aadhaar No: 997690769826;"  

        + "IFSC code: SBIN0012668; "  

        + "Mail ID: dineshreddy2995@gmail.com; "  

        + "A/C No: 20206093051;";  

    // finding Aadhaar Number  

    Pattern p=Pattern.compile("[9]{2}[[0-9]+]{8}[0-6]{2}");  

    Matcher m=p.matcher(x);  

    while(m.find())  

    {  

        System.out.println(m.group());  

    }  

    // finding Account Number  

    Pattern a=Pattern.compile("[[0-9]+]{8}[0-5]{3}");  

    Matcher b=a.matcher(x);  

    while(b.find())  

    {  

        System.out.println(b.group());  

    }  

    //finding email id  

    Pattern c=Pattern.compile("[a-z0-9]+@gmail\\.com");  

    Matcher d=c.matcher(x);  

    while(d.find())  

    {  

        System.out.println(d.group());  

    }  

    //finding mobile number  

    Pattern e=Pattern.compile("[789]{1}[[0-9]+]{7}[38]{2}");  

    Matcher f=e.matcher(x);  

    while(f.find())  

    {  

        System.out.println(f.group());  

    }  

    //finding IFSC code  

    Pattern g=Pattern.compile("[SBIN]{4}[0-9]+{7}");  

    Matcher h=g.matcher(x);  

    while(h.find())  

    {  

        System.out.println(h.group());  

    }
}
```

z) OOPS Concepts:

Data Encapsulation:

Binding data and code in a block called as Data Encapsulation.

Ex: class

Data Abstraction:

Access hidden data & code in a class with permissions.

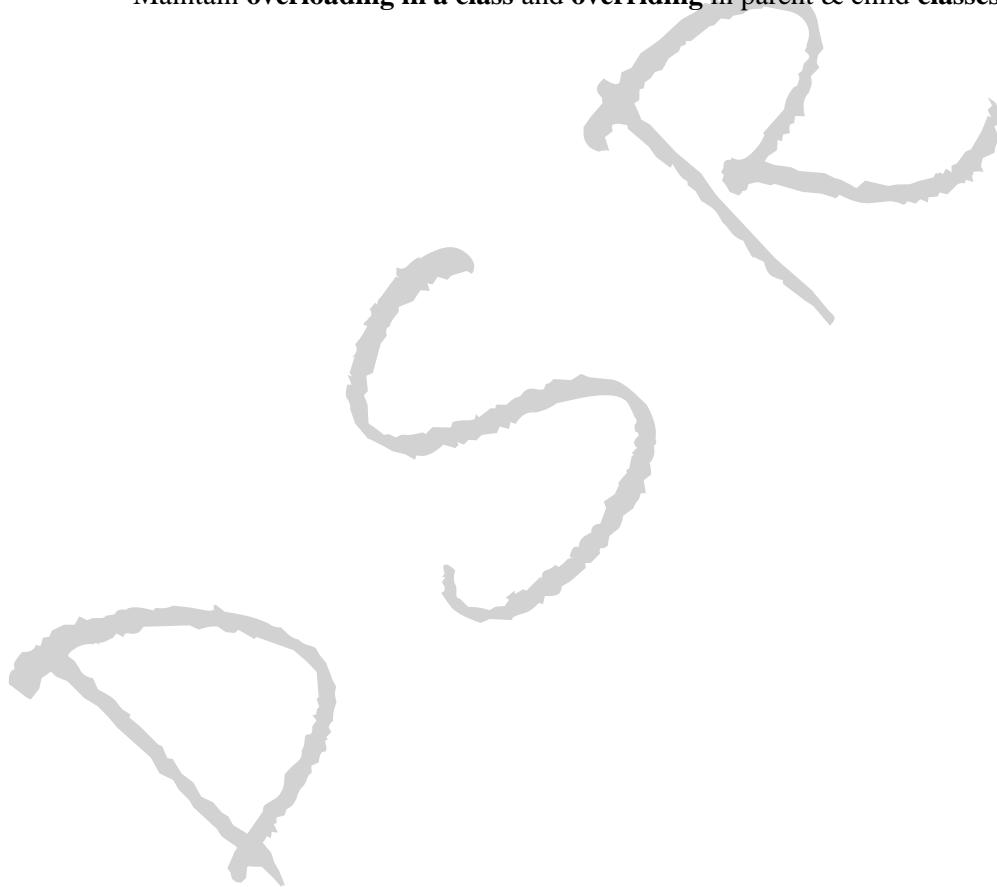
Ex: object

Inheritance:

Creation of “is a” and “has a” relation in between classes.

Polymorphism:

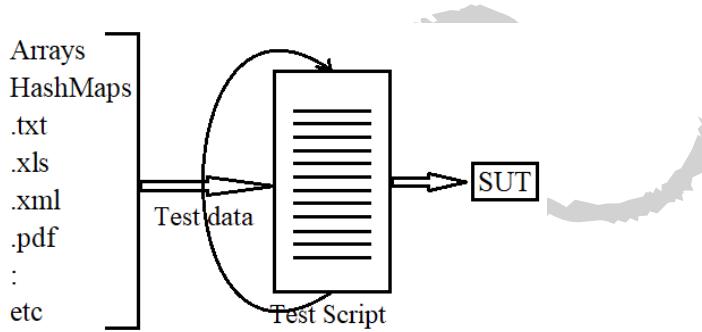
Maintain **overloading in a class** and **overriding** in parent & child classes



VIII. DDT

[Data Driven Testing Framework]

- ❖ When we wrote total automation code in **main()** method of class, we can call that automation code is in **Linear Framework**.
- ❖ When we try to execute **Linear Framework** based test script for multiple test data, the corresponding execution of script is called as **Data Driven Testing Framework**.



a) Using Arrays:

Array is a data structure to maintain multiple data. In java two types of arrays are possible to create, such as **static arrays** and **dynamic arrays**.

Static arrays are size based. We need to specify fixed size while creating these arrays like shown below

```

int[] x=new int[5]; //able to store 5 int values
char[] y=new char[5]; //able to store 5 char values
String[] z=new String[5]; //able to store 5 string values
  
```

To send data to static arrays:

```
int[] c= {10,20,30,40,50}; //initialization
```

x	10	20	30	40
	x[0]	x[1]	x[2]	x[3]

(or)

```

int[] x=new int[5]; //able to store 5 int values
Scanner sc=new Scanner(System.in);
for(int i=0;i<5;i++)
{
    x[i]=sc.nextInt();
}
//Here x[5] is not possible because x is a static array with 5 values as per
declaration so x cannot support 6th value
  
```

Write java code to perform sorting on a static array.

```
public class SortingStaticArray
{
    public static void main(String[] args)
    {
        //Declare static array
        int[] x=new int[5];
        //Fill array with data
        Scanner sc=new Scanner(System.in);
        System.out.println("Fill array");
        for(int i=0;i<5;i++)
        {
            x[i]=sc.nextInt();
        }
        //sorting
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<4;j++)
            {
                if(x[j]>x[j+1]) // > ascending, < descending
                {
                    int temp=x[j];
                    x[j]=x[j+1];
                    x[j+1]=temp;
                }
            }
        }
        //access data in array
        for(int i=0;i<5;i++)
        {
            System.out.println(x[i]);
        }
    }
}
```

We need to follow below syntax to create dynamic arrays.

```
ArrayList<Integer> x=new ArrayList<Integer>(); //Store multiple int values
ArrayList<String> y=new ArrayList<String>(); //Store multiple String values
```

To fill dynamic arrays:

```
Scanner sc=new Scanner(System.in);
ArrayList<Integer> x=new ArrayList<Integer>(); //Store multiple int values
System.out.println("Enter size of array");
int n=sc.nextInt();
System.out.println("Enter values");
for(int i=0;i<n;i++)
{
    x.add(sc.nextInt());
}
//x.add(n+1); accept additional values beyond the given size because x is
dynamic array
```

Write java code to perform sorting on dynamic array

```
public class SortingDynamicArray
{
    public static void main(String[] args)
    {
        //Declare dynamic array
        ArrayList<Integer> x=new ArrayList<Integer>();
        //Fill array with data
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter size array");
        int n=sc.nextInt();
        System.out.println("Fill array");
        for(int i=0;i<n;i++)
        {
            x.add(sc.nextInt());
        }
        //sorting
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n-1;j++)
            {
                if(x.get(j)>x.get(j+1))
                {
                    int temp=x.get(j);
                    x.set(j,x.get(j+1));
                    x.set(j+1,temp);
                }
            }
        }
        //access and display data in array
        for(int i=0;i<n;i++)
        {
            System.out.println(x.get(i));
        }
    }
}
```

Example 1:

Automate user id field testing in Gmail via Data Driven using dynamic arrays.

```
public static void main(String[] args) throws Exception
{
    //Save results in html using extent reports
    ExtentReports er=new ExtentReports("gmailDDT.html",false);
    ExtentTest et=er.startTest("Gmail UID testing");
    //Create dynamic arrays
    ArrayList<String> uids=new ArrayList<String>();
    ArrayList<String> cs=new ArrayList<String>();
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter Number of Iterations");
    int noi=Integer.parseInt(sc.nextLine());
    Date d=new Date();
    SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
```

```

try
{
    //assign data to arrays
    for(int i=0;i<noin;i++)
    {
        System.out.println("Enter UID");
        uids.add(sc.nextLine());
        System.out.println("Enter Criteria");
        cs.add(sc.nextLine());
    }
    for(int i=0;i<noin;i++)
    {
        //Launch Gmail Site
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        ChromeDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.gmail.com");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .name("identifier")));
        //Do login with valid data
        driver.findElement(By.name("identifier")).sendKeys(uids.get(i));
        w.until(ExpectedConditions.elementToBeClickable(By
            .xpath("//span[text()='Next']")));
        driver.findElement(By.xpath("//span[text()='Next']")).click();
        if(uids.get(i).length()==0) //blank user id
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//div[contains(text(),'Enter an email')]")));
            et.log(LogStatus.PASS,"Blank user id test passed");
        }
        else if(cs.get(i).equalsIgnoreCase("Invalid")) //invalid user id
        {
            //invalid with valid email error
            if(uids.get(i).contains(" "))
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                    .xpath("//div[contains(text(),'Enter a valid email')]")));
                et.log(LogStatus.PASS,"Invalid user id test passed
                    (valid mail error)");
            }
            //invalid with couldn't find account error
            else
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                    .xpath("//div[contains(text(),'Google Account')]")));
                et.log(LogStatus.PASS,"Invalid user id test passed
                    (Couldn't find error)");
            }
        }
        else if(cs.get(i).equalsIgnoreCase("Valid")) //valid user id
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .name("password")));
        }
    }
}

```

```

        et.log(LogStatus.PASS,"Valid user id test passed");
    }
    else
    {
        String fname=sdf.format(d)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(fname);
        FileHandler.copy(src, dest);
        et.log(LogStatus.ERROR, "Error in your Code"+et
                           .addScreenCapture(fname));
    }
    //close site
    driver.close();
}
catch(Exception ex)
{
    et.log(LogStatus.FAIL, "Gmail login test failed"+ex.getMessage());
}
//Save results
er.endTest(et);
er.flush();
}

```

Note:

Defect (mismatch in between expected and actual)

- Test case/Test script related defect
- Test data related defect
- Test environment related defect
- Software or sprint coding related defect (Bug)

Example 2:

Automate user id and password field testing in Gmail via Data Driven using dynamic arrays.

```

//Save results in html using extent reports
ExtentReports er=new ExtentReports("gmailDDT.html",false);
ExtentTest et=er.startTest("Gmail UID & PWD testing");
//Create dynamic arrays
ArrayList<String> uids=new ArrayList<String>();
ArrayList<String> uidcs=new ArrayList<String>();
ArrayList<String> pwds=new ArrayList<String>();
ArrayList<String> pwdcs=new ArrayList<String>();
Scanner sc=new Scanner(System.in);
System.out.println("Enter Number of Iterations");
int noi=Integer.parseInt(sc.nextLine());
ChromeDriver driver=null;
int z=0;
//assign data to arrays

```

```

for(int i=0;i<nno;i++)
{
    System.out.println("Enter UID");
    uids.add(sc.nextLine());
    System.out.println("Enter UID Criteria");
    uidcs.add(sc.nextLine());
    if(uidcs.get(i).equalsIgnoreCase("valid"))
    {
        System.out.println("Enter PWD");
        pwds.add(sc.nextLine());
        System.out.println("Enter PWD Criteria");
        pwc.add(sc.nextLine());
    }
}
for(int i=0;i<nno;i++)
{
    try
    {
        //Launch Gmail Site
        System.setProperty("webdriver.chrome.driver"
                           , "D:\\DineshReddy\\chromedriver.exe");
        driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.gmail.com");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                               .name("identifier")));
        //Do login with valid data
        driver.findElement(By.name("identifier")).sendKeys(uids.get(i));
        w.until(ExpectedConditions.elementToBeClickable(By
                                                       .xpath("//span[text()='Next']")));
        driver.findElement(By.xpath("//span[text()='Next']")).click();
        if(uids.get(i).length()==0) //blank user id
        {
            z=z+1;
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                               .xpath("//div[contains(text(),'Enter an email')]")));
            et.log(LogStatus.PASS,"Blank user id test passed");
        }
        else if(uidcs.get(i).equalsIgnoreCase("Invalid")) //invalid user id
        {
            z=z+1;
            //invalid with valid email error
            if(uids.get(i).contains(" "))
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                               .xpath("//div[contains(text(),'Enter a valid email')]")));
                et.log(LogStatus.PASS,"Invalid user id test passed
                                         (valid mail error)");
            }
            //invalid with couldn't find account error
        }
        else
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                               .xpath("//div[contains(text(),'Google Account')]")));
        }
    }
}

```

```

        et.log(LogStatus.PASS,"Invalid user id test passed
                                (Couldn't find error)");
    }
}
else if(uidcs.get(i).equalsIgnoreCase("Valid")) //valid user id
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                .name("password")));
    et.log(LogStatus.PASS,"Valid user id test passed");
    driver.findElement(By.name("password")).sendKeys(pwd.get(i-z));
    w.until(ExpectedConditions.elementToBeClickable(By
                                                .xpath("//span[text()='Next']")));
    driver.findElement(By.xpath("//span[text()='Next']")).click();
    if(pwd.get(i-z).length()==0) //blank password id
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                .xpath("//div[text()='Enter a password']")));
        et.log(LogStatus.PASS,"Blank password test passed");
    }
    else if(pwdcs.get(i-z).equalsIgnoreCase("Invalid")) //invalid pswd
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                .xpath("//div[contains(text(),'Wrong password')]")));
        et.log(LogStatus.PASS,"Invalid password test passed");
    }
    else if(pwdcs.get(i-z).equalsIgnoreCase("Valid")) //valid pswd
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                .xpath("//div[text()='Compose']")));
        et.log(LogStatus.PASS,"Valid password test passed");
    }
    else
    {
        String fname=sdf.format(d)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(fname);
        FileHandler.copy(src, dest);
        et.log(LogStatus.FAIL,"Gmail password login test failed"
                            +et.addScreenCapture(fname));
    }
}
else
{
    String fname=sdf.format(d)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(fname);
    FileHandler.copy(src, dest);
    et.log(LogStatus.FAIL, "Gmail user id login test failed"+et
                            .addScreenCapture(fname));
}
//close site
driver.close();
}

```

```

catch(Exception ex)
{
    //close site
    driver.close();
    et.log(LogStatus.ERROR, "Error in your user id Code"+ex.getMessage());
}
//Save results
er.endTest(et);
er.flush();

```

Note 1:

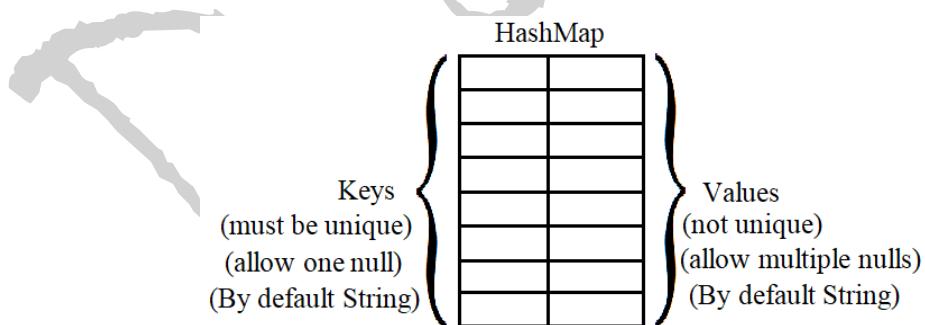
In general, garbage collection can work at end of test script execution. Here all variables, arrays and objects can go to die. While running next time, we can get new memory for variables, array and objects. Due to this reason we are always need to fill arrays with test data.

Note 2:

When our scenario related to multiple fields, we need to take more number of arrays to maintain huge test data along with criteria. As per coding, it is more complex. Due to this reason, we can think about Data Driven testing using Hash maps.

b) DDT using HashMaps:

HashMap is also a data structure to store data as pairs. Here, every pair is a combination of key and value.



- By default it will store 16 pairs and if we give 17th pair then it will extend the memory.

To create HashMap, we need to follow below syntax.

```

HashMap<xxxx,xxxx> x=new HashMap<xxxx,xxxx>();
    ↓   ↓   ↓   ↓   ↓   ↓
  class in Type Type Name to Type Type
  java.util of keys of values of keys of values
  of JDK

```

```

public static void main(String[] args)
{
    //create HashMap
    HashMap<Integer, String> hm=new HashMap<Integer, String>();
    //Insert data as pairs into HashMap
    hm.put(101, "Dinesh");
    hm.put(102, "Reddy");
    //Get data from HashMap
    for(Map.Entry<Integer, String> e:hm.entrySet())
    {
        System.out.print(e.getKey());
        System.out.println(" "+e.getValue());
    }
}

```

From the above example code, **put()** method is useful to store data as key & value pair.

entrySet():

entrySet() method is useful to collect all pairs in HashMap.

```

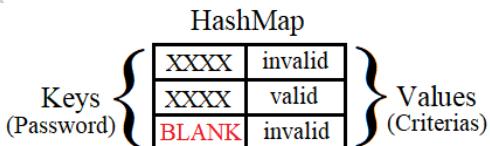
for(Map.Entry<Integer, String> e:hm.entrySet())
{
    =====
}

//Here e is an object to "Map.Entry" class to represent each pair (entry).
//Here hm is an object of "HashMap" class.

```

Here, “Map” is outer class and “Entry” is inner class.

e.getKey() → //to get key of current pair
e.getValue() → //to get value of current pair



Example:

Automation password field Testing in Gmail via Data Driven Testing by using HashMap.

```

//Get test data
HashMap<String, String> pswds=new HashMap<String, String>();
Scanner sc=new Scanner(System.in);
System.out.println("Enter test data size");
int noi=Integer.parseInt(sc.nextLine());
for(int i=0;i<noi;i++)
{
    System.out.println("Enter Password");
    String x=sc.nextLine();
}

```

```

System.out.println("Enter password criteria");
String y=sc.nextLine();
pswds.put(x, y);
}
//Create html report file
ExtentReports er=new ExtentReports("Gamil.html",false);
ExtentTest et=er.startTest("Gmail password testing");
//Data Driven Testing
ChromeDriver driver=null;
for(Map.Entry<String, String> e:pswds.entrySet())
{
    try
    {
        //Launch Site
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.gmail.com");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
.name("identifier")));
        //Do login with valid data
        driver.findElement(By.name("identifier")).sendKeys("dineshry143");
        w.until(ExpectedConditions.elementToBeClickable(By
.xpath("//span[text()='Next']")));
        driver.findElement(By.xpath("//span[text()='Next']")).click();
        w.until(ExpectedConditions.visibilityOfElementLocated(By
.name("password")));
        driver.findElement(By.name("password")).sendKeys(e.getKey());
        w.until(ExpectedConditions.elementToBeClickable(By
.xpath("//span[text()='Next']")));
        driver.findElement(By.xpath("//span[text()='Next']")).click();
        if(e.getKey().length()==0) //blank password id
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
.xpath("//div[text()='Enter a password']")));
            et.log(LogStatus.PASS,"Blank password test passed");
        }
        else if(e.getValue().equalsIgnoreCase("Invalid")) //invalid password id
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
.xpath("//div[contains(text(),'Wrong password')]")));
            et.log(LogStatus.PASS,"Invalid password test passed");
        }
        else if(e.getValue().equalsIgnoreCase("Valid")) //valid password id
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
.xpath("//div[text()='Compose']")));
            et.log(LogStatus.PASS,"Valid password test passed");
        }
    }
    else
    {
        Date d=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
    }
}

```

```

        String fname=sdf.format(d)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(fname);
        FileHandler.copy(src, dest);
        et.log(LogStatus.FAIL, "Gmail password login test failed"+et
                .addScreenCapture(fname));
    }
    //close site
    driver.close();
}
catch(Exception ex)
{
    et.log(LogStatus.ERROR,"Error in your password Code"+ex.getMessage());
}
//Save results
er.endTest(et);
er.flush();
}

```

Note 1:

Like Arrays, Hashmaps can get memory in RAM which is not permanent in general, garbage collector can clean data in RAM at the end of corresponding block of code execution.

Due to this reason, we need to enter data to HashMap for every time of execution.

Note 2:

In HashMap pairs of entries are accessible randomly (not using index).

Note 3:

We are able to change keys and values of existing entries in a HashMaps.

```

//change value using key
hm.replace(102, "virat kohli"); //102 virat kohli
//change key using key
hm.put(103, hm.remove(102)); //103 virat kohli

```

c) Using “.txt” file:

- Files are permanent because they are saved in HDD. So, we need to enter test data for every time of execution.
- We can maintain test data in **.txt** file as lines.
- We can use notepad or any other editor to create **.txt** file.
- In every line of text in **.txt** file, we can use “,” as separator. Due to this reason **.txt** file is also called as **“.csv”** file (**Comma Separated Values**).

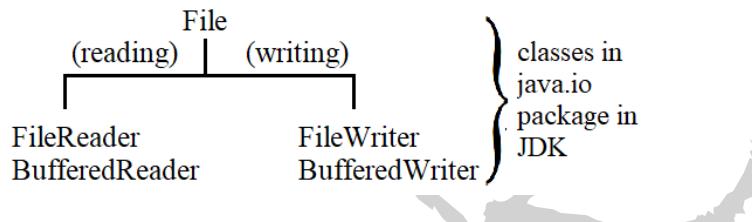
way2sms.txt

```

,invalid,XXXXXXX,valid
9491,invalid,XXXXXXX,valid
8446551112,invalid,XXXXXXX,valid
9491260836,invalid,XXXXXXX,valid
9491947838,valid,,invalid                                blank value
9491947838,valid,aiusiav,invalid
9491947838,valid,XXXXXXX,valid

```

- To work with .txt files, JDK can provide below classes.



Example:

Automate **way2sms** site login operation by taking required test data from above specified text file.

```

//open text file for data reading
File f=new File("way2sms.txt");
FileReader fr=new FileReader(f);
BufferedReader br=new BufferedReader(fr);
//create HTML reports file
ExtentReports er=new ExtentReports("way2smsDDT.html",false);
ExtentTest et=er.startTest("way2sms login testing");
//Data driven testing
ChromeDriver driver=null;
ChromeOptions co=new ChromeOptions();
co.addArguments("disable-notifications");
String s="";
while((s=br.readLine())!=null)
{
    String[] t=s.split(",");
    try
    {
        //Launch site
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        driver=new ChromeDriver(co);
        driver.manage().window().maximize();
        driver.get("http://www.way2sms.com");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                       .name("mobileNo")));
        //Do login
        driver.findElement(By.name("mobileNo")).sendKeys(t[0]);
        driver.findElement(By.name("password")).sendKeys(t[2]);
    }
}

```

```

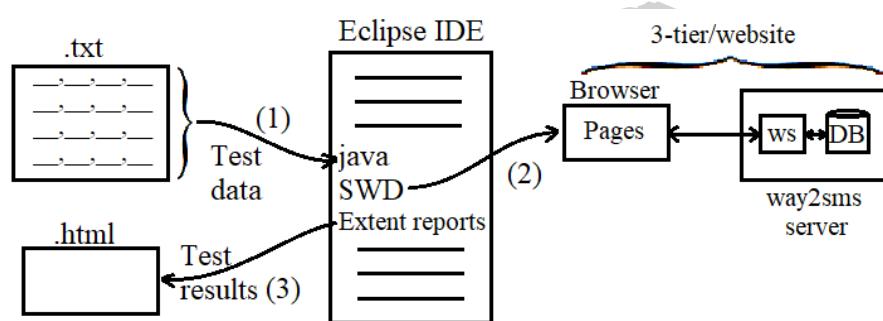
driver.findElement(By.xpath("//button[contains(text(),'Login')])[1]").click();
//validations
if(t[0].length()==0 && t[1].equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[text()='Enter your mobile number']")));
    et.log(LogStatus.PASS, "Blank mobile number test passed");
}
else if(t[0].length()<10 && t[1].equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[text()='Enter valid mobile number']")));
    et.log(LogStatus.PASS, "Wrong size mobile number test passed");
}
else if(t[1].equalsIgnoreCase("invalid"))
{
    try
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[contains(text(),'Incorrect number')]")));
        et.log(LogStatus.PASS, "Invalid mobile number test passed");
    }
    catch(Exception ex)
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[contains(text(),'number is not register')]")));
        et.log(LogStatus.PASS, "Mobile no: not registered test passed");
    }
}
else if(t[1].equalsIgnoreCase("valid") && t[2].length()==0 &&
        t[3].equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[text()='Enter password'])[1]")));
    et.log(LogStatus.PASS, "Blank password test passed");
}
else if(t[1].equalsIgnoreCase("valid") && t[3].equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[contains(text(),'Incorrect number')]")));
    et.log(LogStatus.PASS, "Invalid password test passed");
}
else if(t[1].equalsIgnoreCase("valid") && t[3].equalsIgnoreCase("valid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//div[text()='SendSMS']")));
    et.log(LogStatus.PASS, "Login test passed");
}
else
{
    et.log(LogStatus.FAIL,"Login test failed");
}
//close site
driver.close();
}

```

```

        catch(Exception ex)
        {
            //close site
            driver.close();
            et.log(LogStatus.ERROR, "Error in code"+ex.getMessage());
        }
    }
//close text file
br.close();
fr.close();
//Save Results in HTML
er.endTest(et);
er.flush();

```



Note 1:

After getting a line of text from text file, we need to split that line of text into pieces depends on “,”.

```

String s="";
while((s=br.readLine())!=null)
{
    String[] t=s.split(",");
}

```

Ex:

s="9491,Invalid,XXXX,valid";

t	9491	Invalid	XXXX	valid
	t[0]	t[1]	t[2]	t[3]
	Mobile No	Mobile criteria	Password	Password criteria

Note 2:

```

while((s=br.readLine())!=null)
{
    _____
}

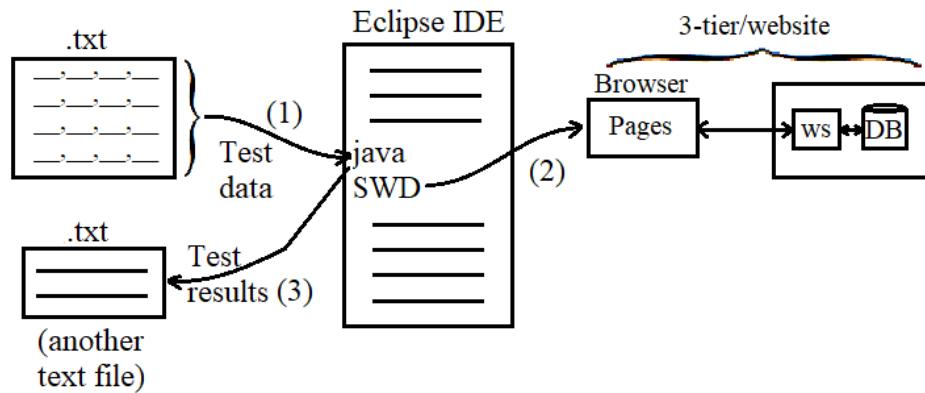
```

Read line by line from 1st line to last line in a .txt file.

- Due to above reasons, .txt files are called as sequential files.

***Note 3:**

If we are interested, we can able to maintain test results in another text file because text files can allow either reading or writing at a time.



- For data reading

```
File f=new File("existing .txt file");
FileReader fr=new FileReader(f);
BufferedReader br=new BufferedReader(fr);
```

- For data writing

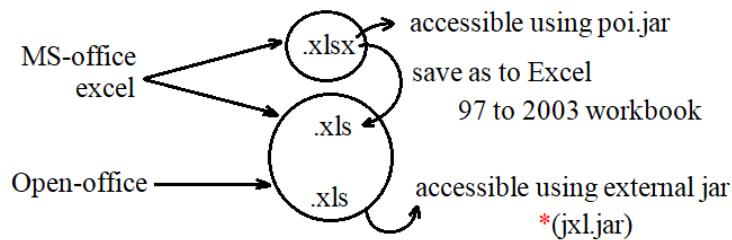
```
File f1=new File("existing/new .txt file");
FileWriter fw=new FileWriter(f1);
BufferedWriter bw=new BufferedWriter(fw);
```

- Write data to file

```
String s="";
while((s=br.readLine())!=null)
{
    _____
    _____
    bw.write("xxxxxxxxxxxxxxxxxxxx");
    bw.newLine();
    _____
    _____
}

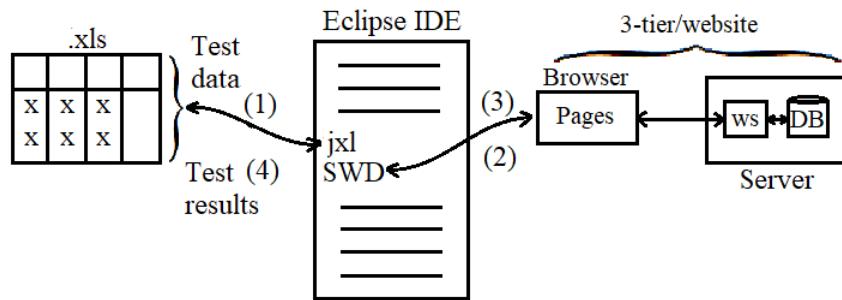
//close text file
br.close();
fr.close();
//save text file
bw.close();
fw.close();
```

d) Using “.xls” file:



“.xls” files are useful for test data reading and test results writing.

- In general, Arrays & HashMaps are maintaining data in RAM but RAM is not permanent memory. “.txt” files are saved in HDD as permanent but these files can support either reading or writing at a time. To maintain test data & test results as permanent in a single file, we can use **.xls** (MS-office/Open-office) file.



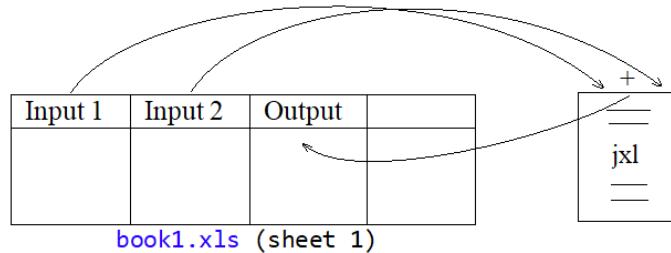
- ❖ To configure **jxl** we need to follow below navigation:

- Go to Google site
- Enter “**jxl download**” & click search
- Go to “java2s.com”
- Click on **jxl** link to start download
- Paste that download in personal folder
- Extract that downloaded file
- Go to eclipse IDE
- Right click on project folder & click “properties”
- Then click on “Java build path” and go to “libraries”.
- Click on “add external jars”
- Browse path of **jxl** jar and click “open”
- Click “apply & close”

- ❖ We need to save excel file as **.xls** in MS-office/Open-office (“Save as” Excel 97-2003 Workbook”).

- ❖ Every .xls file is called as workbook. By default every workbook is having 3 work sheets. Every worksheet is having **65,536** rows and **256** columns. The area of a row & column is called cell, cells are having data.

Example 1:



```

public static void main(String[] args) throws Exception
{
    //Open ".xls" file for test data reading
    File f=new File("book1.xls");
    Workbook rwb=Workbook.getWorkbook(f);
    Sheet rsh=rwb.getSheet(0); //0 means Sheet1
    int nour=rsh.getRows(); //Count of used rows
    //Open Same excel file for writing
    WritableWorkbook wwb=Workbook.createWorkbook(f,rwb);
    WritableSheet wsh=wwb.getSheet(0); //0 means Sheet1
    for(int i=1;i<nour;i++)
    {
        int x=Integer.parseInt(rsh.getCell(0,i).getContents());
        int y=Integer.parseInt(rsh.getCell(1,i).getContents());
        int z=x+y;
        Number n=new Number(2,i,z);
        wsh.addCell(n);
    }
    //Save excel file
    wwb.write();
    rwb.close();
    wwb.close();
}

```

Note 1:

We need to follow below rules while using .xlsx files into .xls files.

- Save excel file in project folder to use file name instead of path.
- Save as “Excel 97 to 2003 Workbook” for .xls extension.
- First row for names of columns in every sheet in .xls file.
- *To make used cell/cells as unused, select all those cell/cells and then right click and select delete on corresponding cell’s row/column.
- *Need to close .xls file on desktop before going to run jxl code.

Note 2:

While working with .xls file we can use below **jxl** jar.

```
classDiagram
    Workbook
    Sheet
    WritableWorkbook
    WritableSheet
    Number
    Label
    WritableCellFormat
    Alignment
    WritableFont
    Colour
    }
    jxl jar
```

Example 2:

Automate way2sms site login testing by getting test data from .xls file.

Mobile No:	Mobile Criteria	Password	Password Criteria
BLANK	invalid	XXXX	valid
8446551112	invalid	XXXX	valid
9491260836	invalid	XXXX	valid
9491947838	valid	RedmiNote5Pro	invalid
9491947838	valid	BLANK	invalid
9491947838	valid	XXXX	valid
9491	invalid	XXXX	valid

```
public static void main(String[] args) throws Exception
{
    ChromeDriver driver=null;
    //Stop Notifications
    ChromeOptions co=new ChromeOptions();
    co.addArguments("disable-notifications");
    //Open ".xls" file for test data reading
    File f=new File("way2sms.xls");
    Workbook rwb=Workbook.getWorkbook(f);
    Sheet rsh=rwb.getSheet(0); //0 means Sheet1
    int nour=rsh.getRows(); //Count of used rows
    //Open Same excel file for writing
    WritableWorkbook wwb=Workbook.createWorkbook(f,rwb);
    WritableSheet wsh=wwb.getSheet(0); //0 means Sheet1
    //Set font style, colour and cell alignment for Heading
    WritableFont wf=new WritableFont(WritableFont.TIMES,11,WritableFont.BOLD);
    wf.setColour(Colour.BLUE);
    WritableCellFormat wcf=new WritableCellFormat(wf);
    wcf.setAlignment(Alignment.CENTRE);
    //Set font style, colour and alignment for test passed
    WritableFont wf1=new WritableFont(WritableFont.TIMES,11);
```

```

wf1.setColour(Colour.GREEN);
WritableCellFormat wcf1=new WritableCellFormat(wf1);
wcf1.setAlignment(Alignment.CENTRE);
//Set font style, colour and alignment for test failed
WritableFont wf2=new WritableFont(WritableFont.TIMES,11);
wf2.setColour(Colour.RED);
WritableCellFormat wcf2=new WritableCellFormat(wf2);
wcf2.setAlignment(Alignment.CENTRE);
//Create result column
int nouc=wsh.getColumns();
//Take results heading as date and time format
Date d=new Date();
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
String x="Results on "+sdf.format(d);
Label l=new Label(nouc,0,x,wcf);
wsh.addCell(l);
for(int i=1;i<nour;i++)
{
    String mbno=rsh.getCell(0,i).getContents();
    String mbnoc=rsh.getCell(1,i).getContents();
    String pswd=rsh.getCell(2,i).getContents();
    String pswdc=rsh.getCell(3,i).getContents();
    try
    {
        //Launch site
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        driver=new ChromeDriver(co);
        driver.manage().window().maximize();
        driver.get("http://www.way2sms.com");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                       .name("mobileNo")));
        //Do login
        driver.findElement(By.name("mobileNo")).sendKeys(mbno);
        driver.findElement(By.name("password")).sendKeys(pswd);
        driver.findElement(By.xpath("//button[contains(text(),'Login')])[1]"))
                           .click();
        //validations
        if(mbno.length()==0 && mbnoc.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                       .xpath("//b[text()='Enter your mobile number']")));
            Label l1=new Label(nouc,i,"Blank mobile number test passed",
                               wcf1);
            wsh.addCell(l1);
        }
        else if(mbno.length()<10 && mbnoc.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                                                       .xpath("//b[text()='Enter valid mobile number']")));
            Label l1=new Label(nouc,i,"Wrong size mobile number test passed",
                               wcf1);
            wsh.addCell(l1);
        }
    }
}

```

```

else if(mbnoc.equalsIgnoreCase("invalid"))
{
    try
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[contains(text(),'Incorrect number')]")));
        Label l1=new Label(nouc,i,"Invalid mobile number test passed",wcf1);
        wsh.addCell(l1);
    }
    catch(Exception ex)
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[contains(text(),'number is not register')]")));
        Label l1=new Label(nouc,i,"Mobile no: not register test passed",wcf1);
        wsh.addCell(l1);
    }
}
else if(mbnoc.equalsIgnoreCase("valid") && pswd.length()==0 &&
        pswdc.equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[text()='Enter password'][1]")));
    Label l1=new Label(nouc,i,"Blank password test passed",wcf1);
    wsh.addCell(l1);
}
else if(mbnoc.equalsIgnoreCase("valid") && pswdc.equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[contains(text(),'Incorrect number')]")));
    Label l1=new Label(nouc,i,"Invalid password test passed",wcf1);
    wsh.addCell(l1);
}
else if(mbnoc.equalsIgnoreCase("valid") && pswdc.equalsIgnoreCase("valid"))
{
    String image=sdf.format(d)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(image);
    FileHandler.copy(src,dest);
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//div[text()='SendSMS']")));
    Label l1=new Label(nouc,i,"Login test passed",wcf1);
    wsh.addCell(l1);
}
else
{
    String image=sdf.format(d)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(image);
    FileHandler.copy(src,dest);
    Label l1=new Label(nouc,i,"Login test failed "+image,wcf2);
    wsh.addCell(l1);
}
//close site
driver.close();
}

```

```

    catch(Exception ex)
    {
        //close site
        driver.close();
        Label l1=new Label(nouc,i,"Error in code"+ex.getMessage(),wcf2);
        wsh.addCell(l1);
    }
}
CellView cv=rsh.getColumnView(nouc);
cv.setAutoSize(true);
wsh.setColumnView(nouc, cv);
//Save excel file
wwb.write();
//close file
rwb.close();
wwb.close();
}

```

Note 1:

Open an excel file for reading data

```

File f=new File("path of.xls file");
Workbook rwb=Workbook.getWorkbook(f);

```

Here: Workbook is a singleton class in **jxl**

Note 2:

```

Sheet rsh=rwb.getSheet(sheet index/"sheet name");//sheet index starts with '0'
```

Here: Sheet is a class in **jxl**

Note 3:

Get count of used rows and columns in a sheet.

```

int nour=rsh.getRows(); //Count of used rows
int nouc=wsh.getColumns(); //Count of used columns

```

Note 4:

Write only

```

File f=new File("path of.xls file");
WritableWorkbook wwb=Workbook.createWorkbook(f);

```

Read and write

```

File f=new File("path of.xls file");
Workbook rwb=Workbook.getWorkbook(f);
WritableWorkbook wwb=Workbook.createWorkbook(f,rwb);

```

Note 5:

By default it is in string and converting it into integer.

```
int x=Integer.parseInt(rsh.getCell(column,row).getContents());
```

Directly take data as string

```
String x=rsh.getCell(column,row).getContents();
```

Note 6:

```
//Font Style, size and colour  
WritableFont wf=new WritableFont(WritableFont.TIMES,11,WritableFont.BOLD);  
wf.setColour(Colour.BLUE);  
  
//Cell alignment, colour and wrapttext  
WritableCellFormat wcf=new WritableCellFormat(wf);  
//Here WritableCellFormat(wf) means adding font decorations to cell  
wcf.setAlignment(Alignment.CENTRE);  
wcf.setBackground(Colour.BLACK);  
wcf.setWrap(true);  
  
Label l=new Label(column,row,"xxxxx",wcf);  
//Here Label(---,---,---,wcf) means adding font decorations & cell decorations  
to corresponding cell  
wsh.addCell(l);
```

Note 7:

Auto fit Column

```
CellView cv=rsh.getColumnView(nouc);  
cv.setAutosize(true);  
wsh.setColumnView(nouc, cv);
```

Auto fit Row

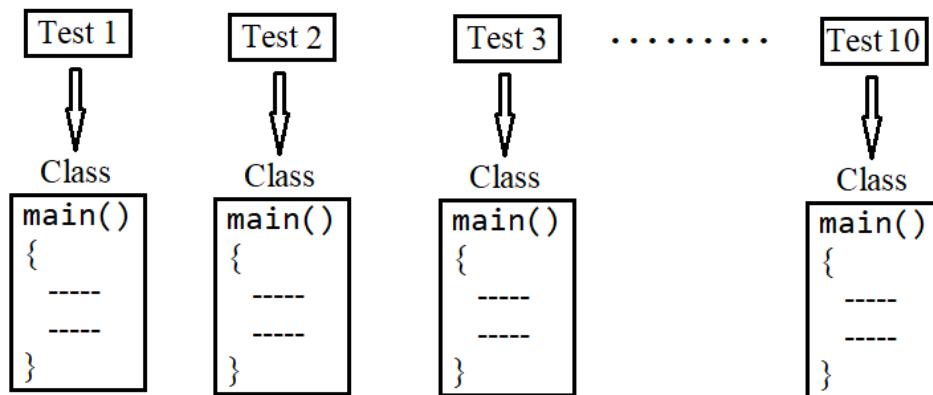
```
CellView cv=rsh.getRowView(nour);  
cv.setAutosize(true);  
wsh.setRowView(nour, cv);
```

IX. KWD

[Key Word Driven Framework]

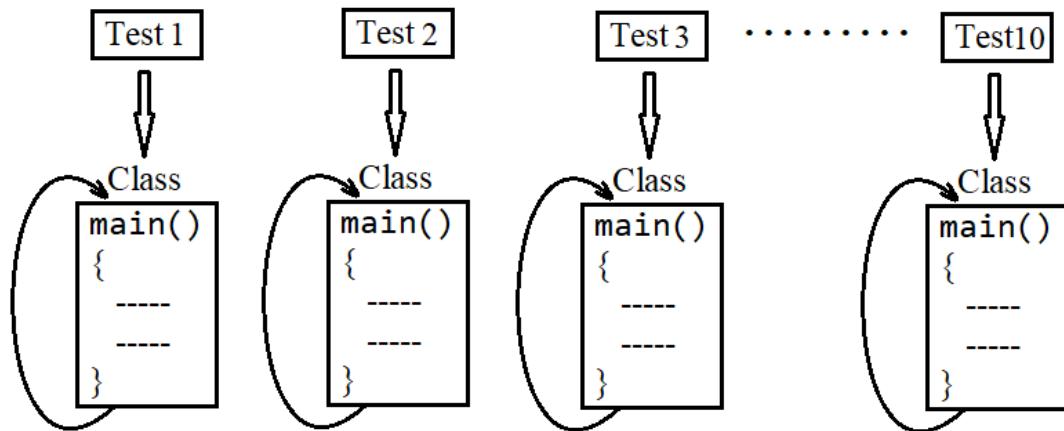
To improve code reusability in automation, we can use **Key Word Driven Framework**. In this framework we can implement an Excel file with tests and steps, a methods class with unique names to methods and finally a runner class with logic to integrate Excel file and methods.

➤ In Linear Framework



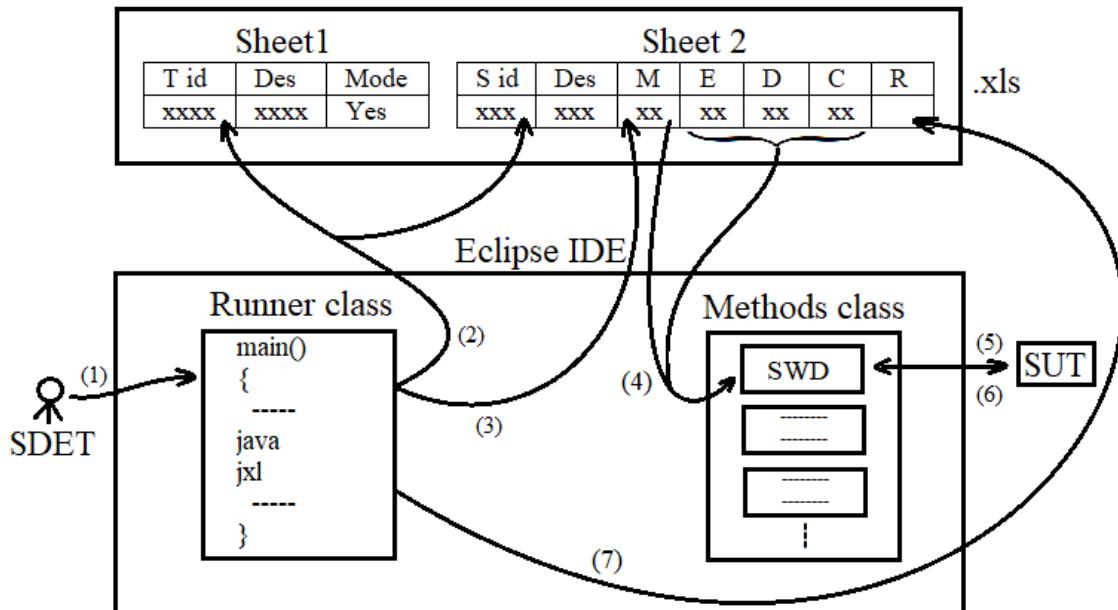
- Every class is running one time, but not running for multiple test data
- No code reusability in between classes.

➤ In Data Driven Testing Framework



- No code reusability in between classes, but every class can run for multiple test data by taking from Arrays/HashMaps/.txt files and .xls files.

➤ In Key Word Driven Framework



- Pure coding based framework
- Easy to maintain for future modifications
- We need to follow below process to implement **Key Word Driven Framework** for a website testing using selenium.

Step 1: Install JDK8 (Create JAVA_HOME environment variable and extend path variable value).

Step 2: Download and launch Eclipse IDE (create a new folder and use it as workspace folder for Eclipse IDE)

Step 3: Create new Java project with a package in Eclipse IDE (check the version of java compiler and JRE of corresponding java project).

Step 4: Download and associate required jars to java project (for a website testing, we need SWD jars, Sikulix api jars ... etc and jxl).

Step 5: Save a new excel file as .xls (Excel 97 to 2003 Workbook) in project folder.

Step 6: Go to sheet 1 of excel file for a test.

Example:

Test id	Description	Mode
TEST_1	Validate Login	Yes
TEST_2	Validate Registration	No
TEST_3	Validate Forgot Password	No
TEST_4	Validate Send SMS	No
TEST_5	Validate Logout	No

Step 7: Go to sheet 2 of excel file for steps to each test.

Example:

Step id	Step Description	Method Name Keyword	Element Locators	Test Data	Criteria
TEST_1	launch site	launch	chrome	http://www.way2sms.com	N/A
TEST_1	fill mobile number	fill	(//input[@id='mobileNo'])[1]		N/A
TEST_1	fill password	fill	(//input[@name='password'])[1]		N/A
TEST_1	click login	click	(//button[contains(text(),'Login'))][1]	N/A	N/A
TEST_1	validate login	validateLogin	//b[text()='Enter your mobile number']	N/A	MBNO_BLANK
TEST_1	close site	closeSite	N/A	N/A	N/A
TEST_1	launch site	launch	chrome	http://www.way2sms.com	N/A
TEST_1	fill mobile number	fill	(//input[@id='mobileNo'])[1]		9491 N/A
TEST_1	fill password	fill	(//input[@name='password'])[1]		N/A
TEST_1	click login	click	(//button[contains(text(),'Login'))][1]	N/A	N/A
TEST_1	validate login	validateLogin	//b[text()='Enter valid mobile number']	N/A	MBNO_WRONG_SIZE
TEST_1	close site	closeSite	N/A	N/A	N/A
TEST_1	launch site	launch	chrome	http://www.way2sms.com	N/A
TEST_1	fill mobile number	fill	(//input[@id='mobileNo'])[1]		8446551112 N/A
TEST_1	fill password	fill	(//input[@name='password'])[1]	uadvdvuu	N/A
TEST_1	click login	click	(//button[contains(text(),'Login'))][1]	N/A	N/A
TEST_1	validate login	validateLogin	//b[contains(text(),'Incorrect number or password!')]	//b[contains(text(),'number is not register')]	MBNO_INVALID
TEST_1	close site	closeSite	N/A	N/A	N/A
TEST_1	launch site	launch	chrome	http://www.way2sms.com	N/A
TEST_1	fill mobile number	fill	(//input[@id='mobileNo'])[1]		9491260836 N/A
TEST_1	fill password	fill	(//input[@name='password'])[1]		N/A
TEST_1	click login	click	(//button[contains(text(),'Login'))][1]	N/A	N/A
TEST_1	validate login	validateLogin	//b[contains(text(),'Incorrect number or password!')]	//b[contains(text(),'number is not register')]	MBNO_INVALID
TEST_1	close site	closeSite	N/A	N/A	N/A
TEST_1	launch site	launch	chrome	http://www.way2sms.com	N/A
TEST_1	fill mobile number	fill	(//input[@id='mobileNo'])[1]		9491947838 N/A
TEST_1	fill password	fill	(//input[@name='password'])[1]		N/A
TEST_1	click login	click	(//button[contains(text(),'Login'))][1]	N/A	N/A
TEST_1	validate login	validateLogin	//b[text()='Enter password']][1]	N/A	PSWD_BLANK
TEST_1	close site	closeSite	N/A	N/A	N/A
TEST_1	launch site	launch	chrome	http://www.way2sms.com	N/A
TEST_1	fill mobile number	fill	(//input[@id='mobileNo'])[1]		9491947838 N/A
TEST_1	fill password	fill	(//input[@name='password'])[1]	svhvajhsads	N/A
TEST_1	click login	click	(//button[contains(text(),'Login'))][1]	N/A	N/A
TEST_1	validate login	validateLogin	//b[contains(text(),'Incorrect number or password!')]	N/A	PSWD_INVALID
TEST_1	close site	closeSite	N/A	N/A	N/A
TEST_1	launch site	launch	chrome	http://www.way2sms.com	N/A
TEST_1	fill mobile number	fill	(//input[@id='mobileNo'])[1]		9491947838 N/A
TEST_1	fill password	fill	(//input[@name='password'])[1]		N/A
TEST_1	click login	click	(//button[contains(text(),'Login'))][1]	N/A	N/A
TEST_1	validate login	validateLogin	//div[text()='SendSMS']	N/A	ALL_VALID
TEST_1	close site	closeSite	N/A	N/A	N/A

Note:

Create copy files to above .xls file.

Step 8: Develop bodies to methods.

- Right click on package in project.
 - Select “New” & Select “Class”
 - Enter a name
- Ex:** MyMethods1
- Don’t select **main()** method option.
 - Click ok

```
public class MyMethods1
{
    public WebDriver driver;
    public WebDriverWait wait;
    public String launch(String e, String d, String c) throws Exception
    {
        try
        {
            if(e.equalsIgnoreCase("chrome"))
            {
                System.setProperty("webdriver.chrome.driver",
                                   "D:\\DineshReddy\\chromedriver.exe");
                ChromeOptions co=new ChromeOptions();
                co.addArguments("disable-notifications");
                driver=new ChromeDriver(co);
                driver.manage().window().maximize();
            }
            else if(e.equalsIgnoreCase("firefox"))
            {
                System.setProperty("webdriver.gecko.driver",
                                   "D:\\DineshReddy\\geckodriver.exe");
                driver=new FirefoxDriver();
            }
            else if(e.equalsIgnoreCase("ie"))
            {
                System.setProperty("webdriver.ie.driver",
                                   "D:\\DineshReddy\\IEDriverServer.exe");
                driver=new InternetExplorerDriver();
            }
            else if(e.equalsIgnoreCase("edge"))
            {
                System.setProperty("webdriver.edge.driver",
                                   "D:\\DineshReddy\\MicrosoftWebDriver.exe");
                driver=new EdgeDriver();
            }
            else
            {
                return("Unknown browser");
            }
        }
    }
}
```

```

        driver.get(d);
        driver.manage().window().maximize();
        return("Done");
    }
    catch(Exception ex)
    {
        return("Error "+ex.getMessage());
    }
}
public String fill(String e, String d, String c) throws Exception
{
    try
    {
        wait=new WebDriverWait(driver,20);
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(e)));
        driver.findElement(By.xpath(e)).sendKeys(d);
        return("Done");
    }
    catch(Exception ex)
    {
        return("Error "+ex.getMessage());
    }
}
public String click(String e, String d, String c) throws Exception
{
    try
    {
        wait.until(ExpectedConditions.elementToBeClickable(By.xpath(e)));
        driver.findElement(By.xpath(e)).click();
        return("Done");
    }
    catch(Exception ex)
    {
        return("Error "+ex.getMessage());
    }
}
public String validateLogin(String e, String d, String c) throws Exception
{
    try
    {
        if(c.equalsIgnoreCase("MBNO_BLANK"))
        {
            wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(e)));
            return("Passed");
        }
        else if(c.equalsIgnoreCase("MBNO_WRONG_SIZE"))
        {
            wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(e)));
            return("Passed");
        }
    }
}

```

```

    else if(c.equalsIgnoreCase("MBNO_INVALID"))
    {
        try
        {
            wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(e)));
            return("Passed");
        }
        catch(Exception ex)
        {
            wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(d)));
            return("Passed");
        }
    }
    else if(c.equalsIgnoreCase("PSWD_BLANK"))
    {
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(e)));
        return("Passed");
    }
    else if(c.equalsIgnoreCase("PSWD_INVALID"))
    {
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(e)));
        return("Passed");
    }
    else if(c.equalsIgnoreCase("ALL_VALID"))
    {
        wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(e)));
        return("Passed");
    }
    else
    {
        String temp=this.screenshot();
        return("Test failed & goto"+temp);
    }
}
catch(Exception ex)
{
    return("Error "+ex.getMessage());
}
}
public String closeSite(String e, String d, String c)
{
    driver.close();
    return("Done");
}
public String screenshot() throws Exception
{
    Date d=new Date();
    SimpleDateFormat s=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
    String x=s.format(d)+".png";
    File src=((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
    File dest=new File(x);
    FileHandler.copy(src,dest);
    return(x);
}
}

```

Step 9: Developing runner class

After completion of excel file sheets with tests, steps and methods class or classes, we need to develop a runner class in **main()** method like shown below:

```
public static void main(String[] args) throws Exception
{
    //Connect to Excel file
    File f=new File("smsdata.xls");
    //Open ".xls" file for test data reading
    Workbook rwb=Workbook.getWorkbook(f);
    Sheet rsh=rwb.getSheet(0); //0 means Sheet1(tests)
    int nour=rsh.getRows(); //Count of used rows
    Sheet rsh1=rwb.getSheet(1); //1 means Sheet2(steps)
    int nour1=rsh1.getRows();
    //Open Same excel file for writing
    WritableWorkbook wwb=Workbook.createWorkbook(f,rwb);
    WritableSheet wsh=wwb.getSheet(0); //0 means Sheet1
    int nouc=wsh.getColumns();
    WritableSheet wsh1=wwb.getSheet(1); //1 means Sheet2
    int nouc1=wsh1.getColumns();
    //Set font style, colour and cell alignment for Heading
    WritableFont wf=new WritableFont(WritableFont.TIMES,11,WritableFont.BOLD);
    wf.setColour(Colour.RED);
    WritableCellFormat wcf=new WritableCellFormat(wf);
    wcf.setAlignment(Alignment.CENTRE);
    //Set font style, colour and alignment for test passed
    WritableFont wf1=new WritableFont(WritableFont.TIMES,11);
    wf1.setColour(Colour.GREEN);
    WritableCellFormat wcf1=new WritableCellFormat(wf1);
    wcf1.setAlignment(Alignment.CENTRE);
    //Set font style, colour and alignment for test failed
    WritableFont wf2=new WritableFont(WritableFont.TIMES,11);
    wf2.setColour(Colour.RED);
    wf2.setItalic(true);
    WritableCellFormat wcf2=new WritableCellFormat(wf2);
    wcf2.setAlignment(Alignment.CENTRE);
    //Take results heading as date and time format
    Date dt=new Date();
    SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
    String x="Results on "+sdf.format(dt);
    //set name to result column in sheet 1
    Label l=new Label(nouc,0,x,wcf);
    wsh.addCell(l);
    //set name to result column in sheet 2
    Label l1=new Label(nouc1,0,x,wcf);
    wsh1.addCell(l1);
    //create object to methods class
    MyMethods1 mm=new MyMethods1();
    //collect methods info using methods class object
    Method[] m=mm.getClass().getMethods();
```

```

//keyword driven
//1st row (index is 0) have names of column in sheet 1
for(int i=1;i<nour;i++)//from 2nd row (index=1)
{
    int flag=0;
    //Get testid and mode from sheet 1
    String testid=rsh.getCell(0, i).getContents();
    String mode=rsh.getCell(2, i).getContents();
    if(mode.equalsIgnoreCase("yes"))
    {
        //1st row (index is 0) have names of column in sheet 2
        for(int j=1;j<nour1;j++)
        {
            //Get stepid from sheet 2
            String stepid=rsh1.getCell(0,j).getContents();
            if(stepid.equalsIgnoreCase(testid))
            {
                //take step details from sheet 2
                String mn=rsh1.getCell(2,j).getContents();
                String e=rsh1.getCell(3,j).getContents();
                String d=rsh1.getCell(4,j).getContents();
                String c=rsh1.getCell(5,j).getContents();
                int t=mn.length;
                for(int k=0;k<t;k++)
                {
                    //compare each method name with method key in excel sheet
                    if(m[k].getName().equals(mn))
                    {
                        String r=(String) m[k].invoke(mm,e,d,c);
                        Label l2=new Label(nouc1,j,r,wcf1);
                        wsh1.addCell(l2);
                        // if browser name is unknown
                        if(r.equalsIgnoreCase("Unknown browser"))
                        {
                            wwb.write();
                            wwb.close();
                            rwb.close();
                            System.exit(0);//stop run
                        }
                        // if any error or validation fail
                        if(r.contains("failed") || r.contains("Error"))
                        {
                            flag=1;
                            Label l3=new Label(nouc1,j,r,wcf2);
                            wsh1.addCell(l3);
                        }
                        break; //terminate from for loop of k
                    } // if closing
                } //for k closing
            } // if closing
        else
        {
            break; //terminate from for loop of j if stepid != testid
        }
    } //for j closing
}

```

```

        if(flag==0)
        {
            Label l2=new Label(nouc,i,testid+" Passed",wcf1);
            wsh.addCell(l2);
        }
        else
        {
            Label l2=new Label(nouc,i,testid+" failed",wcf2);
            wsh.addCell(l2);
        }
    } // if closing
} // for i closing
// Auto size/fit results column in sheet 1
CellView cv=rsh.getColumnView(nouc);
cv.setAutoSize(true);
wsh.setColumnView(nouc,cv);
// Auto size/fit results column in sheet 2
CellView cv1=rsh1.getColumnView(nouc1);
cv1.setAutoSize(true);
wsh1.setColumnView(nouc1,cv1);
//Save & close file
wwb.write();
wwb.close();rwb.close();
}

```

Note 1:

MyMethods1 mm=new MyMethods1();

Here: MyMethods1 is a class, which have reusable methods with automation code.

mm is an object to MyMethods1 class.

Method[] m=mm.getClass().getMethods();

Here: Method[] is a class in “java.lang.reflect” package.

mm.getClass().getMethods() is to collect all methods in “MyMethods1” class into an array.

Note 2:

String r=(String) m[k].invoke(mm,e,d,c);

Here: r is a return value of method after execution

In m[k], m is the methods collection/array and k is the index of corresponding method to be executed.

mm is an object of “MyMethods1” class, which have methods.

e,d,c are values to arguments of corresponding method.

Note 3:

Sometimes we need to maintain separate methods with bodies for some elements instead of **fill()** and **click()**.

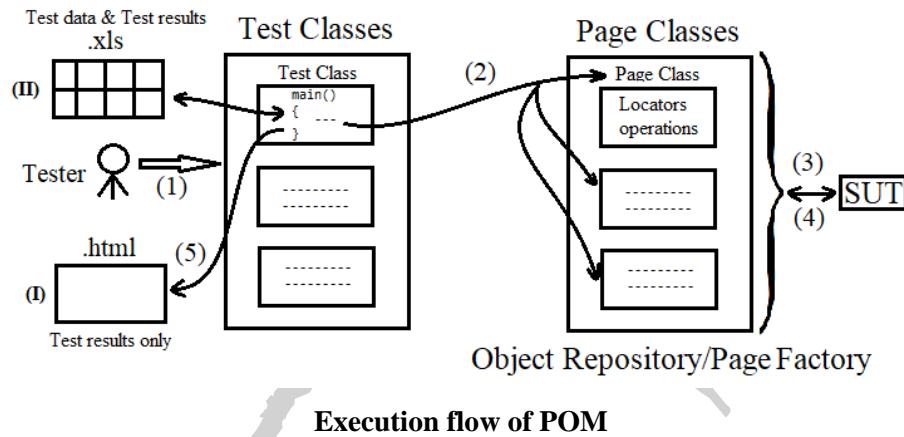
For example dropdown, calendars, dropdowns, sliders, pop-ups, alerts... etc.

```
public String select(String e, String d, String c) throws Exception
{
    String t=driver.findElement(By.xpath(e)).getTagName();
    if(t.equalsIgnoreCase("select"))
    {
        -----
        -----
        return("Done");
    }
    else
    {
        -----
        -----
        return("Done");
    }
}
```

X. POM

[Page Object Model Framework]

- ❖ Framework is a process of automation. POM framework is a standard process for website automation using Selenium Web Driver. From this framework, we need to implement page classes and test classes.



- ❖ POM is also called as object repository framework/page factory framework.
- ❖ This idea will go to **outdate** as per **Simon Stewart announcement**.

Case Study:

Implement test scripts for below scenarios related to Gmail site in POM framework.

Test Scenario 1:

- Launch Gmail site
- Enter user id & click next
- If user id is valid
 - ✓ Password will be displayed
- If user id is invalid
 - ✓ Error message will be displayed
- If user id is blank
 - ✓ Error message will be displayed
- Close site

Test data for Test Scenario 1:

Gmailwithpom.txt	
,	invalid
xxxxxx	invalid
xxxxxx	invalid
xxxxxx	valid

→ Put test results in HTML file using ExtentReports.

Test Scenario 2:

- Launch Gmail site
- Enter user id as valid e.g.(dineshreddy2995@gmail.com)
- Click next
- Enter password and click next
- If password is valid
 - ✓ Compose will be displayed
- If password is invalid
 - ✓ Error message will be displayed
- If password is blank
 - ✓ Error message will be displayed
- Close site

Test data for Test Scenario 2:

Gmailwithpom	
Password	Criteria
,	invalid
xxxxx	invalid
xxxxx	valid

→ Put test results in next to last column in Excel file.

Step 1: Install JDK8 (Create JAVA_HOME environment variable and extend path variable value).

Step 2: Download and launch Eclipse IDE (create a new folder and use it as workspace folder for Eclipse IDE)

Step 3: Create new Java project with a package in Eclipse IDE (check the version of java compiler and JRE of corresponding java project).

Step 4: Download and associate required jars to java project (for a website testing, we need SWD jars, Extent Reports, jxl ...etc).

***Step 5:** Create two packages in java project, such as page classes and test classes.

Step 6: Developing Page classes

- Open a page in website under testing.
- Apply inspect on corresponding page elements to get details.
- Right click pages package in Eclipse IDE project.
- Select “New” and select “Class”.
- Enter a name (Class name starts with uppercase) & Click “finish”.
- Type locators to elements and operators to elements like shown below.

```
package pageclasses;

public class HomePage
{
    @FindBy(name="identifier")
    public WebElement uid;

    @FindBy(xpath="//span[text()='Next']")
    public WebElement uidNext;

    @FindBy(xpath="//div[contains(text(),'Enter an email')]")
    public WebElement uidBlankError;

    @FindBy(xpath="//div[contains(text(),'Enter a valid email')]")
    public WebElement uidWithSpacesError;

    @FindBy(xpath="//div[contains(text(),'find your Google Account')]")
    public WebElement uidInvalidError;

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

    public void fillUid(String x)
    {
        uid.sendKeys(x);
    }

    public void uidClickNext()
    {
        uidNext.click();
    }
}
```

```

package pageclasses;

public class LoginPage
{
    @FindBy(name="password")
    public WebElement pswd;

    @FindBy(xpath="//span[text()='Next ']")
    public WebElement pswdNext;

    @FindBy(xpath="//div[text()='Enter a password']")
    public WebElement pswdBlankError;

    @FindBy(xpath="//div[contains(text(),'Wrong password')]")
    public WebElement pswdInvalidError;

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

    public void fillPswd(String x)
    {
        pswd.sendKeys(x);
    }

    public void pswdClickNext()
    {
        pswdNext.click();
    }
}

package pageclasses;

public class ComposePage
{
    @FindBy(xpath="//div[text()='Compose']")
    public WebElement compose;

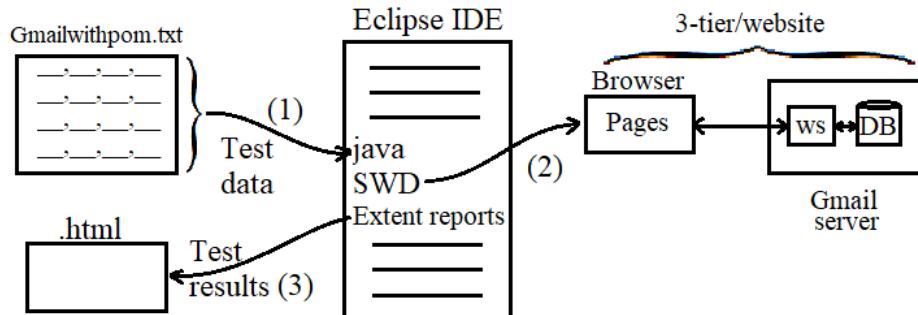
    public ComposePage(WebDriver driver)
    {
        PageFactory.initElements(driver,this);
        //here PageFactory is a static class in SWD jars and this is an
        //object to current class
    }
}

```

Step 7: Developing Test classes

- Right click Test package in Eclipse IDE project.
- Select “New” and select “Class”.
- Enter a name (Class name starts with uppercase).

- Select main() method and Click “finish”.
- Provide code for main() method like shown below.



```

package testclasses;

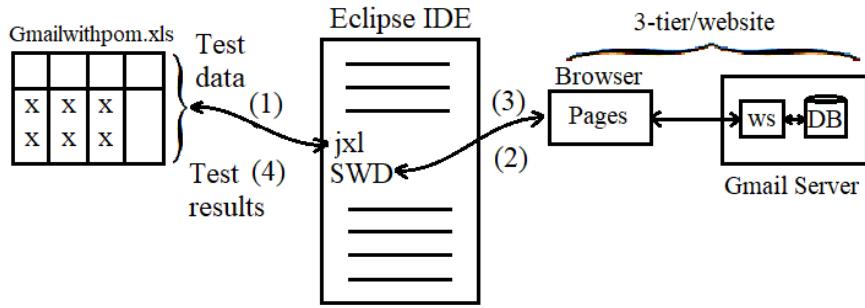
public class Test1
{
    public static void main(String[] args) throws Exception
    {
        //Open ".txt" file for test data reading
        File f=new File("Gmailwithpom.txt");
        FileReader fr=new FileReader(f);
        BufferedReader br=new BufferedReader(fr);
        //create HTML reports file
        ExtentReports er=new ExtentReports("GmailwithPOM.html",false);
        ExtentTest et=er.startTest("Gmail UID testing");
        //Data Driven with Page Object Model
        ChromeDriver driver=null;
        String s="";
        while((s=br.readLine())!=null)
        {
            String[] t=s.split(",");
            try
            {
                //Launch site
                System.setProperty("webdriver.chrome.driver",
                    "D:\\DineshReddy\\chromedriver.exe");
                driver=new ChromeDriver();
                driver.manage().window().maximize();
                driver.get("http://www.gmail.com");
                //Create objects to page classes
                HomePage hp=new HomePage(driver);
                LoginPage lp=new LoginPage(driver);
                //Automation Code
                WebDriverWait w=new WebDriverWait(driver,20);
                w.until(ExpectedConditions.visibilityOf(hp.uid));
                hp.fillUid(t[0]);
                w.until(ExpectedConditions.elementToBeClickable(hp.uidNext));
                hp.uidClickNext();
            }
        }
    }
}

```

```

//validations
if(t[0].length()==0 && t[1].equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOf(hp.uidBlankError));
    et.log(LogStatus.PASS, "Blank user id test passed");
}
else if(t[0].contains(" ") && t[1].equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOf(hp
        .uidWithSpacesError));
    et.log(LogStatus.PASS, "Invalid user id test passed
        (valid mail error)");
}
else if(t[1].equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOf(hp
        .uidInvalidError));
    et.log(LogStatus.PASS, "Invalid user id test passed
        (Couldn't find error)");
}
else if(t[1].equalsIgnoreCase("valid"))
{
    w.until(ExpectedConditions.visibilityOf(lp.pswd));
    et.log(LogStatus.PASS, "Valid user id test passed");
}
else
{
    Date d=new Date();
    SimpleDateFormat sdf=new SimpleDateFormat(
        "dd-MM-yyyy-hh-mm-ss");
    String c=sdf.format(d)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(c);
    FileHandler.copy(src,dest);
    et.log(LogStatus.FAIL,"UID test failed"+et
        .addScreenCapture(c));
}
//close site
driver.close();
}
catch(Exception ex)
{
    //close site
    driver.close();
    et.log(LogStatus.ERROR, "Error in code"+ex.getMessage());
}
}
//close text file
br.close();
fr.close();
//Save Results in HTML
er.endTest(et);
er.flush();
}
}

```



```

package testclasses;

public class Test3
{
    public static String Screenshot(ChromeDriver driver) throws Exception
    {
        Date d=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yy-hh-mm-ss");
        String fname=sdf.format(d)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(fname);
        FileHandler.copy(src, dest);
        return fname;
    }
    public static void main(String[] args) throws Exception
    {
        ChromeDriver driver=null;
        //Open ".xls" file for test data reading
        File f=new File("Gmailwithpom1.xls");
        Workbook rwb=Workbook.getWorkbook(f);
        Sheet rsh=rwb.getSheet(0); //0 means Sheet1
        int nouri=rsh.getRows(); //Count of used rows
        //Open Same excel file for writing
        WritableWorkbook wwb=Workbook.createWorkbook(f,rwb);
        WritableSheet wsh=wwb.getSheet(0); //0 means Sheet1
        //Set font style, colour and cell alignment for Heading
        WritableFont wf=new WritableFont(WritableFont.TIMES,11,WritableFont.BOLD);
        wf.setColour(Colour.BLUE);
        WritableCellFormat wcf=new WritableCellFormat(wf);
        wcf.setAlignment(Alignment.CENTRE);
        //Set font style, colour and alignment for test passed
        WritableFont wf1=new WritableFont(WritableFont.TIMES,11);
        wf1.setColour(Colour.GREEN);
        WritableCellFormat wcf1=new WritableCellFormat(wf1);
        wcf1.setAlignment(Alignment.CENTRE);
        //Set font style, colour and alignment for test failed
        WritableFont wf2=new WritableFont(WritableFont.TIMES,11);
        wf2.setColour(Colour.RED);
        WritableCellFormat wcf2=new WritableCellFormat(wf2);
        wcf2.setAlignment(Alignment.CENTRE);
        int nouc=wsh.getColumns();
        //Take results heading as date and time format
        Date d=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
    }
}

```

```

String x="Results on "+sdf.format(d);
Label l=new Label(nouc,0,x,wcf);
wsh.addCell(l);
for(int i=1;i<nour;i++)
{
    String pswd=rsh.getCell(0,i).getContents();
    String pswdc=rsh.getCell(1,i).getContents();
    try
    {
        //Launch Gmail Site
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.gmail.com");
        //Create objects to page classes
        HomePage hp=new HomePage(driver);
        LoginPage lp=new LoginPage(driver);
        ComposePage cp=new ComposePage(driver);
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOf(hp.uid));
        //Do login with valid data
        hp.fillUid("xxxxxx@gmail.com");
        w.until(ExpectedConditions.elementToBeClickable(lp.pswd));
        lp.pswdClickNext();
        w.until(ExpectedConditions.visibilityOf(lp.pswd));
        lp.fillPswd(pswd);
        w.until(ExpectedConditions.elementToBeClickable(lp.pswdNext));
        lp.pswdClickNext();
        if(pswd.length()==0) //blank password
        {
            w.until(ExpectedConditions.visibilityOf(lp.pswdBlankError));
            Label l2=new Label(nouc,i,"Blank pswd test passed",wcf1);
            wsh.addCell(l2);
        }
        else if(pswdc.equalsIgnoreCase("Invalid")) //invalid password
        {
            w.until(ExpectedConditions.visibilityOf(lp
                .pswdInvalidError));
            Label l2=new Label(nouc,i,"Invalid pwd test passed",wcf1);
            wsh.addCell(l2);
        }
        else if(pswdc.equalsIgnoreCase("Valid")) //valid password
        {
            w.until(ExpectedConditions.visibilityOf(cp.compose));
            Label l2=new Label(nouc,i,"Valid pswd test passed",wcf1);
            wsh.addCell(l2);
        }
        else
        {
            Label l2=new Label(nouc,i,"Gmail login test failed",wcf2);
            wsh.addCell(l2);
        }
        driver.close();
    }
}

```

```

        catch(Exception ex)
        {
            //close site
            driver.close();
            Label l1=new Label(nouc,i,"Error in code"+ex.getMessage(),wcf2);
            wsh.addCell(l1);
        }
    }
    CellView cv=rsh.getColumnView(nouc);
    cv.setAutoSize(true);
    wsh.setColumnView(nouc, cv);
    //Save excel file
    wwb.write();
    //close file
    rwb.close();
    wwb.close();
}
}

```

Note 1:

In Page Object Model framework we can create page classes with locators and operations related to elements of corresponding pages in a website. Here we need to use below annotations related to Selenium.

- @FindBy
- @FindBys
- @FindAll

(In java/jars, init cap will come only for classes, exceptions & annotations)

Note 2:

“@FindBy” annotation is related to one element where as “@FindBys” and “@FindAll” are related to collection of elements.

Example 1:

```

@FindBy(name="identifier")
public WebElement uid;
//Here uid is DOM for an element

```

Example 2:

```

@FindBy({@FindBy(className="raDiv"), @FindBy(tagName="div")})
public List<WebElement> el;

```

Here: collection of elements, which are matched with both criteria's like className & tagName.

Example 3:

```

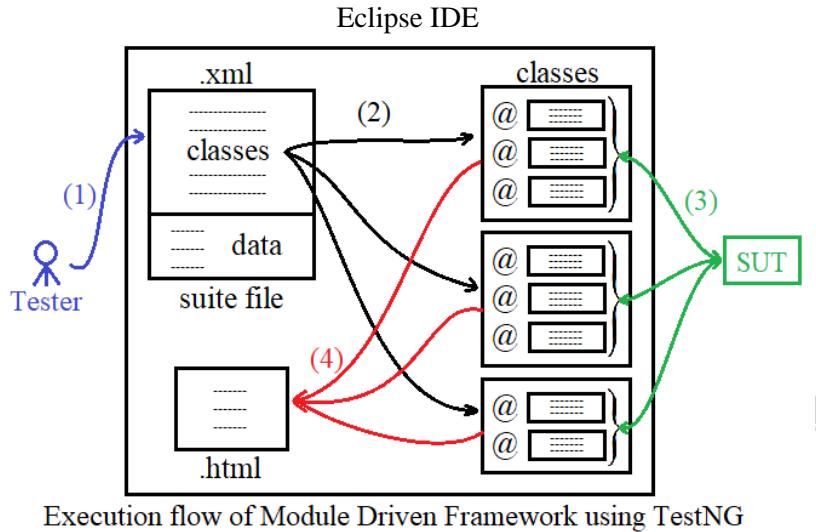
@FindBy({@FindBy(className="raDiv"), @FindBy(tagName="div")})
public List<WebElement> el;

```

Here: collection of elements, which are matched with any one or both criteria's.

XI. Module Driven Framework using “TestNG”

[NG stands for Next Generation]



- ❖ TestNG is inspired from Junit (Junit will be used by Java Developers in unit testing).
- ❖ TestNG is available as “External Jar” and **built-in plugin to Eclipse IDE**. So, Eclipse IDE people can use TestNG plugin.
- ❖ TestNG can provide annotations, assertions, reporting, listeners & xml suite file.

Module Driven Framework development process:

Step 1: Download and install JD8 and create “JAVA_HOME” environment variable and update “path” variable value.

Step 2: Download and launch Eclipse IDE, provide personal folder as workspace.

Step 3: Create java project & check for java compiler and JRE as latest versions.

Step 4: Install TestNG plugin to Eclipse IDE

- Go to help menu in Eclipse IDE
- Select Eclipse market place
- Enter “TestNG for eclipse” in find box and click Go/Enter
- Click install button for “TestNG for eclipse”
- Select TestNG checkbox and deselect remaining checkboxes
- Click confirm
- Accept license agreement & Click next until finish
- Restart Eclipse IDE

Step 5: Select TestNG for project

- Right click on previously created TestNG project
- Go to properties
- Select java build path and select libraries
- Click on “Add Library”
- Select “TestNG” & click “Finish”
- Click “apply & close”

Step 6: Associate required jars

Ex: Selenium Web Driver jars... etc.

Step 7: Developing Scripts

- Right click on project which is related to TestNG
- Select “New” and select “Package”
- Enter a name and click “finish”
- Right click on that package
- Select “TestNG” and then create “TestNG class”
- Enter a name to class and click finish
- Write automation code in corresponding class’s annotated method

```
public class Test01
{
    @Test
    public void f()
    {
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        ChromeOptions co=new ChromeOptions();
        co.addArguments("disable-notifications");
        ChromeDriver driver=new ChromeDriver(co);
        driver.manage().window().maximize();
        driver.get("http://www.way2sms.com");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By.name("mobileNo")));
        if(driver.getTitle().contains("Free SMS"))
        {
            Reporter.Log("Title test passed");
            Assert.assertTrue(true);
        }
        else
        {
            Reporter.Log("Title test failed");
            Assert.assertTrue(false);
        }
        driver.close();
    }
}
```

- TestNG classes can allow more than one “@Test” annotated methods with priorities. Here priorities can start with ‘0’ or ‘1’.
- If there is no priorities for methods, TestNG can run corresponding methods by following alphabetical order of method names.

```
public class Test02
{
    public ChromeDriver driver;
    public WebDriverWait w;
    @Test(priority=1)
    public void launch()
    {
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        ChromeOptions co=new ChromeOptions();
        co.addArguments("disable-notifications");
        driver=new ChromeDriver(co);
        driver.manage().window().maximize();
        driver.get("http://www.way2sms.com");
        WebDriverWait w=new WebDriverWait(driver,20);
        w.until(ExpectedConditions.visibilityOfElementLocated(By.name("mobileNo")));
    }

    @Test(priority=2)
    public void validateLogin()
    {
        if(driver.getTitle().contains("Free SMS"))
        {
            Reporter.Log("Title test passed");
            Assert.assertTrue(true);
        }
        else
        {
            Reporter.Log("Title test failed");
            Assert.assertTrue(false);
        }
    }

    @Test(priority=3)
    public void closeSite()
    {
        driver.close();
    }
}
```

- From previous examples, TestNG based classes are having one or more annotated methods (no **main()** method). So, we need to follow different way to run those classes like:

- ✓ Right click on TestNG based class
- ✓ Click Run As TestNG Test

- By default TestNG can create a built result file in **html** format. TestNG can maintain this **html** file in “**test-output**” folder in corresponding project folder.

Ex: D:\DineshReddy\TestNG\test-output\index.html

- TestNG can maintain previous results in “old” folder in ‘test-output’ folder.
- “@Test” annotation in TestNG can provide properties/attributes like:
 - priority → To execute methods in order
 - enabled → To skip corresponding methods from execution

```
@Test(enabled=false)
public void method()
{
    -----
    -----
    -----
}
```

dependsOnMethods → Run a method when related all methods are passed, if any one related method failed then this method cannot run.

```
@Test(dependsOnMethods= {"related method name1,name2,..."})
public void method()
{
    -----
    -----
    -----
}
```

dependsOnGroups → Run a method when all methods listed in corresponding groups are passed

```
@Test(groups= {"smoketest"})
public void method1()
{
    -----
    -----
    -----
}

@Test(groups= {"smoketest"})
public void method2()
{
    -----
    -----
    -----
}

@Test(dependsOnGroups= {"smoketest"})
public void method3()
{
    -----
    -----
    -----
}
```

dataProvider → To provide data to method for data driven
alwaysRun → To run corresponding method everytime (without skip & even if every method failed). By default “alwaysRun” is true.
expectedExceptions → To pass a method when specified exception were raised.

```
@Test(expectedExceptions= {NoSuchElementException.class})
public void method()
{
    -----
    -----
    -----
}
```

invocationCount → To run corresponding method for specified number of times

```
@Test(invocationCount=5)
public void method()
{
    -----
    -----
    -----
}
```

timeOut → When corresponding method execution need to finish with in specified time. If method execution took more than specified time, TestNG can show that method as failed.

```
@Test(timeOut=5000)
public void method()
{
    -----
    -----
    -----
}
```

Step 8: Developing suite file

To run multiple classes (which have multiple annotated methods), we can use xml suite file this file can allow us to run classes sequentially or parallel.

- Right click on package in project
- Select “TestNG” and then “create TestNG class”
- Enter a name to suite file with .xml extension
- Click “finish”
- Click on “source” in bottom side of suite file
- Change **xml** file by adding classes to be executed

Ex 1:

```
<suite name="Suite" parallel="false">
    <test name="Test">
        <classes>
            <class name="mypack.Test01"/>
            <class name="mypack.Test02"/>
        </classes>
    </test>
</suite>
```

Here: `parallel="false"` means sequential execution & `mypack` is a package name in TestNG project & `Test4`, `Test5` are classes under that package.

Ex 2:

```
<suite name="Suite" parallel="classes" thread-count="2">
    <test name="Test">
        <classes>
            <class name="mypack.Test01"/>
            <class name="mypack.Test02"/>
        </classes>
    </test>
</suite>
```

Here: `parallel="classes"` `thread-count="2"` means TestNG will run all the methods in the same class in the same thread, but each class will be run in a separate thread.

Ex 3:

```
<suite name="Suite" parallel="tests" thread-count="2">
    <test name="Test">
        <classes>
            <class name="mypack.Test01"/>
        </classes>
    </test>
    <test name="Test1">
        <classes>
            <class name="mypack.Test02"/>
        </classes>
    </test>
</suite>
```

Here: `parallel="test"` `thread-count="2"` means TestNG will run `<test>` tags in parallel based on thread-count.

- One .xml file is having only one `<suite>` tag
- One `<suite>` tag is having multiple `<test>` tags
- One `<test>` tag is having multiple `<class>` tags
- One class is having multiple annotated methods.

Step 9: Adding different “TestNG” annotations to methods in classes for code reusability.

In general, test engineers can implement automation code by writing in multiple classes with multiple @Test annotated methods. To run @Test annotated methods in order, test engineers can use @Test annotated properties like priority, dependsOnMethods, enabled, groups, dependsOnGroups...etc.

To improve code reusability in between classes and in corresponding class, we need to use other annotations:

@BeforeSuite

The annotated methods will be run only once before all tests in this suite have run.

@AfterSuite

The annotated method will be run only once after all tests in this suite have run.

@BeforeTest

The annotated method will be run before running all classes in <test> tag.

@AfterTest

The annotated method will be run after running all classes in <test> tag.

The above four will work for classes.

@BeforeClass

The annotated method will be run only once before the first test method in the current class is invoked.

@AfterClass

The annotated method will be run only once after all the test methods in the current class have run.

@BeforeGroups

The annotated method will be run before running all the methods related to groups in current class.

@AfterGroups

The annotated method will be run after running all the methods related to groups in current class.

@BeforeMethod

The annotated method will be run before running each @Test annotated method in current class.

@AfterMethod

The annotated method will be run after running each @Test annotated method in current class.

The above six methods will work for corresponding class, which have methods with above annotations.

Example 1:

Suite file:

```
<suite name="Suite" parallel="false">
    <test name="Test">
        <classes>
            <class name="mypack.Test1"/>
            <class name="mypack.Test2"/>
        </classes>
    </test>
</suite>
```

Test1:

```
public class Test1
{
    @Test(priority=1)
    public void method1()
    {
        System.out.println("in method1 in Test1");
    }

    @Test(priority=2)
    public void method2()
    {
        System.out.println("in method2 in Test1");
    }

    @BeforeSuite
    public void beforeSuite()
    {
        System.out.println("in BeforeSuite");
    }

    @BeforeTest
    public void beforeTest()
    {
        System.out.println("in BeforeTest");
    }

    @BeforeClass
    public void beforeClass()
    {
        System.out.println("in BeforeClass in Test1");
    }
    @BeforeMethod
    public void beforeMethod()
    {
        System.out.println("in BeforeMethod in Test1");
    }
    @AfterMethod
    public void afterMethod()
    {
        System.out.println("in AfterMethod in Test1");
    }
}
```

```

@AfterClass
public void afterClass()
{
    System.out.println("in AfterClass in Test1");
}

@AfterTest
public void afterTest()
{
    System.out.println("in AfterTest");
}

@AfterSuite
public void afterSuite()
{
    System.out.println("in AfterSuite");
}

```

Test2:

```

public class Test2
{
    @Test(priority=3)
    public void method1()
    {
        System.out.println("in method1 in Test2");
    }

    @Test(priority=4)
    public void method2()
    {
        System.out.println("in method2 in Test2");
    }

    @BeforeClass
    public void beforeClass()
    {
        System.out.println("in BeforeClass in Test2");
    }

    @BeforeMethod
    public void beforeMethod()
    {
        System.out.println("in BeforeMethod in Test2");
    }

    @AfterMethod
    public void afterMethod()
    {
        System.out.println("in AfterMethod in Test2");
    }
}

```

```

    @AfterClass
    public void afterClass()
    {
        System.out.println("in AfterClass in Test2");
    }
}

```

Output:

```

in BeforeSuite
in BeforeTest
in BeforeClass in Test1
in BeforeMethod in Test1
in method1 in Test1
in AfterMethod in Test1
in BeforeMethod in Test1
in method2 in Test1
in AfterMethod in Test1
in AfterClass in Test1
in BeforeClass in Test2
in BeforeMethod in Test2
in method1 in Test2
in AfterMethod in Test2
in BeforeMethod in Test2
in method2 in Test2
in AfterMethod in Test2
in AfterClass in Test2
in AfterTest
in AfterSuite

```

Note 1:

In xml suite file, parallel attribute can take “tests” or “classes” or “methods” to run code in parallel. Here we need to add **thread-count** attribute.

Note 2:

While using TestNG framework we have two ways to implement scripts.

Way1	Way2
<ul style="list-style-type: none"> ❖ Using <code>@Test</code> annotation methods in classes. ❖ Using properties with <code>@Test</code> annotation. <p>➤ No more code reusability</p> <p>➤ Easy to implement</p>	<ul style="list-style-type: none"> ❖ Using <code>@Test</code> annotation along with other annotations for methods in classes. <p>➤ Code reusability will be possible, but bit of confusion in execution flow</p>

Step 10: Develop test scripts for Data Driven (parameterization).

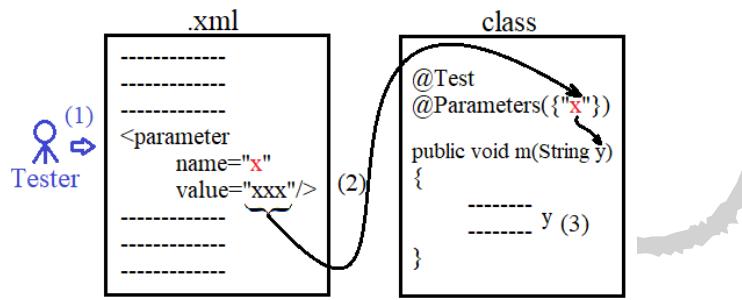
Parameterizations in TestNG are 2 types.

(1) @Parameters

(2) @DataProvider

(1) @Parameters:

This TestNG annotation can integrate xml suite file and @Test annotated methods in classes for data transfer.



Example 1:

Way2sms website login functionality test automation by using valid and invalid data.

Suite file:

```
<suite name="Suite" parallel="false">
    <test name="Test1">
        <parameter name="uid" value="" />
        <parameter name="uidc" value="invalid" />
        <parameter name="pswd" value="XXXXXXXXXX" />
        <parameter name="pswdc" value="valid" />
        <classes>
            <class name="mypack.Test6" />
        </classes>
    </test>
    <test name="Test2">
        <parameter name="uid" value="9491" />
        <parameter name="uidc" value="invalid" />
        <parameter name="pswd" value="XXXXXXXXXX" />
        <parameter name="pswdc" value="valid" />
        <classes>
            <class name="mypack.Test6" />
        </classes>
    </test>
    <test name="Test3">
        <parameter name="uid" value="8446551112" />
        <parameter name="uidc" value="invalid" />
        <parameter name="pswd" value="XXXXXXXXXX" />
        <parameter name="pswdc" value="valid" />
        <classes>
            <class name="mypack.Test6" />
        </classes>
    </test>
```

```

<test name="Test4">
    <parameter name="uid" value="9491260836"/>
    <parameter name="uidc" value="invalid"/>
    <parameter name="pswd" value="XXXXXXXXXX"/>
    <parameter name="pswdc" value="valid"/>
    <classes>
        <class name="mypack.Test6"/>
    </classes>
</test>
<test name="Test5">
    <parameter name="uid" value="9491947838"/>
    <parameter name="uidc" value="valid"/>
    <parameter name="pswd" value="" />
    <parameter name="pswdc" value="invalid"/>
    <classes>
        <class name="mypack.Test6"/>
    </classes>
</test>
<test name="Test6">
    <parameter name="uid" value="9491947838"/>
    <parameter name="uidc" value="valid"/>
    <parameter name="pswd" value="aiusiaav"/>
    <parameter name="pswdc" value="invalid"/>
    <classes>
        <class name="mypack.Test6"/>
    </classes>
</test>
<test name="Test7">
    <parameter name="uid" value="9491947838"/>
    <parameter name="uidc" value="valid"/>
    <parameter name="pswd" value="XXXXXXXXXX"/>
    <parameter name="pswdc" value="valid"/>
    <classes>
        <class name="mypack.Test6"/>
    </classes>
</test>
</suite>

```

Class:

```

public class Test6
{
    public ChromeDriver driver;
    public WebDriverWait w;
    @BeforeMethod
    public void launch()
    {
        System.setProperty("webdriver.chrome.driver",
                           "D:\\DineshReddy\\chromedriver.exe");
        ChromeOptions co=new ChromeOptions();
        co.addArguments("disable-notifications");
        driver=new ChromeDriver(co);
        driver.manage().window().maximize();
        driver.get("http://www.way2sms.com");
    }
}

```

```

@Test
@Parameters({"uid","uidc","pswd","pswdc"})
public void doLogin(String a, String b, String c, String d)
{
    w=new WebDriverWait(driver,20);
    w.until(ExpectedConditions.visibilityOfElementLocated(By.name("mobileNo")));
    driver.findElement(By.name("mobileNo")).sendKeys(a); //Do login
    driver.findElement(By.name("password")).sendKeys(c);
    driver.findElement(By.xpath("//button[contains(text(),'Login')][1]")).click();
    try
    {
        //validations
        if(a.length()==0 && b.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[text()='Enter your mobile number']")));
            Reporter.Log("Blank mobile number test passed");
            Assert.assertTrue(true);
        }
        else if(a.length()<10 && b.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[text()='Enter valid mobile number']")));
            Reporter.Log("Wrong size mobile number test passed");
            Assert.assertTrue(true);
        }
        else if(b.equalsIgnoreCase("invalid"))
        {
            try
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                    .xpath("//b[contains(text(),'Incorrect number')]")));
                Reporter.Log("Invalid mobile number test passed");
                Assert.assertTrue(true);
            }
            catch(Exception ex)
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                    .xpath("//b[contains(text(),'not register')]")));
                Reporter.Log("Mobile number not registered test passed");
                Assert.assertTrue(true);
            }
        }
        else if(b.equalsIgnoreCase("valid") && c.length()==0 &&
                d.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[text()='Enter password'][1]")));
            Reporter.Log("Blank password test passed");
            Assert.assertTrue(true);
        }
        else if(b.equalsIgnoreCase("valid") && d.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[contains(text(),'Incorrect number')]")));

```

```

        Reporter.Log("Invalid password test passed");
        Assert.assertTrue(true);
    }
    else if(b.equalsIgnoreCase("valid") && d.equalsIgnoreCase("valid"))
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//div[text()='SendSMS ']")));
        Reporter.Log("Login test passed");
        Assert.assertTrue(true);
    }
    else
    {
        Date date=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
        String sname=sdf.format(date)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(sname);
        FileHandler.copy(src, dest);
        sname="D:\\DineshReddy\\TestNG\\"+sname;
        String image="<img src=\"file:///"+sname+"\" alt=\"\" />";
        Reporter.Log("Login test failed");
        Reporter.Log(image);
        Assert.assertTrue(false);
    }
}
catch(Exception ex)
{
    Reporter.Log("Error in code"+ex.getMessage());
    Assert.assertTrue(false);
}
}
@AfterMethod
public void closeSite()
{
    driver.close();
}
}

```

(2) @DataProvider:

Like “@Parameters”, “@DataProvider” annotations is useful to parameterize @Test annotated methods in corresponding class.

Example 1:

Way2sms login functional test by using valid & invalid data.

Suite file:

```

<suite name="Suite" parallel="false">
    <test name="Way2sms Login Testing">
        <classes>
            <class name="mypack.Test8"/>
        </classes>
    </test>
</suite>

```

Class:

```
public class Test8
{
    public ChromeDriver driver;
    public WebDriverWait w;

    @DataProvider(name="way2smsdata")
    public Object[][] smsData()
    {
        //Rows-Number of times has to be repeated
        //columns-Number of parameters in each test data
        Object[][] data=new Object[7][4];
        //1st row
        data[0][0]="";
        data[0][1]="invalid";
        data[0][2]="XXXXXXX";
        data[0][3]="valid";
        //2nd row
        data[1][0]="9491";
        data[1][1]="invalid";
        data[1][2]="XXXXXXX";
        data[1][3]="valid";
        //3rd row
        data[2][0]="9491260836";
        data[2][1]="invalid";
        data[2][2]="XXXXXXX";
        data[2][3]="valid";
        //4th row
        data[3][0]="8446551112";
        data[3][1]="invalid";
        data[3][2]="XXXXXXX";
        data[3][3]="valid";
        //5th row
        data[4][0]="9491947838";
        data[4][1]="valid";
        data[4][2]++;
        data[4][3]="invalid";
        //6th row
        data[5][0]="9491947838";
        data[5][1]="valid";
        data[5][2]="ysfdfylcj";
        data[5][3]="invalid";
        //7th row
        data[6][0]="9491947838";
        data[6][1]="valid";
        data[6][2]="XXXXXXX";
        data[6][3]="valid";
        //return array
        return(data);
    }
}
```

```

@BeforeMethod
public void launch()
{
    System.setProperty("webdriver.chrome.driver",
                       "D:\\DineshReddy\\chromedriver.exe");
    ChromeOptions co=new ChromeOptions();
    co.addArguments("disable-notifications");
    driver=new ChromeDriver(co);
    driver.manage().window().maximize();
    driver.get("http://www.way2sms.com");
}

@Test(dataProvider="way2smsdata")
public void doLogin(String a, String b, String c, String d)
{
    w=new WebDriverWait(driver,20);
    w.until(ExpectedConditions.visibilityOfElementLocated(By.name("mobileNo")));
    //Do login
    driver.findElement(By.name("mobileNo")).sendKeys(a);
    driver.findElement(By.name("password")).sendKeys(c);
    driver.findElement(By.xpath("//button[contains(text(),'Login')][1]")).click();
    try
    {
        //validations
        if(a.length()==0 && b.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[text()='Enter your mobile number']")));
            Reporter.Log("Blank mobile number test passed");
            Assert.assertTrue(true);
        }
        else if(a.length()<10 && b.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[text()='Enter valid mobile number']")));
            Reporter.Log("Wrong size mobile number test passed");
            Assert.assertTrue(true);
        }
        else if(b.equalsIgnoreCase("invalid"))
        {
            try
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                    .xpath("//b[contains(text(),'Incorrect number')]")));
                Reporter.Log("Invalid mobile number test passed");
                Assert.assertTrue(true);
            }
            catch(Exception ex)
            {
                w.until(ExpectedConditions.visibilityOfElementLocated(By
                    .xpath("//b[contains(text(),'not register')]")));
                Reporter.Log("Mobile number not registered test passed");
                Assert.assertTrue(true);
            }
        }
    }
}

```

```

        else if(b.equalsIgnoreCase("valid") && c.length()==0 &&
                 d.equalsIgnoreCase("invalid"))
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[text()='Enter password'][1]")));
        Reporter.Log("Blank password test passed");
        Assert.assertTrue(true);
    }
    else if(b.equalsIgnoreCase("valid") && d.equalsIgnoreCase("invalid"))
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[contains(text(),'Incorrect number')]")));
        Reporter.Log("Invalid password test passed");
        Assert.assertTrue(true);
    }
    else if(b.equalsIgnoreCase("valid") && d.equalsIgnoreCase("valid"))
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//div[text()='SendSMS']")));
        Reporter.Log("Login test passed");
        Assert.assertTrue(true);
    }
    else
    {
        //Take current date and time as image name
        Date date=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
        String sname=sdf.format(date)+".png";
        File src=driver.getScreenshotAs(OutputType.FILE);
        File dest=new File(sname);
        FileHandler.copy(src, dest);
        sname="D:\\DineshReddy\\TestNG\\"+sname;
        String image="![\"\"](\"file://")";
        Reporter.Log("Login test failed");
        Reporter.Log(image);
        Assert.assertTrue(false);
    }
}
catch(Exception ex)
{
    Reporter.log("Error in code"+ex.getMessage());
    Assert.assertTrue(false);
}
}

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

```

Example 2:

- *Associate “jxl” jar to project

Mobile No:	Mobile Criteria	Password	Password Criteria
BLANK	invalid	XXXX	Valid
8446551112	invalid	XXXX	Valid
9491260836	invalid	XXXX	Valid
9491947838	valid	RedmiNote5Pro	Invalid
9491947838	valid	BLANK	Invalid
9491947838	valid	XXXX	Valid
9491	invalid	XXXX	Valid

way2sms.xls

Suite file:

```
<suite name="Suite" parallel="false">
    <test name="Way2sms Login Testing with Excel data">
        <classes>
            <class name="mypack.Test9"/>
        </classes>
    </test>
</suite>
```

Class:

```
public class Test9
{
    public ChromeDriver driver;
    public WebDriverWait w;

    @DataProvider(name="way2smsdata")
    public Object[][] smsData()
    {
        //Connect & open Excel file for test data reading
        File f=new File("way2sms.xls");
        Workbook rwb=Workbook.getWorkbook(f);
        Sheet rsh=rwb.getSheet(0);
        int nour=rsh.getRows();
        //Rows-Number of times has to be repeated
        //columns-Number of parameters in each test data
        Object[][] data=new Object[nour-1][4];
        for(int i=1;i<nour;i++)
        {
            data[i-1][0]=rsh.getCell(0,i).getContents();
            data[i-1][1]=rsh.getCell(1,i).getContents();
            data[i-1][2]=rsh.getCell(2,i).getContents();
            data[i-1][3]=rsh.getCell(3,i).getContents();
        }
        //close file
        rwb.close();
        //return array
        return(data);
    }
}
```

(or)

```
int nouc=rsh.getColumns();
Object[][] data=new Object[nour-1][nouc];
for(int i=1;i<nour;i++)
{
    for(int j=0;j<nouc;j++)
    {
        data[i-1][j]=rsh.getCell(j,i).getContents();
    }
}
//close file
rwrb.close();
//return array
return(data);
}

@BeforeMethod
public void launch()
{
    System.setProperty("webdriver.chrome.driver",
                       "D:\\DineshReddy\\chromedriver.exe");
    ChromeOptions co=new ChromeOptions();
    co.addArguments("disable-notifications");
    driver=new ChromeDriver(co);
    driver.manage().window().maximize();
    driver.get("http://www.way2sms.com");
}

@Test(dataProvider="way2smsdata")
public void doLogin(String a, String b, String c, String d)
{
    w=new WebDriverWait(driver,20);
    w.until(ExpectedConditions.visibilityOfElementLocated(By.name("mobileNo")));
    //Do login
    driver.findElement(By.name("mobileNo")).sendKeys(a);
    driver.findElement(By.name("password")).sendKeys(c);
    driver.findElement(By.xpath("//button[contains(text(),'Login')][1]")).click();
    try
    {
        //validations
        if(a.length()==0 && b.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[text()='Enter your mobile number']")));
            Reporter.Log("Blank mobile number test passed");
            Assert.assertTrue(true);
        }
        else if(a.length()<10 && b.equalsIgnoreCase("invalid"))
        {
            w.until(ExpectedConditions.visibilityOfElementLocated(By
                .xpath("//b[text()='Enter valid mobile number']")));
            Reporter.Log("Wrong size mobile number test passed");
            Assert.assertTrue(true);
        }
    }
}
```

```

else if(b.equalsIgnoreCase("invalid"))
{
    try
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[contains(text(),'Incorrect number')]")));
        Reporter.Log("Invalid mobile number test passed");
        Assert.assertTrue(true);
    }
    catch(Exception ex)
    {
        w.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//b[contains(text(),'not register')]")));
        Reporter.Log("Mobile number not registered test passed");
        Assert.assertTrue(true);
    }
}
else if(b.equalsIgnoreCase("valid") && c.length()==0 &&
        d.equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[text()='Enter password'][1]")));
    Reporter.Log("Blank password test passed");
    Assert.assertTrue(true);
}
else if(b.equalsIgnoreCase("valid") && d.equalsIgnoreCase("invalid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//b[contains(text(),'Incorrect number')]")));
    Reporter.Log("Invalid password test passed");
    Assert.assertTrue(true);
}
else if(b.equalsIgnoreCase("valid") && d.equalsIgnoreCase("valid"))
{
    w.until(ExpectedConditions.visibilityOfElementLocated(By
        .xpath("//div[text()='SendSMS']")));
    Reporter.Log("Login test passed");
    Assert.assertTrue(true);
}
else
{
    Date date=new Date();
    SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
    String ssname=sdf.format(date)+".png";
    File src=driver.getScreenshotAs(OutputType.FILE);
    File dest=new File(ssname);
    FileHandler.copy(src, dest);
    ssname="D:\\DineshReddy\\TestNG\\"+ssname;
    String image=<img src=\"file:///"+ssname+"\" alt=\"\" />";
    Reporter.Log("Login test failed");
    Reporter.Log(image);
    Assert.assertTrue(false);
}
}

```

```

        catch(Exception ex)
    {
        Reporter.Log("Error in code"+ex.getMessage());
        Assert.assertTrue(false);
    }
}

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

```

Note 3:

Reporter & Assert are static classes related to TestNG. Here TestNG can provide an interface like “ITestListener”, which consists of methods declarations. To customize TestNG html results file content, we need to implement bodies for methods in “ITestListener”.

➤ Implementing bodies to “ITestListener”

```

public class Test1 implements ITestListener //Test1 is an implementing class
{
    @Override
    public void onTestStart(ITestResult result)
    {
        System.out.println("Method Testing Started");
    }

    @Override
    public void onTestSuccess(ITestResult result)
    {
        System.out.println("Method Testing Passed");
    }

    @Override
    public void onTestFailure(ITestResult result)
    {
        System.out.println("Method Testing Failed");
    }

    @Override
    public void onTestSkipped(ITestResult result)
    {
        System.out.println("Method Test has been skipped");
    }

    @Override
    public void onTestFailedButWithinSuccessPercentage(ITestResult result)
    {
        System.out.println("Passed with success percentage");
    }
}

```

```

@Override
public void onStart(ITestContext context)
{
    System.out.println("Main Test Started");
}

@Override
public void onFinish(ITestContext context)
{
    System.out.println("Main Test Ended");
}
}

```

➤ **Using ITestListener in Classes:**

```

@Listeners(mypack.Test1.class)
public class Test2
{
    @Test(priority=0)
    public void method1()
    {
        System.out.println("Method 1 in Test");
    }

    @Test(priority=1)
    public void method2()
    {
        System.out.println("Method 2 in Test");
    }

    @Test(priority=2)
    public void method3()
    {
        System.out.println("Method 3 in Test");
    }
}

```

Here: mypack is a package name in TestNG project; Test1 is a class which is having implemented bodies to “declared methods” in ITestListener interface.

➤ **Output:**

```

Main Test Started
Method Testing Started
Method 1 in Test
Method Testing Passed
Method Testing Started
Method 2 in Test
Method Testing Passed
Method Testing Started
Method 3 in Test
Method Testing Passed
Main Test Ended

```

Step 11: Run TestNG based classes from command prompt via suite file.

- Create a new folder. (Ex: D:\DineshReddy\Jars)
- Paste “all used jars in Test class’s methods” including “TestNG jar with all dependencies” in that folder (download TestNG jars with all dependencies in “jar-download.com”).
- Open notepad & type below like commands.

```
D:  
cd D:\DineshReddy\TestNG  
java -cp "D:\DineshReddy\Jars\*;D:\DineshReddy\TestNG\bin" org.testng.TestNG "D:\DineshReddy\TestNG\src\KeySuite.xml"
```

- Save file on desktop with filename.**bat** (change file type to all files).
- Double click on **.bat** file and observe output in index.html in “test-output” folder under project folder.

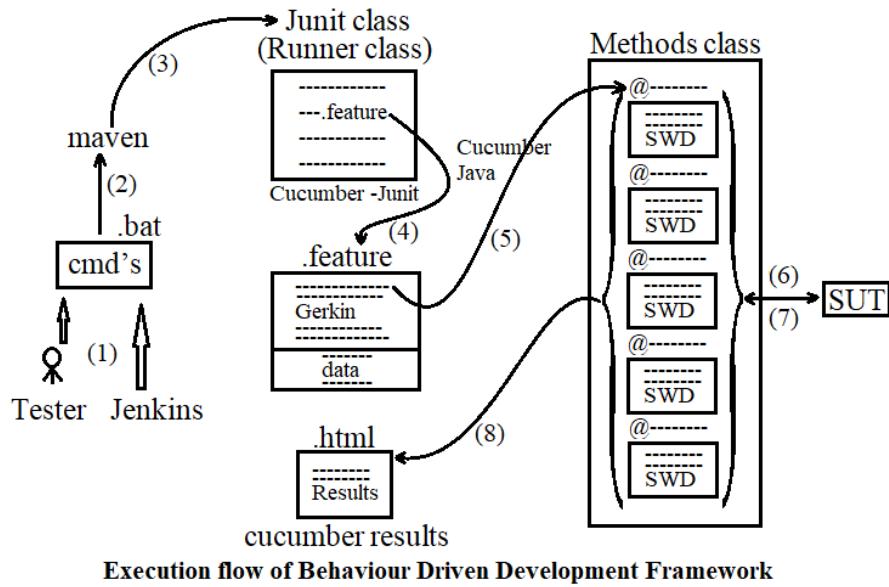
Note:

TestNG framework can support failed test execution instead of all tests. While running real testing we can run suite file directly which is in **src** folder under project folder but while running re/regression testing we can use previously failed tests info in “**path of project folder\test-output\testing.failed.xml**”.

XII. BDD

[Behaviour Driven Development Framework]

a) BDD Framework execution flow:



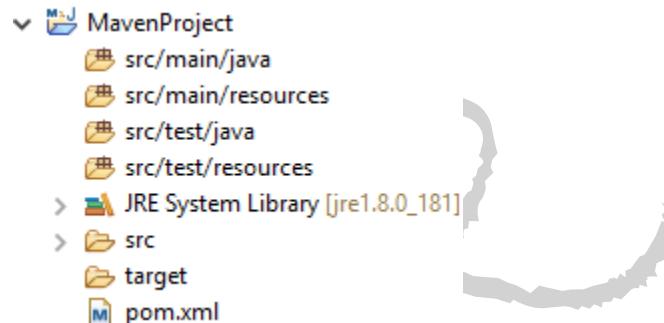
b) BDD Framework development:

- ✓ **Step 1:** Download and Install JDK8 (Create JAVA_HOME environment variable and extend “path” variable).
- ✓ **Step 2:** Download and launch eclipse IDE (provide a personal folder as workspace).
- ✓ **Step 3:** Create maven project
 - Open Eclipse IDE
 - Go to file menu
 - Select “new”
 - Select “others”
 - Open “maven”
 - Select “maven project”
 - Click “next”
 - Check “create a simple project” option & click next
 - Enter company name as “group id” and project name as “artefact id”
 - Click finish
 - Right click on created maven project

- Go to properties
- Check for latest compiler and latest JRE as [1.8], if not in latest version select it to latest versions.
- Click “Apply & close”

Note 1:

After completion of a maven project we can get below like folder structure.



Note 2:

If we didn't get above like folder structure in maven project, we need to follow like below:

- Right click on maven project
- Select “maven”
- Select “update project”
- Select “force update” checkbox
- Click ok

Note 3:

Maven is inbuilt in Eclipse oxygen & in latest versions.

✓ Step 4: Extend “pom.xml” for required jars

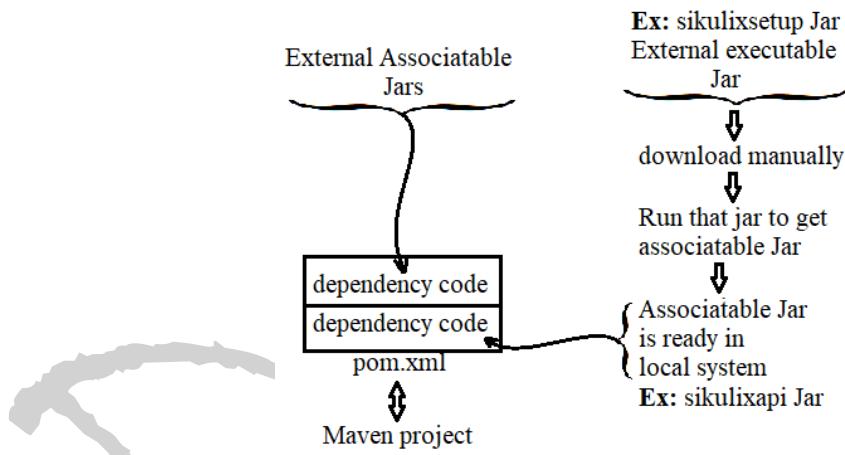
- We need to write below like code in “pom.xml” for required jars.
- ```
<project xmlns="http://maven.apache.org/POM/4.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
 http://maven.apache.org/xsd/maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>dinesh.reddy</groupId>
 <artifactId>BehaviourDrivenDevelopment</artifactId>
 <version>0.0.1-SNAPSHOT</version>
 <dependencies>
 <dependency>
 <groupId>org.seleniumhq.selenium</groupId>
 <artifactId>selenium-java</artifactId>
 <version>3.141.59</version>
 <scope>test</scope>
 </dependency>
```

```

<dependency>
 <groupId>io.cucumber</groupId>
 <artifactId>cucumber-java</artifactId>
 <version>3.0.2</version>
 <scope>test</scope>
</dependency>
<dependency>
 <groupId>io.cucumber</groupId>
 <artifactId>cucumber-junit</artifactId>
 <version>3.0.2</version>
 <scope>test</scope>
</dependency>
<dependency>
 <groupId>junit</groupId>
 <artifactId>junit</artifactId>
 <version>4.12</version>
 <scope>test</scope>
</dependency>
</dependencies>
</project>

```

- Sometimes we need to use local system jar in maven project.



- From above diagram dependency code is different for external jar and local system jar.

#### Ex 1: Dependency code for external jar:

```

<dependency>
 <groupId>junit</groupId>
 <artifactId>junit</artifactId> ┌─────────┐ Mandatory
 <version>4.12</version> ┌─────────┐
 <scope>test</scope> ──────────> Optional
</dependency>

```

#### Ex 2: Dependency code for external jar:

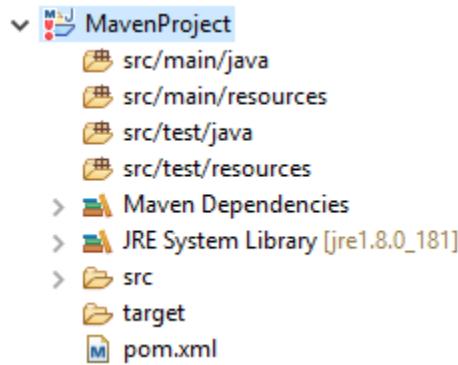
```

<dependency>
 <groupId>sikulix</groupId>
 <artifactId>sikulix</artifactId> ┌─────────┐ Sample Details
 <version>2.0</version> ┌─────────┐
 <scope>system</scope> ──────────> Mandatory
 <systemPath>D:\DineshReddy\sikulixapi.jar</systemPath>
</dependency>

```

#### Note 4:

After adding jars to the maven project then the folder structure will get as shown below.



#### ✓ Step 5: Developing feature files

In BDD framework one feature file is having one or more manual test scenarios. In every test scenario, we can get multiple steps like operations and observations (test cases) but the corresponding operations and observations related to test scenarios in a feature file are in **Gherkin** language.

**Gherkin** is an English based descriptive language. Testers can use this language keywords like **Given**, **Then**, **When**, **And**, **But**...etc. while writing operations and observations related sentences in feature files.

Just to know correctness of our sentences with **Gherkin** keywords, we can use anyone of the following two ways:

- Tidy Gherkin (plug-in to chrome browser)
- \*Cucumber Eclipse plug-in to Eclipse IDE

To install cucumber plugin to Eclipse IDE we need to follow below navigation

- Click on help menu in Eclipse IDE
- Select “install new software”
- Click “add”
- Enter name as “cucumber”
- Enter location as “<http://cucumber.github.com/cucumber-eclipse/update-site>”
- Click “ok”
- Select **Cucumber Eclipse Plugin** checkbox
- Click “next” until “finish”
- Restart Eclipse IDE after completion of installation

When we are strong in **Gherkin** sentences formation, above plugin installation is not required.

**Note:** If any error while installing **cucumber eclipse plugin**, then you do one thing. Before trying to install **cucumber eclipse plugin** you need to install one more plugin (**Eclipse PDE plugin**) by navigating to Eclipse market place in Eclipse help menu and search for **Eclipse PDE plugin** and install it. After restarting Eclipse IDE then try to install **cucumber eclipse plugin**.

To create a feature file before going to type **Gherkin** sentences, we need to follow below navigation:

- Right click on **src/test/resources** in maven project
- Select “new”
- Select “package”
- Enter a name to that package (Ex: features)
- Click finish
- Right click on created package
- Select “new”
- Select “file”
- Enter a name to that file with **.feature**(Ex: feature1.feature)
- Click “finish”
- Delete suggestions
- Write Gherkin sentences in that feature file using below keywords

| <b>Gherkin Keywords<br/>(Case sensitive)</b> | <b>Purpose</b>                                                                            |
|----------------------------------------------|-------------------------------------------------------------------------------------------|
| <b>Feature:</b>                              | To write current feature/module name.                                                     |
| <b>Scenario:</b>                             | To write current test name                                                                |
| <b>Scenario Outline:</b>                     | To write current data driven test name                                                    |
| <b>Given</b>                                 | To write precondition                                                                     |
| <b>When</b>                                  | To write operation                                                                        |
| <b>Then</b>                                  | To write observation/condition                                                            |
| <b>And</b><br><b>But</b>                     | To add multiple lines for Given, When & Then                                              |
| “ ”                                          | To enclose data                                                                           |
| < >                                          | To enclose variable                                                                       |
| #                                            | To write comments                                                                         |
| <b>Examples:</b>                             | To provide multiple data related to corresponding scenario outline                        |
| /                                            | To separate data in Examples:                                                             |
| <b>Background:</b>                           | To perform operations before each Scenario/Scenario Outline in corresponding feature file |

We need to follow below rules while developing feature file with Gherkin keywords

**Rule 1:** One feature file can allow **Feature:** as one time only.

**Rule 2:** No limit on number of feature files in a project.

**Rule 3:** Every feature file can allow one or more Scenario's and Scenario Outline's.

**Rule 4:** Under every Scenario/Scenario Outline, we need to write operations and observations to related sentences with Gherkin keywords like **Given**, **Then**, **When**, **And**, **But**...etc. But all keywords utilization is not mandatory.

**Rule 5:** When we took **Scenario Outline:**, we need to attach **Examples:** with multiple data as mandatory.

**Rule 6:** To skip some scenarios from execution, we need to use **@tags** in feature files.

**Rule 7:** We need to create feature files under a package in **src/test/resources** of maven project.

#### Example 1:

```
Feature: Gmail Login Test

 Scenario: Validate Title
 Given launch site
 Then Title will be "Gmail" for homepage
 When close site
```

#### Example 2:

```
Feature: Gmail Login Test

 Background:
 Given launch site

 Scenario Outline: Validate userid field
 When we enter userid as "<userid>"
 And click userid next button
 Then validate userid output for "<userid>" with "<criteria>"
 When close site

 Examples:
 | userid | criteria |
 | jsjdhvvdsv | invalid |
 | dineshry143 | valid |

 Scenario Outline: Validate password field
 When we enter userid as "dineshry143"
 And click userid next button
 When we enter password as "<password>"
 And click password next button
 Then validate password output for "<password>" with "<criteria>"
 When close site

 Examples:
 | password | criteria |
 | jsjdhvvdsv | invalid |
 | DINESHREDDY | valid |
```

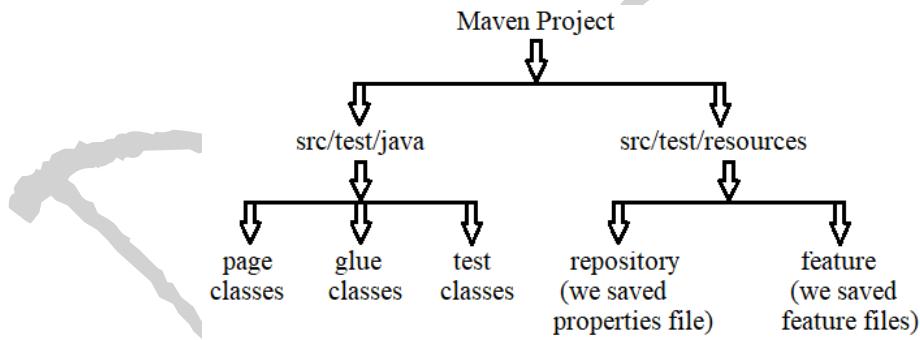
✓ **Step 6:** Developing properties file

- Right click on **src/test/resources** in maven project
- Select “new”
- Select “package”
- Enter a name to that package (Ex: repository)
- Click “finish”
- Right click on created package
- Select “new”
- Select “file”
- Enter a name with **.properties** as extension (Ex: Test1.properties)
- Click “finish”

Type properties with reusable data like

```
url=http://www.gmail.com
ChromePath=D:\DineshReddy\chromedriver.exe
FirefoxPath=D:\DineshReddy\geckodriver.exe
IEPath=D:\DineshReddy\iedriverserver.exe
OperaPath=D:\DineshReddy\operadriver.exe
```

✓ **Step 7:** Developing page classes



While creating page classes we need to follow below navigation

- Right click on **src/test/java** in maven project
- Select “New”
- Select “Package”
- Enter a name to that package (Ex: pageclasses)
- Click “finish”
- Right click on that created package
- Select “new”
- Select “class”

- Enter a name to that class (Ex: HomePage)
- Click “finish”

### Class 1:

```
public class HomePage
{
 @FindBy(name="identifier")
 public WebElement uid;

 @FindBy(xpath="//span[text()='Next ']")
 public WebElement uidNext;

 @FindBy(xpath="//div[contains(text(),'Enter an email')]")
 public WebElement uidBlankError;

 @FindBy(xpath="//div[contains(text(),'Enter a valid email')]")
 public WebElement uidWithSpacesError;

 @FindBy(xpath="//div[contains(text(),'find your Google Account')]")
 public WebElement uidInvalidError;

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

 public void fillUid(String x)
 {
 uid.sendKeys(x);
 }

 public void clickUidNext()
 {
 uidNext.click();
 }
}
```

### Class 2:

```
public class LoginPage
{
 @FindBy(name="password")
 public WebElement pswd;

 @FindBy(xpath="//span[text()='Next ']")
 public WebElement pswdNext;

 @FindBy(xpath="//div[text()='Enter a password']")
 public WebElement pswdBlankError;

 @FindBy(xpath="//div[contains(text(),'Wrong password')]")
 public WebElement pswdInvalidError;
}
```

```

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

public void fillPswd(String x)
{
 pswd.sendKeys(x);
}

public void clickPswdNext()
{
 pswdNext.click();
}
}

```

### Class 3:

```

public class ComposePage
{
 @FindBy(xpath="//div[text()='Compose']")
 public WebElement compose;

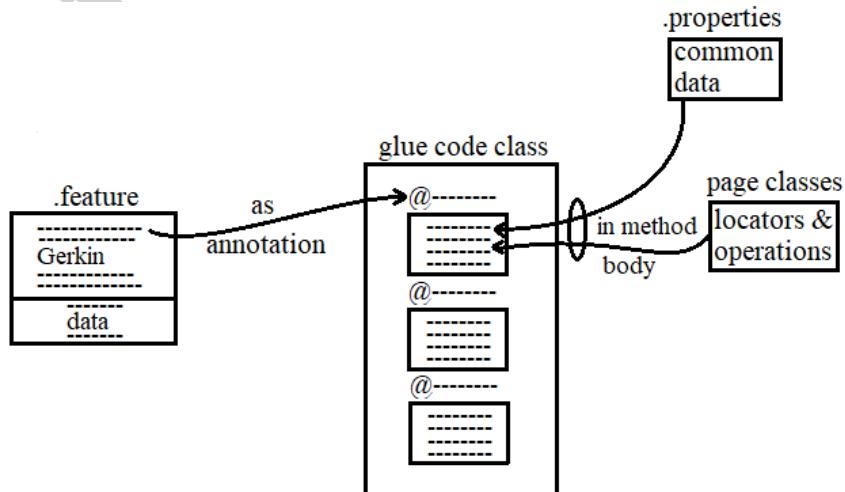
 public ComposePage(WebDriver driver)
 {
 PageFactory.initElements(driver, this);
 }
}

```

### Note:

The above 3 page classes are saved in **pageclasses** package in **src/test/java** in maven project.

### ✓ Step 8: Developing glue code classes (Step Definition classes)



While creating glue classes we need to follow below navigation

- Right click on **src/test/java** in maven project
- Select “new”

- Select “package”
- Enter a name to that package (Ex: glueclasses)
- Click “finish”
- Right click on that created package
- Select “new”
- Select “class”
- Enter a name to that class (Ex: Class1)
- Click “finish”

Develop methods to that created class with respect to above diagram as shown like below code:

```

public class Class1
{
 public WebDriver driver;
 public WebDriverWait wait;
 public Scenario s;
 public Properties p;
 public HomePage hp;
 public LoginPage lp;
 public ComposePage cp;

 @Before
 public void method1(Scenario s) throws Exception
 {
 this.s=s;
 FileInputStream fi=new FileInputStream(
 "D:\\DineshReddy\\MavenProject\\src\\test\\resources\\repository\\Test1.properties");
 p=new Properties();
 p.load(fi);
 }

 @Given("^launch site$")
 public void method1()
 {
 System.setProperty("webdriver.chrome.driver",p.getProperty("ChromePath"));
 driver=new ChromeDriver();
 driver.manage().window().maximize();
 wait=new WebDriverWait(driver,20);
 hp=new HomePage(driver);
 lp=new LoginPage(driver);
 cp=new ComposePage(driver);
 driver.get(p.getProperty("url"));
 }

 @Then("^Title will be \"(.*)\" for homepage$")
 public void method2(String x)
 {
 wait.until(ExpectedConditions.visibilityOf(hp.uid));
 String t=driver.getTitle();
 }
}

```

```

 if(t.equalsIgnoreCase(x))
 {
 s.write("Gmail title test passed");
 }
 else
 {
 byte[] b=((TakesScreenshot)driver).getScreenshotAs(OutputType.BYTES);
 s.embed(b,"Gmail title test failed");
 Assert.fail();
 }
}

@When("^close site$")
public void method3()
{
 driver.close();
}

@When("^we enter userid as \"(.*)\"$")
public void method4(String x)
{
 wait.until(ExpectedConditions.visibilityOf(hp.uid));
 hp.fillUid(x);
}

@And("^click userid next button$")
public void method5()
{
 wait.until(ExpectedConditions.elementToBeClickable(hp.uidNext));
 hp.clickUidNext();
}

@Then("^validate userid output for \"(.*)\" with \"(.*)\"$")
public void method6(String x, String y)
{
 try
 {
 //validations
 if(x.length()==0 && y.equalsIgnoreCase("invalid"))
 {
 wait.until(ExpectedConditions.visibilityOf(hp.uidBlankError));
 s.write("Blank user id test passed");
 }
 else if(y.equalsIgnoreCase("invalid"))
 {
 wait.until(ExpectedConditions.visibilityOf(hp.uidInvalidError));
 s.write("Invalid user id test passed");
 }
 else if(y.equalsIgnoreCase("valid"))
 {
 wait.until(ExpectedConditions.visibilityOf(lp.pswd));
 s.write("Valid user id test passed");
 }
 }
}

```

```

 else
 {
 byte[] b=((TakesScreenshot)driver).getScreenshotAs(OutputType.BYTES);
 s.embed(b,"UID test failed");
 Assert.fail();
 }
 }
catch(Exception ex)
{
 s.write("Error in code"+ex.getMessage());
}
}

@When("^we enter password as \"(.*)\"$")
public void method7(String x)
{
 wait.until(ExpectedConditions.visibilityOf(lp.pswd));
 lp.fillPswd(x);
}

@And("^click password next button$")
public void method8()
{
 wait.until(ExpectedConditions.elementToBeClickable(lp.pswdNext));
 lp.clickPswdNext();
}

@Then("^validate password output for \"(.*)\" with \"(.*)\"$")
public void method9(String x, String y)
{
 try
 {
 if(x.length()==0) //blank password
 {
 wait.until(ExpectedConditions.visibilityOf(lp.pswdBlankError));
 s.write("Blank password test passed");
 }
 else if(y.equalsIgnoreCase("Invalid")) //invalid password
 {
 wait.until(ExpectedConditions.visibilityOf(lp.pswdInvalidError));
 s.write("Invalid password test passed");
 }
 else if(y.equalsIgnoreCase("Valid")) //valid password
 {
 wait.until(ExpectedConditions.visibilityOf(cp.compose));
 s.write("Valid password test passed");
 }
 else
 {
 byte[] b=((TakesScreenshot)driver).getScreenshotAs(OutputType.BYTES);
 s.embed(b,"Gmail password login test failed");
 Assert.fail();
 }
 }
}

```

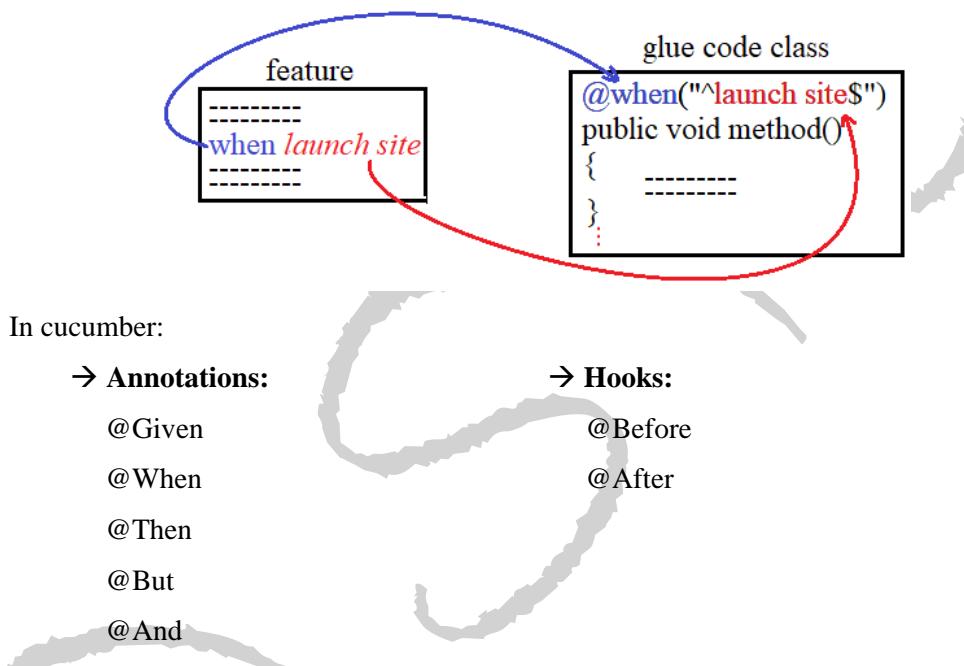
```

 catch(Exception ex)
 {
 s.write("Error in code"+ex.getMessage());
 }
 }
}

```

### Note 1:

While developing glue code class we need to write method bodies for **unique lines** in scenario's in feature files. Here Gherkin keywords can work like cucumber annotations and corresponding lines can work like arguments to annotations.



### Note 2:

**@Before** annotated method can run just before **Background:** of every Scenario. If there is no **Background:**, **@Before** annotated method can run just before corresponding Scenario's in feature file.  
**@After** annotated method can run just after every Scenario in feature file.

### Note 3:

Scenario class is related to cucumber jars. This class methods are useful to generate test results with screenshots as byte.

Ex:

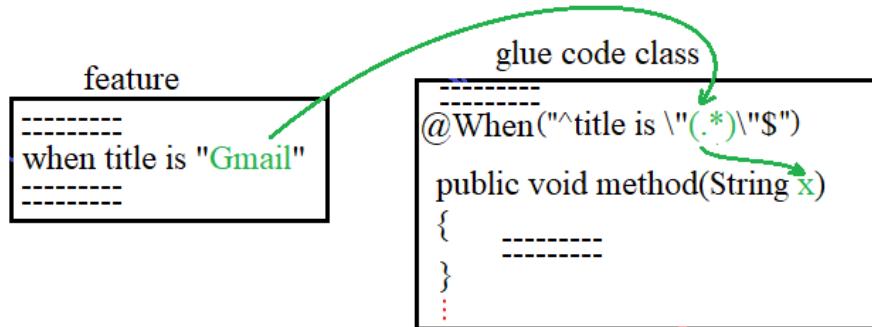
```

public Scenario s;

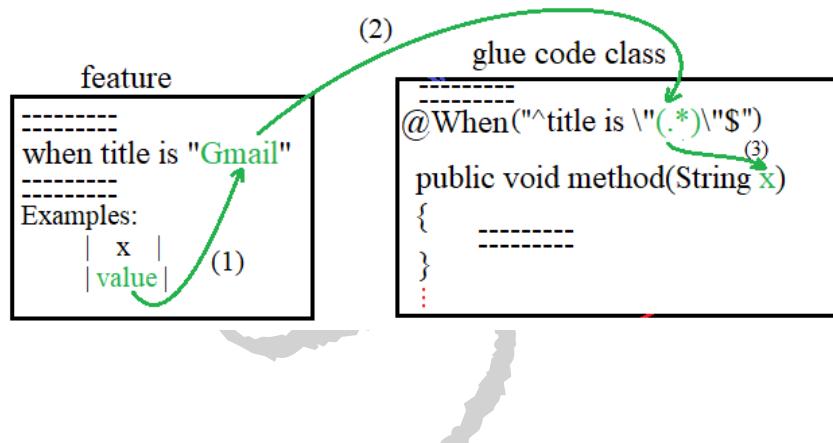
byte[] b=((TakesScreenshot)driver).getScreenshotAs(OutputType.BYTES);
s.embed(b,"message...");

```

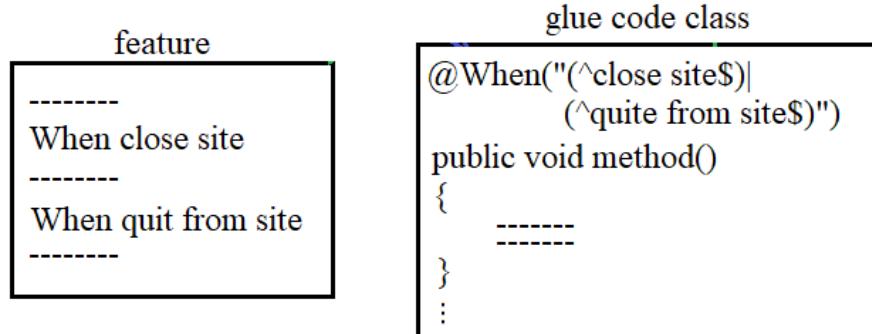
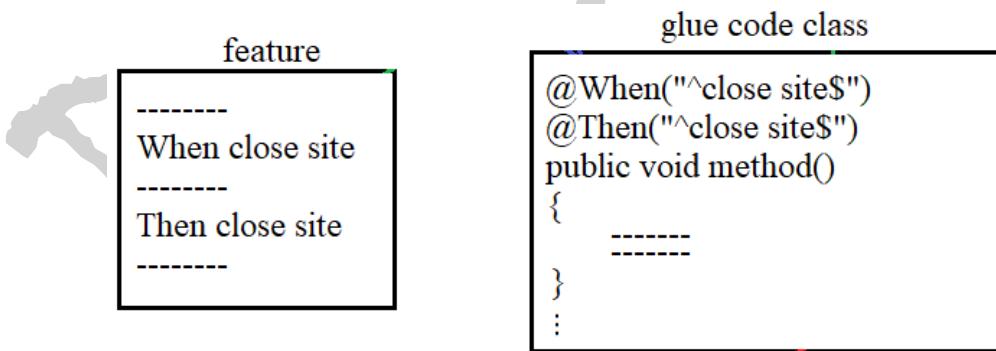
**Note 4:**



**Note 5:**



**Note 6:**



### Note 7:

**DataTable** is a class in cucumber like **Scenario**. It is useful to get data from a sentence in feature file instead of “Example:” in Scenario Outline.

#### ✓ Ex 1:

##### Feature file:

```
Feature: Way2SMS login
```

```
Background:
```

```
Given launching site
```

```
Scenario: login to way2sms
When user enters credentials to login
| 9491947838 | DIXXSHXXXX |
When user click login button
Then login success message will be displayed
When closing site
```

##### Glue class:

```
public class Way2SMSglue
{
 public WebDriver driver;
 public WebDriverWait wait;
 public Scenario s;
 public Properties p;
 public Way2SMSpage way;
 @Before
 public void method1(Scenario s) throws Exception
 {
 this.s=s;
 FileInputStream fi=new FileInputStream(
 "D:\\DineshReddy\\ MavenProject\\src\\test\\resources\\repository\\Test1.properties");
 p=new Properties();
 p.load(fi);
 }
 @Given("^launching site$")
 public void method1()
 {
 System.setProperty("webdriver.chrome.driver",p.getProperty("ChromePath"));
 driver=new ChromeDriver();
 driver.manage().window().maximize();
 wait=new WebDriverWait(driver,20);
 way=new Way2SMSpage(driver);
 driver.get(p.getProperty("way2smsurl"));
 }
 @When("^user enters credentials to login$")
 public void method2(DataTable dt)
 {
 wait.until(ExpectedConditions.visibilityOf(way.mobile));
 List<List<String>> data=dt.asLists();
 way.fillmobile(data.get(0).get(0));
 way.fillpassword(data.get(0).get(1));
 }
}
```

```

@When("^user click login button$")
public void method3()
{
 way.clicklogin();
}

@Then("^login success message will be displayed$")
public void method4()
{
 wait.until(ExpectedConditions.visibilityOf(way.sendSMS));
}

@When("^closing site$")
public void method5()
{
 driver.close();
}

```

in above code, 0 | xxxx | xxxx |

#### ✓ Ex 2:

##### Feature file:

```

Feature: Way2SMS login
Background:
Given launching site

Scenario: login to way2sms
When user enters credentials to login
| Mobile No: | Password | → //header lines
| 9491947838 | DIXXXHXXXXX | → //data lines
When user click login button
Then login success message will be displayed
When closing site

```

##### Glue class:

```

public class Way2SMSglue2
{
 public WebDriver driver;
 public WebDriverWait wait;
 public Scenario s;
 public Properties p;
 public Way2SMSpage way;
 @Before
 public void method1(Scenario s) throws Exception
 {
 this.s=s;
 FileInputStream fi=new FileInputStream(
 "D:\\DineshReddy\\MavenProject\\src\\test\\resources\\repository\\Test1.properties");
 p=new Properties();
 p.load(fi);
 }
}

```

```

@Given("^launching site$")
public void method1()
{
 System.setProperty("webdriver.chrome.driver",p.getProperty("ChromePath"));
 driver=new ChromeDriver();
 driver.manage().window().maximize();
 wait=new WebDriverWait(driver,20);
 way=new Way2SMSpage(driver);
 driver.get(p.getProperty("way2smsurl"));
}

@When("^user enters credentials to login$")
public void method2(DataTable dt)
{
 wait.until(ExpectedConditions.visibilityOf(way.mobile));
 List<Map<String, String>> data=dt.asMaps();
 way.fillmobile(data.get(0).get("Mobile No"));
 way.fillpassword(data.get(0).get("Password"));
}

@When("^user click login button$")
public void method3()
{
 way.clicklogin();
}

@Then("^login success page will be displayed$")
public void method4()
{
 wait.until(ExpectedConditions.visibilityOf(way.sendSMS));
}

@When("^closing site$")
public void method5()
{
 driver.close();
}

```

(OR)

```

@When("^user enters credentials to login$")
public void method2(DataTable dt)
{
 wait.until(ExpectedConditions.visibilityOf(way.mobile));
 List<List<String>> data=dt.asLists();
 way.fillmobile(data.get(1).get(0));
 way.fillpassword(data.get(1).get(1));
}

```

0              1  
in above code, 0 | username | password |  
1 | XXXXXXXX | XXXXXXXX |  
      (1,0)        (1,1)

✓ Ex 3:

**Feature file:**

Feature: Facebook Automation

Background:

Given launch facebook site

Scenario: Facebook Registration

When user enters credentials to sign up

| first name | last name | mobile number | password   | day | month | year | gender |
|------------|-----------|---------------|------------|-----|-------|------|--------|
| Dinesh     | Reddy     | 9491947838    | DINESHs2   | 29  | Nov   | 1995 | male   |
| xxxxxx     | xxxxx     | xxxxxxxxxx    | xxxxxxxxxx | 19  | Oct   | 1997 | female |
| xxxxxx     | xxxxx     | xxxxxxxxxx    | xxxxxxxxxx | 30  | Dec   | 1996 | female |

When user click sign up button

When closing facebook site

**Page class:**

```
public class Facebookpage
{
 @FindBy(name="firstname")
 public WebElement firstname;

 @FindBy(name="lastname")
 public WebElement lastname;

 @FindBy(xpath="//input[@name='reg_email__']")
 public WebElement mobilenumber;

 @FindBy(xpath="//input[@name='reg_passwd__']")
 public WebElement password;

 @FindBy(xpath="//select[@name='birthday_day']")
 public WebElement day;

 @FindBy(xpath="//select[@name='birthday_month']")
 public WebElement month;

 @FindBy(xpath="//select[@name='birthday_year']")
 public WebElement year;

 @FindBy(xpath="(//button[text()='Sign Up'])[1]")
 public WebElement signup;

 @FindBy(xpath="(//input[@type='radio'])[1]")
 public WebElement femalegender;

 @FindBy(xpath="(//input[@type='radio'])[2]")
 public WebElement malegender;

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

```
public void fillfirstname(String x)
{
 firstname.sendKeys(x);
}

public void filllastname(String x)
{
 lastname.sendKeys(x);
}

public void fillmobilenumber(String x)
{
 mobilenumber.sendKeys(x);
}

public void fillpassword(String x)
{
 password.sendKeys(x);
}

public void selectday(String x)
{
 Select s=new Select(day);
 s.selectByVisibleText(x);
}

public void selectmonth(String x)
{
 Select s=new Select(month);
 s.selectByVisibleText(x);
}

public void selectyear(String x)
{
 Select s=new Select(year);
 s.selectByVisibleText(x);
}

public void clicksignup()
{
 signup.click();
}

public void selectgender(String x)
{
 if(x.equalsIgnoreCase("male"))
 {
 malegender.click();
 }
 else
 {
 femalegender.click();
 }
}
```

### Glue class:

```
public class Facebookglue
{
 public WebDriver driver;
 public WebDriverWait wait;
 public Scenario s;
 public Properties p;
 public Facebookpage fb;

 @Before
 public void method1(Scenario s) throws Exception
 {
 this.s=s;
 FileInputStream fi=new FileInputStream(
 "D:\\DineshReddy\\MavenProject\\src\\test\\resources\\repository\\Test1.properties");
 p=new Properties();
 p.load(fi);
 }

 @Given("^launch facebook site$")
 public void method1()
 {
 System.setProperty("webdriver.chrome.driver",p.getProperty("ChromePath"));
 driver=new ChromeDriver();
 driver.manage().window().maximize();
 wait=new WebDriverWait(driver,20);
 fb=new Facebookpage(driver);
 driver.get(p.getProperty("fburl"));
 }

 @When("^user enters credentials to sign up$")
 public void method2(DataTable dt) throws Exception
 {
 wait.until(ExpectedConditions.visibilityOf(fb.firstname));
 List<Map<String, String>> data=dt.asMaps();
 for(int i=0;i<data.size();i++)
 {
 fb.fillfirstname(data.get(i).get("first name"));
 fb.filllastname(data.get(i).get("last name"));
 fb.fillmobilenumber(data.get(i).get("mobile number"));
 fb.fillpassword(data.get(i).get("password"));
 fb.selectday(data.get(i).get("day"));
 fb.selectmonth(data.get(i).get("month"));
 fb.selectyear(data.get(i).get("year"));
 fb.selectgender(data.get(i).get("gender"));
 }
 }

 @When("^user click sign up button$")
 public void method3()
 {
 fb.clicksignup();
 }
}
```

```

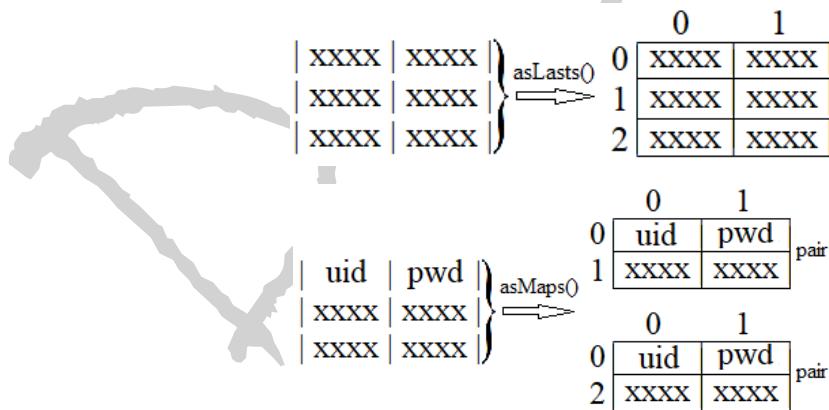
@When("^closing facebook site$")
public void method5() throws Exception
{
 Thread.sleep(5000);
 driver.close();
}
}

(OR)

@When("^user enters credentials to sign up$")
public void method2(DataTable dt) throws Exception
{
 wait.until(ExpectedConditions.visibilityOf(fb.firstname));
 List<List<String>> data=dt.asLists();
 for(int i=1;i<data.size();i++)
 {
 fb.fillfirstname(data.get(i).get(0));
 fb.filllastname(data.get(i).get(1));
 fb.fillmobilenumber(data.get(i).get(2));
 fb.fillpassword(data.get(i).get(3));
 fb.selectday(data.get(i).get(4));
 fb.selectmonth(data.get(i).get(5));
 fb.selectyear(data.get(i).get(6));
 fb.selectgender(data.get(i).get(7));
 }
}

```

From above 3 examples, **asLists()** and **asMaps()** methods can take data from feature file like shown below.



#### ✓ Step 9: Developing runner classes

After completion of **pom.xml** file extension with **dependencies** code, **feature files** development with **Gherkin sentences**, **properties** file development with **reusable common data**, **page class's** development with **element locators & operations** and **glue code class's** development then we can go to develop **runner class's** like shown below:

- Right click on **src/test/java** in maven project
- Select “new”

- Select “package”
- Enter a name to that package (Ex: tests)
- Click “finish”
- Right click on that created package
- Select “new”
- Select “class”
- Enter a name to that class (Ex: Runner1)
- Click “finish”

```
package tests;
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;
@RunWith(Cucumber.class)
@CucumberOptions(features= {"path of .feature file","path of .feature file"},
 glue= {"classpath:package name of glue classes"},
 monochrome=true,
 plugin={"pretty","html:target\\result"}) //target is a built in folder
//result is our own folder to save results (Ex: index.html)
public class Runner1
{

}
```

*From the above runner class @RunWith is a class level annotation related to junit.*

**@RunWith(Cucumber.class)**

Here: @RunWith is related to junit & Cucumber is related to cucumber-junit

*@CucumberOptions annotation related to cucumber-java is also a class level annotation. It is useful to define tests via properties:*

|            |          |
|------------|----------|
| features   | name     |
| dryrun     | plugin   |
| glue       | snippets |
| junit      | strict   |
| monochrome | tags     |

#### Note 1:

“features” property is mandatory property. It is useful to specify path of features files to be executed.

```

@CucumberOptions(features= {"path of .feature file","path of .feature file"})


```

#### Note 2:

“plugin” property is useful to specify location for results. Here results are two types.

Such as 1) HTML (Hyper Text Mark-up Language)

2) JSON (Java Script Object Notation)

```

@CucumberOptions(plugin={"pretty","html:target"})
//we are able to use json in the place of html:


```

#### Note 3:

“monochrome=true” is useful to get output in Eclipse console also.

#### Note 4:

“glue” property is used to specify path of glue code classes, when runner classes and glue code classes are in different packages.

```

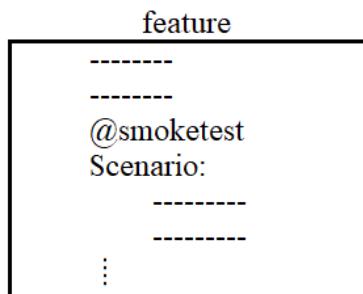
@CucumberOptions(glue={"classpath:package name of glue classes "})
//It will run scenarios of feature files, which are tagged with @smoketest


```

#### Note 5:

“tags” property is useful to skip some Scenarios/Scenario Outlines of feature files from execution.

Ex:



```

@CucumberOptions(tags={"@smoketest"})
//It will run scenarios of feature files, which are tagged with @smoketest


```

```

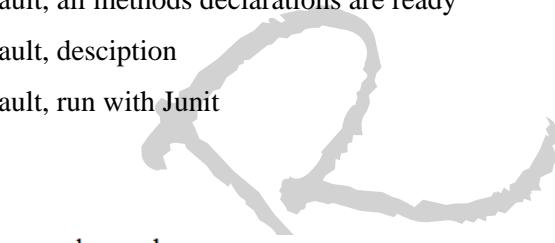
@CucumberOptions(tags={"~@smoketest"})
//It will run scenarios of feature files, which are not tagged with @smoketest


```

#### Note 6:

- |                 |                                                  |
|-----------------|--------------------------------------------------|
| “dryRun=true”   | → By default, run can proceed                    |
| “strict=true”   | → By default, all methods are ready              |
| “snippets=true” | → By default, all methods declarations are ready |
| name="xxxxx"    | → By default, description                        |
| “junit=true”    | → By default, run with Junit                     |

#### Note 7:



glue code

```

else
{

 Assert.fail();
}
```

Static class  
in Junit

#### Runner class 1

```


@CucumberOptions(features= {"path of .feature file",
 "path of .feature file"},
 plugin={"pretty","html:target\\result",
 "rerun:target\\failed.txt"})

```

to store failed  
Scenarios info

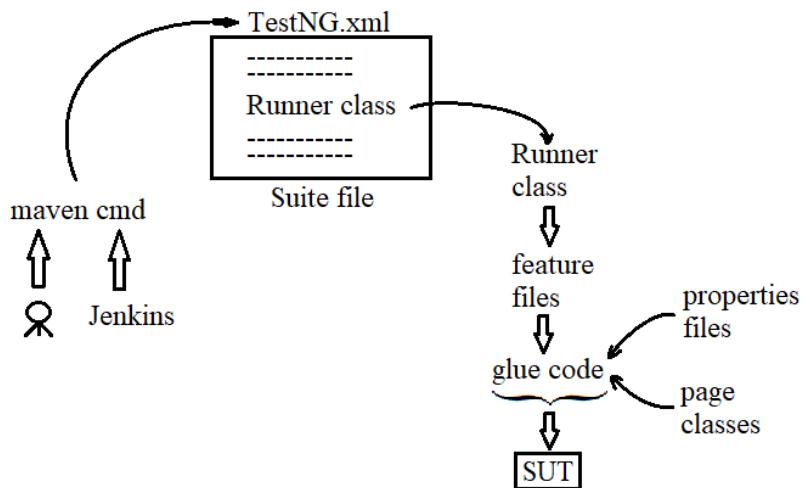
#### Runner class 2 (Retesting)

```


@CucumberOptions(features= {"@target\\failed.txt"},
 plugin={"pretty","html:target\\failedresults"})

```

#### Note 8:



To run BDD framework by using testNG in the place of junit follow below steps:

- ✓ **Step 1:**
  - Right click on maven project
  - Go to properties
  - Select “java build path”
  - And click “add library”
  - Select “TestNG”
  - Click “next” and click “finish”
  - Click “apply and close”
- ✓ **Step 2:**
  - Go to **pom.xml**
  - Add following dependency's before </dependencies> tag and click save

```
<dependency>
<groupId>org.testng</groupId>
<artifactId>testng</artifactId>
<version>6.14.3</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>io.cucumber</groupId>
<artifactId>cucumber-testng</artifactId>
<version>3.0.2</version>
</dependency>
```
  - Wait for some time until getting jars from internet (to check status after clicking save you can observe right side down corner)

✓ **Step 3:**

- Right click on **tests** package under **src/test/java**
- Select “TestNG”
- Select “Create TestNG class”
- Enter a name to xml file along with **.xml** extension
- Click “finish”
- And enter your **runner class package name.runner class name** in **<class name="tests.Runner1"/>** shown below.

```
<suite name="Suite" parallel="false">
<test name="Test">
<classes>
<class name="tests.Runner1"/> //runnerclassespackagename.runnerclassname
</classes>
</test>
</suite>
```

✓ **Step 4:**

- After completion of above steps then write runner class like shown below

```
@CucumberOptions(features= {"src/test/resources/feature"},
glue= {"glueclasses"},
monochrome=true,
plugin={"pretty", "html:target\\result",
"rerun:target\\failedresult.txt"})
```

```
public class Runner1 extends AbstractTestNGCucumberTests
{}
```

✓ **Step 5:** Now run your **.xml** file by following below navigation

- Right click on **.xml** file
- Select “Run As”
- Select “TestNG Suite”

## IMPORTS

### Selenium Web Driver Imports:

```
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxProfile;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.io.FileHandler;
import org.openqa.selenium.opera.OperaDriver;
import org.openqa.selenium.opera.OperaOptions;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.safari.SafariDriver;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.FluentWait;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.support.ui.ExpectedConditions;

WebDriverWait w;
TakesScreenshot ts;
ChromeDriver cd;
OperaDriver od;
ChromeOptions co;
SafariDriver sd;
By by;
Dimension d;
Select se;
Keys k;
DesiredCapabilities dc;
FileHandler fh;
JavascriptExecutor js;

WebDriver wd;
ExpectedConditions ec;
InternetExplorerDriver ie;
OperaOptions oo;
EdgeDriver ed;
FirefoxDriver fd;
WebElement we;
Point p;
Actions a;
FluentWait fw;
OutputType ot;
PageFactory pag;
FirefoxProfile fp;
```

### Java Imports & Languages (JDK):

```
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.datatransfer.DataFlavor;
import java.awt.datatransfer.StringSelection;
import java.awt.event.KeyEvent;
import java.awt.image.BufferedImage;
import java.io.File;
```

```
import java.lang.reflect.Method;
import java.text.SimpleDateFormat;
import java.time.Duration;
import java.time.temporal.ChronoUnit;
import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import javax.imageio.ImageIO;
import java.io.FileInputStream;
import java.util.Properties;
import java.util.Map;
```

#### Imports:

```
Scanner sc;
Properties p;
Map m;
BufferedImage bi;
TimeUnit tu;
Robot r;
Date dt;
Method[] met;
KeyEvent ke;
StringSelection ss;
ChronoUnit cu;
```

```
FileInputStream fi;
List l;
File f;
ImageIO ii;
Duration du;
Iterator it;
SimpleDateFormat sdf;
ArrayList al;
Toolkit tk;
DataFlavor df;
Runtime rt;
```

#### Languages:

```
System s;
String st;
boolean b;
byte bs;
```

```
Thread t;
int i;
char c;
short s;
```

```
double d;
long l;
float f;
```

#### Sikulix Imports:

```
import org.sikuli.script.Button;
import org.sikuli.script.Key;
import org.sikuli.script.Location;
import org.sikuli.script.Match;
import org.sikuli.script.Region;
import org.sikuli.script.Screen;
import org.sikuli.script.ScreenImage;
```

Screen scr;	Location lo;	Match m;	
Button bu;	Region rg;	Key key;	ScreenImage si;

#### FreeTTS Imports:

```
import com.sun.speech.freetts.Voice;
import com.sun.speech.freetts.VoiceManager;

VoiceManager vm; Voice v;
```

### JXL Imports:

```
import jxl.CellView;
import jxl.Sheet;
import jxl.Workbook;
import jxl.format.Alignment;
import jxl.format.Colour;
import jxl.format.UnderlineStyle;
import jxl.write.Label;
import jxl.write.WritableCell;
import jxl.write.WritableCellFormat;
import jxl.write.WritableFont;
import jxl.write.WritableSheet;
import jxl.write.Number;

Workbook wb; Sheet she;
WritableSheet ws; WritableFont wf;
Alignment al; Colour col;
Number nu; CellView cv;
 WritableCell ww;
 Label lb;
 WritableCellFormat wcf;
 UnderlineStyle ud;
```

### Extent Reports Imports:

```
import com.relevantcodes.extentreports.ExtentReports;
import com.relevantcodes.extentreports.ExtentTest;
import com.relevantcodes.extentreports.LogStatus;

ExtentReports er; ExtentTest et; LogStatus ls;
```

### TestNG Imports:

```
import org.testng.Assert;
import org.testng.ITestListener;
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterGroups;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeGroups;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

@Test
@BeforeTest
@AfterClass
@BeforeMethod
@Parameters
Reporter r; @BeforeSuite
 @AfterTest
 @BeforeGroups
 @AfterMethod
Assert a; @AfterSuite
 @BeforeClass
 @AfterGroups
 @DataProvider
ITestListener it;
```

### Cucumber Imports:

```
import cucumber.api.Scenario;
import cucumber.api.java.Before;
import cucumber.api.java.en.And;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;
import io.cucumber.datatable.DataTable
import cucumber.api.CucumberOptions;
import cucumber.api.testng.AbstractTestNGCucumberTests;
import cucumber.api.junit.Cucumber;

@And @Given
@When @Then
@After @CucumberOptions
DataTable dt; AbstractTestNGCucumberTests abt;
```