

Contents

I.	SDET(Software Development Engineer in Testing).....	1
A.	Manual Testing V/S Test Automation.....	1
B.	Scope of Automation.....	1
C.	Functional Test Automation.....	2
	Note1.....	2
	Note2.....	3
II.	Selenium Web Driver.....	5
A.	Types of Software:-.....	5
B.	Configure Selenium web driver in Tester Software:-.....	6
	Step1:- Download and install JDK8	7
	Note:-	7
	Step2:- Download and Launch Eclipse IDE.....	7
	Note1:-	7
	Note2:-	7
	Note3:-	8
	Note4:-	8
	Note5:-	8
	Step3:-Download and associate Selenium web Driver Jars	8
	Step4:- Download and use browser Drivers.....	9
	Example1:.....	10
	Step1:-	10
	Step2:-	10
	Step3:-	10
	Example2:.....	10
	Step1:-	10
	Step2:-	10
	Step3:-	10
	Example3:.....	11
	Step1:-	11
	Step2:-	11
	Step3:-	11
	Example4:.....	11
	Step1:-	11
	Step2:-	11
	Step3:-	11
	Example5:.....	11

Step1:-	11
Step2:-	12
Step3:-	12
Step5:- Locating Elements in webpages for Automation.....	12
Test Scenario.....	12
Test Script:	12
Note1:-	15
Note2:-	15
Note3:-	15
C. Types of xPath:-	16
Syntax1:-	17
Syntax2:-	18
Syntax3:-	18
Syntax4:-	19
Syntax5:-	19
Syntax6:-	20
Syntax7:-	20
Syntax8:-	21
Syntax9:-	21
Syntax10:-	22
Syntax11:-	23
Syntax12:-	23
Syntax13:-	24
Syntax14:-	24
Syntax15:-	25
Syntax16:-	25
Syntax17:-	26
Syntax18:-	26
Note:-	27
D. “WebDriver” Class Methods:-	28
1. get():-	28
2. close():-	28
3. quit():-	28
4. getTitle():-	28
5. getPageSource():-	28
6. getCurrentUrl():-	28
7. findElement():-	28
8. click():-	28
9. sendKeys():-	28
10. getAttribute():-	29

11. getText():-	29
12. getCssValue():-.....	29
13. switchTo().frame():-.....	29
Way1:-	29
Way2:-	29
Way3:-	29
14. switchTo().defaultContent():-	30
Example1:-.....	30
Example2:-	31
Example3:-.....	31
15. switchTo().parentFrame():-	31
16. switchTo().activeElement():-	32
17. switchTo().window():-	32
18. switchTo().getWindowHandles():-	32
19. switchTo().alert():-.....	33
a) dismiss():-	33
b) accept():-.....	33
c) getText():-	33
d) sendKeys():-	33
Syntax1 :-.....	33
Syntax2 :-.....	34
Syntax3 :-.....	34
Syntax4 :-.....	34
Syntax5 :-.....	35
20. clear():-.....	36
21. manage().window().Maximize().....	36
22. manage().window().getSize().....	36
23. manage().window().setSize().....	36
24. manage().window().getPosition().....	36
25. manage().window().setPosition().....	36
26. manage().window().deleteAllCookies().....	36
27. manage().window().getCookies().size().....	37
Example	37
28. manage().window().getCookies().size().....	37
29. navigate().To(),navigate().back(),navigate().forward(),navigate().refresh():-.....	37
Note:-	37
30. findElements():-	37
Example1 :-.....	39

Example2:-.....	39
Example3:-.....	39
Example4:-.....	40
Example5:-.....	40
Example6:-.....	40
Note:-	40
31. getScreenshotAs():-.....	41
E. “Select” Class Methods:-	42
1. selectByVisibleText():-	42
2. selectByIndex():-	42
3. selectByValue():-	42
4. deselectAll():-	42
5. deselectByVisibleText():-	43
6. deselectByIndex():-	43
7. deselectByValue():-	43
8. getoptions().size():-	43
9. isMultiple():-	43
F. “Actions” Class Methods:-	44
1. Click on element:-	44
2. Double Click on element:-	44
3. Right Click on element:-	44
4. Move Mouse Pointer to an element:-	44
5. Fill an element:-	44
6. Handle cache Element (or) Auto Complete Element_-:-	45
7. Automate Drag and Drop element:-	45
8. Handle Sliders:-	46
9. Automate drop downs:-	46
10. Element Scrolling:-	48
G. “Javascpritexecutor” Class Methods:-	49
1. Highlight Element:-	49
2. Click an Element:-	49
3. Disable an Element:-	50
4. Page Scrolling	50
5. Create Alert.....	50
6. Fill Element.....	50
7. JavaScript in SWD.....	51
Note1:-	51
Note2:-	51
H. “WebDriverWait”or “FluentWait”Classes Methods:-	52
a) Thread.sleep():-	52

b)	ImplicitWait():-	52
c)	ExplicitWait Concept(ConditionalWait)/ WebDriver Wait:-	53
d)	Fluent Wait Concept (or) Conditional Wait :-	53
I.	Handling Ajax Calls and Angular Js Controls in Webpages:-	54
a)	Example1.....	54
b)	Example2.....	54
J.	Handling Browser Notifications:-	55
a)	For Chrome Browser: -	55
b)	For Firefox Browser:-	55
c)	For IE browser:-	56
K.	Handling SSL Certificates:-	57
a)	For Chrome Browser: -	57
b)	For Firefox Browser: -	57
c)	For Internet Explorer Browser: -	58
	Note1:-	58
	Note2:-	58
III.	JAVA ROBOT.....	59
	Example 1:-	59
	Test Script :-.....	59
	Example 2:-	61
	Test Script :-.....	61
	Example 3:-	62
	Note 1:-	63
	Test Scripts :-.....	63
	Note 2:-	64
	Note 3:-	64
	Example 4:-	64
	Test Script:-.....	64
IV.	SIKULI X.....	67
A.	Configure “SIKULI’ in Tester Computer	67
B.	Configure “SIKULI’ in Tester Computer	68
C.	“Screen” Class Methods:-	68
1.	Click on Element:-	68
2.	Double click On Element:-	68
3.	Right click On Element:-	68
4.	Mouse Move to Element:-.....	68
5.	Drag Drop an Element:-.....	68
6.	Existence of an Element:-	68
7.	Wheel ():	69
8.	Find an Element:-	71

9. Find ALL Elements:-	71
10. type data to Elements:-	71
11. capture Screen/ Region of Screen:-	71
12. keyDown() or keyUp():-	73
13. mouseDown() or mouseUp():-	73
NOTE:-	73
V. CORE JAVA.....	77
A. Variables in Java language.....	78
B. Data types in java.....	78
C. Primitive Data Type:-	78
Note 1:	79
Note 2:	79
D. Derived Data Type	79
Note 1:	79
Note 2:	79
E. User Defined Data Type (Class).....	80
F. Types of classes	86
a) Instance classes :-.....	86
b) Static classes :-	86
Example 1:.....	86
Example 2:.....	86
Example 3:.....	87
Example4:-.....	87
Example4:-.....	88
c) Interface classes :-.....	89
Example 1:-	89
Example of interface classes	89
d) Abstract classes :-.....	89
Example:-.....	90
G. Ways to create object to Classes:-	91
H. Inheritance in JAVA(is a relation)	92
a) Single inheritance:-.....	92
b) Multiple inheritance:-	92
c) Multi-level inheritance:-	93
d) Hybrid inheritance:-	93
e) "has a" relation:-	94
I. Operators in JAVA.....	95
a) Unary Operator.....	95
b) Binary Operator.....	95

Example	95
c) Ternary Operator	95
J. "String" Class methods:-	96
a) length():-	96
Syntax	96
b) equals():-	97
Syntax	97
c) contains():-	97
Syntax	97
d) equalsIgnoreCase ():-	97
Syntax	97
e) toLowerCase():-	97
Syntax	97
f) toUpperCase():-	97
Syntax	97
g) charAt():-	98
Example 1:-	98
Example2:-	98
Example 3:-	98
h) split():-	98
Example 1:-	98
Note:.....	99
Example2:.....	99
i) trim():-	99
Example:-	99
j) replace():-	99
Syntax 1:.....	99
Syntax 2:.....	99
K. Read data from keyboard:-	100
L. Type casting, Type conversation & Java Generics	102
M. Control statements in java.....	103
a) if - else	103
Example 1	103
Example 2	104
Example 3	104
b) if - else if - else	105
Example 4	105
c) Nested if:-	107

Example 5.....	107
Note.....	109
d) Switch case statement:-	109
Example 6.....	109
Example 7.....	110
N. Loops in java.....	111
a) For Loop:-	111
Example 1:-	112
Example 2:-	113
Example 3:-	113
Example 4:-	114
Example5.....	114
Example 6.....	115
Example 7.....	115
b) while Loop:-	116
c) do while Loop:-	116
*** Note:-	116
Example 8:.....	117
Test scripts1:-.....	117
Test scripts2:-.....	118
O. Arrays In Java.....	120
⇒ Static Array	120
Example:	120
Test scripts:-.....	120
⇒ Dynamic Array	121
Syntax:	121
Example:	121
Test script:-	121
Example:	122
Test script:-	122
Example:	123
Test script:-	123
• Data Driven Testing.....	125
Example:	125
Test Script:-	125
Note 1:-	126
Note 2:-	126

P. Using text files:-	127
Rules:-	127
Example 1:-	128
Test Script:-	129
Note 1:-	130
Q. Extent Reports:-	131
Navigation:-	131
Example:	132
Test Script:-	133
Note 1:-	134
R. JXL:-.....	135
Navigation:-	135
Example:	136
Test Script:-	136
Note 1:-	137
Note 2:-	137
Rules:-	137
Example:	137
Test Script:-	138
Note 1:-	139
S. Keyword Driven Framework:-.....	140
Keyword driven framework process:-	141
Step 1:-	141
Step 2:-	141
Step 3:-	141
Step 4:-	141
Step 5-	141
Step 6-	141
Step 7-	142
Example:	142
Test Script:-	143
Note 1:-	144
Step 8-	145
Test Script:-	145
Note 1:-	146
Note 2:-	147
Note 3:-	148

Note 4:-	148
Note 5:-	149
Example:	150
Example:-	150
Note 6:-	150
Note 7:-	151
T. POM (Page Object Model)	152
Step 1:-	153
Step 2:-	153
Step 3:-	153
Step 4:-	153
Step 5:-	153
Step 6:-	153
Note:-	154
Step 7:-	154
Navigation:-	154
Step 8:-	157
Navigation:-	157
Step 9:-	157
Step 10:-	157
Example:-	158
U. Module-driven framework	159
Step 1:-	160
Step 2:-	160
Step 3:-	160
Step 4:-	160
Step 5:-	160
Test scripts:-	160
Note 1:-	161
@BeforeSuite :-	161
@AfterSuite :-	161
@BeforeTest :-	161
@AfterTest :-	161
@BeforeGroup :-	162
@AfterGroup :-	162
@BeforeClass :-	162
@AfterClass :-	162

@BeforeMethod :-	162
@AfterMethod :-	162
Note 2:-	162
@Test:-	162
alwaysRun:-	163
dependsOnGroups :-	163
Example:-	163
dependsOnMethods :-	164
Example:-	164
enabled:-	165
Example:-	165
expectedExceptions :-	165
groups:-	166
Example:-	166
invocationCount :-	167
Example:-	167
invocationTimeOut :-	167
Example:-	167
priority:-	168
Example:-	168
timeOut:-	169
Example:-	169
Note 3:-	170
Step 6:-	170
Navigation:-	170
Note 1:-	171
Note 2:-	171
Step 7:-	171
Note 1:-	171
Step 8:-	172
Example:-	173
V. BDD framework using cucumber:-	174
Step 1:-	175
Step 2:-	175
Step 3:-	175
Note 1:-	175
Note 2:-	175

Note 3:-	176
Note 4:-	176
Step 4:-	176
Step 5:-	177
Example:-	177
Test Script:-	177
Example:-	178
Test Script:-	178
Note 1:-	178
Note 2:-	178
Note 3:-	178
Step 6:-	178
Note:-	181
Step 7:-	181
Navigation:-	181
Test Script:-	181
Note 1:-	182
Note 2:-	183
Example:-	183
Format	185
Example:-	185
Step 8:-	185
Change 1:-	185
Change 2:-	185
Test Script:-	185
Note 1:-	186
Example:-	186
Note 2:-	186
VI. APPIUM	187
A. Types of software	187
B. Types of mobile software	187
C. About Appium :-	189
i). Tester computer OS is windows	189
a) Configure Mobile app test setup	190
Step 1:- Download and install JDK8	190
Step 2:- Download & install Android SDK	190
Note 1:-	190

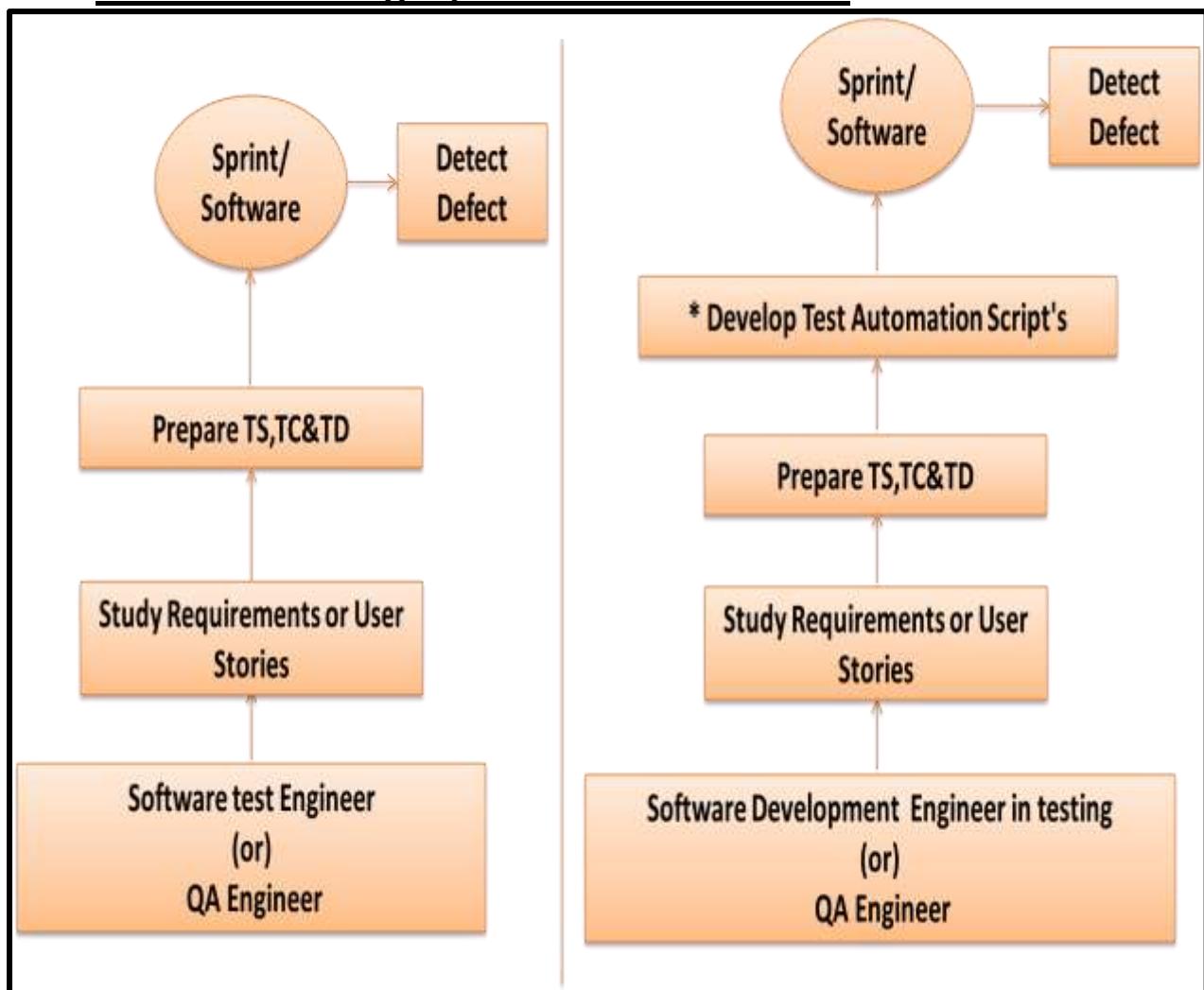
Note 2:-	191
Note 3:-	192
Note 4:-	193
Step 3:- AVD(Android Virtual Device) Creation:	193
Step 4:- Connect real device to tester computer.	195
Step 5:- adb & aapt commands	196
1. adb devices	196
2. adb -s device name shell pm list packages -f	196
3. adb -s device name shell pm list packages -f -s	197
4. adb -s device name shell pm list packages -f -3	197
5. adb -s device name shell pm list packages -f -u	197
6. adb -s device name pull path of app.apk	198
7. adb -s device name install of apk in tester computer	198
8. adb -s device name uninstall path of package name AVD/ARD	198
9. aapt dump badging path of apk in computer	199
Note 1:-	199
Note 2:-	199
Note 3:-	200
Step 6:-	200
Step 7:-	200
Navigation:-	200
Note 1:-	200
Note 2:-	201
Step 8:-	201
Example 1:-	201
Test Script:	201
Example 2:-	202
Test Script:	202
Example 3:-	203
Test Script:	203
Note 1:-	204
Note 2:-	205
Note 3:-	205
1. CloseApp():-	206
Syntax:-	206
2. rotate():-	206
Syntax:-	206

3. pinch():-	206
Syntax:-	206
4. hideKeyboard():-	206
Syntax:-	206
5. openNotifications():-	206
Syntax:-	206
6. pressKeyCode():-	206
Syntax:-	206
7. zoom():-	206
Syntax:-	206
8. context():-	206
Syntax:-	206
9. swipe():-	207
Syntax:-	208
ii). Tester computer OS is MAC (APPLE):-.....	209
Note 1:-	210
VII. SOAP UI.....	213
A. Types of Web Services:-	213
i). Soap based.....	213
ii). Rest based.....	213
B. Soap Based Services v/s Restful Services:-	214
C. Configure SOAP UI TOOL in Tester Computer:-	214
D. Example WSDL's & WADL's:-.....	214
E. Soap based test automation.....	215
Step 1:- Collect service information from developers.....	215
Step 2:- Create Project in SoapUI Tool	215
Step 3:- Create Test Cases:	216
Step 4:- parameterize Request:	217
Step 5:- Insert Assertion:.....	218
Step 6:- Insert Groovy Script:.....	218
Example 1:-	219
Test Script:	220
Example 2:-	220
Test Script:	221
Example 3:-	222
Test Script:	222
Note1.....	223
F. Restable Service automation	224

Step 1:- Collect service information from developers.....	224
Step 2:- Create Project.....	224
Step 3:- Create Test Suite.....	224
Step 4:- Write Groovy Script:-.....	226

I. SDET(Software Development Engineer in Testing)

A. Manual Testing V/S Test Automation

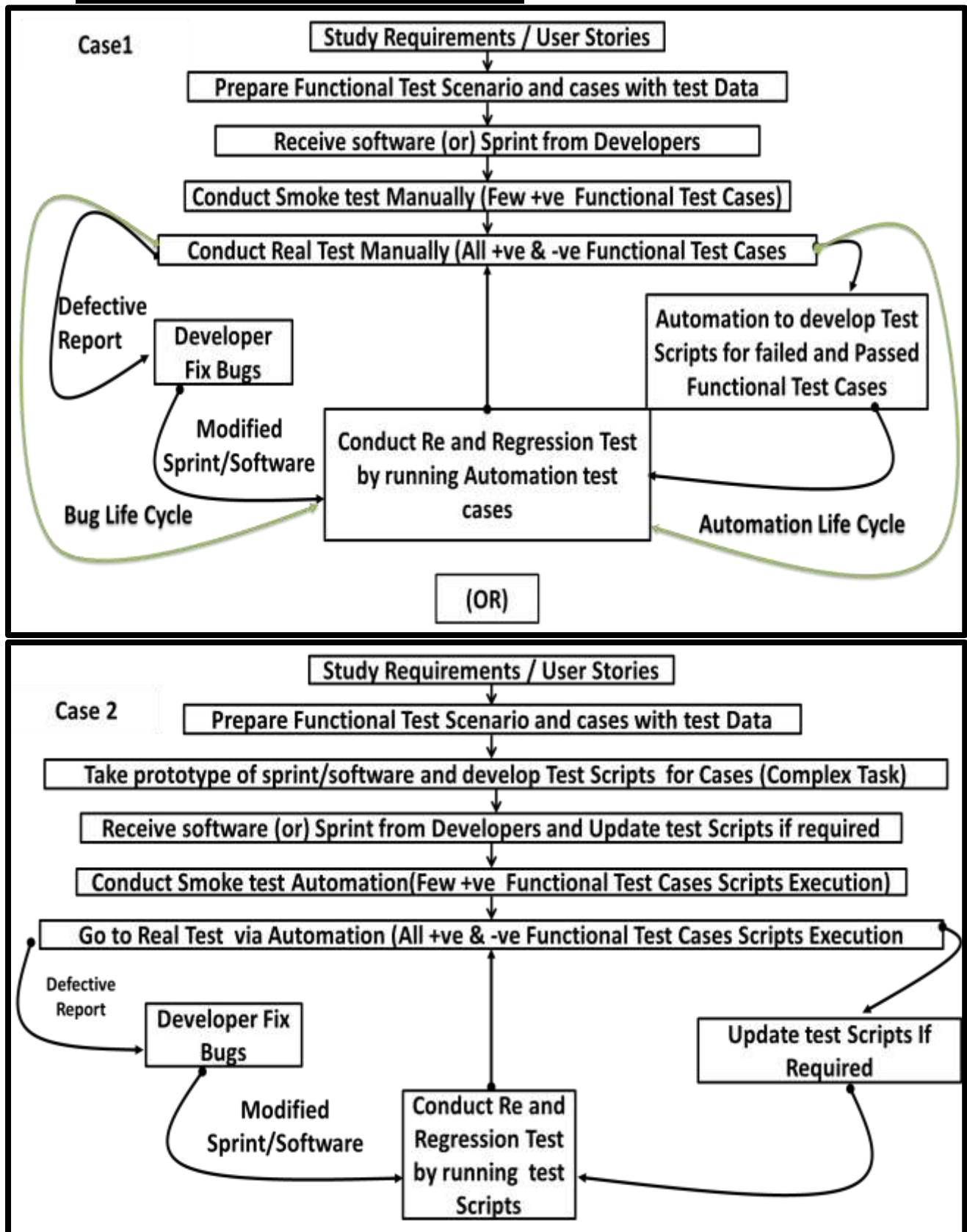


B. Scope of Automation

- ⇒ In General testing process is Cycle Process. In this Process tester need to execute test more than one time to detect defect in software and to confirm those defect are correctly fixed in modified software.
- ⇒ But 100% Test Automation is not possible because testing tools are available in market functional testing and performance testing.
- ⇒ No more well-known tools for other non-functional Testing like usability, compatibility, H/w configuration testing ...etc.

Types of Cases	Manual/Automation	Comments
Functional Testing	Automation	Tools are available like QTP, Selenium
Performance Testing	Automation	Tools are available like Load Runner, Jmeter
Other Non-Functional Testing Like Usability, compatibility, etc.	Manual	No Well Known Tools are in Market

C. Functional Test Automation



Note1

- ⇒ In above two ways of functional test automation, First way is specifying test automation Real and regression testing levels.
- ⇒ Second way is specifying test automation for smoke test, Real test and re and regression test
- ⇒ In second way SDET'S need to work as expert because they need to prepare test scripts depends in Prototypes. Instead of Complete sprint/Software

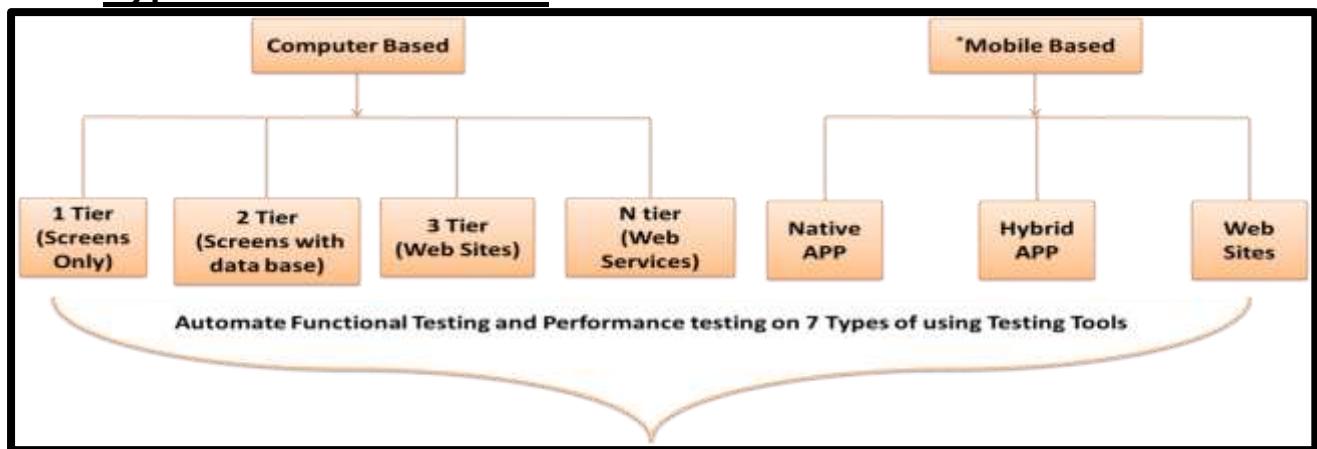
Note2

- ⇒ In above two ways of automation first way more adapted for functional testing.
- ⇒ Second way is more adapted for Performance testing.
- ⇒ In general Performance test is costly to conduct manually because it needs big set up to create load.
- ⇒ Due to this reason we can go for second way of automation for performance because automation tools can prepare virtual load.

Software Testing	Execution Levels	Smoke	Real	Re & Regression	Comments
Functional Testing		Manual	Manual	Automation	Automation can save time
Performance Testing		Automation	Automation	Automation	Manually Conducting is Costly affair
Other Non-Functional Testing (Usability, Compatibility ... Etc.)		Manual	Manual	Manual	No Well-Known Tools in Market

II. Selenium Web Driver

A. Types of Software:-

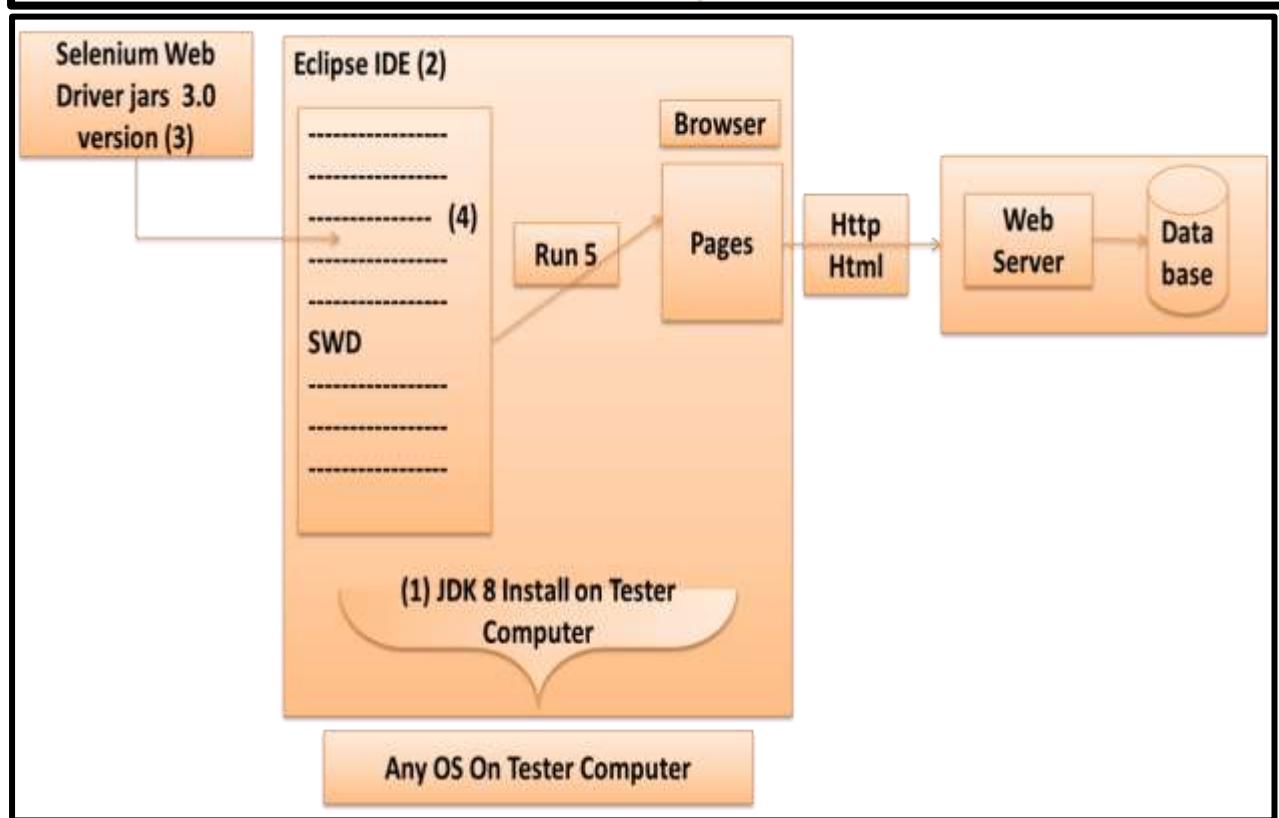
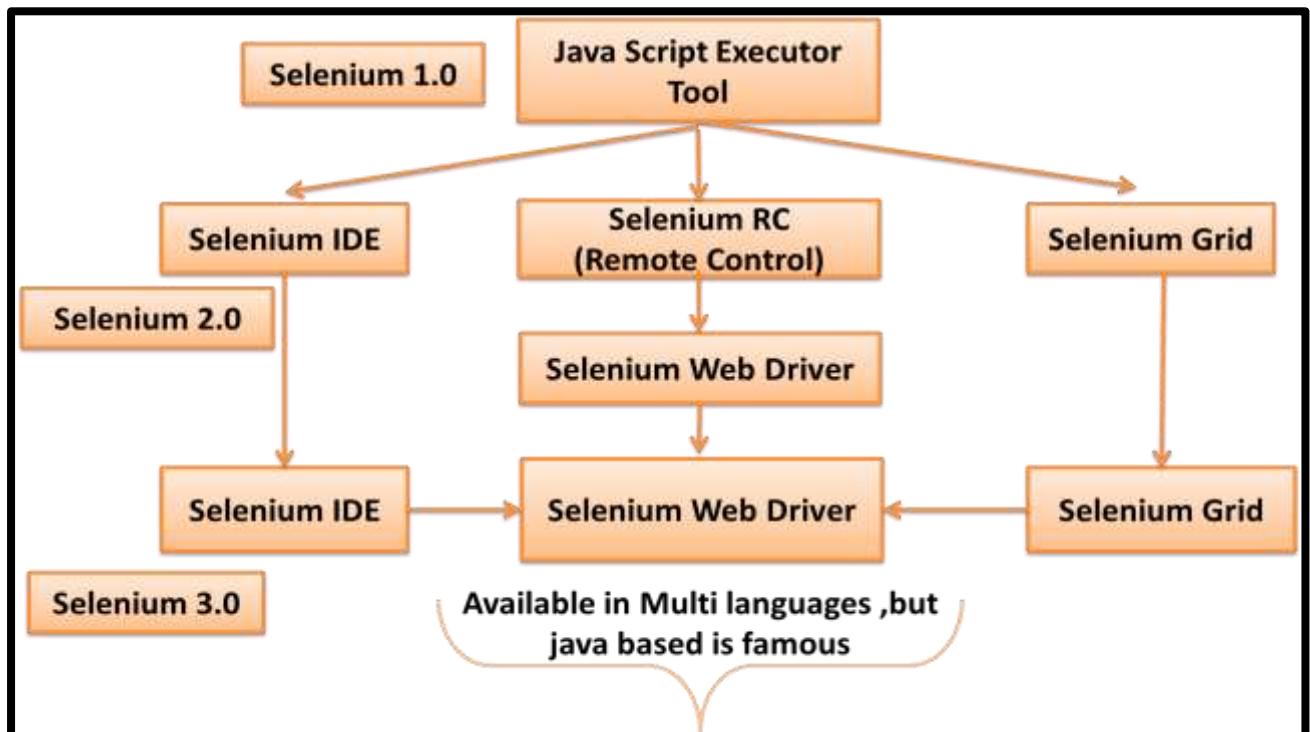


Types of Software	For Functional Testing	For Performance Testing
1 Tier Software (or) Standalone (or) Desktop Software	Java Robot / Auto It / Winium Core java	Load Runner
2 Tier Software (or) Client Server Software	Java Robot / Auto It / Winium SQL Core Java	Load Runner
3 Tier Software (or) Web Sites	Selenium Web Driver Java Robot/Auto It/Winium Sikuli / Protractor SQL Core Java	Load Runner Jmeter
N Tier Software (or) Web series	SopUI SQL Groovy	Load Runner
Native APP, Hybrid APP & Web sites in Android & IOS Mobile	Selenium Web Driver Appium Core Java	-----

B. Configure Selenium web driver in Tester

Software:-

- ⇒ Selenium web driver is used to automate Functional Test on web sites in computer.
- ⇒ Selenium web driver is also useful to automate from mobile apps and sites by integration with Appium. (Android and IOs)
- ⇒ Selenium Web Driver was developed by Simon Stewart.
- ⇒ Selenium Web Driver is available various language C# .Net, Java, Python, Ruby, Perl, JavaScript and PHP, but java based Selenium Web Driver is More Famous because of platform independent.
- ⇒ Open Source (No Need To Purchase License)
- ⇒ Java Script Executor Tool.



Step1:- Download and install JDK8

- ⇒ JDK stands for java development kit
- ⇒ Developed by sun micro systems & take over by oracle.
- ⇒ Need to download and install with respective to corresponding tester computer OS & bit size (32 Bit Or 64 Bit).
- ⇒ To download & install JDK 8 we need to follow below navigation
- ⇒ Go to Google site
 - ↓ Search for JDK 8 Download
 - ↓ Go to Oracle.com
 - ↓ Accept License arrangement
 - ↓ Click on JDK download with respective to O.S and Bit Size
 - ↓ Wait until download complete.
 - ↓ Paste that download in personal folder.
 - ↓ Double click on that download to start installation.
 - ↓ Click next until finish.
 - ↓ Go to copy path of JDK.
Ex: - C:\ProgramFiles\Java\JDK1.8.0
 - ↓ Right Click on My Computer
 - ↓ Go to Advance System Settings.
 - ↓ Select Environment Variables.
 - ↓ Click new in system Variables.
 - ↓ Enter Variable Name as JAVA_HOME
 - ↓ Paste Path in variable value
Ex: - C:\ProgramFiles\Java\JDK1.8.0
 - ↓ Select "Path" variable and click EDIT
 - ↓ Add Path of JDK\Bin at end of existing Path value
 - ↓ Click OK
 - ↓ Close Computer Properties window.

Note:-

1. Selenium Web Driver (Java Based) is best Solution For web sites Functional Test Automation, so we need install JDK and Environment Variable Creation and updation.
2. To ensure the correctness of JDK Installation and Configuration we can go to run java and javac commands at Command Prompt.

Step2:- Download and Launch Eclipse IDE

- ↓ Go to www.google.com.
- ↓ Enter ECLIPSE JUNO Download and click search.
- ↓ Go to eclipse.org site.
- ↓ Click on bit size w.r.t OS for Java Developers.
- ↓ Wait until download to complete.
- ↓ Paste that download in personal folder
- ↓ Extract download in Personal folder
- ↓ Open that Extracted folder
- ↓ Create Shortcut to eclipse software to desktop.

Note1:-

When we launch eclipse for first time it can ask for work space folder to store daily scripts.

- ⇒ Launch Eclipse Ide by Eclipse Icon on desktop
- ⇒ Browse Personal folder path as work space
- ⇒ Select do not ask again check box Click ok.

Note2:-

Eclipse IDE can allow us to change workspace if required.

- ↓ File menu in eclipse IDE.
- ↓ Switch Work Space.
- ↓ Other.
- ↓ Browser another folder as work space.

↓ Click OK.

Note3:-

We need to create java project in Eclipse IDE before going to scripts writing

↓ File menu in Eclipse IDE
↓ New
↓ Other
↓ Java Project
↓ Enter Name of Project as single word in lower case
↓ Click finish
↓ Right Click on created Project
↓ Select properties
↓ Select Java Compiler
↓ Set to latest (1.7)
↓ Select Java build Path
↓ Select Libraries
↓ Remove Existing JRE and Add Latest JRE
↓ Using add Library Option
↓ Click Ok
↓ Click yes to save changes.

Note4:-

We need to create package by following below navigation

↓ Right click on Project in eclipse ide in package Explore
↓ Select New
↓ Select Package
↓ Enter name of package as single word as lower case
↓ Click finish

Note5:-

We need to create Class with main () following below navigation

↓ Right click package in eclipse ide in package Explore
↓ Select New option
↓ Select Class option
↓ Enter name of class as single word with Initcap
↓ Select main() option
↓ Click finish

In main (), we write automation code related selenium web driver

Step3:-Download and associate Selenium web Driver

Jars

→ Java based Selenium web driver is available in jars files.

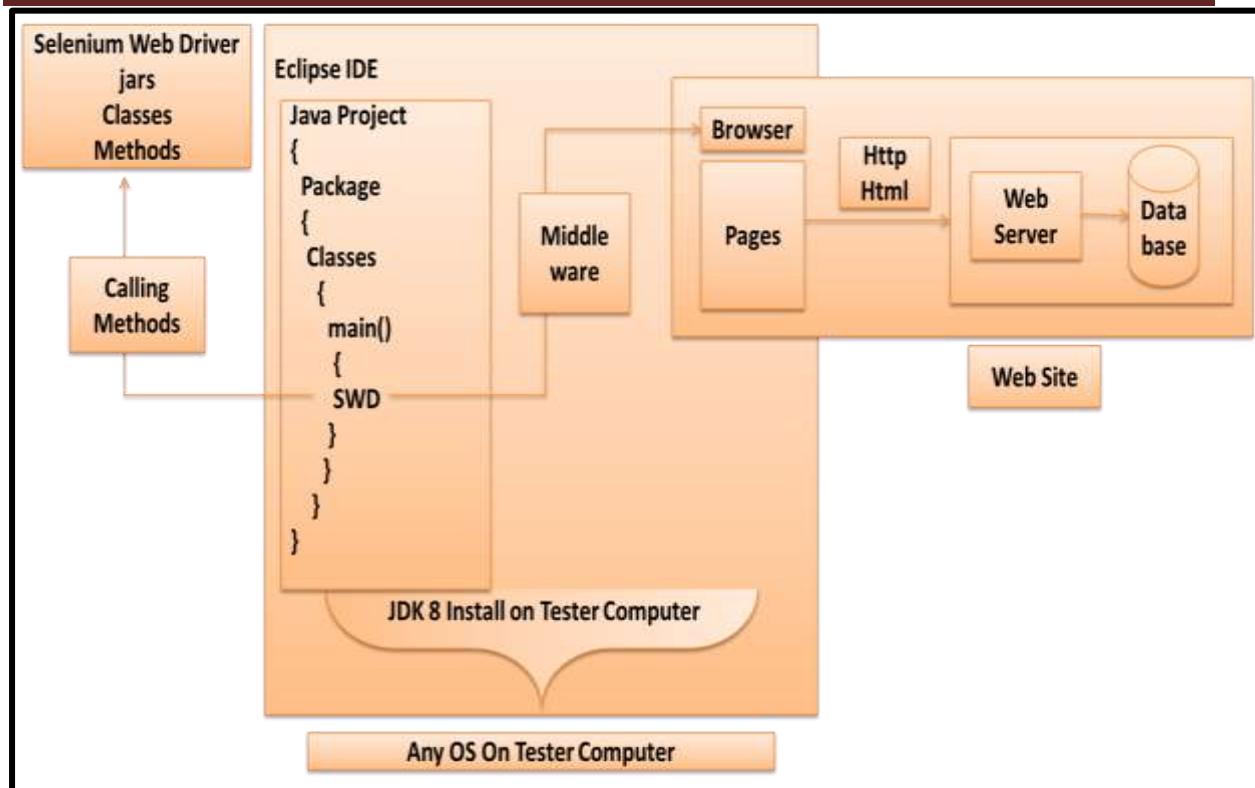
→ Every jar file consists of Classes.

→ Every classes file consists of Methods to automate web pages.

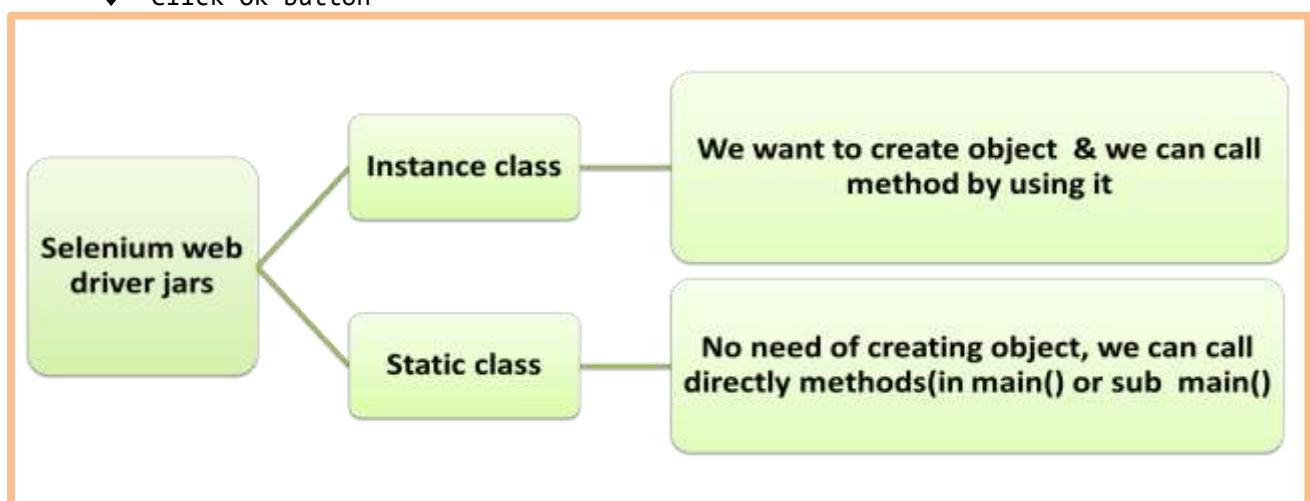
→ We need to use methods in main () of the classes

We need to follow below navigation to download and associate selenium web driver jars

↓ Go to Google site
↓ Enter Selenium web driver download and Click Search
↓ Go to seleniumhq.org official web site
↓ Go to download Page

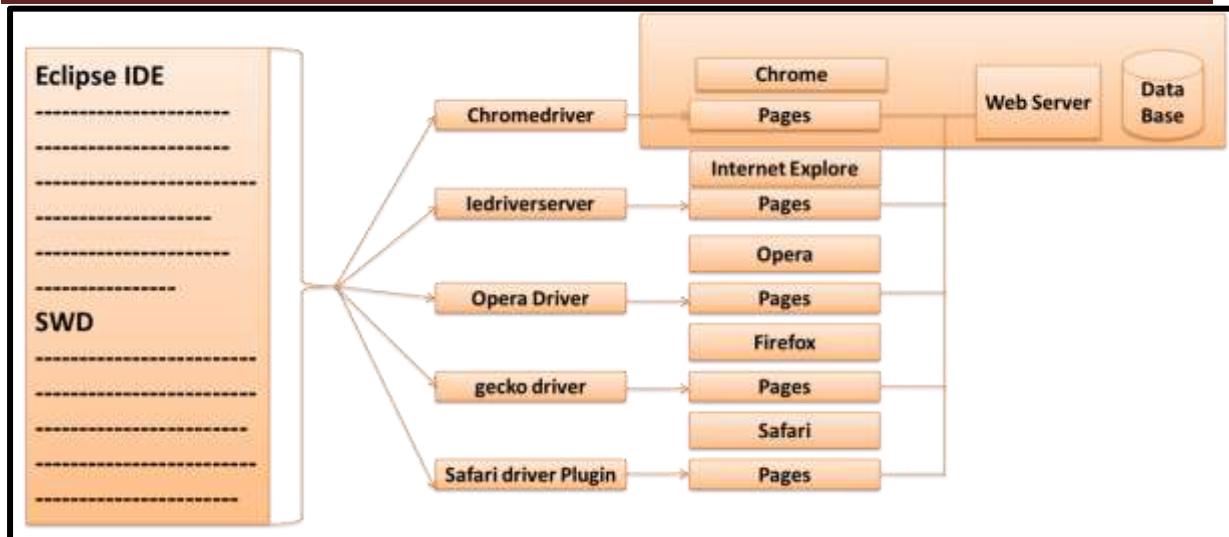


- ↓ Click on download for selenium web driver of java
- ↓ Paste that download in personal folder
- ↓ extract that download (one jar file is outside of lib folder and more jar files in side of lib folder)
- ↓ Right click on project in eclipse
- ↓ Go to Properties
- ↓ Go to java build path
- ↓ Select libraries.
- ↓ Select add external libraries jar button.
- ↓ Browse the path of jar files in lib folder inside of lib folder and outside of lib folder.
- ↓ Click Ok button



Step4:- Download and use browser Drivers

- ⇒ To integrate scripts in eclipse IDE and web pages in a browser, we can use browser driver software
- ⇒ This software is executable to launch browser by integrating with scripts in Eclipse IDE.

**Example1:**

Launch Chrome browser (ver57.0)

Step1:-

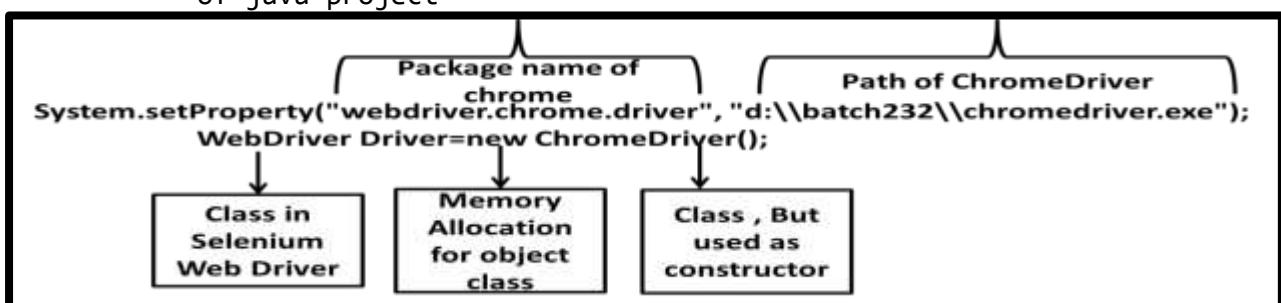
Download and install chrome browser in tester computer (Latest Version)

Step2:-

- ↓ Download chrome driver from seleniumhq.org site
- ↓ Paste that download in personal folder
- ↓ Extract that download
- ↓ Remember path of chromedriver.exe download

Step3:-

Write below like selenium code in main() in class in package of java project

**Example2:**

Launch Internet Explorer browser

Step1:-

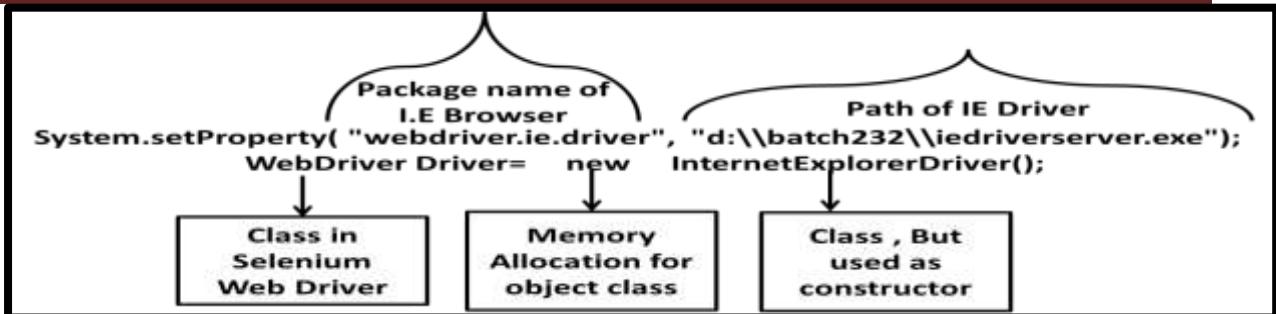
Check the availability of Internet Explorer browser, because Internet Explorer is in Built in windows OS (version like IE7 or IE8 or IE9 or IE10)

Step2:-

- ↓ Download Iedriverserver from seleniumhq.org site
- ↓ Paste that download in personal folder
- ↓ Extract that download
- ↓ Remember path of Iedriverserver.exe download

Step3:-

Write below like selenium code in main() in class in package of java project

**Example3:**

Launch Mozilla Firefox browser

Step1:-

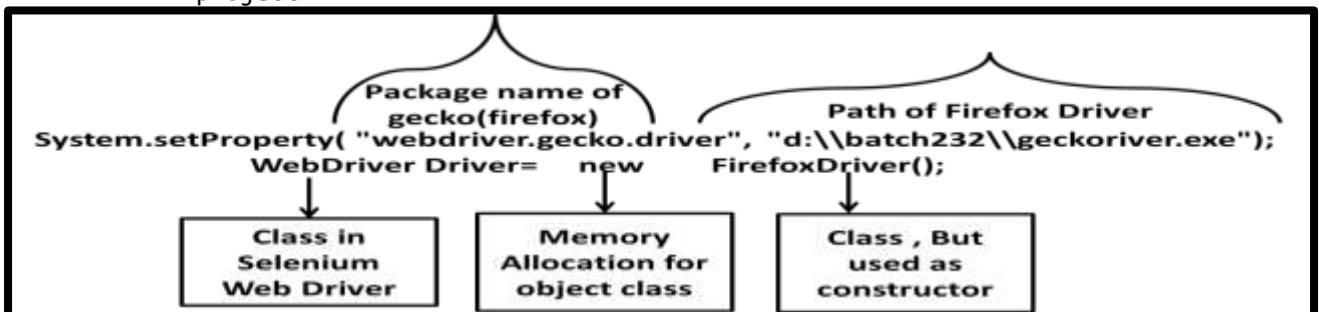
Download and install Firefox browser in tester computer (Latest Version45/46/47)

Step2:-

- ↓ Download gecko driver from seleniumhq.org site
- ↓ Paste that download in personal folder
- ↓ Extract that download
- ↓ Remember path of geckodriver.exe download

Step3:-

Write below like selenium code in main() in class in package of java project

**Example4:**

Launch opera browser

Step1:-

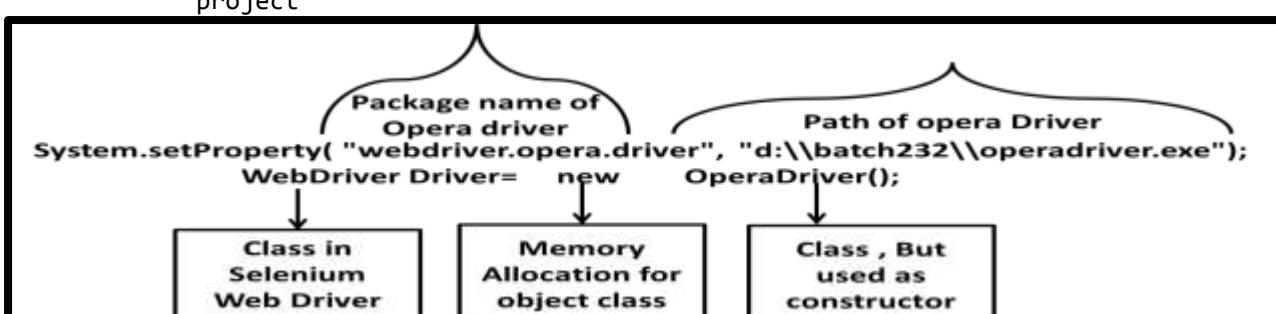
Download and install opera browser in tester computer (Latest Version)

Step2:-

- ↓ Download operadriver from seleniumhq.org site
- ↓ Paste that download in personal folder
- ↓ Extract that download
- ↓ Remember path of operadriver.exe download

Step3:-

Write below like selenium code in main() in class in package of java project

**Example5:**

Launch safari browser. (Only in Apple MAC).

Step1:-

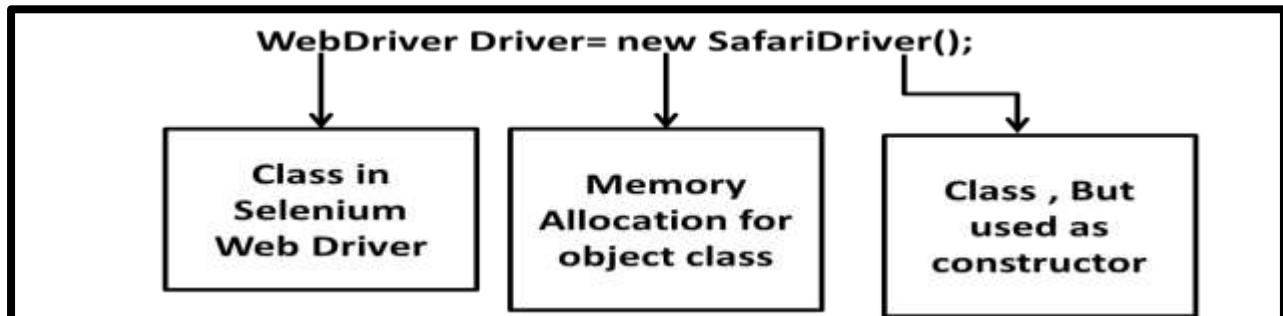
Check the availability of safari browser in Apple MAC Computer because safari browser is in built in Apple Mac Computer.

Step2:-

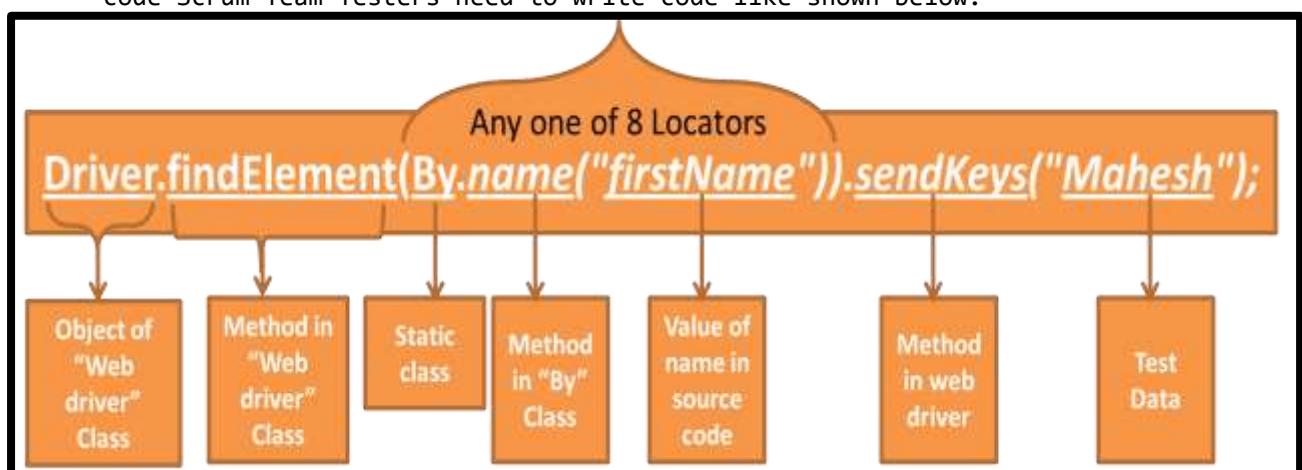
- ↓ Open seleniumhq.org site using safari browser.
- ↓ Click on latest release of safari driver.
- ↓ Click install to add safari driver as plugin to safari driver.

Step3:-

Write below like selenium code in main() in class in package of java project

**Step5:- Locating Elements in webpages for Automation**

After Completion of auto machine test environment establishment corresponding Selenium team testers can concentrate on auto machine code writing using selenium (java based) in Eclipse IDE. In that Auto machine code Scrum Team Testers need to write code like shown below.



From the above example code selenium can support 8 locators to locate elements in web pages before operating?

- * ⇒ By.name()
- ⇒ By.id()
- ⇒ By.className()
- ⇒ By.tagName()
- ⇒ By.linkText()
- ⇒ By.partialLinkText()
- ⇒ By.xpath()
- ⇒ By.cssSelector()

Test Scenario

Launch Mecurytours.com site

Click Register Link

Fill Fields

Close Site

Test Script:

```

package stevejobs;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

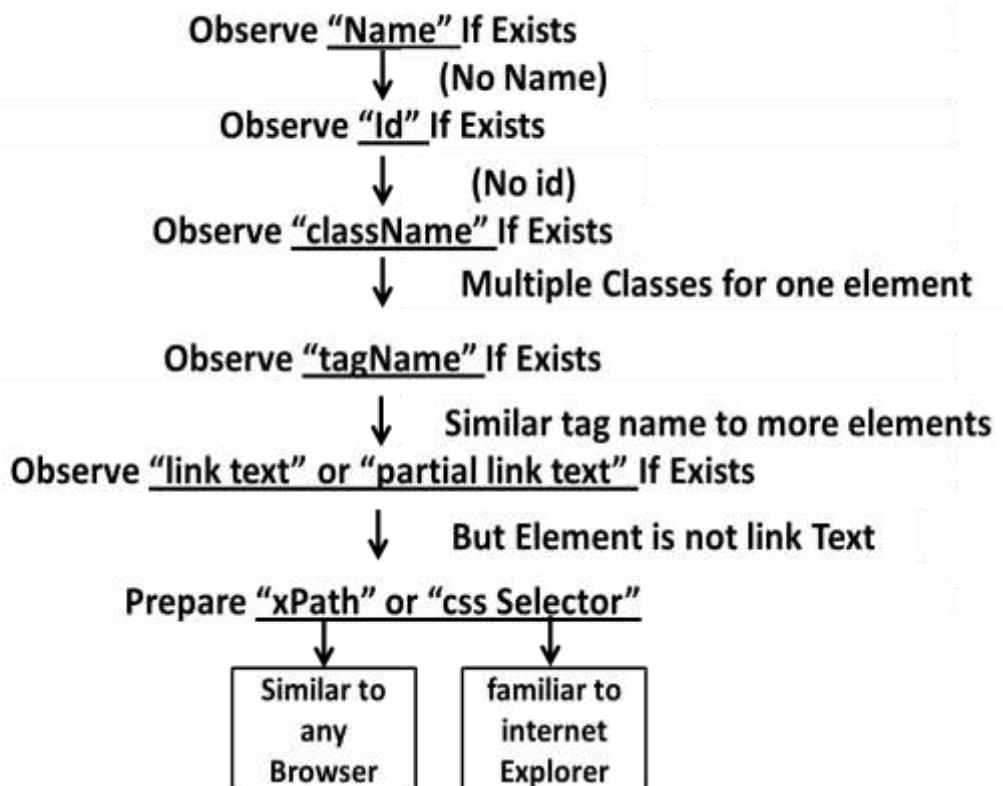
```

```
public class MecuryTours
{
    public static void main(String[] args) throws InterruptedException
    {
        //launch site using ChromeDriver
        System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
        WebDriver Driver=new ChromeDriver();
        //Open web site of tours
        Driver.get("http://newtours.demoaut.com/");
        Thread.sleep(5000);
        // Opening Registration link
        Driver.findElement(By.linkText("REGISTER")).click();
        Thread.sleep(5000);
        Driver.findElement(By.name("firstName")).sendKeys("Mahesh");
        Driver.findElement(By.name("lastName")).sendKeys("ayyappa");
        Driver.findElement(By.name("phone")).sendKeys("9705955572");
        Driver.findElement(By.name("userName")).sendKeys("maheh@gmail.com");
        Driver.findElement(By.name("address1")).sendKeys("D.no16-17-27/A");
        Driver.findElement(By.name("address2")).sendKeys("3rd Crossroad,
Guntur");
        Driver.findElement(By.name("city")).sendKeys("Guntur");
        Driver.findElement(By.name("state")).sendKeys("Andhra Pradesh");
        Driver.findElement(By.name("postalCode")).sendKeys("522001");
        Thread.sleep(5000);
        Select s=new Select(Driver.findElement(By.name("country")));
        s.selectByVisibleText("INDIA");
        Driver.findElement(By.name("email")).sendKeys("Mahesh124");
        Driver.findElement(By.name("password")).sendKeys("ayyappa");
        Driver.findElement(By.name("confirmPassword")).sendKeys("ayyappa");
        Driver.findElement(By.name("register")).click();
        Thread.sleep(5000);
        Driver.close();
    }
}
```

Variation	Description	Sample
<code>By.className</code>	finds elements based on the value of the "class" attribute	<code>driver.findElement(By.className("Element_ClassName"))</code>
<code>By.cssSelector</code>	finds elements based on the driver's underlying CSS Selector engine	<code>driver.findElement(By.cssSelector("Element_CssSelector"))</code>
<code>By.id</code>	locates elements by the value of their id attribute	<code>driver.findElement(By.id("Element_ID"))</code>
<code>By.linkText</code>	finds a link element by the exact text it displays	<code>driver.findElement(By.linkText("Element_LinkText"))</code>
<code>By.name</code>	locates elements by the value of the "name" attribute	<code>driver.findElement(By.name("Element_Name"))</code>
<code>By.partialLinkText</code>	locates elements that contain the given link text	<code>driver.findElement(By.partialLinkText("Element_PartialLinkText"))</code>
<code>By.tagName</code>	locates elements by their tag name	<code>driver.findElement(By.tagName("Element_TagName"))</code>
<code>By.xpath</code>	locates elements by their XPath	<code>driver.findElement(By.xpath("Element_Xpath"));</code>

While locating elements in webpages, we can use any one of 8 ways in selenium. To choose best locator for an element, we can follow below algorithm

Apply "Inspect Element" option on element in page for source code



Note1:-

In general SDET's can use different inspector in different browser to get details of elements in web page

Browser	Inspector	Need for installation
Firefox(45,46,47)	Inspect element	Inbuilt Facility
	Fire bug	Need to install
	Fire path	Need to install
	Etc.	
Chrome	Inspect	Inbuilt Facility
	Ospy	Need to install
	XPath helper	Need to install
	Etc.	
Internet Explorer	Tool Menu→Developer Tools	Inbuilt Facility
	Dom Inspector Document object Model	Need to install
	Etc.	

Note2:-

If we want to install any plugin for Firefox browser, we need to follow below navigation.

- ↓ Open Firefox browser
- ↓ Go to open menu option
- ↓ Select Add-ons
- ↓ Enter Plug name in search box
 - Example:-Firebug and click search button
 - Fire path and click search button
- ↓ Click install required Plugin
- ↓ Restart Firefox browser

Note3:-

If we want to install any plugin for chrome browser, we need to follow below navigation.

- ↓ Open chrome browser
- ↓ Go to Google page
- ↓ Enter requirement plugin for chrome browser
- ↓ Example:-Ospy for chrome or Xpathhelper for chrome
- ↓ Click search button
- ↓ Go to Chrome web store
- ↓ Click Add to chrome
- ↓ Restart Chrome browser

C. Types of xPath:-

X-Path is address of an element in web page source code

X-Path is two types such as

⇒ Absolute X-path

⇒ Relative X-path

⇒ Absolute x-path is an address of an element by forming tags from html to target element tag.

Example:-1

```
<html>
  <body>
    <input>
    <input>
    <input>
  </body>
</html>
```

→ Absolute xPath

/html/body/input/input/input

(OR)

/html/body/input[3]

Example:-2

```
<html>
  <body>
    <div>
      <input>
      <p> </p>
    </div>
    <div>
      <input>
      <p> </p>
    </div>
  </body>
</html>
```

Absolute xPath

/html/body/div/div/input

(OR)

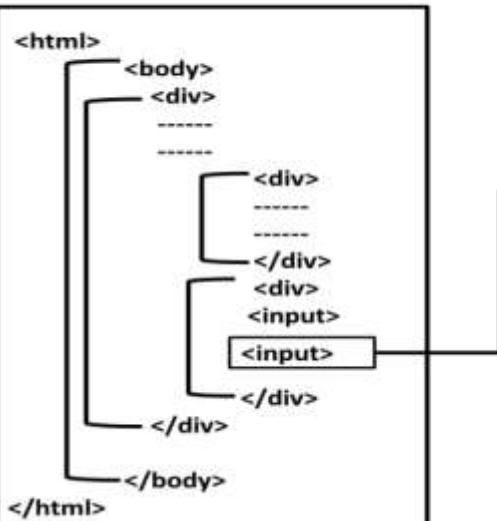
/html/body/div[2]/input

Example:-3

```
<html>
  <body>
    <div>
      <div>
        -----
        -----
      </div>
      <div>
        -----
        -----
      </div>
      <div>
        <div>
          <input>
          <input>
        </div>
      </div>
    </body>
</html>
```

Absolute xPath

/html/body/div[3]/div/input[2]

Example:-4

Absolute xPath
`/html/body/div/div[2]/input[2]`
Or
`/html//div/div[2]/input[2]`

In above Absolute xPath concept “/” can define relation between parent tag and child tag. Whereas “//” can define relation between grandparent and child tags.

In General SDET are not interested to use absolute ‘xPath’ to locate element in web page because absolute xPath of an element will be changed with respect to changes in html sources. Due to this reason, SDET can go to relative xPath to locate elements in dynamically changes in web pages.

In general relative x-path is syntax based.

Syntax1:-

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

The screenshot shows the Firebug interface with the 'Highlight' tab selected. The XPath expression `//input[@type='button']` is entered in the 'XPath:' field. The DOM tree on the left shows a hierarchy of div elements. A blue box highlights the `<input class="button button br2" type="button" onclick="javascript:register();" value="Register Here"/>` element. Arrows point from the highlighted element in the DOM tree to the relative XPath expression and then to the final syntax `//tagname[@attribute='value']`.

Syntax2:-

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

Here * means any tag name in xPath concepts in selenium

The screenshot shows the Firebug developer tool interface. In the top navigation bar, 'XPath' is selected. Below it, the XPath expression `//*[@type='button']` is entered. The DOM tree on the left shows a portion of a login page with various HTML elements like `<div>`, `<p>`, `<input>`, and `<form>`. Two specific `<input>` elements are highlighted with a blue selection bar at the bottom of the tree. One is located in a search bar with the value 'Find Mobile Location', and the other is a button labeled 'Locate'. To the right of the DOM tree, a diagram illustrates the XPath expression. It starts with a box labeled `//*[@type='button']`, which has two arrows pointing down to a second box labeled `//*[@attribute='value']`.

Syntax3:-

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

To find an element when both conditions are true

This screenshot of Firebug shows a more complex XPath query. The 'XPath' tab is active, and the expression `//*[@type='button'][@value='Register Here']` is typed in. The DOM tree on the left shows the same login page structure as the previous screenshot. A specific button with the value 'Register Here' is highlighted with a blue selection bar. To the right, a diagram shows the breakdown of the XPath. It starts with a box labeled `//*[@attribute='value'][@attribute='value']`, which has four arrows pointing down to a final box labeled `//*[@type='button'][@value='Register Here']`.

Syntax4:-

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

To find an element when any one conditions is true (Like toggle button by click caption of submit and it will change to logout for one button)

```


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



//*[@type='button' OR @value='Register Here']


```

The diagram shows the breakdown of the XPath expression `//*[attribute='value' OR attribute='value']` into `//*[type='button' OR value='Register Here']`. Arrows point from the top box to each part of the bottom box.

Syntax5:-

```
(//*[attribute='value']) [INDEX]
```

If we are finding many elements of attribute values to target of our element we can use index option like below

```


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



(//*[type='button']) [2]

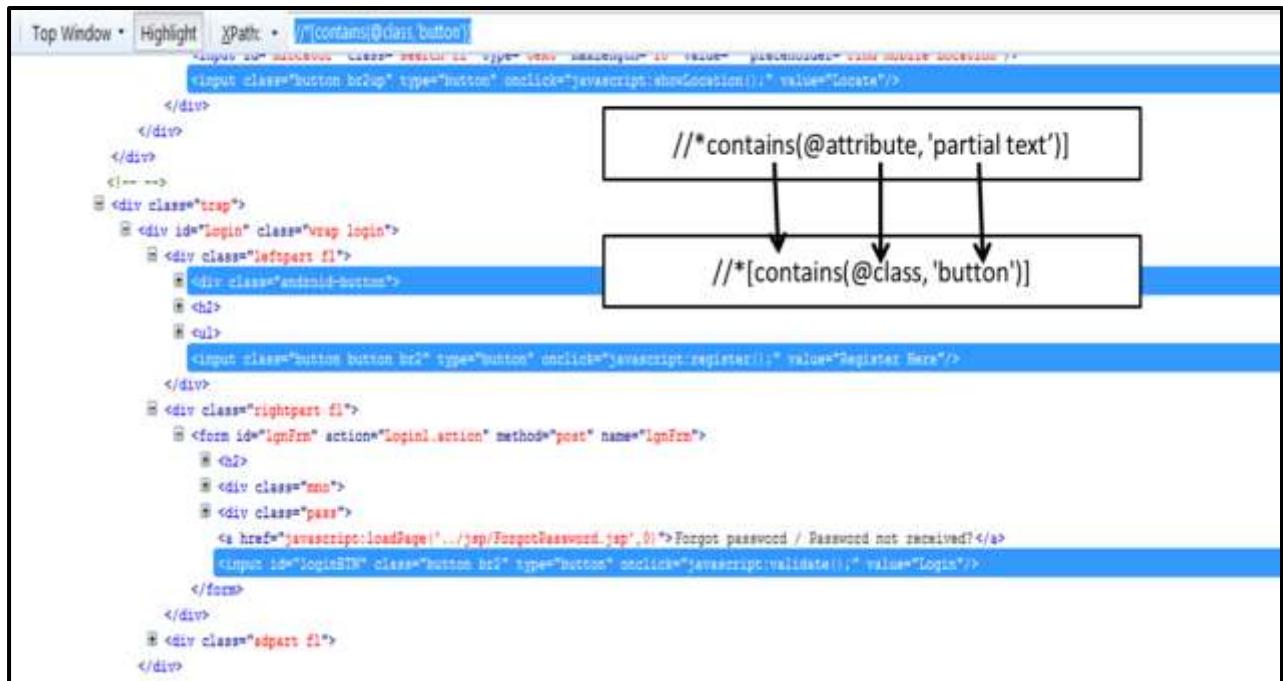

```

The diagram shows the breakdown of the XPath expression `(//*[attribute='value']) [index]` into `(//*[type='button']) [2]`. Arrows point from the top box to each part of the bottom box.

Syntax6:-

//*[contains (@attribute, 'partial text')]

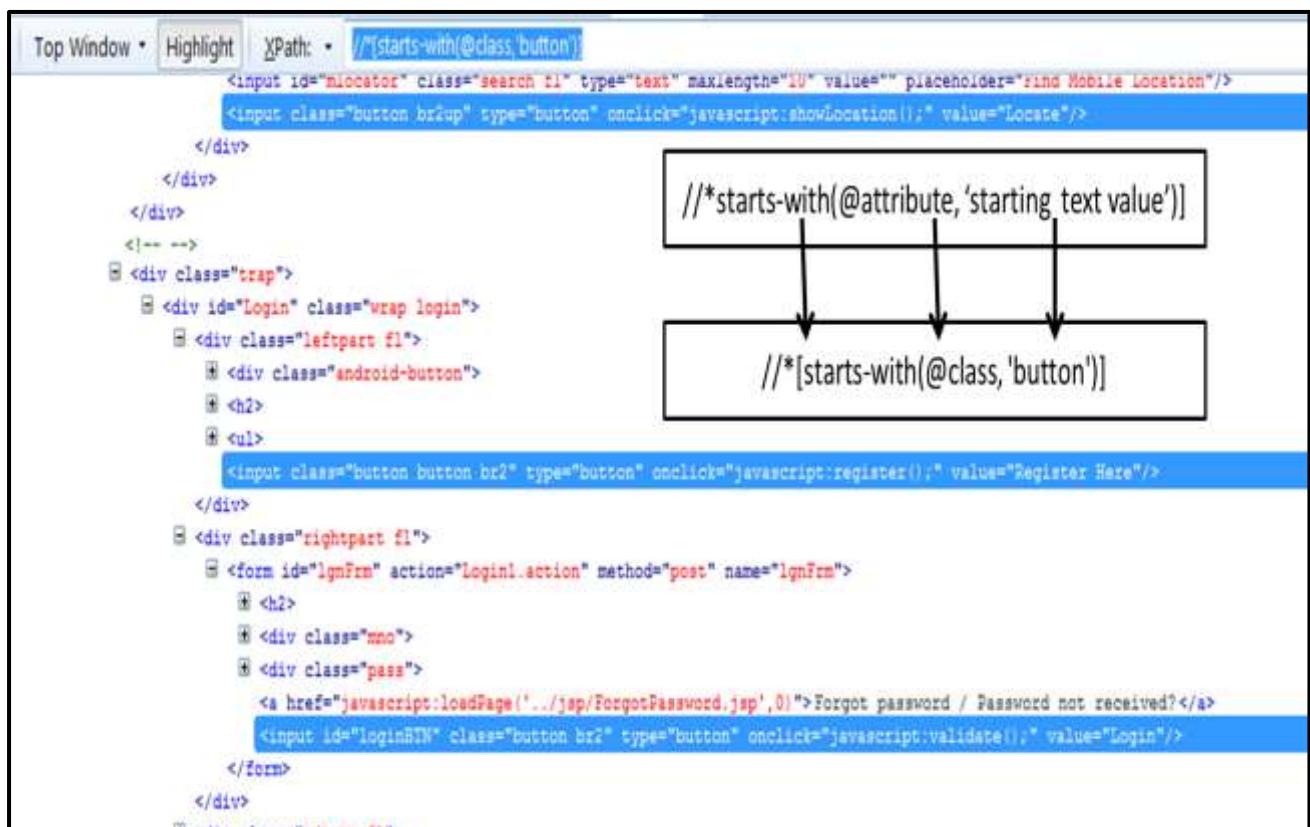
The contain feature has an ability to find the element with partial text, when attribute values are length is high then we can use partial text with contains feature as shown below.



Syntax7:-

//*[starts-with (@attribute, 'starting text')]

Start-with function finds the element whose attribute value matches the starting text of element.



Syntax8:-

```
//*[text() ='text value']
//*[contains (text(), 'partial text')]
//*[starts-with (text(), 'starting text')]
When we are find value of text without having any attribute like class name, tag name, id, name, Etc.
Here a text value is between starting tag to ending tag of source element.
```

The screenshot shows the Firebug developer tools interface. The 'Highlight' tab is selected, displaying the XPath expression `//*[text()='Send Free SMS']`. The DOM tree on the left shows various HTML elements. A blue box highlights the text node `<h3>Send Free SMS</h3>` within the `<li class='mid'>` element. To the right, a flowchart diagram illustrates the XPath expression: it starts with a box containing `//*[text() ='text value']`, which points down to another box containing `//*[text()='Send Free SMS']`.

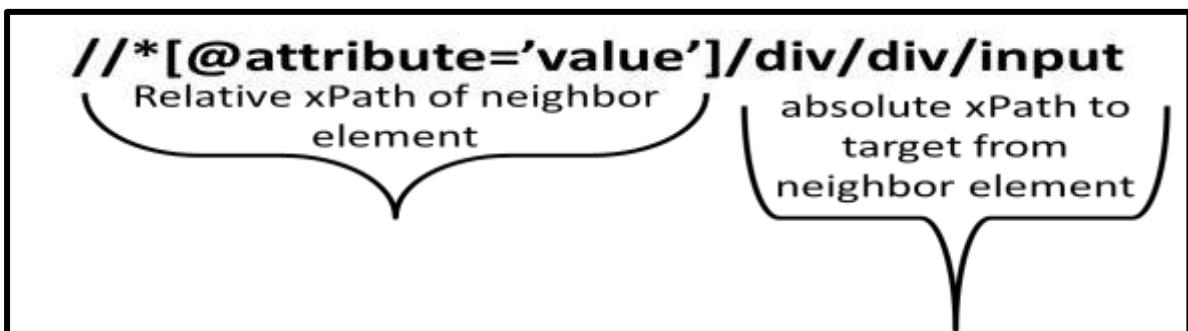
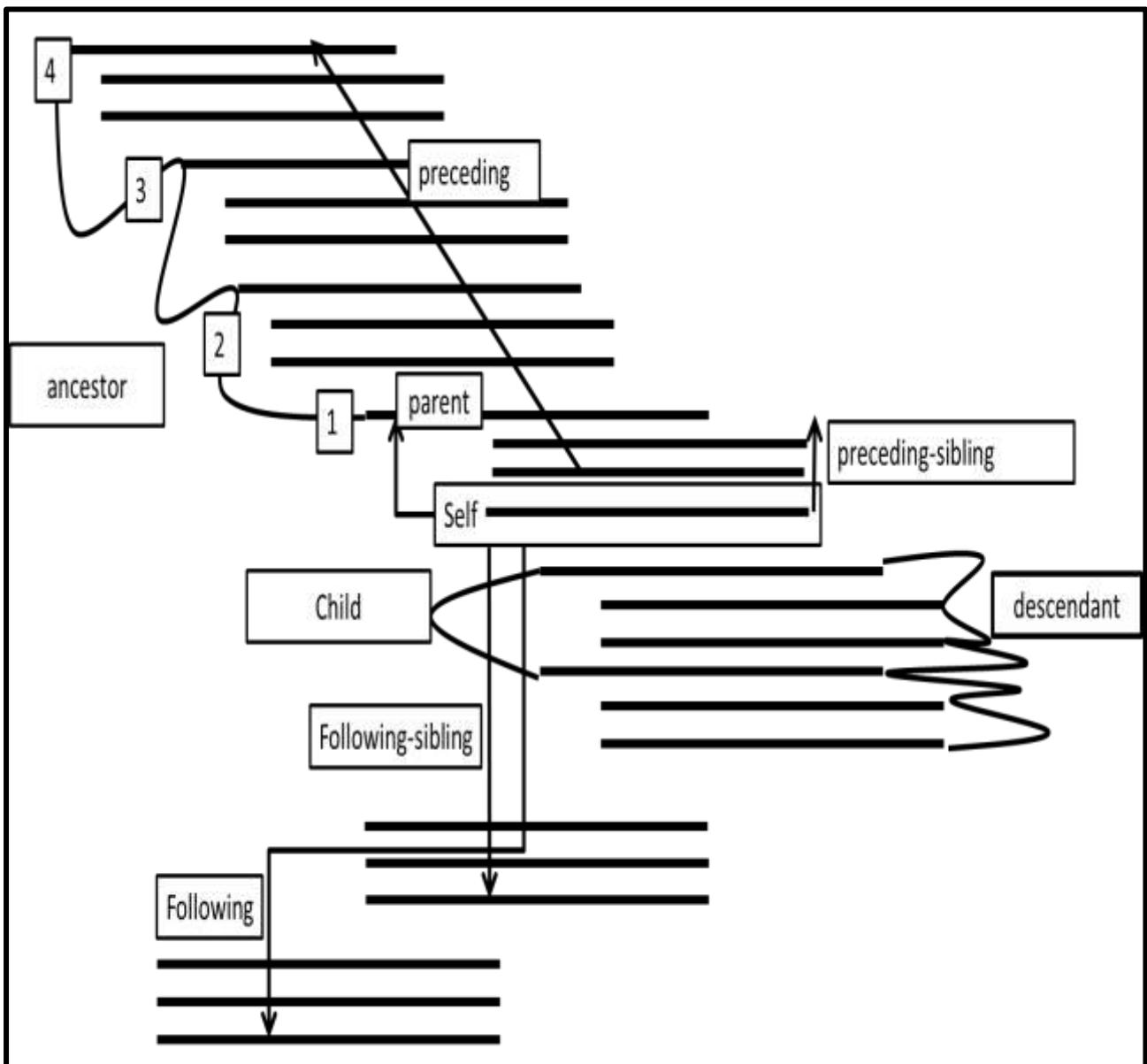
Syntax9:-

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

The screenshot shows the Firebug developer tools interface. The 'Highlight' tab is selected, displaying the XPath expression `(//*[@type='button'])[-last()]`. The DOM tree on the left shows various HTML elements. A blue box highlights the button element `<input id='loginBTN' type='button' value='Login' onclick="javascript:validate()"/>` within the `<form id='lgnFrm'>`. To the right, a flowchart diagram illustrates the XPath expression: it starts with a box containing `(//*[@attribute='value'])[-last()] or (//*[@attribute='value'])[-last ()-1]`, which points down to another box containing `(//*[@type='button'])[-last()]`.

Syntax10:-

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

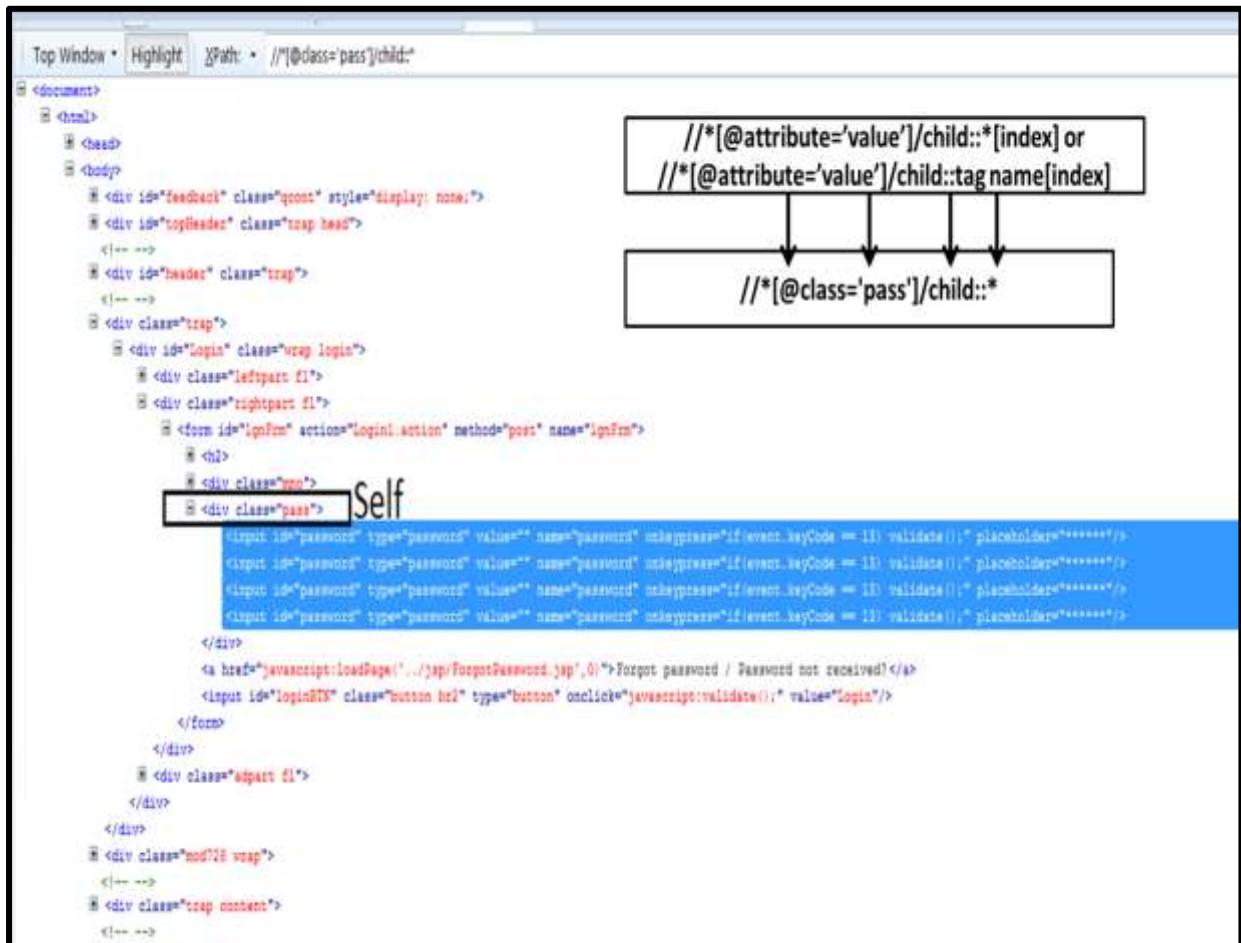
**xPath Axes Relation Flow Diagram**

Syntax11:-

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

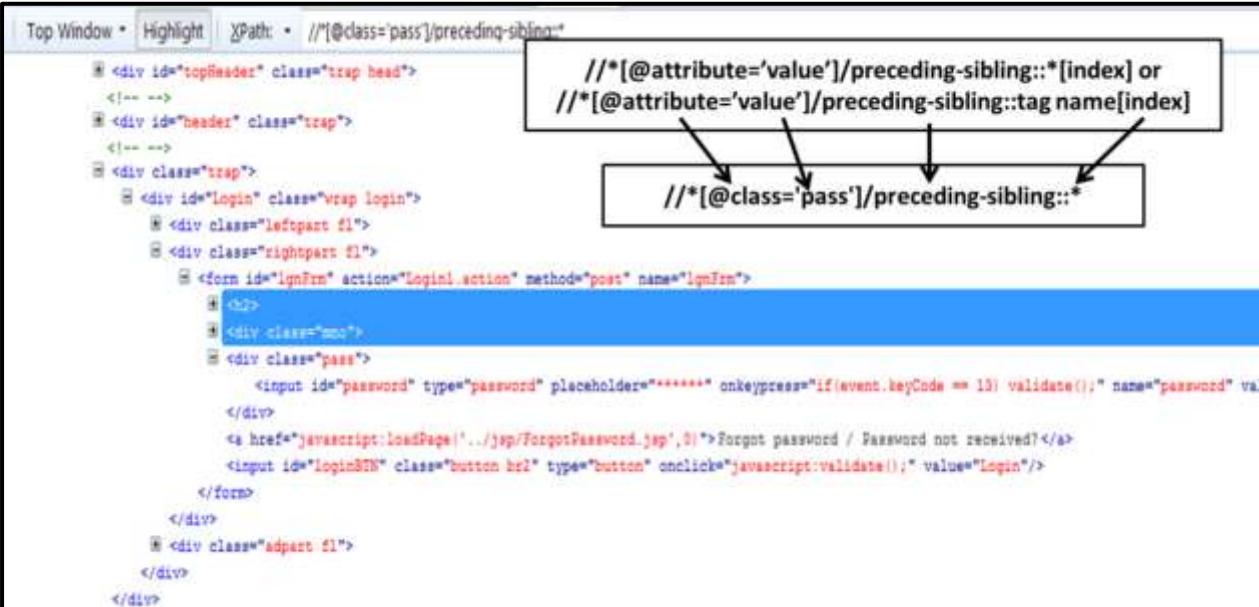
**Syntax12:-**

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



Syntax13:-

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



The screenshot shows the Firebug DOM panel with the 'Highlight' tab selected. The URL is `http://[redacted]/preceding-sibling*`. The DOM tree is displayed, and a specific node under the `<div class='pass'>` element is highlighted in blue. A callout box at the top right contains the XPaths for preceding-sibling selection. Arrows point from this box to the original syntax examples and the highlighted node.

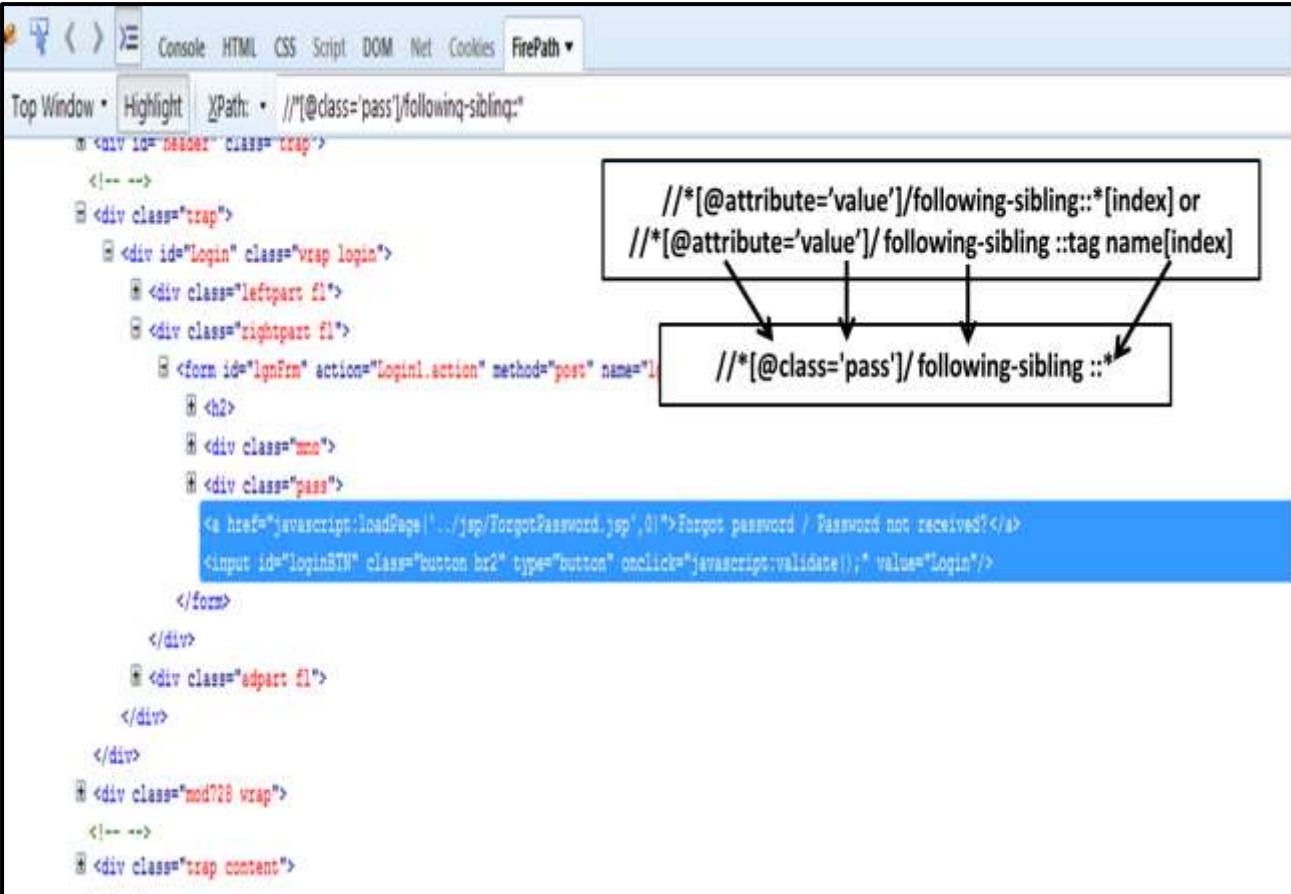
```

<div id="topHeader" class="trap head">
<!-- -->
<div id="header" class="trap">
<!-- -->
<div class="trap">
  <div id="Login" class="trap login">
    <div class="leftpart fl">
    <div class="rightpart fl">
      <form id="lgnFrm" action="LogIn.action" method="post" name="lgnFrm">
        <div>
          <div class="mc">
            <div class="pass">
              <input id="password" type="password" placeholder="*****" onkeypress="if(event.keyCode == 13) validate();" name="password" value="1234567890"/>
            </div>
            <a href="javascript:loadPage('../jsp/ForgotPassword.jsp',0)">Forgot password / Password not received?</a>
            <input id="loginBTN" class="button br1" type="button" onclick="javascript:validate();" value="Login"/>
          </form>
        </div>
      </div>
      <div class="adpart fl">
      </div>
    </div>
  </div>
</div>

```

Syntax14:-

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



The screenshot shows the Firebug DOM panel with the 'Highlight' tab selected. The URL is `http://[redacted]/following-sibling*`. The DOM tree is displayed, and a specific node under the `<div class='pass'>` element is highlighted in blue. A callout box at the top right contains the XPaths for following-sibling selection. Arrows point from this box to the original syntax examples and the highlighted node.

```

<div id="header" class="trap">
<!-- -->
<div class="trap">
  <div id="Login" class="trap login">
    <div class="leftpart fl">
    <div class="rightpart fl">
      <form id="lgnFrm" action="LogIn.action" method="post" name="lgnFrm">
        <div>
          <div class="mc">
            <div class="pass">
              <input id="password" type="password" placeholder="*****" onkeypress="if(event.keyCode == 13) validate();" name="password" value="1234567890"/>
            </div>
            <a href="javascript:loadPage('../jsp/ForgotPassword.jsp',0)">Forgot password / Password not received?</a>
            <input id="loginBTN" class="button br2" type="button" onclick="javascript:validate();" value="Login"/>
          </form>
        </div>
      </div>
      <div class="adpart fl">
      </div>
    </div>
  </div>
</div>

```

Syntax15:-

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

Top Window • Highlight XPath: /*[@class='pass']/ancestor::*

```
<document>
  <html>
    <head>
    <body>
      <div id="feedback" class="qcont" style="display: none;">
      <div id="topHeader" class="trap_head">
        <!-- -->
      <div id="header" class="trap">
        <!-- -->
      <div class="trap">
        <div id="Login" class="trap_login">
          <div class="leftpart fl">
          <div class="rightpart fl">
            <form id="lgnfrm" action="Login.action" method="post" name="lgnfrm">
              </div>
            <div class="adpart fl">
              </div>
            </div>
          <div class="mod723 vrap">
            <!-- -->
          <div class="trap_content">
```

Syntax16:-

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

Top Window • Highlight XPath: /*[@class='pass']/descendant::*

```
<document>
  <html>
    <head>
    <body>
      <div id="feedback" class="qcont" style="display: none;">
      <div id="topHeader" class="trap_head">
        <!-- -->
      <div id="header" class="trap">
        <!-- -->
      <div class="trap">
        <div id="Login" class="trap_login">
          <div class="leftpart fl">
          <div class="rightpart fl">
            <form id="lgnfrm" action="Login.action" method="post" name="lgnfrm">
              <h2>
                <div class="mso">
                  <div class="pass">
                    <input id="password" type="password" placeholder="*****" onkeypress="if(event.keyCode == 13) validate();" name="password" value="" />
                  </div>
                <a href="javascript:loadPage('.../jsp/ForgetPassword.jsp',0)">Forgot password / Password not received?</a>
                <input id="loginbtn" class="button hr" type="button" onclick="javascript:validate();" value="Login"/>
              </form>
            </div>
          <div class="adpart fl">
            </div>
          </div>
        <div class="mod723 vrap">
          <!-- -->
        <div class="trap_content">
          <!-- -->
        <div class="trap">
```

Syntax17:-

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

```

Top Window * Highlight XPath: * //[@class='pass']/preceding::*
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Raybits - India's largest free messaging destination</title>
    <link href="http://www.raybits.com/images/favicon.ico" type="image/x-icon" rel="shortcut icon" />
  </head>
  <body>
    <div id="header">
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
                                                                                        <div>
              //*[@attribute='value']/preceding::*[index] or
              //*[@attribute='value']/preceding::tag name[index]
              //*[@class='pass']/preceding::*[index]
            
```

Syntax18:-

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

```

Top Window * Highlight XPath: * //[@class='pass']/following::*
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Raybits - India's largest free messaging destination</title>
    <link href="http://www.raybits.com/images/favicon.ico" type="image/x-icon" rel="shortcut icon" />
  </head>
  <body>
    <div id="header">
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
          //*[@attribute='value']/following::*[index] or
          //*[@attribute='value']/following::tag name[index]
          //*[@class='pass']/following::*[index]
        
```

Relative xPath (Axes relation)
Syntax1:- //tagname[@attribute='value']
Syntax2:- //*[@attribute='value']
Syntax3:- //*[@attribute='value'] [@attribute='value']
Syntax4:- //*[@attribute='value' OR @attribute='value']
Syntax5:- (//*[@attribute='value']) [INDEX]
Syntax6:- //*[contains (@attribute, 'partial text')]
Syntax7:- //*starts-with (@attribute, 'starting text')
Syntax8:- //*text() ='text value' Or /*contains (text(), 'partial text') Or /*starts-with (text(), 'starting text')
Syntax9:- (//*[@attribute='value'])[last ()] or (*[@attribute='value']) [last ()-1]
Syntax10:- //*[@attribute='value']/div/div/input
Syntax11:- //*[@attribute='value']/parent::* or /*[@attribute='value']/parent::tag name
Syntax12:- //*[@attribute='value']/child::*[index] or /*[@attribute='value']/child::tagname [index]
Syntax13:- //*[@attribute='value']/ preceding-sibling::*[index] or /*[@attribute='value']/ preceding-sibling:: tagname [index]
Syntax14:- //*[@attribute='value']/ following -sibling::*[index] or /*[@attribute='value']/ following-sibling:: tagname [index]
Syntax15:- //*[@attribute='value']/ ancestor::*[index] or /*[@attribute='value']/ancestor:: tagname [index]
Syntax16:- //*[@attribute='value']/ descendant::*[index] or /*[@attribute='value']/ descendant:: tagname [index]
Syntax17:- //*[@attribute='value']/ preceding::*[index] or /*[@attribute='value']/ preceding:: tagname [index]
Syntax18:- //*[@attribute='value']/ following::*[index] or /*[@attribute='value']/ following:: tagname [index]

Note:-

From the Relative xPath syntax's ,xPath concept is working as a language it have symbols, methods and axes

xPath Symbols	xPath methods	xPath Axes
[]	text()	parent
()	contains()	child
,	last()	ancestor
@	starts-with	descendant
=		following
*		following-sibling
//		preceding
/		preceding-sibling
Or		
-		
::		

D. “WebDriver” Class Methods:-

1. get():-

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

Example: - driver.get (“URL of web site”);

2. close():-

We can use this method to close launched web site.

Example: - driver.close ();

3. quit():-

We can use this method to close active browser window including related browser windows (or) tabs.

Example: - driver.quit ();

4. getTitle():-

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

Example: -

```
String x=driver.getTitle();
System.out.println(x);
```

5. getPageSource():-

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

Example: -

```
String x=driver.getPageSource();
System.out.println(x);
```

6. getCurrentUrl():-

we can use this method for to get URL of active browser window

Example: -

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

7. findElement():-

we can use this method to locate element in web page by giving name/id/xPath/className>tagName/ linkText/partialLinkText/cssSelector.

Example: -

```
Driver.findElement(By.XXXX("XXXX").operation());
```

8. click():-

we can use this method to operate (Or) click on locate element in webpage.

Example: -

```
Driver.findElement(By.XXXX("XXXX")).click();
```

9. sendKeys():-

we can use this method to send data to located element, apply functional keys on located element and also apply combination of functional keys on located element

Example: -

```
Driver.findElement(By.XXXX("XXXX")).sendKeys("xxxx");
Driver.findElement(By.XXXX("XXXX")).sendKeys("xxxx"),keys.Tab);
Here Keys is static class in selenium web driver ,so we can use directly.
Driver.findElement(By.XXXX("XXXX")).sendKeys("xxxx"),keys.chord('keys.control,"a"));
Keys.chord is a combination of functional keys in selenium web driver class
```

10. getAttribute():-

We can use this method to get value of specified attribute of located elements

Example: -

```
String x= driver.findElement(By.XXX("xxxx")).getAttribute("xxxx")
By.XXX is 8 ways to locate element
getAttribute("xxxx") from source code
```

11. getText():-

We can use this method to get the text of located element.

```
<a Class="Need-To Help"> Find My account </a>
```

Example: -

```
String x=driver.findElement(By.XXX("xxxx")).getText();
String x=driver.findElement(By.XXX("xxxx")).getAttribute("Value");
```

12. getCssValue():-

We can use this method to get style related attribute value of located Element

Example:- color, border-style, border-width,.....Etc

```
// Launch browser
System.out.println("webdriver.chrome.driver","Path of ChromeDriver");
WebDriver Driver=new ChromeDriver();
//Launch site in that browser
Driver.get("http://www.google.com");
Thread.sleep(50000);
String x=Driver.findElement(By.name("Email")).getAttribute("type");
String y=Driver.findElement(By.name("Email")).getText();
String z=Driver.findElement(By.name("Email")).getCssValue("color");
System.out.println(x);
System.out.println(y);
System.out.println(z);
Driver.close();
```

13. switchTo().frame():-

We can use this method to change focus of "WebDriver" class object from page level to frame level.

```
Driver.switchTo().frame("name of frame"/Index/Frame Object").
```

From the above syntax, we need to follow 3 ways switch to frame in page.

Way1:-

- ↓ Apply "firebug"/Inspect Element on an element in page in Firefox.
 - ↓ Scroll Up Source code and search for iframe tag <iframe name="xxx">.
 - ↓ Search for "name" of iframes in source code.
 - ↓ Write code like shown below.
- ```
Driver.switchTo().frame("name of frame");
```

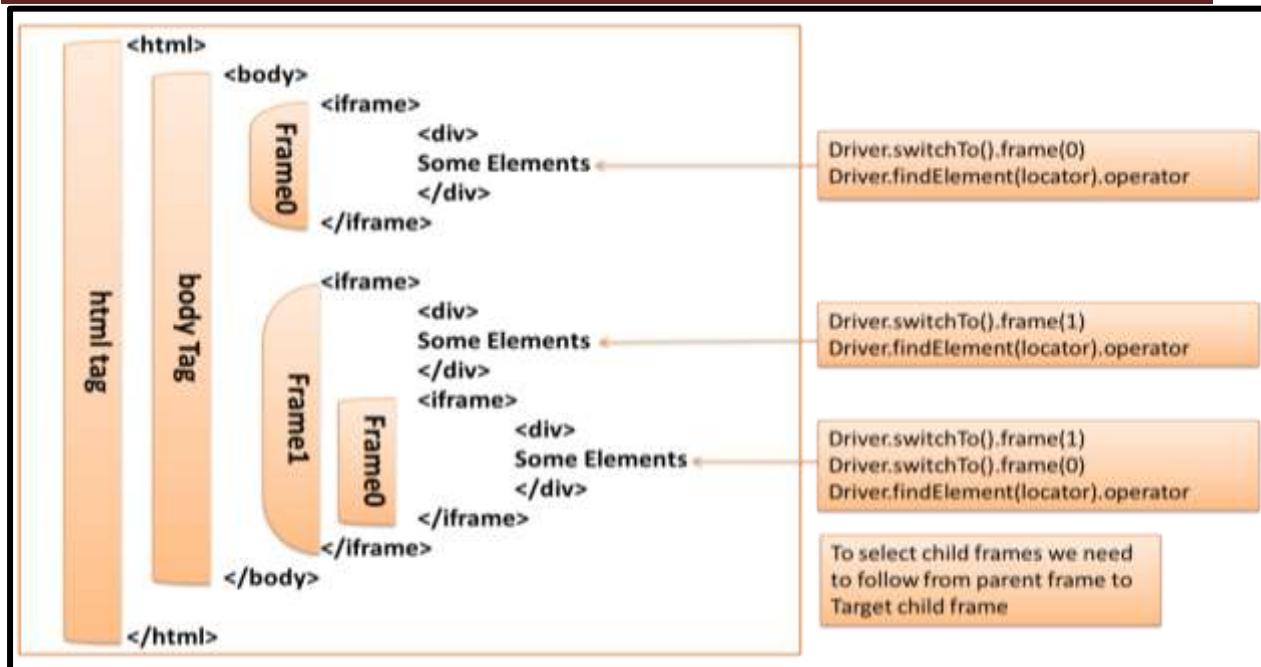
**Way2:-**

- ↓ Apply "firebug /inspect element" on an element in page in Firefox
- ↓ Scroll Up for source code for iframe tag <iframe="xxx">
- ↓ Search for iframe tag <iframe>
- ↓ Identify index of <iframe> tag via manual counting

**Way3:-**

- ↓ Apply "firebug /inspect element "on an element in page in Firefox
- ↓ Scroll Up for source code for iframe tag <iframe="xxx">
- ↓ Search for iframe tag <iframe>
- ↓ Prepare locator tag

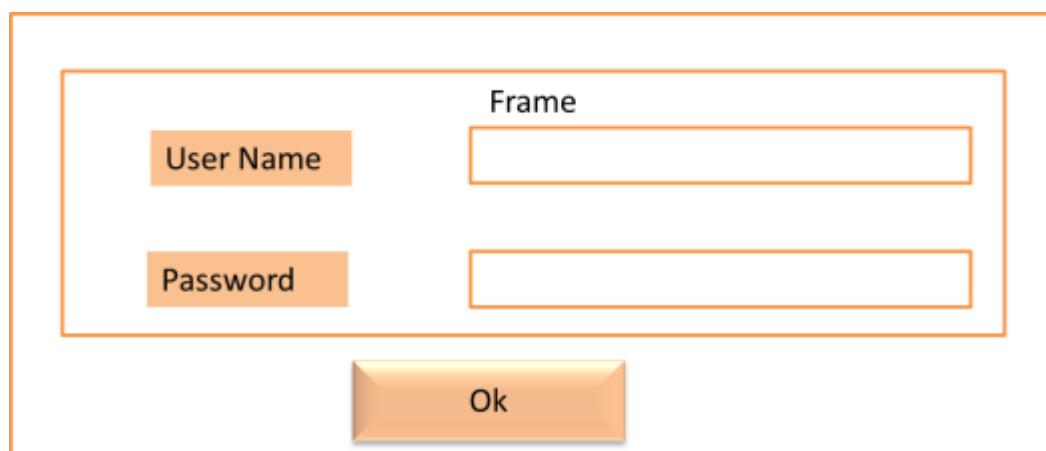
```
WebElement e= driver.findElement(locator);
Driver.switchTo().frame(e)
```



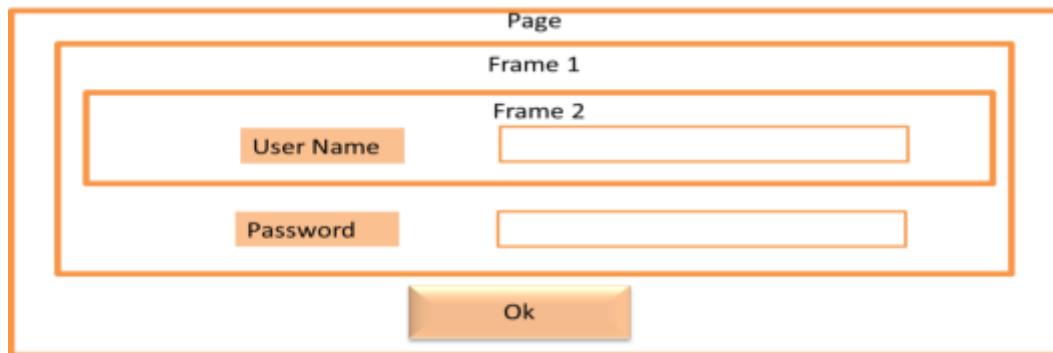
#### 14. switchTo( ).defaultContent( ):-

We can use this method to change focus of webdriver class object from current frame to web page.

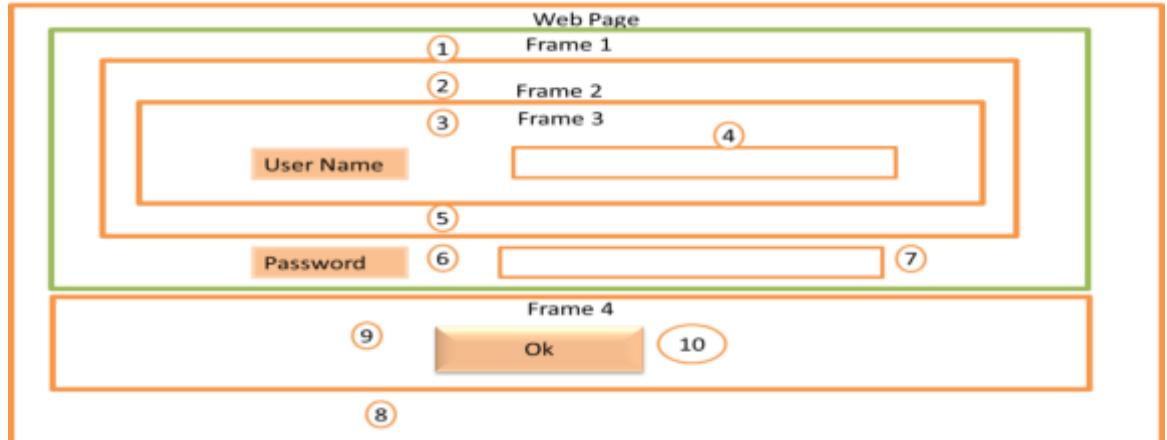
##### Example1:-



```
// Launch browser
System.out.println("webdriver.chrome.driver","Path of Chrome Driver");
WebDriver Driver=new ChromeDriver();
//Launch site in that browser
Driver.get("http://www.google.com");
Thread.sleep(5000);
Driver.switchTo().frame("Frame");
Driver.findElement(By.name("Username")).sendKeys("xxxx");
Driver.findElement(By.name("Password")).sendKeys("xxxx");
Driver.switchTo().DefaultContent()
Driver.findElement(By.name("Ok")).click();
```

**Example2:-**

```
System.out.println("webdriver.chrome.driver","Path of Chrome Driver");
WebDriver Driver=new ChromeDriver();
//Launch site in that browser
Driver.get("http://www.google.com");
Thread.sleep(50000);
Driver.switchTo().frame("Frame1");
Driver.switchTo().frame("Frame2");
Driver.findElement(By.name("Username")).sendKeys("xxxx");
Driver.switchTo().parentFrame(); // Switch to Frame 1
Driver.findElement(By.name("Password")).sendKeys("xxxx");
Driver.switchTo().DefaultContent();
Driver.findElement(By.name("Ok")).click();
```

**Example3:-**

```
System.out.println("webdriver.chrome.driver","Path of Chrome Driver");
WebDriver Driver=new ChromeDriver();
//Launch site in that browser
Driver.get("http://www.google.com");
Thread.sleep(50000);
Driver.switchTo().frame("Frame1");
Driver.switchTo().frame("Frame2");
Driver.switchTo().frame("Frame3");
Driver.findElement(By.name("Username")).sendKeys("xxxx");
Driver.switchTo().parentFrame(); // Switch to Frame 2
Driver.switchTo().parentFrame(); // Switch to Frame 1
Driver.findElement(By.name("Password")).sendKeys("xxxx");
Driver.switchTo().DefaultContent()
Driver.switchTo().frame("Frame4");
Driver.findElement(By.name("Ok")).click();
Driver.switchTo().DefaultContent()
```

**15. switchTo().parentFrame():-**

We can use this method to change focus to "WebDriver" Class object from current frame to Parent of that frame

**Syntax:-**

```
Driver.switchTo().parentFrame();
```

## **16. switchTo( ).activeElement( ):-**

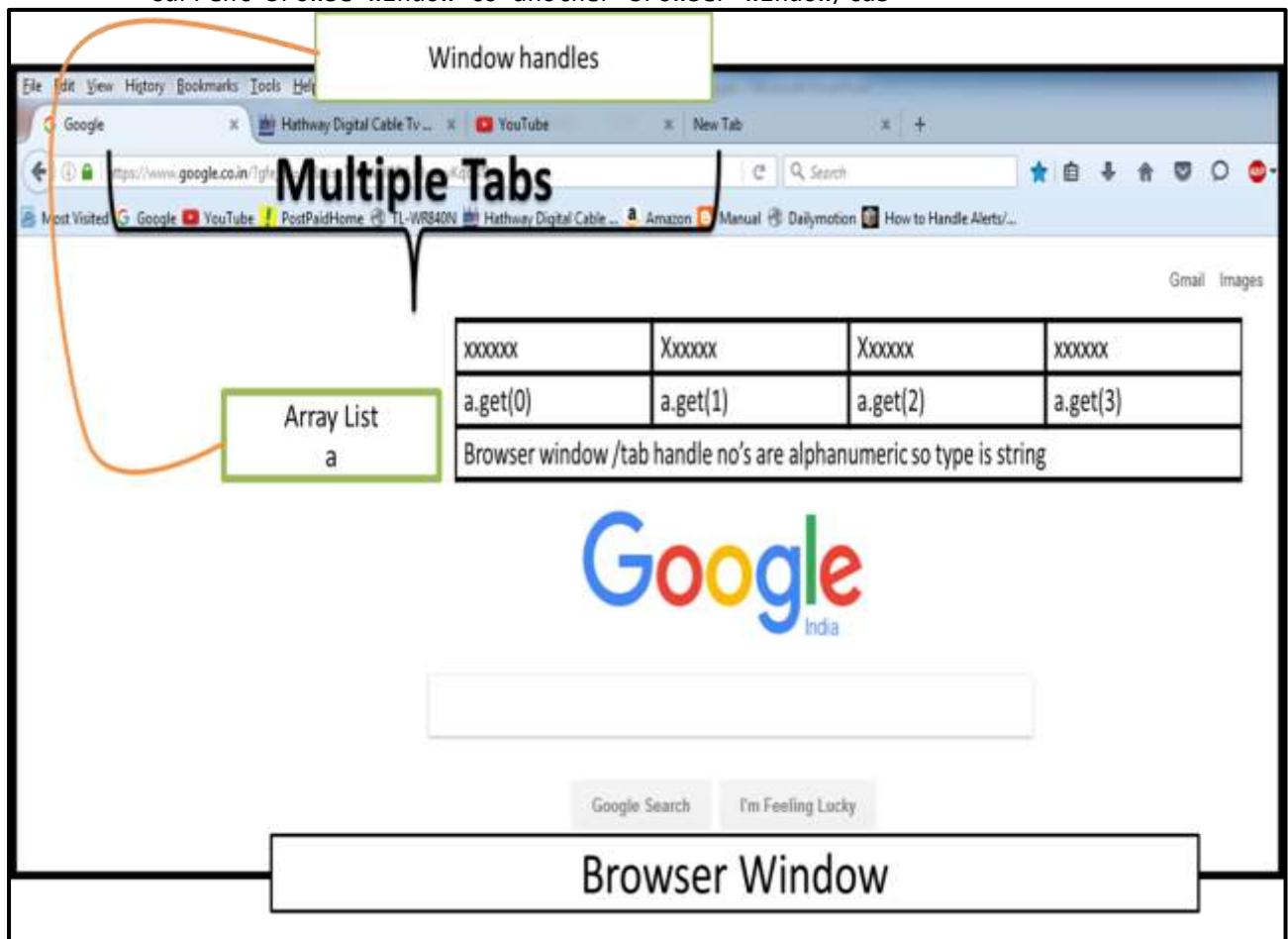
We can use this method to send focus to active element in a webpage for operations

### Syntax:-

```
Driver.switchTo().activeElement().operations();
 .sendKeys();
 .click();
 .Etc
```

## **17. switchTo( ).window( ):-**

We can use this method to change 'WebDriver' Class object focus to from current browser window to another browser window/tab



### Syntax:-

```
ArrayList<string> a=new ArrayList<string>(Driver.getWindowHandler());
Driver.switchTo().window(a.get(1));
Driver.switchTo().window(a.get(0));
ArrayList is class in JDK in Java.util.
ArrayList<string> is called javaGenerics.
```

## **18. switchTo( ).getWindowHandles( ):-**

We can use this method to get handle numbers of browser window/tabs into an "ArrayList"

### Syntax:-

```
ArrayList<string> a=new ArrayList<string>(Driver.getWindowHandler());
```

### Example:-

```
System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
WebDriver driver=new ChromeDriver();
driver.get("http://site24.way2sms.com/content/index.html?");
Thread.sleep(5000);
driver.findElement(By.xpath("//*[contains(@src,'android-
button.png')]]")).click();
Thread.sleep(5000);
ArrayList<String> a=new ArrayList<String>(driver.getWindowHandles());
```

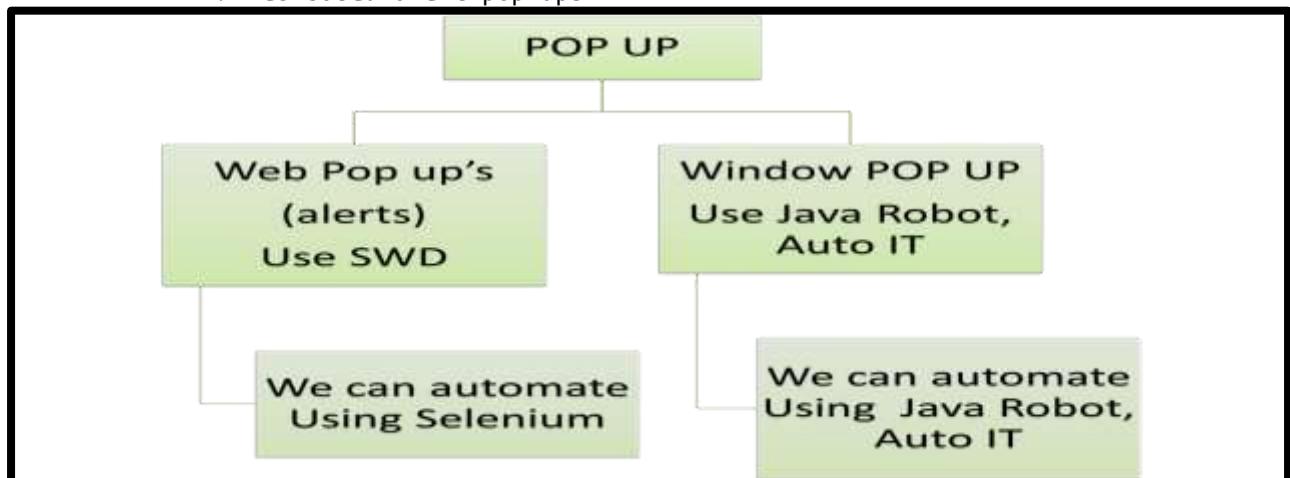
```
driver.switchTo().window(a.get(1));
System.out.println(a.size());
for(int i=0;i<a.size();i++)
{
System.out.println(a.get(i));
}
```

### 19. switchTo( ).alert( ):-

While automatic website we can get pop-up which are like web pop, window pop

Pop up are two types

1. Windows based alert pop ups
2. Web based alert pop ups



There are the four methods that we would be using along with the Alert

#### a) dismiss():-

The dismiss() method clicks on the “Cancel” button as soon as the pop up window appears.

#### b) accept():-

The accept() method clicks on the “Ok” button as soon as the pop up window appears.

#### c) getText():-

The getText() method returns the text displayed on the alert box.

#### d) sendKeys():-

The sendKeys() method send data to alert , apply functional keys on alert and also apply combination of functional keys on alert

While handling web pop ups (or) alerts by using selenium webdriver we can write like below code

#### Syntax1 :-

Dismiss() (Or)accept()



`Driver.switchTo().alert().dismiss() or Driver.switchTo().alert().accept();`

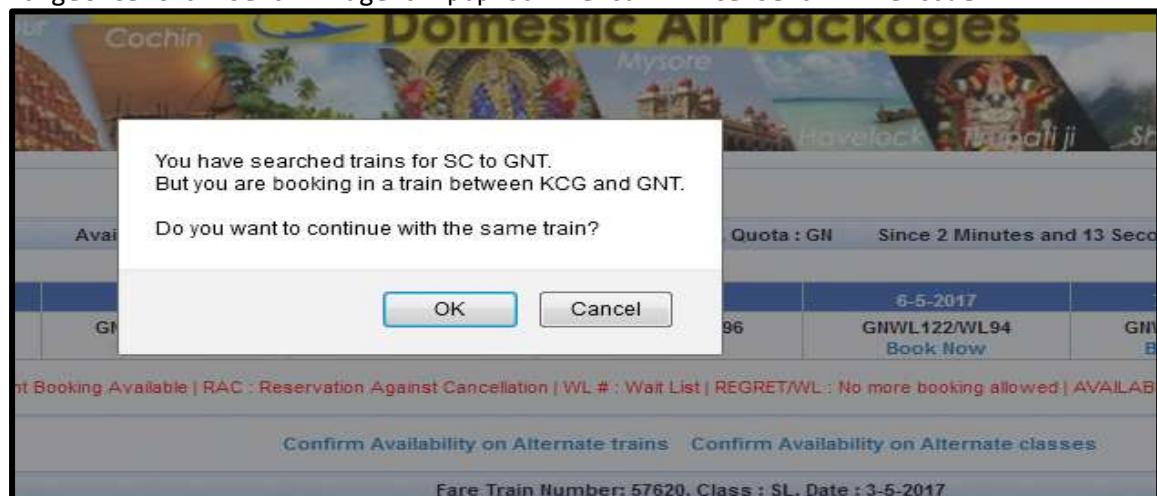
**Syntax2 :-**

Below image ok for acceptance and cancel is for dismiss we can write below like code

```
Driver.switchTo().alert().dismiss(); //ok
Driver.switchTo().alert().accept(); //cancel
```

**Syntax3 :-**

To get text of below image of pop box we can write below like code



```
String x=Driver.switchTo().alert().getText();
Driver.switchTo().dismiss();
```

**Syntax4 :-**

To fill data in message box we can write below code

```
[],
{
 \n";
 at) {
 ie +
 ed: " + input.checked)
 }
},
```

**LOG IN**

---

User Name

Password

Remember me

Login

Cancel

playing messages. Also usage of the plug-in is so g' used by plug-in to 'Images' folder in the root directory. That's all!

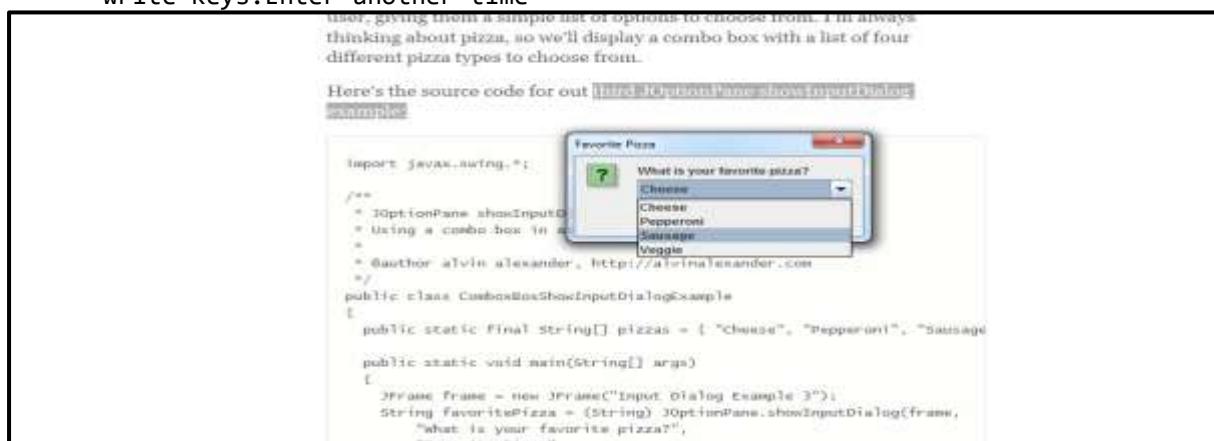
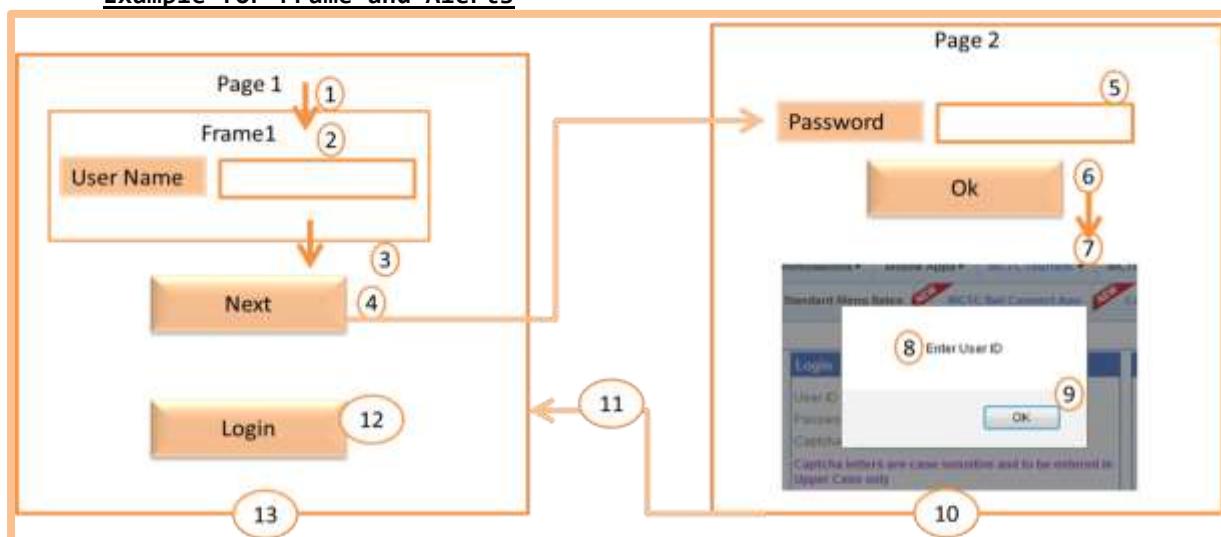
```
ay-1.4.1.min.js"></script>
js" type="text/javascript"></script>
rel="stylesheet" type="text/css">
```

```
Driver.switchTo().alert().sendKeys("Username data",keys.Tab,"Password Data",Keys.Enter);
```

**Syntax5 :-**

To select 3 item in dropdown in a messagebox we can like below like code  
`Driver.switchTo().alert.sendKeys(Keys.Down, Keys.Down, Keys.Down, Keys.Enter, Keys.Enter);`

Keys.Enter is repeated twice because to select third item in dropdown to select it we use first time Keys.Enter, later to Accept Message box so we write Keys.Enter another time

**Example for Frame and Alerts**

```

System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
WebDriver driver=new ChromeDriver ();
driver.get("http://site24.way2sms.com/content/index.html?");
Thread.sleep(5000);
Driver.switchTo().frame("Frame1");// switching to frame
Driver.findElement(By.name("Username")).sendKeys("xxxx");
Driver.switchTo().DefaultContent() /switching to page
Driver.findElement(By.name("Next")).click();
Thread.sleep(5000);
ArrayList<string> a=new ArrayList<string>(Driver.getWindowHandler());
Driver.switchTo().window(a.get(1)); //switching to 2 tab /window
Driver.findElement(By.name("Password")).sendKeys("xxxx");
Driver.findElement(By.name("Ok")).click();
String x=Driver.switchTo().alert().getText();
System.out.println(x);
Driver.switchTo().dismiss();
Driver.close //closing 2 tab/window
Driver.switchTo().window(a.get(0)); //switching to 1 tab /window
Thread.sleep(5000);
Driver.findElement(By.name("Login")).click();
Driver.close //closing 1 tab/window

```

**20. clear( ):-**

we can use this method to clear Existing Data in Located Element

**Syntax**

```
Driver.findElement(By.XXXX("xxxx")).Clear();
```

**21. manage( ).window( ).Maximize( ):**

we can use this method to maximize active browser window

**Syntax**

```
Driver.manage().window().maximize();
```

**22. manage( ).window( ).getSize( ):**

We can use this method to get width and height of active browser window

**Syntax:-**

```
int x= driver.manage().window().getSize.getWidth();
int y= driver.manage().window().getSize.getHeight();
System.out.println(x+ " " +y);
```

**23. manage( ).window( ).setSize( ):**

We can use this method to change size of (width and height) active browser window

**Syntax:-**

```
Dimension d=new Dimension(200,100) // Dimension (width, height)
Dimension is class in Selenium Web driver jars
Driver.manage().window().setSize(d);
```

**24. manage( ).window( ).getPosition( ):**

We can use this method to get the position of active browser window on desktop.

**Example:**

```
int x= driver.manage().window().getPosition.getx();
int y= driver.manage().window().getPosition.gety();
System.out.println(x+ " " +y);
```

**25. manage( ).window( ).setPosition( ):**

We can use this method to change Position of active browser window

**Syntax:-**

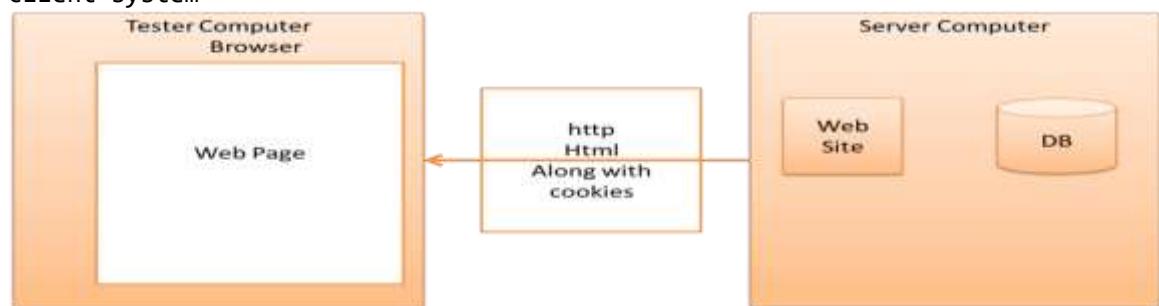
```
Point p=new Point(200,100) // Pixels (x, y)
Point is class in Selenium Web driver jars
Driver.manage().window().setPosition(p);
```

**26. manage( ).window( ).deleteAllCookies( ):**

Cookie is a server side code which is running in client system to share client details to server system.

In general Cookies are loading from server to client along with page source code in html.

We can this to ‘deleteAllCookies()’ method to delete all existing in client system

**Syntax**

```
Driver.manage().deleteAllCookies();
```

**27. manage( ).window( ).getCookies( ).size( ):**

We can use this method to count of existing cookies in client system

**Syntax:-**

```
int x=driver.manage().getCookies().size();
System.out.println(x);
```

**Example**

```
System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
WebDriver driver=new ChromeDriver();
//Delete All Cookies
driver.manage().deleteAllCookies();
// Launching site
driver.get("http://site24.way2sms.com/content/index.html?");
if(driver.manage().getCookies().size()!=0)
{
 System.out.println("Cookies Are Loaded");
}
else
{
 System.out.println("Cookies Are Not Loaded");
}
driver.close();
```

**28. manage( ).window( ).getCookies( ).size( ):**

We can use this method to define wait state in execution as implicit.  
implicit wait means wait if required.

**Syntax:-**

```
Driver.manage().timeouts.implicitlyWait(timeValue, TimeUnit.XXXX);
```

**Example:**

```
Driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

**29. navigate().To(), navigate().back(), navigate().forward(), navigate().refresh():-**

we can use this method to automate browser related operations like to, back, forward, refresh... Etc

```
// launch browser
System.setProperty("webdriver.chrome.driver",d:\\chromedriver.exe");
WebDriver Driver=new ChromeDriver();
// Launch site
Driver.get("http://www.google.in/");
Thread.sleep(2000);
Driver.navigate().to("http://www.gmail.com");
Thread.sleep(2000);
Driver.navigate().back(); // Back to google.com
Thread.sleep(2000);
Driver.navigate().forward(); // Forward to gmail.com
Thread.sleep(2000);
Driver.navigate().refresh(); // Refreshing Gmail Page
Thread.sleep(2000);
Driver.close();
```

**Note:-**

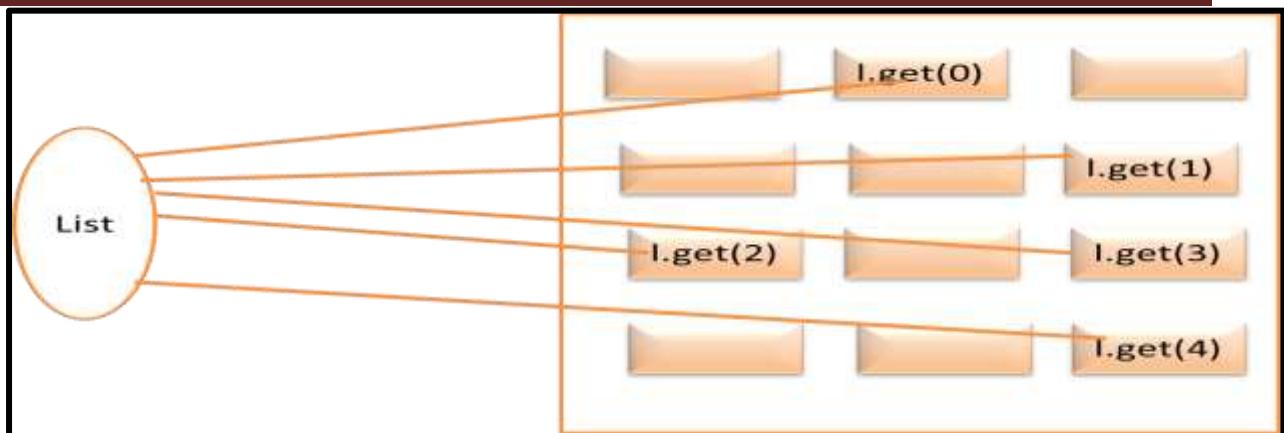
get() is a method is used to create a session by launching a website in browser window where as

navigate().to() is a method to use redirect other site in a same session.

**30. findElements( ):-**

We can use this method to collect multiple similar elements in a webpage. FindElements is also called as Bucket Testing or raids Testing.

```
List<WebElement> l=driver.findElements(locator);
⇒ collection all matching elements with given locator of specified
attribute
```



From the above syntax we can use different locators to locate multiple similar elements .But most of Test Automates can use By.tagName() as a locator to locate multiple similar elements.

In general, html language can provides different tag related to various types of elements.

| Tag Name | Type of element       |                |
|----------|-----------------------|----------------|
| <a>      | Link                  |                |
| <b>      | Bold                  |                |
| <i>      | Italic                |                |
| <Table>  | Table                 |                |
| <Tr>     | Row in table          |                |
| <td>     | Column in table       |                |
| <Select> | Dropdown              |                |
| <img>    | Image                 |                |
| <Span>   | Plain text            |                |
| <Input>  | Text                  | search         |
|          | Password              | Button         |
|          | Submit                | url            |
|          | Reset                 | Month          |
|          | Date                  | Radio          |
|          | Time                  | Range          |
|          | Week                  | Tel            |
|          | Number                | Image          |
|          | Checkbox              | File           |
|          | Hidden                | Datetime-local |
|          | Email                 | Color          |
| <div>    | Ajax Control          |                |
| <iframe> | Frames in a page      |                |
| <u>      | Underline             |                |
| <p>      | Paragraph             |                |
| <h>      | Heading               |                |
| <Form>   | Forms                 |                |
| <br>     | Break in text in page |                |
| etc.     |                       |                |

**Example1:-**

Count Number of link is active browser window page

**Solution:-**

```
package stevejobs;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class Test3
{
 public static void main(String[] args)
 {
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.get("http://www.amazon.in/");
 List<WebElement> l=Driver.findElements(By.tagName("a"));
 int count=l.size(); // no of links in active browsers
 System.out.println(count);
 }
}
Output is 356 Links
```

**Example2:-**

Click on 7<sup>th</sup> link in active browser window page

**Solution:-**

```
package stevejobs;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class Test4
{
 public static void main(String[] args)
 {
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.get("http://www.google.in/");
 List<WebElement> l=Driver.findElements(By.tagName("a"));
 int count=l.size(); // no of links in active browsers
 System.out.println(count);
 l.get(6).click(); //click on 7th link
 }
}
```

**Example3:-**

click on 2<sup>nd</sup> link in a 4<sup>th</sup> column, in a 5<sup>th</sup> row in 2<sup>nd</sup> table in active browser window page

**Solution**

```
List<WebElement> tl=driver.findElements(By.tagName("table"));
List<WebElement> rl=tl.get(1).findElements(By.tagName("tr")); // 2nd table
List<WebElement> cl=rl.get(4).findElements(By.tagName("td")); //5th row
List<WebElement> ll=cl.get(3).findElements(By.tagName("a")); // 4th column
ll.get(1).click();
```

**Example4:-**

Count no of radio buttons in active browser window

**Solution**

```
List<WebElement> l=driver.findElements(By.tagName("input"));
int c=0;
for(int i=0;i<l.getSize();i++)
{
String x=l.getAttribute("type");
If(x.equals("radio"))
{
 C=c+1;
}
}
System.out.println(c);
OR
List<WebElement> l=driver.findElements(By.xpath("//input[@type='radio']));
System.out.println(l.size());
```

**Example5:-**

Fill 3<sup>rd</sup> text box with "India" in 4<sup>th</sup> column in 2<sup>nd</sup> row in 3<sup>rd</sup> table in webpage

**Solution**

```
List<WebElement> tl=driver.findElements(By.tagName("table"));
List<WebElement> rl=tl.get(2).findElements(By.tagName("tr")); //3rd table
List<WebElement> cl=rl.get(1).findElements(By.tagName("td")); //2th row
List<WebElement>
textl=cl.get(3).findElements(By.xpath("//input[@type='text']")); // 4th column
Textl.get(2).sendKeys("India");
```

**Example6:-**

Select "India" in 3<sup>rd</sup> dropdown in 2<sup>nd</sup> column in 1<sup>st</sup> row in 4<sup>th</sup> table

**Solution**

```
List<WebElement> tl=driver.findElements(By.tagName("table"));
List<WebElement> rl=tl.get(3).findElements(By.tagName("tr")); //3rd table
List<WebElement> cl=rl.get(0).findElements(By.tagName("td")); //2th row
List<WebElement> ddl=cl.get(2).findElements(By.tagName("Select")); // 4th column
Select s=new Select(ddl.get(2));
s.selectByVisibleText("India");
```

**Note:-**

1. In html, some tags are useful to develop one type of elements for each tag

**For Example:-**

<a> is useful to devolve to only

2. In html, some tags are useful to develop multiple types of elements for one tag

**For Example :-**

<input> tag is useful to develop text, radio, checkbox, submit and password etc.

3. Sometime developers can develop one type of elements for multiple tags

**For Example :-**

images are included in web pages using <img>, <input> and <div> tag

### **31. getScreenshotAs( ):-**

We can use this method to get screenshot of current active browser page. Here we need .to save screenshot of web as “png” image “Portable Network Graphics”

#### Syntax:-

```
File f1=((TakesScreenshot)Driver).getScreenshotAs(OutputType.FILE);
File f2=new File("Path of file in hdd");
FileUtils.copyFile(f1, f2);
```

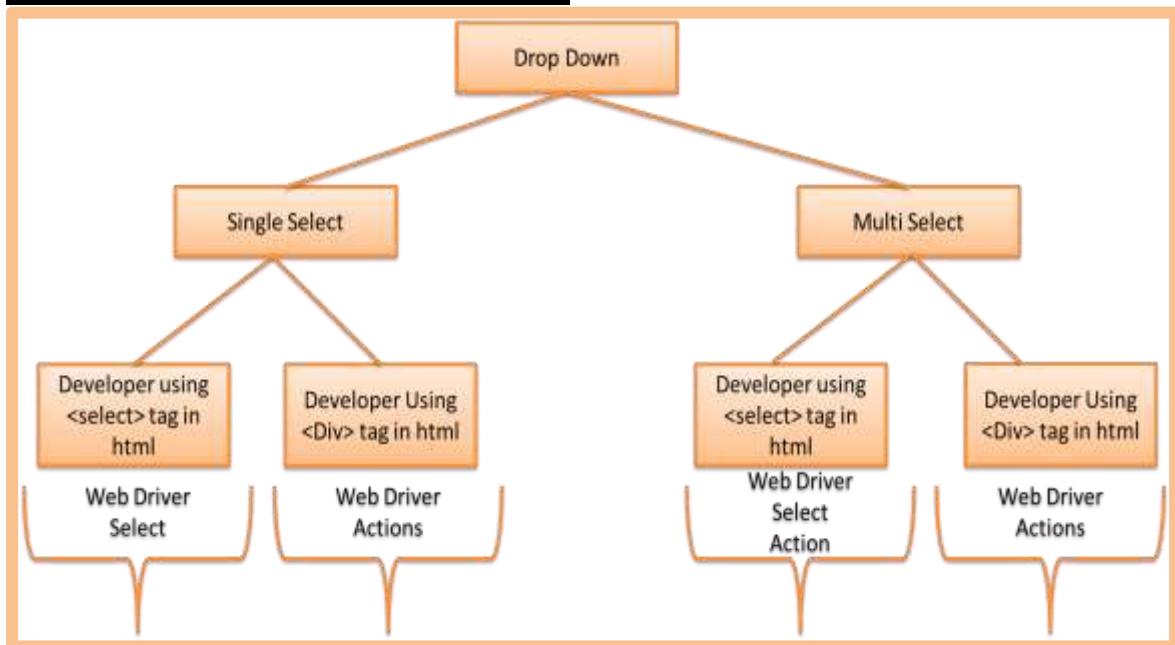
#### Example Program:-

```
package stevejobs;
import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class GmailImage
{
 public static void main(String[] args) throws Exception
 {
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.get("http://www.google.com");
 Thread.sleep(1000);
 Driver.findElement(By.name("q")).sendKeys("Steve jobs",Keys.ENTER);
 Thread.sleep(5000);
 File f1=((TakesScreenshot)Driver).getScreenshotAs(OutputType.FILE);
 File f2=new File("E:\\pic.png");
 FileUtils.copyFile(f1, f2);
 }
}
```

### “WebDriver” Class Methods

|                        |                                           |
|------------------------|-------------------------------------------|
| get( )                 | manage( ).window( ).Maximize( )           |
| close( )               | manage( ).window( ).setSize( )            |
| quit( )                | manage( ).window( ).getSize( )            |
| getTitle()             | manage( ).window( ).getPosition( )        |
| getPageSource          | manage( ).window( ).setPosition( )        |
| getCurrentUrl( )       | manage( ).window( ).deleteAllCookies( )   |
| findElement( )         | manage( ).window( ).getCookies( ).size( ) |
| click( )               | manage( ).window( ).getCookies( ).size( ) |
| sendKeys( )            | switchTo( ).frame( )                      |
| navigate( ).To( )      | switchTo( ).defaultContent( )             |
| navigate( ).back       | switchTo( ).parentFrame( )                |
| navigate( ).forward( ) | switchTo( ).activeElement( )              |
| navigate( ).refresh( ) | switchTo( ).window( )                     |
| findElements( )        | switchTo( ).getWindowHandles( )           |
| getAttribute()         | switchTo( ).alert( )                      |
| getText( )             | clear( )                                  |
| getCssValue( )         |                                           |

## E.“Select” Class Methods:-



From the above diagram Select Class is useful to Automate single select drop down which is developed using <select> html

### 1. selectByVisibleText( ):-

We can use this method to select an item in dropdown by giving by visible text of that item

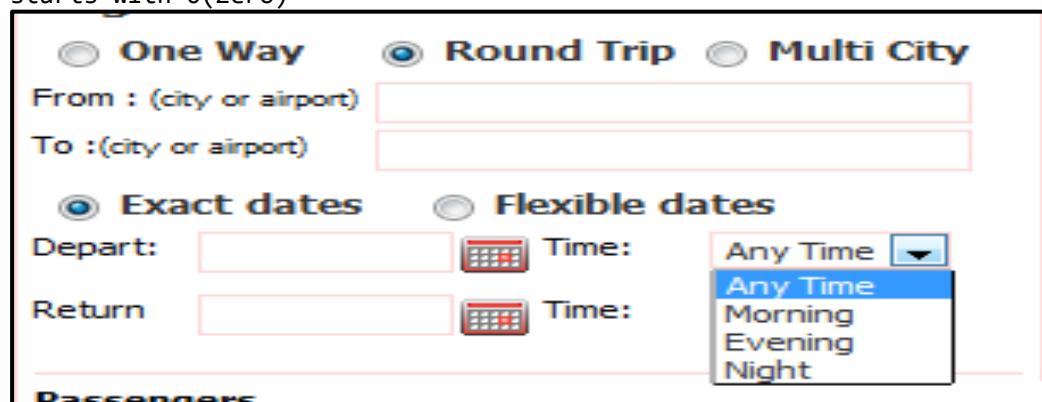
```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Select s=new Select(e);
s.selectByVisibleText("night");
```

### 2. selectByIndex( ):-

We can use this method to select an item in dropdown by giving by that item index.

```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Select s=new Select(e);
s.selectByIndex(3);
```

To select night we need to give index as 3 because in java index starts with 0(Zero)



### 3. selectByValue( ):-

We can use this method to select an item in dropdown by giving by that item value in source code.

```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Select s=new Select(e);
s.selectByValue("xxxx");
```

### 4. deselectAll( ):-

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

```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Select s=new Select(e);
s.deselectAll();
```

## 5. deselectByVisibleText( ):-

We can use this method to deselect a selected item in dropdown by giving by visible text of that item

```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Select s=new Select(e);
s.deselectByVisibleText("night");
```

## 6. deselectByIndex( ):-

We can use this method to deselect a selected item in dropdown by giving by that item index.

```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Select s=new Select(e);
s.deselectByIndex(1);
```



## 7. deselectByValue( ):-

We can use this method to deselect selected item in dropdown by giving by that item value in source code.

```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Select s=new Select(e);
s.deselectByValue("xxxx");
```

## 8. getoptions( ).size( ):-

We can use this method to get count of items in dropdown

Syntax:  
int x= s.getOptions().size();  
System.out.println(x);

## 9. isMultiple( ):-

We can use this method to check given dropdown is single select or multiple select

Syntax:-

```
if(s.isMultiple())
{
 System.out.println("Multiple Select");
}
else
{
 System.out.println("Single select");
}
```

## F. “Actions” Class Methods:-

This is class methods are useful to handle different situations in webpages while automation

### 1. Click on element:-

In general click() method in “WebDriver” class is useful to automate click operation on element .But when element was developed using html with css(Cascading style sheets), JavaScript, AJAX(Asynchronous JavaScript) and Angularjs like technology, we are not able to use click method in webdriver class. Here we can “click()” method of “Actions” class.

#### Syntax:

```
Actions a=new Actions(driver);
WebElement e=driver.findElement(By.xpath(locator));
a.click(e).build().perform();
Actions and WebElement are selenium webdriver class
Actions are indirect Automation
```

### 2. Double Click on element:-

We can use this method to perform double click on an element. We can use below code

#### Syntax:

```
Actions a=new Actions(driver);
WebElement e=driver.findElement(By.xpath(locator));
a.doubleClick(e).build().perform();
Actions and WebElement are selenium webdriver class
Actions are indirect Automation
```

### 3. Right Click on element:-

We can use this method to perform right click on an element .We can use below code

#### Syntax:

```
Actions a=new Actions(driver);
WebElement e=driver.findElement(By.xpath(locator));
a.contextClick(e).build().perform();
Actions and WebElement are selenium webdriver class
Actions are indirect Automation
While we Right Click we will get a menu it is called as context menu
```

### 4. Move Mouse Pointer to an element:-

We can use this method to move mouse pointer to an element in a web page .We can use below code

#### Syntax:

```
Actions a=new Actions(driver);
WebElement e=driver.findElement(By.xpath(locator));
a.moveToElement(e).build().perform();
Actions and WebElement are selenium webdriver class
Actions are indirect Automation
(Or)
WebElement e=driver.findElement(By.xpath(locator));
Int x=e.getLocation().getx();
Int y=e.getLocation().gety();
Actions a=new Actions(driver);
a.moveByOffset(x,y).bulid().perform();
```

### 5. Fill an element:-

We can use this method to fill an element with data in a web page .We can use below code

#### Syntax:

```
Actions a=new Actions(driver);
WebElement e=driver.findElement(By.xpath(locator));
a.click(e).sendKeys("data").bulid().perform();
Actions and WebElement are selenium webdriver class
Actions are indirect Automation
```

## **6.Handle cache Element (or) Auto Complete**

### **Element :-**

Some wen pages having text box but they behaving like a input box .This type of elements are called as cache elements (or) auto complete elements. To handle those elements in auto machine, We can use below code.

#### **Syntax:**

```
package stevejobs;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class Handlecahce
{
 public static void main(String[] args) throws Exception
 {
 System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.get("http://www.google.com");
 Thread.sleep(1000);
 WebElement e=Driver.findElement(By.name("q"));
 Actions a=new Actions(Driver);
 a.click(e).build().perform();
 Thread.sleep(5000);
 a.sendKeys("Kalam");
 Thread.sleep(5000);
 a.sendKeys(Keys.DOWN);
 Thread.sleep(5000);
 a.sendKeys(Keys.DOWN);
 Thread.sleep(5000);
 a.sendKeys(Keys.ENTER);
 }
}
```

## **7.Automate Drag and Drop element:-**

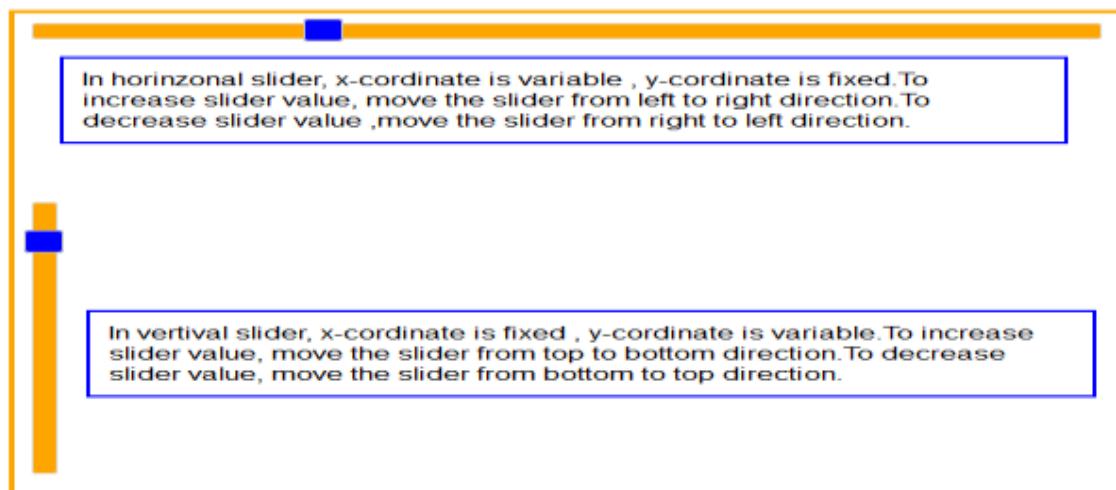
We can use this method to automate drag and drop element ,we can use below code.

#### **Example:**

```
package stevejobs;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class Handlecahce
{
 public static void main(String[] args) throws InterruptedException
 {
 System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.get("http://jqueryui.com/droppable");
 Thread.sleep(1000);
 Driver.switchTo().frame(0);
 WebElement e1=Driver.findElement(By.id("Dragable"));
 WebElement e2=Driver.findElement(By.id("Droppable"));
 Actions a=new Actions(Driver);
 a.dragAndDrop(e1,e2).build().perform();
 driver.switchTo().defaultContent();
 }
}
```

## 8. Handle Sliders:-

In general web pages are having two types of sliders such as Horizontal Sliders and Vertical Sliders



### Example:-

```
package stevejobs;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class Sliders
{
 public static void main(String[] args) throws
InterruptedException
 {
 System.setProperty("webdriver.chrome.driver",
"d:\\chromedriver.exe");
 WebDriver driver=new ChromeDriver();
 driver.get("http://www.jqueryui.com/slider");
 Thread.sleep(5000);
 driver.switchTo().frame(0) ;
 WebElement e=driver.findElement(By.id("slider"));
 int x=e.getLocation().getX();
 int y=e.getLocation().getY();
 Actions a=new Actions(driver);
 a.dragAndDropBy(e, x+50,y).build().perform();
 Thread.sleep(5000);
 a.dragAndDropBy(e, x-50,y).build().perform();
 Thread.sleep(5000); }
}
```

## 9. Automate drop downs:-

When dropdown are using single select dropdown's developer using <div> tags. Multi select dropdowns using <select> tag or using <div> tag. we can use "Action()" class method like shown below.

### Example1:-

Single select dropdown developer using <div> tag

#### Execution Code

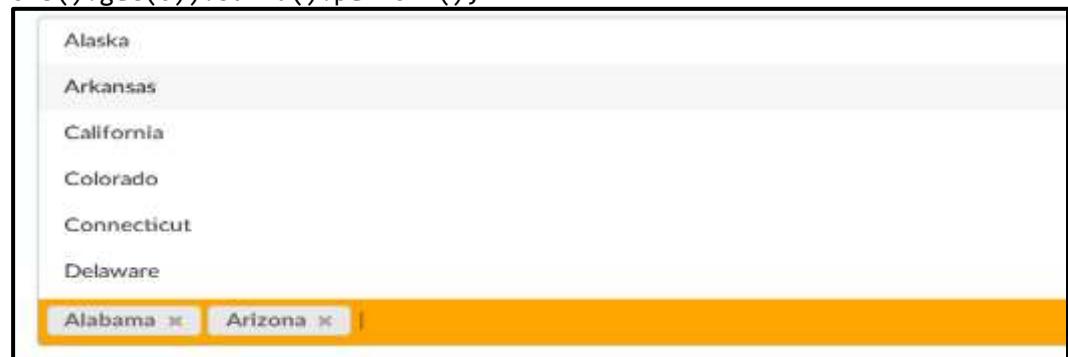
```
WebElement e=Driver.findElement(By.xpath("//*[@id='BirthMonth']"));
Actions a=new Actions(Driver);
a.click(e).build().perform();
Thread.sleep(1000);
a.sendKeys("Feb",Keys.ENTER).build().perform();
Thread.sleep(1000);
```

### Example2:-

Multi select dropdown developer using <select> tag

#### Execution Code

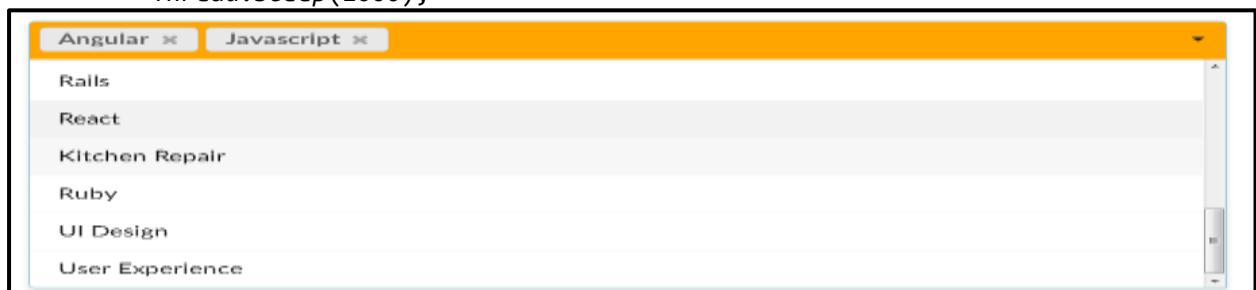
```
WebElement e=Driver.findElement(By.xpath("locator"));
Select s=new Select(e);
Actions a=new Actions(driver);
a.keyDown(Keys.CONTROL).click(s.getOptions().get(3)).click(s.getOptions().get(6)).build().perform();
```



**Example3:-**

**Multi select dropdown developer using <select> tag Execution Code**

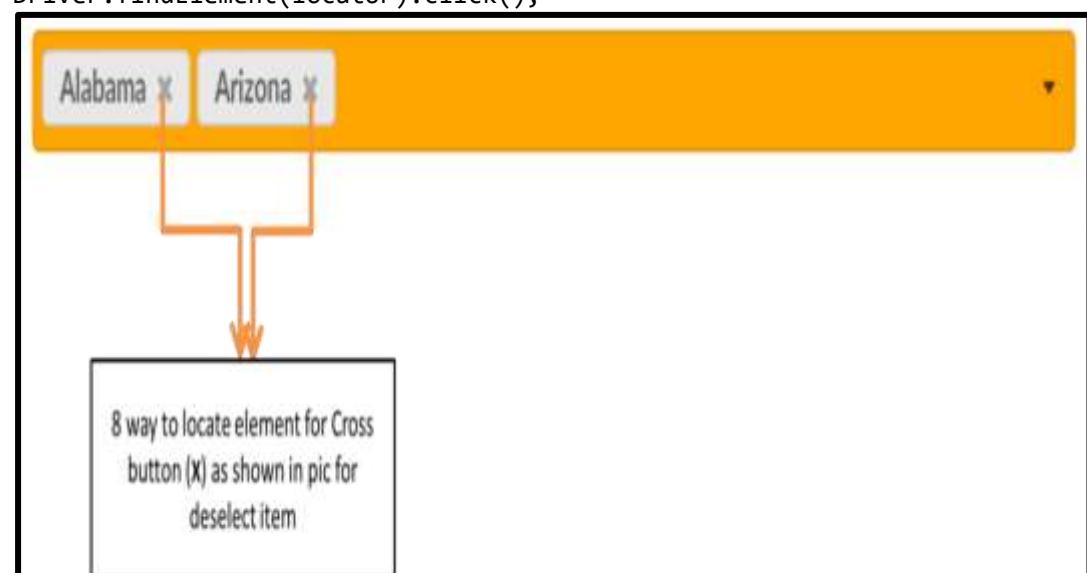
```
WebElement e=Driver.findElement(By.xpath("locator"));
Actions a=new Actions(driver);
a.click(e).build().perform();
a.sendKeys("J",Keys.ENTER).build().perform();
Thread.sleep(1000);
a.sendKeys("A",Keys.ENTER).build().perform();
Thread.sleep(1000);
```



**Note1:-** To deselect selected item in <select> tag dropdown ,we can use deselectAll() deselect ByVisualText(), deselectByIndex(), deselectByValue() in select class

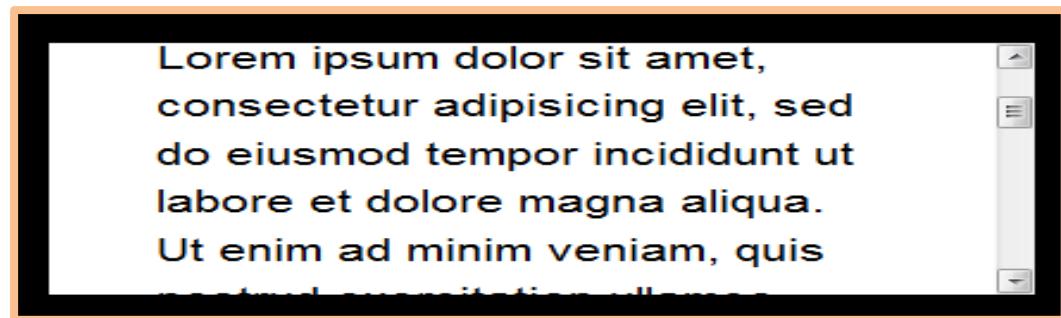
**Note2:-** To deselect selected item in <div> tag ,we can dropdown, We can use below like code

```
Driver.findElement(locator).click();
```



## 10. Element Scrolling:-

In general web page are having scroll when that page is more than desktop. But in rare cases elements are also having scroll effect .To automate this situation we use below like code.



### Syntax:-

```
WebElement e=Driver.findElement(locator);
Actions a=new Actions(Driver);
a.click(e).build().perform();
Thread.sleep(1000);
a.sendKeys(Keys.DOWN,Keys.DOWN).build().perform();
Keys.DOWN,Keys.DOWN Till scroll reaches bottom
```

## “Select” Class Methods

|                               |                                 |
|-------------------------------|---------------------------------|
| <u>selectByVisibleText( )</u> | <u>deselectAll( )</u>           |
| <u>selectByIndex( )</u>       | <u>deselectByVisibleText( )</u> |
| <u>selectByValue( )</u>       | <u>deselectByIndex( )</u>       |
| <u>getoptions( ).size( )</u>  | <u>deselectByValue( )</u>       |
| <u>isMultiple( )</u>          |                                 |

## “Action” Class Methods

|                               |                                 |
|-------------------------------|---------------------------------|
| <u>selectByVisibleText( )</u> | <u>deselectAll( )</u>           |
| <u>selectByIndex( )</u>       | <u>deselectByVisibleText( )</u> |
| <u>selectByValue( )</u>       | <u>deselectByIndex( )</u>       |
| <u>getoptions( ).size( )</u>  | <u>deselectByValue( )</u>       |
| <u>isMultiple( )</u>          |                                 |

## G. “JavascriptExecutor” Class Methods:-

This is useful to automate some operations related to webpages.

### 1. Highlight Element:-

```
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("document.getElementById(\"xxxx\").style.border='2px
dotted blue';");
⇒ Bold Font is JavaScript, JavaScript is end with semicolon
⇒ Locating and operation by JavaScript Only
Or
WebElement e1=Driver.findElement(Locator);
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("arguments[0].style.border='5px dotted green';",e1);
⇒ Bold Font is JavaScript, JavaScript is end with semicolon
⇒ Locating by SWD and operation by JavaScript Only
• dotted - Defines a dotted border
• dashed - Defines a dashed border
• solid - Defines a solid border
• double - Defines a double border
• groove - Defines a 3D grooved border. The effect depends on the
border-color value
• ridge - Defines a 3D ridged border. The effect depends on the
border-color value
• inset - Defines a 3D inset border. The effect depends on the
border-color value
• outset - Defines a 3D outset border. The effect depends on the
border-color value
• none - Defines no border
• hidden - Defines a hidden border
```

### 2. Click an Element:-

```
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("document.getElementById(\"xxxx\").click();");
⇒ Bold Font is JavaScript, JavaScript is end with semicolon
⇒ Locating and operation by JavaScript Only
Or
WebElement e1=Driver.findElement(Locator);
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("arguments[0].click();",e1);
⇒ Bold Font is JavaScript, JavaScript is end with semicolon
⇒ Locating by SWD and operation by JavaScript Only
In general JavaScript can locate elements in webpages in 4 ways whereas
selenium can follow 8 ways
```

| JavaScript                                                                                                                                                                             | Selenium                                                                                                                                                                                                   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Document.getElementById()</code><br><code>Document.getElementsByClassName()</code><br><code>Document.getElementsByName ()</code><br><code>Document.getElementsByTagName()</code> | <code>By.name()</code><br><code>By.id()</code><br><code>By.cssSelector()</code><br><code>By.linkText()</code><br><code>By.partialLinkText()</code><br><code>By.tagName()</code><br><code>By.xpath()</code> |
| <b>So JavaScript is object based</b>                                                                                                                                                   | So java is object oriented                                                                                                                                                                                 |
| <b>Object based having objects</b>                                                                                                                                                     | Objected oriented having Class<br>Class are Two Types They are<br>Instance Class (create object and use)<br>Static class (use directly)                                                                    |

⇒ `Document.getElementById()`  
⇒ `Document.getElementsByClassName()`  
⇒ `Document.getElementsByName ()`  
⇒ `Document.getElementsByTagName()`

**Note:** when we used JavaScript methods like

- ⇒ Document.getElementsByClassName()
- ⇒ Document.getElementsByName()
- ⇒ Document.getElementsByTagName()

If We need to located one element:-  
 js.executeScript("document.getElementById('xxxx')[0].click();");  
 [0] is first Matched Element

### 3. Disable an Element:-

```
//Disable Element
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("document.getElementsByName('locator')[0].setAttribute(
'disable','');");
'→ Represents null, without space
// Enable Element
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("document.getElementsByName('locator')[0].removeAttribute(
'disable');");
Bold Font is JavaScript, JavaScript is end with semicolon
Locating and operation by JavaScript Only
Or
WebElement e1=Driver.findElement(Locator);
JavascriptExecutor js=(JavascriptExecutor)Driver;
//Disable Element
js.executeScript("arguments[0].setAttribute('disable','');",e1);
'→Represents null, in between 'no space
//Enable Element
js.executeScript("arguments[0].removeAttribute('disable');",e1);
Bold Font is JavaScript, JavaScript is end with semicolon
Locating by SWD and operation by JavaScript Only.
```

### 4. Page Scrolling

```
//Scroll to Web Page Top-Bottom
JavascriptExecutor js=(JavascriptExecutor) driver;
js.executeScript("window.scrollTo(0,document.body.scrollHeight);");
//Scroll to Web Page Bottom-Up
js.executeScript("window.scrollTo(document.body.scrollHeight,0);");
// Scroll To Specified Element
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("document.getElementById('xxxx').scrollIntoView();");
Bold Font is JavaScript, JavaScript is end with semicolon
Locating and operation by JavaScript Only
Or
WebElement e=driver.findElement(Locator);
JavascriptExecutor js=(JavascriptExecutor) driver;
js.executeScript("arguments[0].scrollIntoView();",e);
Bold Font is JavaScript, JavaScript is end with semicolon
Locating by SWD and operation by JavaScript Only
```

### 5. Create Alert

```
JavascriptExecutor js=(JavascriptExecutor) driver;
js.executeScript("alert('Program Successfully Completed');");
Thread.sleep(5000);
driver.switchTo().alert().dismiss();
⇒ Bold Font is JavaScript, JavaScript is end with semicolon
```

### 6. Fill Element

We can use "sendKeys()" method in "WebDriver" class or "sendKeys()" method in "Action" class to fill an element with data. In rare cases JavaScript code to fill an element

**Syntax:-**

```
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("document.getElementById('xxxx').value='xxxx';");
Bold Font is JavaScript, JavaScript is end with semicolon
```

⇒ Locating and operation by JavaScript Only  
 Or  
`WebElement e=driver.findElement(Locator);  
 JavascriptExecutor js=(JavascriptExecutor) driver;  
 js.executeScript("arguments[0].value='xxx';",e);`  
 ⇒ Bold Font is JavaScript, JavaScript is end with semicolon  
 ⇒ Locating by SWD and operation by JavaScript Only

## 7. JavaScript in SWD

Run JavaScript code in HTML Source of element. Sometimes, we are able to run developer given JavaScript code of element to automate instead of our own JavaScript code.

### Syntax:-

```
JavascriptExecutor js=(JavascriptExecutor)Driver;
js.executeScript("javascript:validate();");
⇒ Bold Font is JavaScript, JavaScript is end with semicolon
```

### Note1:-

JavaScript language is an object based language so it's having objects only. These objects are useful to call methods and to set data to properties.

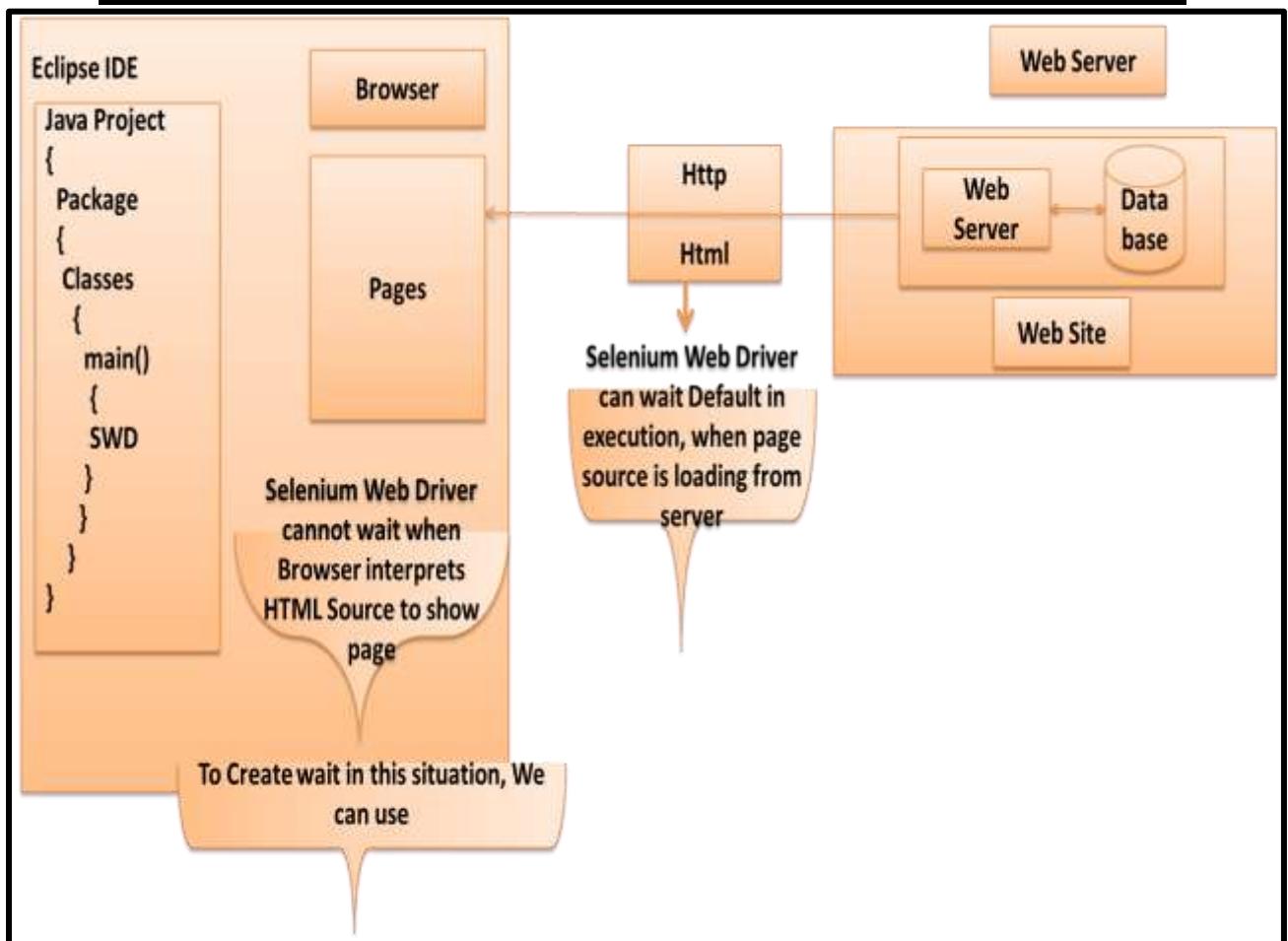
| JavaScript objects                                           | Methods                                                                                          | Properties                                                     |
|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| document<br>arguments[0]<br>window<br>alert<br>. . .<br>etc. | setAttribute()<br>removeAttribute()<br>click()<br>scrollTo()<br>scrollIntoView()<br>.. .<br>etc. | value<br>type<br>style.border<br>scrollHeight<br>. . .<br>etc. |

### Note2:-

Java is object oriented language so we are able to create class and objects and those objects are useful to call properties and methods of class.

JavaScript is object based language. So it consists of built in classes and objects. These inbuilt objects are useful to call properties and Methods.

## H. “WebDriverWait” or “FluentWait” Classes Methods:-



|                                 |        |                 |
|---------------------------------|--------|-----------------|
| ⇒ <code>Thread.sleep( )</code>  | Code   | (Fixed Wait)    |
| ⇒ <code>implicitWait( )</code>  | Method | (implicit Wait) |
| ⇒ <code>WebDriverWait( )</code> | Class  | (Explicit Wait) |
| ⇒ <code>FluentWait( )</code>    | Class  | (Fluent Wait)   |

### a) Thread.sleep( ) :-

It is a java code. Here Thread is static class and sleep() is method. This code is useful to create fixed delay in execution.

#### Syntax:

```
Thread.sleep(Time in milliseconds);
```

### b) ImplicitWait( ) :-

It is a method in “WebDriver” class in selenium. It is useful to create waiting state in execution if required, but this concept of waiting is working for `findElement()` and `findElements()`.

#### Syntax:-

```

```

```

```

```
driver.manage().timeouts().implicitlyWait(150, TimeUnit.SECONDS);
```

```

```

```

```

#### Example:-

```
driver.findElement(By.xpath("//input[@value='Send Free SMS']")).click();
driver.manage().timeouts().implicitlyWait(150, TimeUnit.SECONDS);
```

### c) ExplicitWait Concept(ConditionalWait)/ WebDriver Wait:-

We are able to use this concept to define wait state depends on a condition.

Syntax:-

```
WebDriverWait w=new WebDriverWait(Driver, 100);
w.until(ExpectedConditions.invisibilityOfElementLocated(By.className("raDiv")));
);
```

Example:-

```
package stevejobs;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class Webdriverwait
{
 public static void main(String[] args) throws InterruptedException
 {
 System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.manage().window().maximize();
 Thread.sleep(1000); //Fixed Wait
 Driver.get("http://demos.telerik.com/aspnet-
ajax/ajaxloadingpanel/functionality/explicit-show-hide/defaultcs.aspx");
 Driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
 Driver.findElement(By.linkText("11")).click();
 WebDriverWait w=new WebDriverWait(Driver, 100);
 w.until(ExpectedConditions.invisibilityOfElementLocated(By.className("raDiv")));
 }
}
```

In above code WebDriverWait & Expected conditions are class in Selenium Web Driver Jars.

TimeUnit is class in JDK

### d) Fluent Wait Concept (or) Conditional Wait :-

We can use this concept to define wait state depends on condition by checking as interval by interval

Example:-

```
import org.openqa.selenium.support.ui.FluentWait;
import org.openqa.selenium.support.ui.Wait;
import org.openqa.selenium.support.ui.ExpectedConditions;
import java.util.concurrent.TimeUnit;

Wait<WebDriver> w=new FluentWait<WebDriver>(Driver).pollingEvery(1,
TimeUnit.SECONDS).withTimeout(1, TimeUnit.SECONDS);
w.until(ExpectedConditions.invisibilityOfElementLocated(By.className
("raDiv")));
```

In above code Wait, FluentWait & ExpectedConditions are class in SWD; TimeUnit is a class in JDK

## I. Handling Ajax Calls and Angular Js Controls in Webpages:-

In general developers can use HTML along with CSS, JavaScript, AJAX (Asynchronous JavaScript) and Angular JS for webpages development.

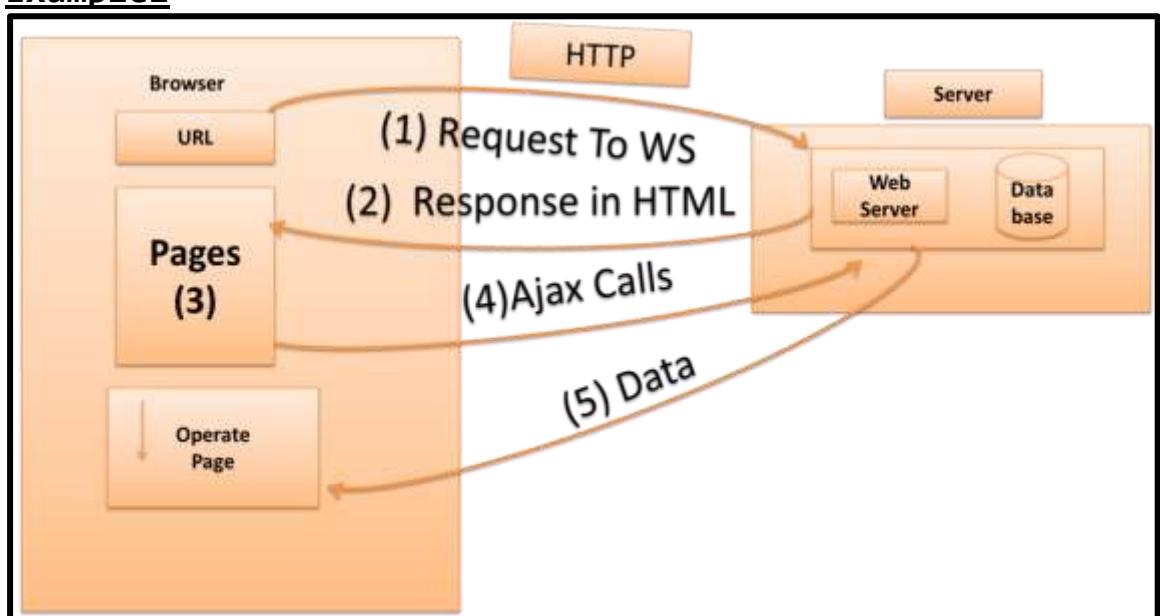
Here HTML is useful to design webpage with elements .To increase look and feel of elements, CSS is useful.

For browser side validations and to showing alerts, JavaScript is useful.

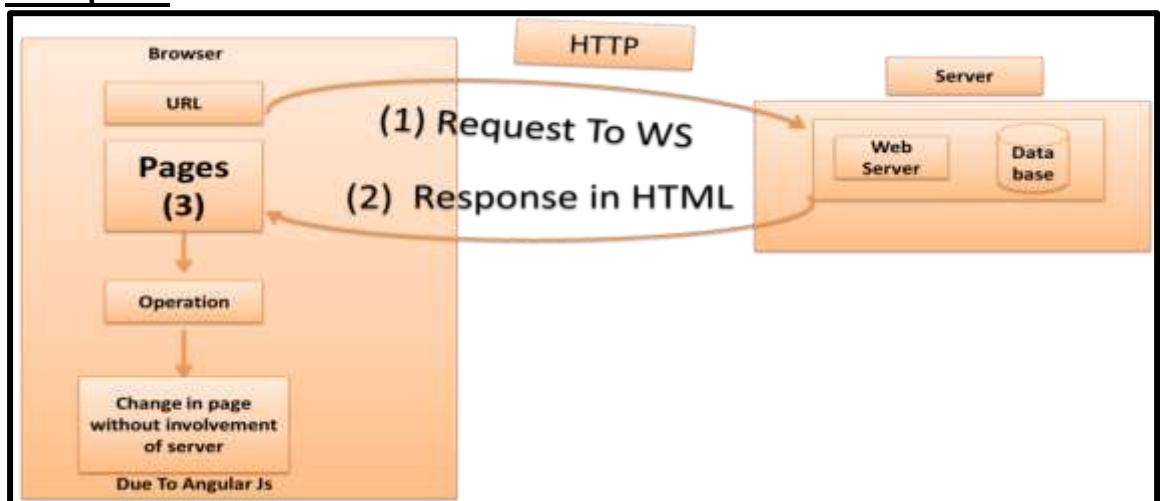
To request data instead of HTML source from server, Ajax calls are useful.

To get dynamic changes in pages without involvement of server, Angular JS is useful.

### a) Example1



### b) Example2



#### Note:-

To automate “Ajax” and “Angular JS” based webpages, we need to use selenium webdriver with proficient on synchronization for time mapping & on xPath to locate dynamic elements.

Some test automaters are going to use protractor with selenium web driver.

#### Note:-

When developers used “Angular JS” in webpages development, corresponding page HTML Source is having “ng” with attributes and values.

“ng” stands for next generation.

## **J. Handling Browser Notifications:-**

While automating website pages using selenium webdriver, browser can show some default notifications. To prevent these notifications in automation, we can use below solutions:

### **a) For Chrome Browser: -**

We need to use below code in tests scripts to stop notifications in chrome browser.

#### **Syntax:**

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
public class ChromeNotifications
{
 public static void main(String[] args) throws
InterruptedException
 {
 ChromeOptions co=new ChromeOptions();
 co.addArguments("--disable-notifications");
System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe")
;
 WebDriver Driver=new ChromeDriver(co);

 }
}
```

### **b) For Firefox Browser:-**

To stop notifications in Firefox browser during automation, we need to follow below Navigation Code:

- ↓ Click
- ↓ Enterfirefox.exe-p in search box or run box
- ↓ Click create profile button
- ↓ Click next
- ↓ Enter a name to profile
- ↓ Click “finish”
- ↓ Select created profile name
- ↓ Click on start Firefox
- ↓ Click open menu
- ↓ Click options
- ↓ Click on content
- ↓ Select “do not disturb me no notification will be shown option”
- ↓ Click Firefox browser
- ↓ Use below like code in daily programs (or) tests scripts.

#### **Syntax:-**

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxProfile;
import org.openqa.selenium.firefox.internal.ProfilesIni;
public class Mozillanotifications1
{
 public static void main(String[] args) throws Exception
 {
ProfilesIni fi=new ProfilesIni();
FirefoxProfile fp=fi.getProfile("mahesh");
```

```

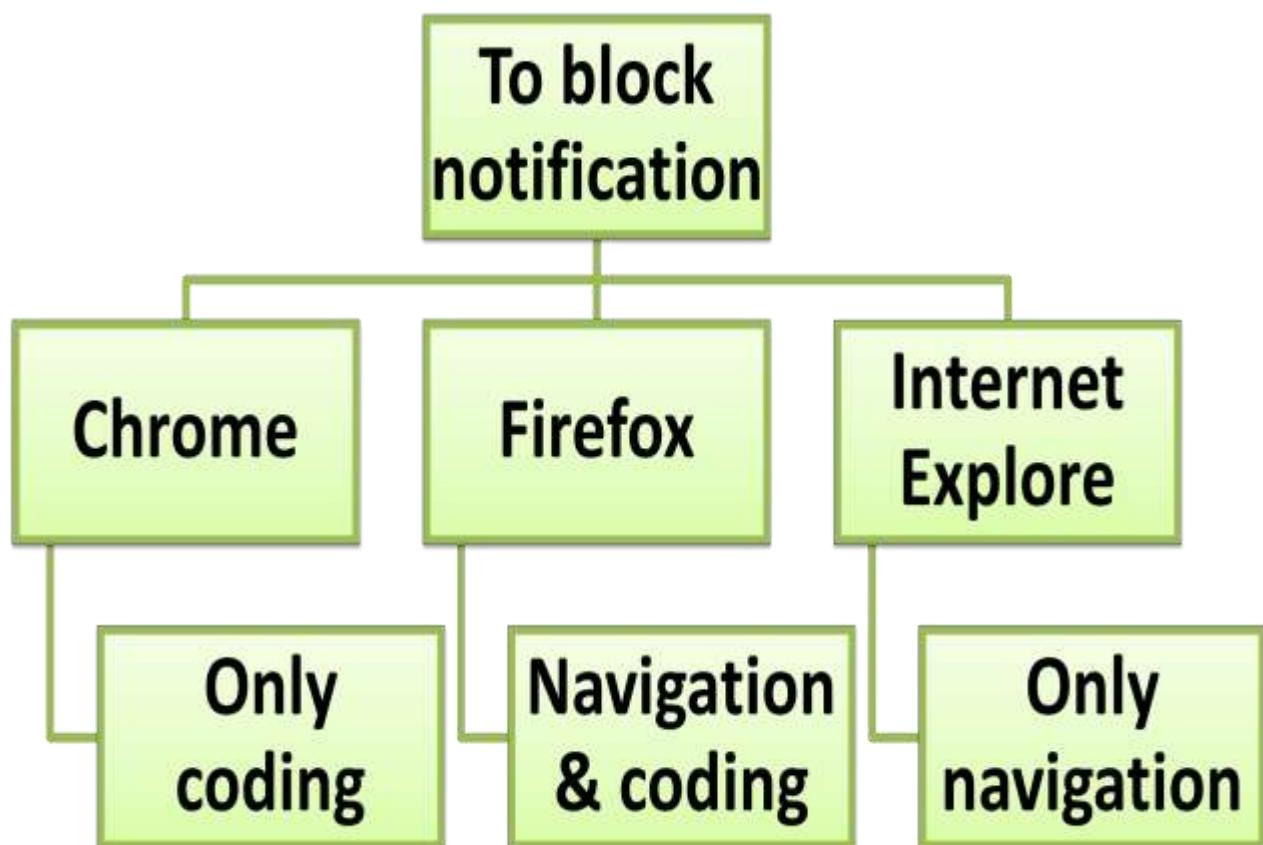
System.setProperty("webdriver.gecko.driver", "D:\\geckodriver.exe"
);
WebDriver Driver=new FirefoxDriver(fp);
Driver.get("http://www.Gmail.com");
Driver.findElement(By.id("identifierId")).sendKeys("gmailid@gmail
.com",Keys.ENTER);
Thread.sleep(1000)
Driver.findElement(By.name("password")).sendKeys("Password2");
Thread.sleep(1000);
Driver.close

```

c) **For IE browser:-**

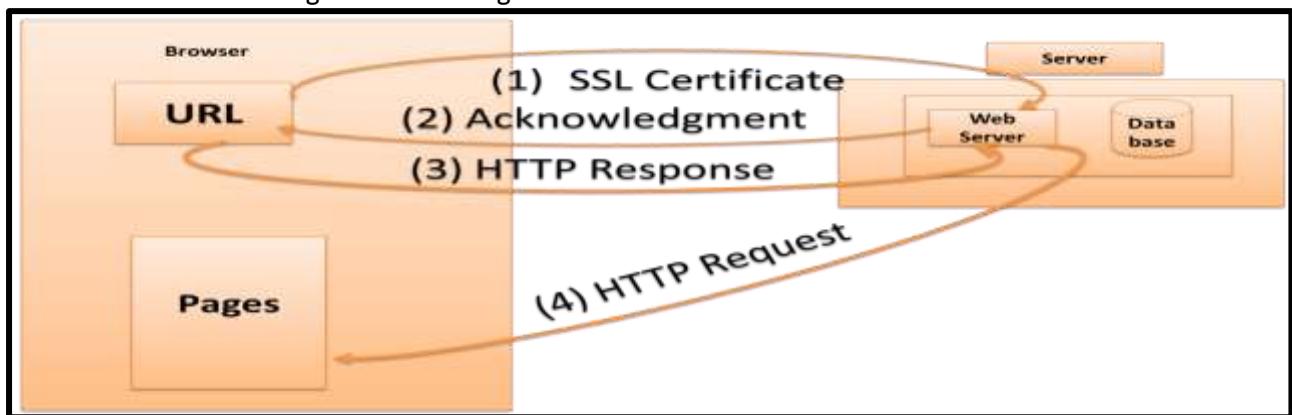
To stop notifications for IE browser during automation, we can follow below like navigation:

- ↓ Open IE browser
- ↓ Tools menu
- ↓ Internet options
- ↓ Privacy
- ↓ Click on settings in “POP-UP BLOCKER”
- ↓ Deselect options under notifications and blocking level.
- ↓ Click “Close”
- ↓ Click “Apply”
- ↓ Click “OK”



## K. Handling SSL Certificates:-

In general browser can send SSL certificates (security socket layer) to your web server to get acknowledgement.



- ⇒ When server was not responded for SSL certificates, corresponding browser can show error page instead of corresponding website homepage.
- ⇒ Due to this reason, we can follow below ways to prevent SSL certificate concept. While automation

### a) For Chrome Browser: -

```

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.remote.DesiredCapabilities;
public class ChromeSSLCer
{
 public static void main(String[] args) throws InterruptedException
 {
 DesiredCapabilities dc=new DesiredCapabilities();
 dc.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
 System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe")
 ;
 WebDriver Driver=new ChromeDriver(dc);

 }
}

```

### b) For Firefox Browser: -

```

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.firefox.FirefoxProfile;
import org.openqa.selenium.firefox.internal.ProfilesIni;
public class MozillaSSlcert
{
 public static void main(String[] args)
 {
 ProfilesIni pi=new ProfilesIni();
 FirefoxProfile fp=pi.getProfile("mahesh");
 System.setProperty("webdriver.gecko.driver",D:\\geckodriver.exe");
 WebDriver driver=new FirefoxDriver(fp);
 fp.setAcceptUntrustedCertificates(true);
 fp.setAssumeUntrustedCertificateIssuer(false);
 driver.get("any web site name");

 ----}
}

```

### c) For Internet Explorer Browser: -

```

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
public class IESSL {
{
 public static void main(String[] args) throws InterruptedException
 {
 DesiredCapabilities dc=new DesiredCapabilities();
 dc.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
 System.setProperty("webdriver.ie.driver","d:\\IEDriverServer1.exe");
 WebDriver Driver=new InternetExplorerDriver();

 }
}

```

#### Note1:-

While automating webpages using selenium webdriver jars we can use below class.

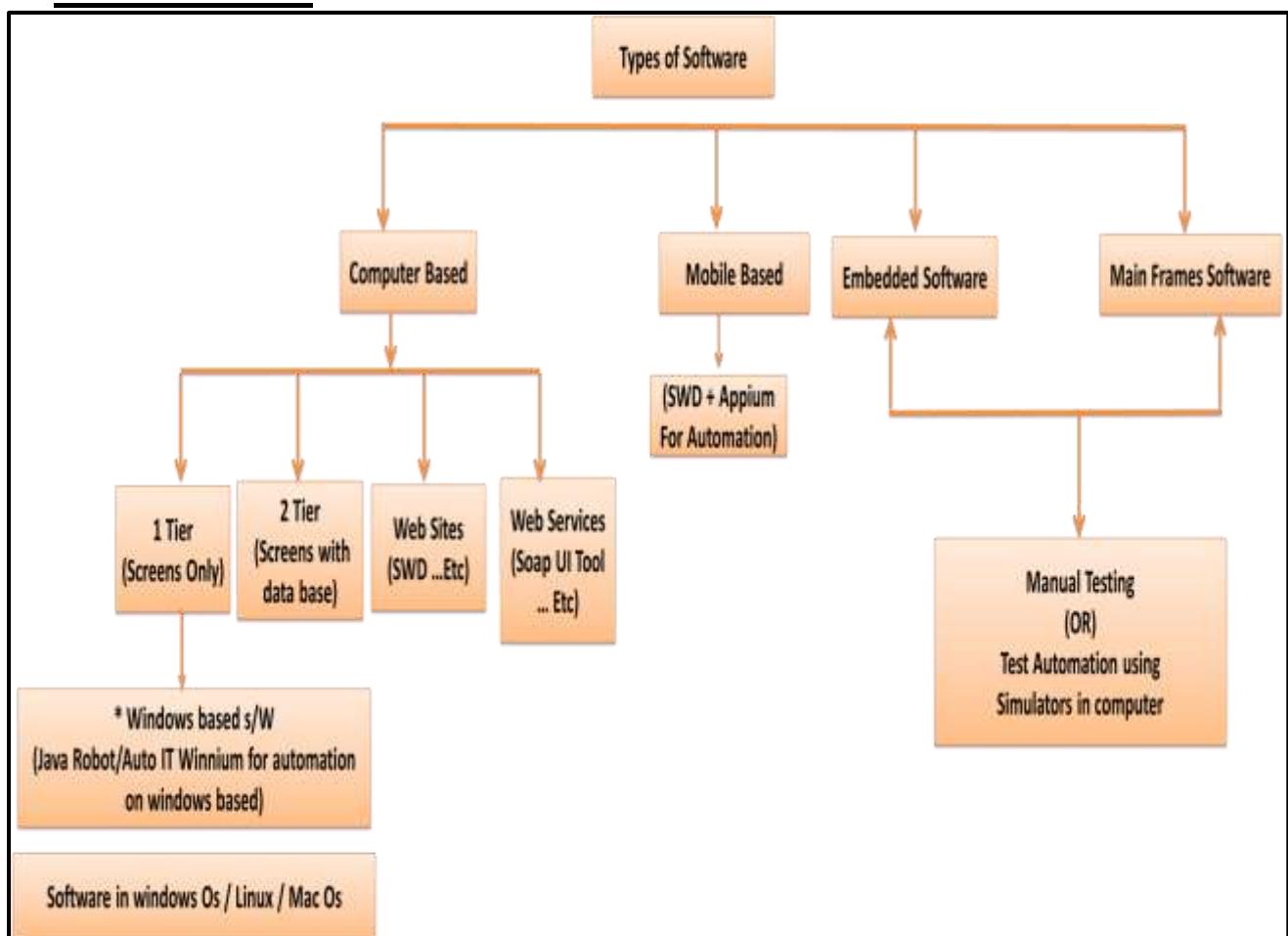
|                         |                      |                       |                      |
|-------------------------|----------------------|-----------------------|----------------------|
| → WebDriver             | → TakesScreenshot    | → Point               | → CapabilityType     |
| → ChromeDriver          | → Dimension          | → WebDriverWait       | → ExpectedConditions |
| → FirefoxDriver         | → JavascriptExecutor | → DesiredCapabilities | → FluentWait         |
| → InternetExploreDriver | → ProfilesIni        | → WebElement          | → Wait               |
| → OperaDriver           | → Actions            | → Select              | → ChromeOptions      |
| → SafariDriver          | → Keys               | → By                  | → FirefoxProfile     |
| → ..... . .             | → ..... . .          | → ..... . .           | → ..... . Etc        |

#### Note2:-

While using above selenium classes in automation we can use few Java (JDK) classes for supporting

- System
- Thread
- List
- ArrayList
- Fileutils
- TimeUnit
- .....ETC

### III. JAVA ROBOT



From the above diagram, Java Robot (or) Auto IT (OR) Winnium is useful to automate windows based software's in windows / Linux / Mac -Computers.

Here Java Robot is more famous "Robot" is class in JDK, so no extra downloading and installation for it. While using Java Robot concept, for windows based software automation, we can use below JDK Classes.

|                                                                                     |                   |                    |
|-------------------------------------------------------------------------------------|-------------------|--------------------|
| In Java .awt Package of SDK<br>(active Window Tool Kit Or abstract Window Tool Kit) | > Runtime         | > (Static Class)   |
|                                                                                     | > Robot           | > (Instance Class) |
|                                                                                     | > KeyEvent        | > (Static Class)   |
|                                                                                     | > Toolkit         | > (Static Class)   |
|                                                                                     | > StringSelection | > (Instance Class) |
|                                                                                     | > DataFlavour     | > (Static Class)   |

#### Example 1:-

- Launch Calculator window
- Enter a value
- Click add
- Enter second value
- Click on Equal
- Get output and display
- Close calculator window

#### Test Script :-

```

import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.datatransfer.DataFlavor;
import java.awt.datatransfer.StringSelection;
import java.awt.event.KeyEvent;

```

```
public class Robotcal
{
 public static void main(String[] args) throws Exception
 {
 Runtime.getRuntime().exec("calc.exe");
 Thread.sleep(5000);
 Robot r=new Robot();
 // Put data in Clipboard value
 StringSelection s= new StringSelection("6819");
 Toolkit.getDefaultToolkit().getSystemClipboard().setContents(s,
 null);
 Thread.sleep(5000);
 // Paste 1st value
 r.keyPress(KeyEvent.VK_CONTROL);
 r.keyPress(KeyEvent.VK_V);
 r.keyRelease(KeyEvent.VK_V);
 r.keyRelease(KeyEvent.VK_CONTROL);
 Thread.sleep(5000);
 // Operating
 r.keyPress(KeyEvent.VK_ADD);
 r.keyRelease(KeyEvent.VK_ADD);
 Thread.sleep(5000);
 // Put data in Clipboard value
 StringSelection s1= new StringSelection("68109");
 Toolkit.getDefaultToolkit().getSystemClipboard().setContents(s1,
 null);
 Thread.sleep(5000);
 // Pasting into screen
 r.keyPress(KeyEvent.VK_CONTROL);
 r.keyPress(KeyEvent.VK_V);
 r.keyRelease(KeyEvent.VK_V);
 r.keyRelease(KeyEvent.VK_CONTROL);
 Thread.sleep(5000);
 // Make Eq
 r.keyPress(KeyEvent.VK_EQUALS);
 r.keyRelease(KeyEvent.VK_EQUALS);
 Thread.sleep(5000);
 // Copying Out Put
 r.keyPress(KeyEvent.VK_CONTEXT_MENU);
 r.keyRelease(KeyEvent.VK_CONTEXT_MENU);
 Thread.sleep(5000);
 r.keyPress(KeyEvent.VK_DOWN);
 r.keyRelease(KeyEvent.VK_DOWN);
 Thread.sleep(5000);
 r.keyPress(KeyEvent.VK_ENTER);
 r.keyRelease(KeyEvent.VK_ENTER);
 Thread.sleep(5000);
 String s2=(String)
 Toolkit.getDefaultToolkit().getSystemClipboard().getData(DataFlavor.
 stringFlavor);
 System.out.println(s2);
 // Close Calc
 r.keyPress(KeyEvent.VK_ALT);
 r.keyPress(KeyEvent.VK_F4);
 r.keyRelease(KeyEvent.VK_F4);
 r.keyRelease(KeyEvent.VK_ALT);
 }
}
```

**Example 2:-**

- ↓ Launch Notepad.
- ↓ Click New in File Menu.
- ↓ Enter Some data.
- ↓ Click save option in file Menu.
- ↓ Enter file name to save that file.
- ↓ Close Notepad.

**Test Script :-**

```

package robot;
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.datatransfer.StringSelection;
import java.awt.event.KeyEvent;
public class Robotnotepad
{
 public static void main(String[] args) throws Exception
 {
 // launch notepad
 Runtime.getRuntime().exec("notepad.exe");
 Thread.sleep(5000);
 Robot r=new Robot();
 // click new in file menu
 r.keyPress(KeyEvent.VK_ALT);
 r.keyPress(KeyEvent.VK_F);
 r.keyRelease(KeyEvent.VK_F);
 r.keyRelease(KeyEvent.VK_ALT);
 Thread.sleep(5000);
 // Click Enter
 r.keyPress(KeyEvent.VK_ENTER);
 r.keyRelease(KeyEvent.VK_ENTER);
 Thread.sleep(5000);
 //Enter Some Text
 StringSelection x=new StringSelection("Hi this Mahesh");
 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);
 Thread.sleep(5000);
 // Save Data
 r.keyPress(KeyEvent.VK_ALT);
 r.keyPress(KeyEvent.VK_F);
 r.keyRelease(KeyEvent.VK_F);
 r.keyRelease(KeyEvent.VK_ALT);
 Thread.sleep(5000);
 r.keyPress(KeyEvent.VK_S);
 r.keyRelease(KeyEvent.VK_S);
 Thread.sleep(5000);
 //Enter file path and save
 StringSelection x1=new StringSelection("D:\\Mahesh.txt");
 Toolkit.getDefaultToolkit().getSystemClipboard().setContents(x1,null);
 r.keyPress(KeyEvent.VK_CONTROL);
 r.keyPress(KeyEvent.VK_V);
 r.keyRelease(KeyEvent.VK_V);
 r.keyRelease(KeyEvent.VK_CONTROL);
 Thread.sleep(5000);
 // Save with file name
 r.keyPress(KeyEvent.VK_ENTER);
 r.keyRelease(KeyEvent.VK_ENTER);
 Thread.sleep(5000);
 String xy=System.getProperty("os.name");
 System.out.println(xy);
 // Close notepad
 }
}

```

```

 r.keyPress(KeyEvent.VK_ALT);
 r.keyPress(KeyEvent.VK_F4);
 r.keyRelease(KeyEvent.VK_F4);
 r.keyRelease(KeyEvent.VK_ALT);
 }
}

```

**Example 3:-**

|                               |                                 |
|-------------------------------|---------------------------------|
| ↓ Launch Gmail Site           | <b>Using Selenium WebDriver</b> |
| ↓ Do Login using valid data   |                                 |
| ↓ Click Compose               |                                 |
| ↓ Fill fields in compose page |                                 |
| ↓ Click attachment            | <b>Using Java Robot</b>         |
| ↓ Automate file Upload window |                                 |
| ↓ Send Mail                   | <b>Using Selenium WebDriver</b> |
| ↓ Do Logout                   |                                 |
| ↓ Close Site                  |                                 |

**Test Scripts:-**

```

package robot;
import java.awt.AWTException;
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.datatransfer.StringSelection;
import java.awt.event.KeyEvent;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class GmailAutomate
{
 public static void main(String[] args) throws Exception
 {
 // Launch Gmail
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Robot r=new Robot();
 Driver.get("http://www.Gmail.co.in");
 // Do Login
 Driver.findElement(By.id("identifierId")).sendKeys("Mailed@gmail.com");
 Driver.findElement(By.id("identifierNext")).click();
 Thread.sleep(1000);
 Driver.findElement(By.name("password")).sendKeys("password");
 Thread.sleep(1000);
 Driver.findElement(By.id("passwordNext")).click();
 Thread.sleep(2000);
 // Automating Notifications Credentials
 if(Driver.findElement(By.xpath("//*[text()='COMPOSE']")).isDisplayed())
 {
 r.keyPress(KeyEvent.VK_ESCAPE);
 r.keyRelease(KeyEvent.VK_ESCAPE);
 }
 Thread.sleep(2000);
 // Click on Compose
 Driver.findElement(By.xpath("//*[text()='COMPOSE']")).click();
 Thread.sleep(2000);
 // Fill Mail Details
 Driver.findElement(By.name("to")).sendKeys("mailed@gmail.com");
 Driver.findElement(By.name("subjectbox")).sendKeys("Wishes");
 Driver.findElement(By.xpath("//[@aria-label='Message Body'][2]")).sendKeys("Hi Sir,"+Keys.ENTER+"How are You sir?",Keys.ENTER, Keys.ENTER,
 "Bye Sir",Keys.ENTER,Keys.ENTER,Keys.ENTER,Keys.ENTER);
 }
}

```

```

// Click On Attachment Files
Driver.findElement(By.xpath("//*[@ data-tooltip='Attach
files']/descendant::*[3]")).click();
Thread.sleep(5000);
// Handle File Upload Window
StringSelection x=new StringSelection("D:\\mahesh\\pic.png");
Toolkit.getDefaultToolkit().getSystemClipboard().setContents(x, null);
Thread.sleep(5000);
// Pasting into screen
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_ENTER);
r.keyRelease(KeyEvent.VK_ENTER);
Thread.sleep(25000);
Driver.findElement(By.xpath("//*[text()='Send']")).click();
Thread.sleep(5000);
Driver.findElement(By.xpath("//*[contains(@title, 'Google
Account')]/parent::*")).click();
Driver.findElement(By.xpath("//*[text()='Sign out']")).click();
Thread.sleep(5000);
Driver.close();}
}

```

### **Note 1:-**

while using Java Robot concept to automate pop-windows and window based controls, we can automate these key combinations are changing for OS to OS. To make Java Robot related programs as platform independent , we need to add below like logic To test

### **Test Scripts :-**

```

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);
 r.keyPress(KeyEvent.VK_ENTER);
 r.keyRelease(KeyEvent.VK_ENTER);
}
else if(System.getProperty("os.name").contains("Mac"))
{
 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_ENTER);
 r.keyRelease(KeyEvent.VK_ENTER);
}
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);
 r.keyPress(KeyEvent.VK_ENTER);
 r.keyRelease(KeyEvent.VK_ENTER);
}
else{ System.exit(0);
}

```

**Note 2:-**

In general Keyboard key combinations are changing w.r.t operating system (OS) for Copy and Paste like Operations.

| Operating System | Copy           | Paste    |
|------------------|----------------|----------|
| Windows          | Ctrl +C        | Ctrl + V |
| Mac              | Meta + C       | Meta + V |
| Linux            | Alt + Ctrl + C | Ctrl + V |

**Note 3:-**

While automating window based screens and Pop Up windows in webpages we use AutoIt or Winnium are alternatives to java robot concept , but they are working in Windows Os only.

**Example 4:-**

|                                        |                     |
|----------------------------------------|---------------------|
| ↓ Launch Seleniqumhq.org site          | Selenium Web Driver |
| ↓ Click Download Link                  |                     |
| ↓ Right Click on a software a downlink |                     |
| ↓ Choose Second option in context Menu | Java Robot          |
| ↓ Close Site                           | Selenium Web Driver |

**Test Script:-**

```
package robot;
import java.awt.Robot;
import java.awt.event.KeyEvent;
import java.util.ArrayList;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

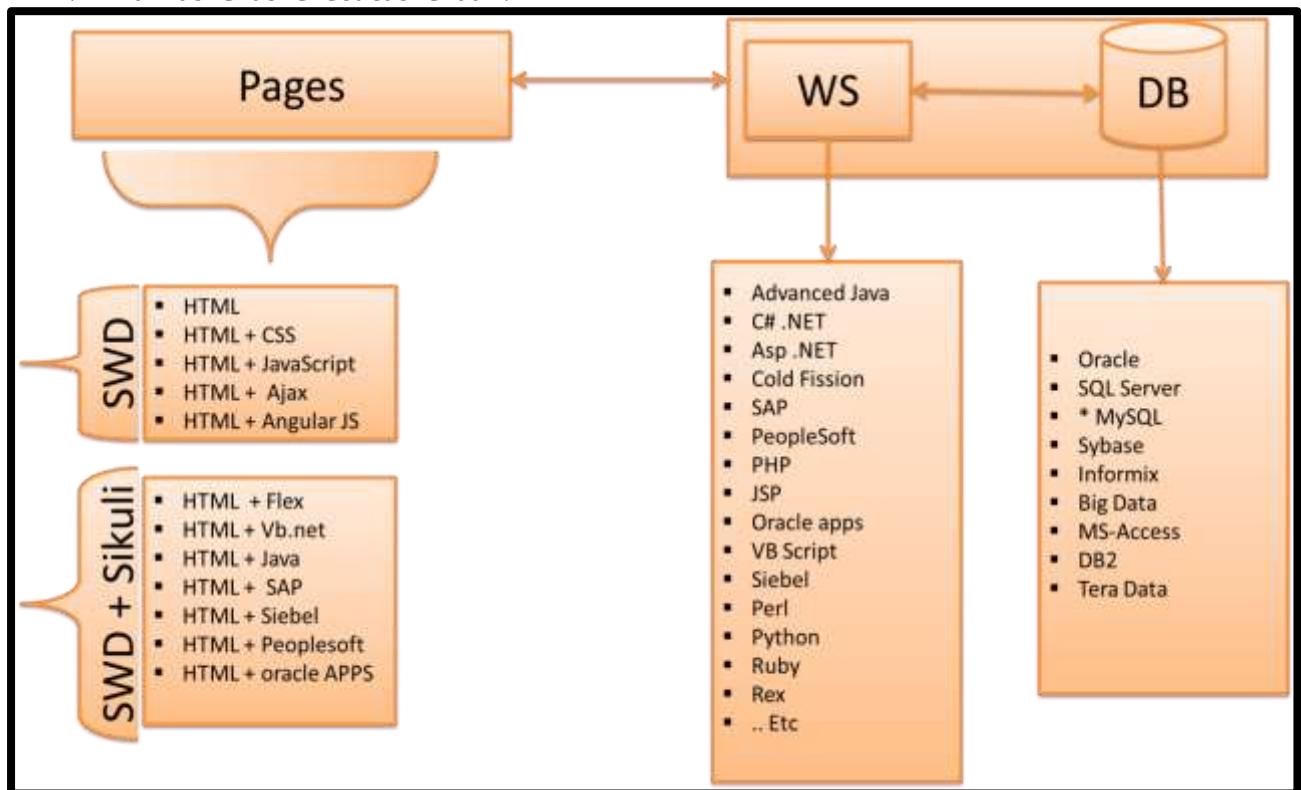
public class SiteAutomateSelenium
{
 public static void main(String[] args) throws Exception
 {
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.get("http://www.seleniumhq.org");
 Thread.sleep(5000);
 Robot r=new Robot();
 // Click Link sing SWD
 Driver.findElement(By.LinkText("Download")).click();
 Thread.sleep(5000);
 // Right Click Link sing SWD
 WebElement e= Driver.findElement(By.LinkText("3.4.0"));
 Actions a=new Actions(Driver);
 a.contextClick(e).build().perform();
 // Chose Second Option
 r.keyPress(KeyEvent.VK_DOWN);
 r.keyRelease(KeyEvent.VK_DOWN);
 Thread.sleep(5000);
 r.keyPress(KeyEvent.VK_DOWN);
 r.keyRelease(KeyEvent.VK_DOWN);
 Thread.sleep(5000);
 r.keyPress(KeyEvent.VK_ENTER);
 r.keyRelease(KeyEvent.VK_ENTER);
 Thread.sleep(5000);
 }
}
```

```
// To save data in hard To select Keep/Discard
r.keyPress(KeyEvent.VK_SHIFT);
r.keyPress(KeyEvent.VK_TAB);
r.keyRelease(KeyEvent.VK_TAB);
r.keyRelease(KeyEvent.VK_SHIFT);
Thread.sleep(5000);
r.keyPress(KeyEvent.VK_SHIFT);
r.keyPress(KeyEvent.VK_TAB);
r.keyRelease(KeyEvent.VK_TAB);
r.keyRelease(KeyEvent.VK_SHIFT);
Thread.sleep(5000);
r.keyPress(KeyEvent.VK_SHIFT);
r.keyPress(KeyEvent.VK_TAB);
r.keyRelease(KeyEvent.VK_TAB);
r.keyRelease(KeyEvent.VK_SHIFT);
Thread.sleep(5000);
r.keyPress(KeyEvent.VK_ENTER);
r.keyRelease(KeyEvent.VK_ENTER);
Thread.sleep(5000);
ArrayList<String> a1=new
ArrayList<String>(Driver.getWindowHandles());
Driver.switchTo().window(a1.get(1));
System.out.println(a1.size());
Driver.close();
Thread.sleep(5000);
Driver.switchTo().window(a1.get(0));
Thread.sleep(5000);
Driver.close();
}
}
```



## IV. SIKULI X

- ⇒ Developed by “Raiman”.
- ⇒ Java Based Open Source Tool.
- ⇒ Useful to automate any type of software screens, but more popular for non html web pages.
- ⇒ Available as executable Jar.



### A. Configure “SIKULI” in Tester Computer

- ↓ Go to Google Site.
- ↓ Enter “Sikuli X” Launch Pad.
- ↓ Go to “Launch pad.net” Site.
- ↓ Click On Sikuli X setup Jar to start download.
- ↓ Paste that download in personal folder.
- ↓ Double click on that download.
- ↓ Select 2nd option in set up window.
- ↓ Click “Setup now “ button.
- ↓ Click “Yes” until “have Fun “Message.
- ↓ Click Ok.
- ↓ Observe the availability of Sikuli X api Jar in personal folder.
- ↓ Launch Eclipse IDE.
- ↓ Right Click on Project.
- ↓ Go to Properties.
- ↓ Java Build Path.
- ↓ Libraries.
- ↓ Add External Jars.
- ↓ Browser Sikuli X API Jar
- ↓ Click Ok

## **B. Configure “SIKULI” in Tester Computer**

To automate non-html elements in web pages we can use Sikuli X. Here we need to locate elements in webpages by matching with .png images.  
To make elements as .png images we can use Snipping Tool( inbuilt in Window Operating System), Qsnap(download and install)...etc.

## **C. “Screen” Class Methods:-**

Screen is a base class provided by Sikuli. We need to create object for this screen class first, then only we can access all the methods provided by Sikuli.

**Syntax:**

```
Screen s=new Screen();
```

### **1. Click on Element:-**

We can use this method to locate an element by matching with .png image and perform click

**Syntax:**

```
s.click("path of image");
```

**Example:**

```
s.click("D:\\Pic1.png");
```

### **2. Double click On Element:-**

We can use this method to locate an element by matching with .png image and perform double click

**Syntax:**

```
s.doubleClick("path of image");
```

**Example:**

```
s.doubleClick("D:\\Pic1.png");
```

### **3. Right click On Element:-**

We can use this method to locate an element by matching with .png image and perform right click

**Syntax:**

```
s.rightClick("path of image");
```

**Example:**

```
s.rightClick("D:\\Pic1.png");
```

### **4. Mouse Move to Element:-**

We can use this method to locate an element by matching with .png image and perform move mouse pointer to that element.

**Syntax:**

```
s.mouseMove("path of image");
```

**Example:**

```
s.mouseMove("D:\\Pic1.png");
```

### **5. Drag Drop an Element:-**

We can use this method to locate two elements by matching with .png images and perform drag and drop operation in b/w those elements.

**Syntax:**

```
s.dragDrop("Path of source image name" , "path target image name");
```

**Example:**

```
s.dragDrop("D:\\Pic1.png" , " D:\\Pic1.png");
```

```
s.dragDrop(arg0, arg1)
```

### **6. Existence of an Element:-**

We can use this method to check that availability of an element by matching with .png image.

**Syntax:**

```
s.exists("path of image");
```

**Example:**

```
Screen s=new Screen()
If (s.exists("d:\\Pic1.png")){
System.out.println("Found")
}
else{
System.out.println("Found")
}
```

## **7.Wheel ():**

We can use this method to move mouse pointer specified location on screen.

**Syntax:**

```
Location l=new Location(int X coordinate, int Y coordinate);
Location l=new Location(x, y)
s.wheel(target, direction, steps)
import org.sikuli.script.Location
```

**Example:**

```
Screen s=new Screen()
Location l=new Location(500,500);
s.wheel(l,Button.LEFT,1);
```

**Example 1:-**

Move mouse pointer to specified Location and Click

**Script**

```
Location l=new Location(500,500);
s.wheel(l,Button.LEFT,1);
```

**Example 2:-**

Move mouse pointer to specified Location and perform double Click

**Script**

```
Location l=new Location(500,500);
s.wheel(l,Button.LEFT,2);
```

**Example 3:-**

Move mouse pointer to specified Location and perform right Click

**Script**

```
Location l=new Location(500,500);
s.wheel(l,Button.RIGHT,1);
```

**Example 4:-**

Scroll Down screen five times

**Script**

```
s.wheel(direction, steps)
s.wheel(Button.WHEEL_DOWN,5);
```

**Example 5:-**

Scroll Down screen five times

**Script**

```
s.wheel(direction, steps)
s.wheel(Button.WHEEL_UP,5);
```

**Note1-**

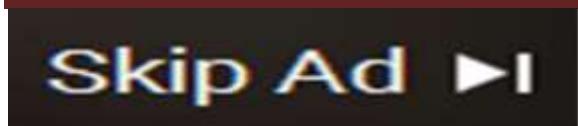
In general , “dragDrop( )”is used to drag and drop in between two elements by matching with .png images. But this method is useful to perform drag and drop in between element and location also. This idea is useful for “sliders”. (Horizontal Or Vertical).

Match e=s.find("pic.png");/// match with slider element

```
Location l1=new Location(e.getX()-10,e.getY());
```

### **Scenario1:**

- ⇒ Launch YouTube site using chrome browser using SWD.
- ⇒ Search for a vide using SWD
- ⇒ Star a video click on link using SWD
- ⇒ Click on Skip Ad if Exist using Sikulix
- ⇒ Wait few seconds using SWD
- ⇒ Click pause using Sikulix
- ⇒ Wait few seconds using SWD
- ⇒ Click Play using Sikulix
- ⇒ Wait few seconds
- ⇒ Decrease volume using Sikulix
- ⇒ Wait for few seconds
- ⇒ Increase volume using Sikulix
- ⇒ Wait for few seconds
- ⇒ Close web Site using SWD

**Test Script:-**

```

package sikuli;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.sikuli.script.Button;
import org.sikuli.script.Location;
import org.sikuli.script.Match;
import org.sikuli.script.Screen;
public class Skuiyoutube
{
 public static void main(String[] args) throws Throwable
 {
 //launch web site
 System.setProperty("webdriver.chrome.driver","D:\\chromedriver.exe");
 WebDriver driver= new ChromeDriver();
 Thread.sleep(5000);
 driver.get("http://www.youtube.com");
 driver.manage().window().maximize();
 Thread.sleep(5000);
 driver.findElement(By.name("search_query")).sendKeys("bahubali 2 trailer");
 Thread.sleep(5000);
 driver.findElement(By.id("search-btn")).click();
 Thread.sleep(5000);
 driver.findElement(By.LinkText("Baahubali 2 - The Conclusion Trailer | Prabhas, Rana Daggubati | SS Rajamouli")).click();
 Thread.sleep(5000);
 Screen s=new Screen();
 Location l=new Location(300,200);
 s.wheel(l,Button.LEFT,0);
 s.click("pause.png");
 Thread.sleep(5000);
 s.wheel(l,Button.LEFT,0);
 s.click("play.png");
 Thread.sleep(5000);
 // volume
 s.wheel(l,Button.LEFT,0);
 s.mouseMove("volume.png");
 Thread.sleep(5000);
 // -- volume
 Match e=s.find("bar.png");
 Location l1=new Location(e.getX()-15,e.getY());
 s.dragDrop(e, l1);
 Thread.sleep(5000);
 // changing location
 s.wheel(l, Button.LEFT,0);
 s.mouseMove("volume.png");
 Thread.sleep(5000);
 //++ volume
 Match e1=s.find("bar.png");
 Location l2=new Location(e1.getX()+30,e1.getY());
 s.dragDrop(e1, l2);
 Thread.sleep(5000);
 s.wheel(l, Button.LEFT,0);
 Thread.sleep(5000);
 driver.close();
 }
}

```

## **8.Find an Element:-**

We can use this method to create a DOM (Document object Model) to an element by matching with .png image

### Syntax:-

```
Match e=s.find("pic.png");// match with slider element
```

## **9.Find ALL Elements:-**

We can use this method to create a DOM (Document object Model) to an element by matching with .png image

### Syntax:-

```
Iterator<Match> l=s.findAll("pic.png");// match with slider element
s.click(l.next()); // click First match Element
(In general "List" in java is allowing random access through index where as "Iterator" in java is allowing sequential access through next() method.)
```

### Example:-

```
import java.util.Iterator;
import org.sikuli.script.Match;
import org.sikuli.script.Screen;
public class AllElements
{
 public static void main(String[] args) throws Exception
 {
 Thread.sleep(5000);
 Screen s=new Screen();
 Iterator<Match> l=s.findAll("bomma1.PNG");
 for(int i=1;i<=10;i++)
 {
 l.next();
 }
 s.click(l.next()); // click 11th matched Element
 }
}
```

## **10.type data to Elements:-**

We can use this method to locate an element by matching with .png image and fill that element with given text

### Scenario1:

Launch Gmail Using SWD.

Fill Elements of Email Id And click next using Sikulix.

Fill Elements of Password And click next using Sikulix.

### Syntax:

```
Screen s=new Screen();
s.type("User.png", "Mahesh.samudrala24@gmail.com");
s.click("Ok.png");
Thread.sleep(5000);
s.type("User.png", "password");
s.click("Ok.png");
```



Above images are Text images for Email Id, Password and Login Button

## **11.capture Screen/ Region of Screen:-**

We can use this method to capture fullscreen or region of screen as screenshot.

### Scenario:

Launch Gmail Site

Search For Stevejobs

Make a Screen Shot Of Screen using Sikulix

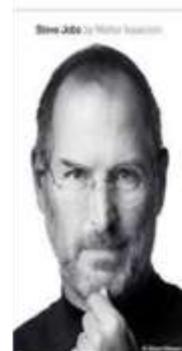
Make a capture of Some Region using Sikulix

### Test Script

```
package sikuli;
import java.awt.image.BufferedImage;
import java.io.File;
```

```
import javax.imageio.ImageIO;
import org.sikuli.script.Key;
import org.sikuli.script.Match;
import org.sikuli.script.Region;
import org.sikuli.script.ScreenImage;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.sikuli.script.Screen;
public class ScreenRegion
{
 public static void main(String[] args) throws Exception
 {
 System.setProperty("webdriver.chrome.driver",
 "d:\\chromedriver.exe");
 WebDriver Driver=new ChromeDriver();
 Driver.manage().window().maximize();
 Driver.get("http://www.google.com");
 Thread.sleep(5000);
 Screen s=new Screen();
 s.type("GSearch", "steve jobs");
 s.keyDown(Key.ENTER);
 s.keyUp(Key.ENTER);
 Thread.sleep(5000);
 ScreenImage si=s.capture();
 BufferedImage bi=si.getImage();
 File f1= new File("d:\\FullScreen.png");
 ImageIO.write(bi,"png",f1);
 // Get a region of screen using Sikulix
 Region r=new Region(823,230,500,240);
 ScreenImage sri=s.capture(r);
 BufferedImage bri=sri.getImage();
 File f2= new File("d:\\mahesh\\region.png");
 ImageIO.write(bri,"png",f2);
 Match e=s.find("stev");
 int st=e.getX();
 System.out.print(st);
 s.click("stev");
 Thread.sleep(5000);
 Driver.close();
 }
}
```

Below is GSearch Image



Beside is Steve image

## **12. keyDown( ) or keyUp( ):-**

We can use this method to automate keyboard keys as individual (or) combination.

### **Syntax:-**

```
import org.sikuli.script.Screen;
import org.sikuli.script.Key;
Screen s=new Screen();
s.keyDown(Key.ALT);
s.keyDown(Key.F4);
s.keyUp(Key.F4);
s.keyUp(Key.ALT);
```

## **13. mouseDown( ) or mouseUp( ):-**

We can use this method to automate to mouse controls.

### **Syntax:-**

```
import org.sikuli.script.Screen;
import org.sikuli.script.Button;
Screen s=new Screen();
s.mouseDown(Button.LEFT);
s.mouseUp(Button.LEFT);
```

### **NOTE:-**

To automate non html elements like flexi ,we can use class & methods of “sikulixapi” jar.

- ⇒ Screen.
- ⇒ Match.
- ⇒ ScreenImage.
- ⇒ BufferedImage.
- ⇒ Button.
- ⇒ Region.

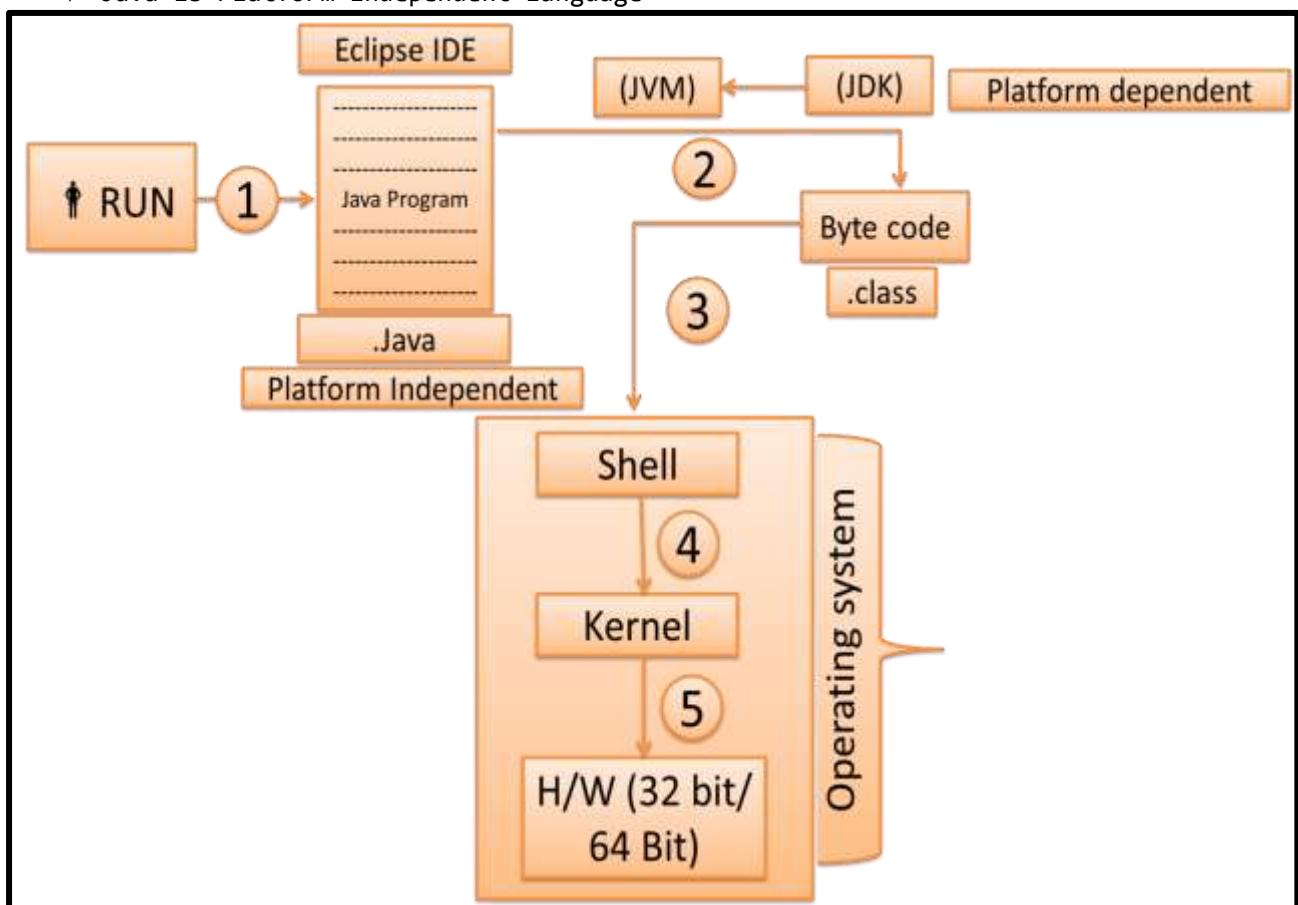






## V. CORE JAVA

- Developed by SUN micro system take over by ORCALE Corporation
- Java is Platform independent language



→ To install JDK for JVM and Java Language, we need to choose JDK w.r.t Operating System & Bit Size(32bit & 64bit)

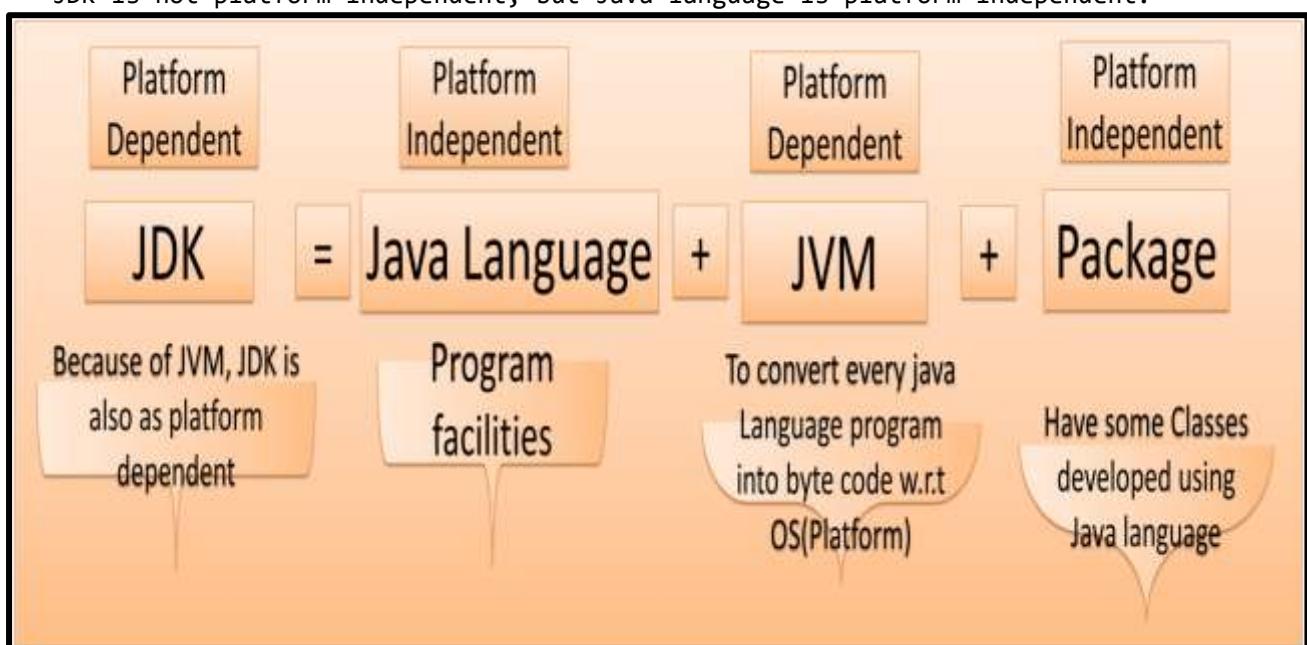
Example:-

JDK 8 for windows x86 for 32 bit size, JDK 8 for windows x64 for 64 bit size.

JDK 8 for Mac Operating System x86 for 32 bit size,

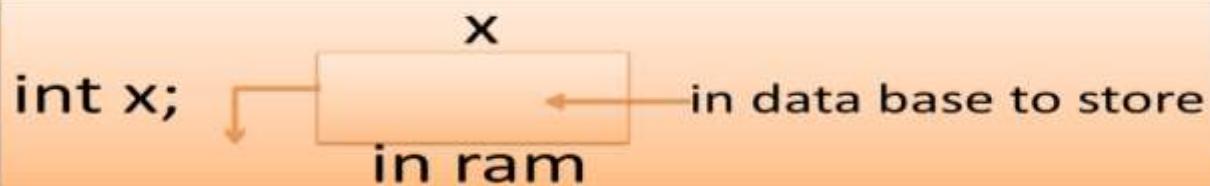
....Etc

JDK is not platform independent, but Java language is platform independent.



## A. Variables in Java language

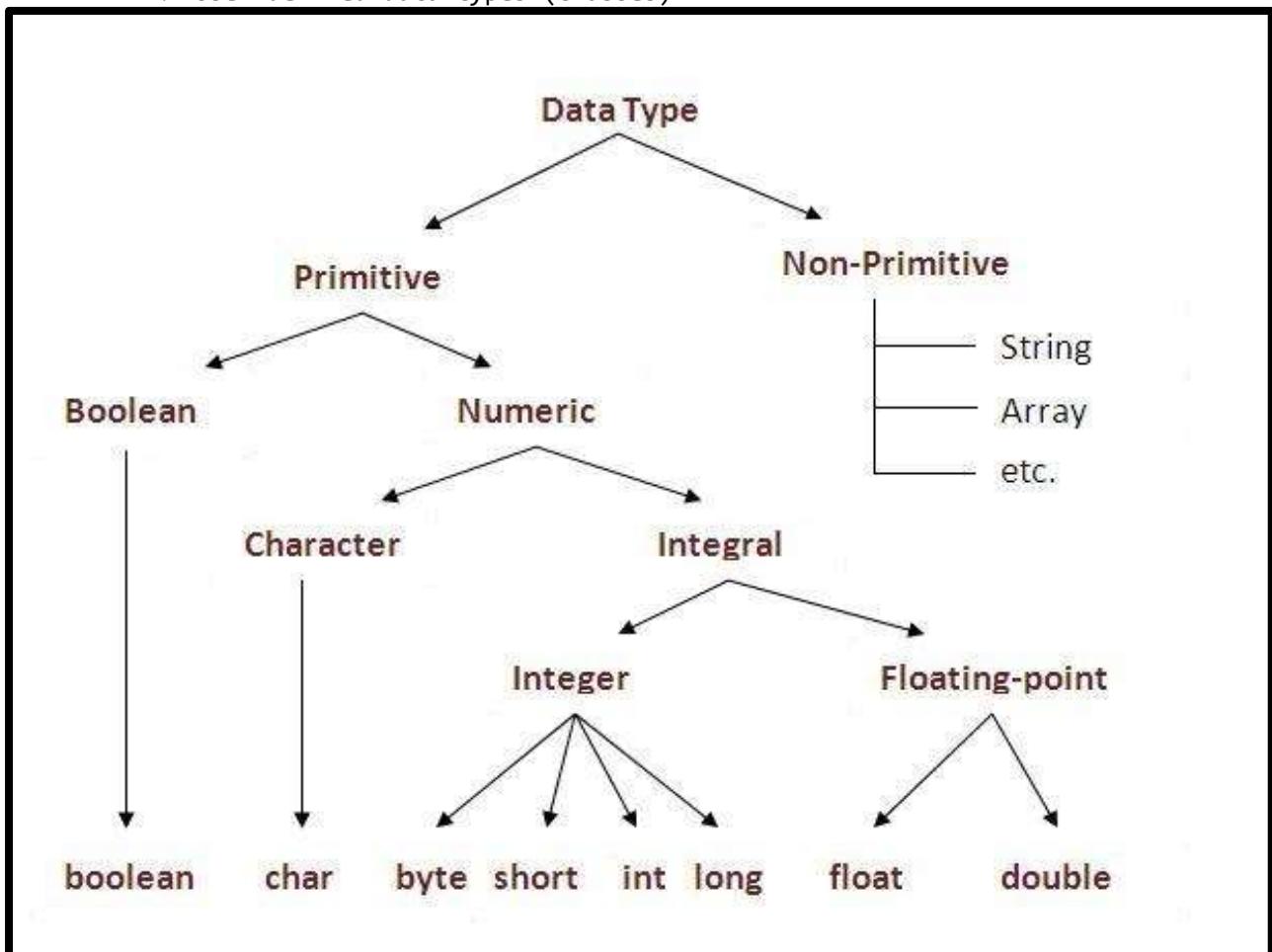
Variable is storage to store one value in specified type.



## B. Data types in java

Data type can specify type of data, in java language data types are "3 types" such as

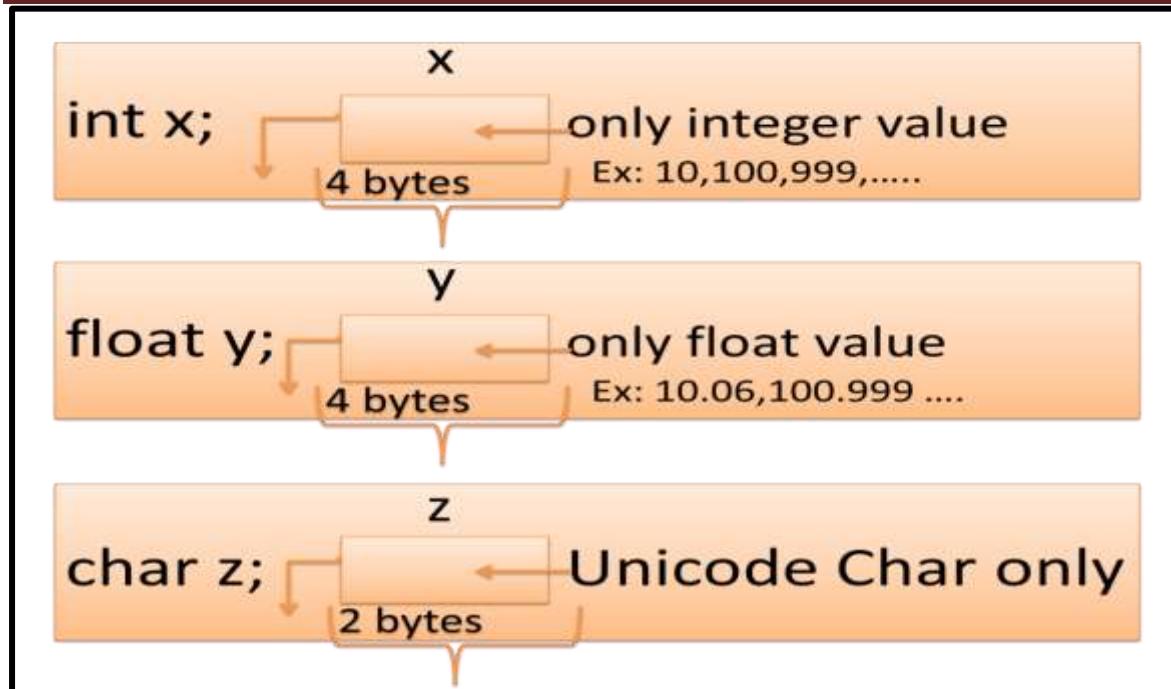
- ⇒ Primitive data types
- ⇒ Derived data types
- ⇒ User defined data types (Classes)



## C. Primitive Data Type:-

This data types are useful to declare variables to store data only. For every data type required memory size is varying.

| Primitive Data Type | Default Value | Default size | Data Range                              |
|---------------------|---------------|--------------|-----------------------------------------|
| Byte                | 0             | 1 byte       | -2 <sup>7</sup> to +2 <sup>7</sup> -1   |
| Short               | 0             | 2 byte       | -2 <sup>15</sup> to +2 <sup>15</sup> -1 |
| Int                 | 0             | 4 byte       | -2 <sup>31</sup> to +2 <sup>31</sup> -1 |
| Long                | 0             | 8 byte       | -2 <sup>63</sup> to +2 <sup>63</sup> -1 |
| Float               | 0.0           | 4 byte       | -2 <sup>31</sup> to +2 <sup>31</sup> -1 |
| Double              | 0.0           | 8 byte       | -2 <sup>63</sup> to +2 <sup>63</sup> -1 |
| boolean             | False         | 1 bit        | 0    1                                  |
| Char                | Blank space   | 2 byte       | Unicode                                 |

**Note 1:**

When we declare variables using primitive data type, we can get memory for those variables with default data, with respect to data type as show in above table.

**Note 2:**

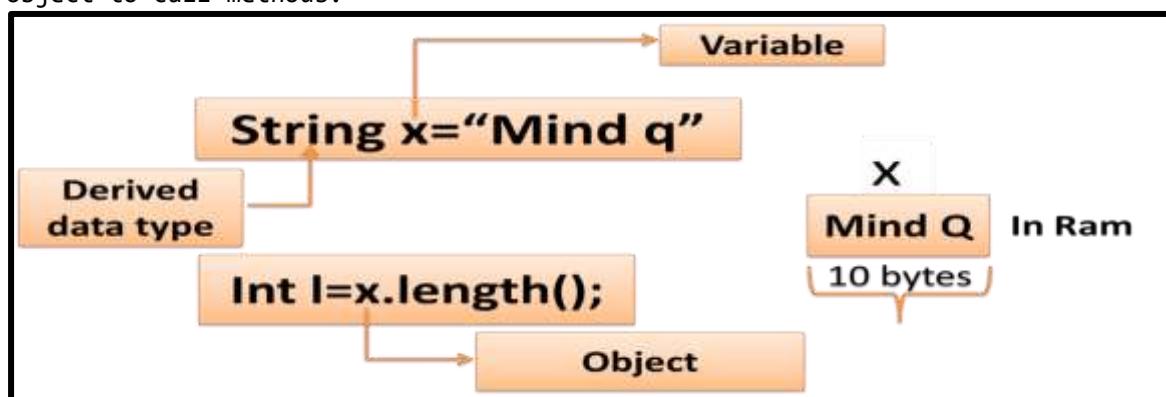
While working with “char” type variable we can use single quotes.

Example:-

```
int x=99;
float y=10000.26;
char z='$';
```

**D. Derived Data Type**

This data types are useful to declare variables, but they can behave like object to call methods.

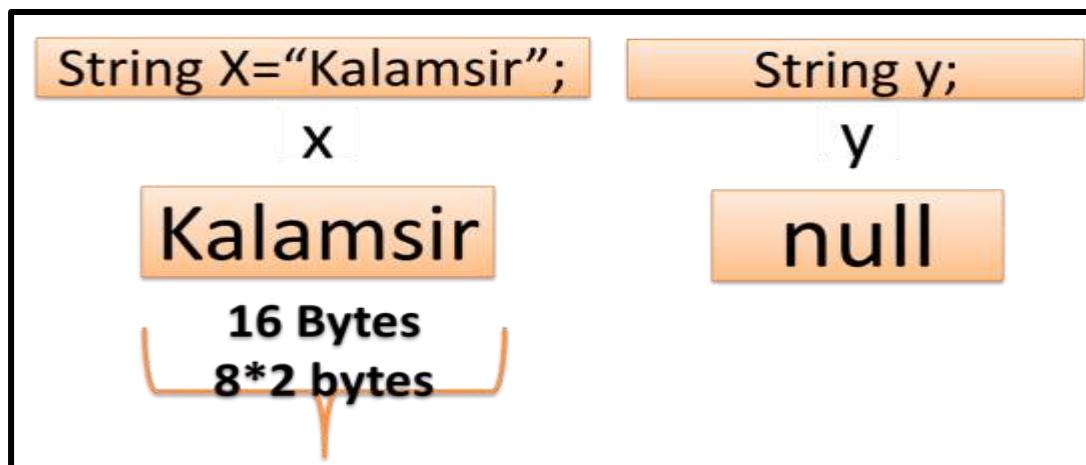
**Note 1:**

In java language we can get multiple derived data types like

- ⇒ String
- ⇒ Integer
- ⇒ Float
- ⇒ ...Etc

**Note 2:**

While using “String” derived data type variables we can get memory for those variables depends on size of given data if we did not provide any data, String variable can get null (no value).



## E. User Defined Data Type (Class)

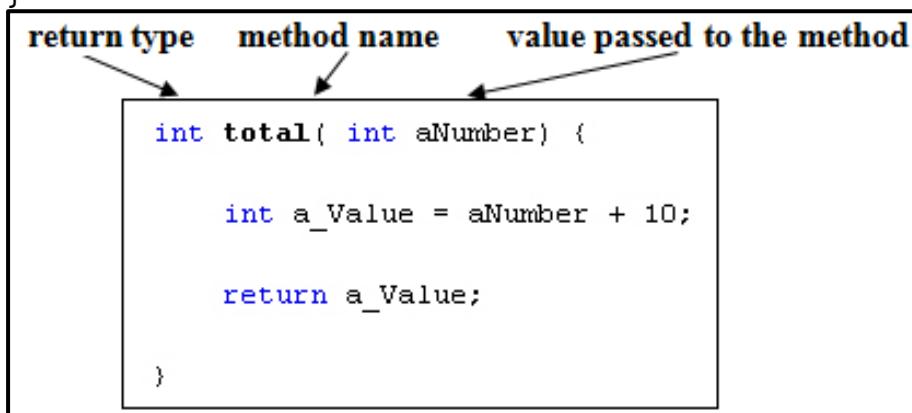
Java language can allow us to create user defined data types/classes. Here every class is working as a “template/Structure/blue print”. In general every class is having data members for storage & methods for operations but not compulsory because some classes are having methods only & some more classes are having data members only.

### Example 1:-

```
package java;
public class mindq
{
// data members
 int x;
 float y;
 char z;
 boolean w;
 String s;
 // Methods
 public void display()
 {
 System.out.println(x);
 System.out.println(y);
 System.out.println(z);
 System.out.println(w);
 System.out.println(s);
 }
}
```

From the above example, class is having data members with methods but those methods are following specific syntax.

```
public returntype methodname(arguments)
{
}
```



There are different types of return types in java they are

- ⇒ void
- ⇒ int
- ⇒ float
- ⇒ String
- ⇒ ..etc

There are different types of arguments

- ⇒ float
- ⇒ int
- ⇒ String
- ⇒ ..etc

In above syntax void represents “no return”, if we want return anything from method we need to specify type & we need to use return statement to use that method.

**Example**

```
public int add(int x, int y)
{
 int z;
 z=x+y;
 return(z);
}
```

After completion of class creation we need to concentrate on access of that class data members & methods via on object. In general object is an instance to class. To create object for a class, we can follow below syntax.

**mindq obj=new mindq();**

Declaration and memory allocation with default data

OR

**mindq obj**

Declaration Only

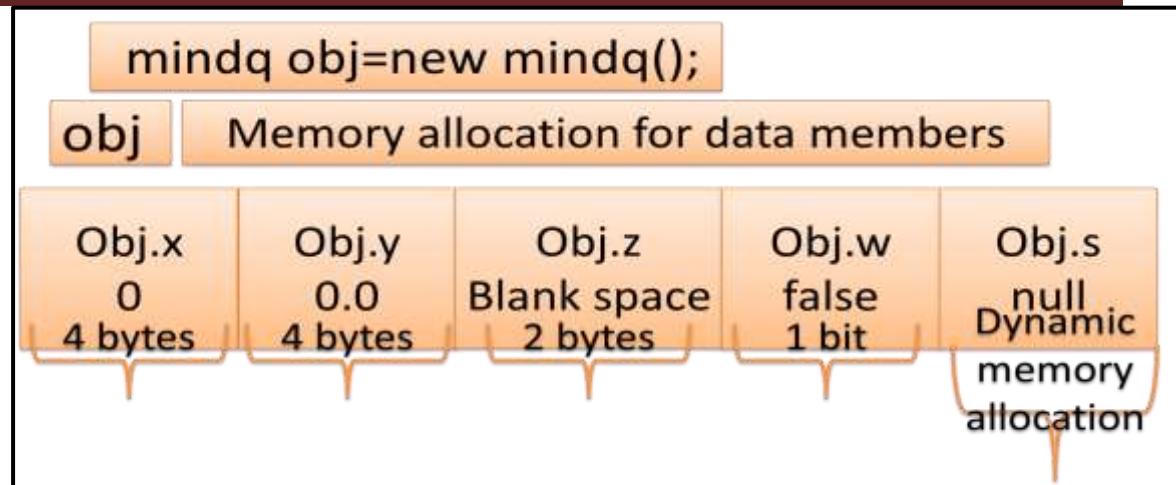
**obj=new mindq();**

memory allocation with default data

When object was ready we are able to access data members & methods in class by using object.

**Example 1:**

```
public class DerivedMain
{
 public static void main(String[] args)
 {
 mindq obj=new mindq();
 obj.display();
 }
}
```



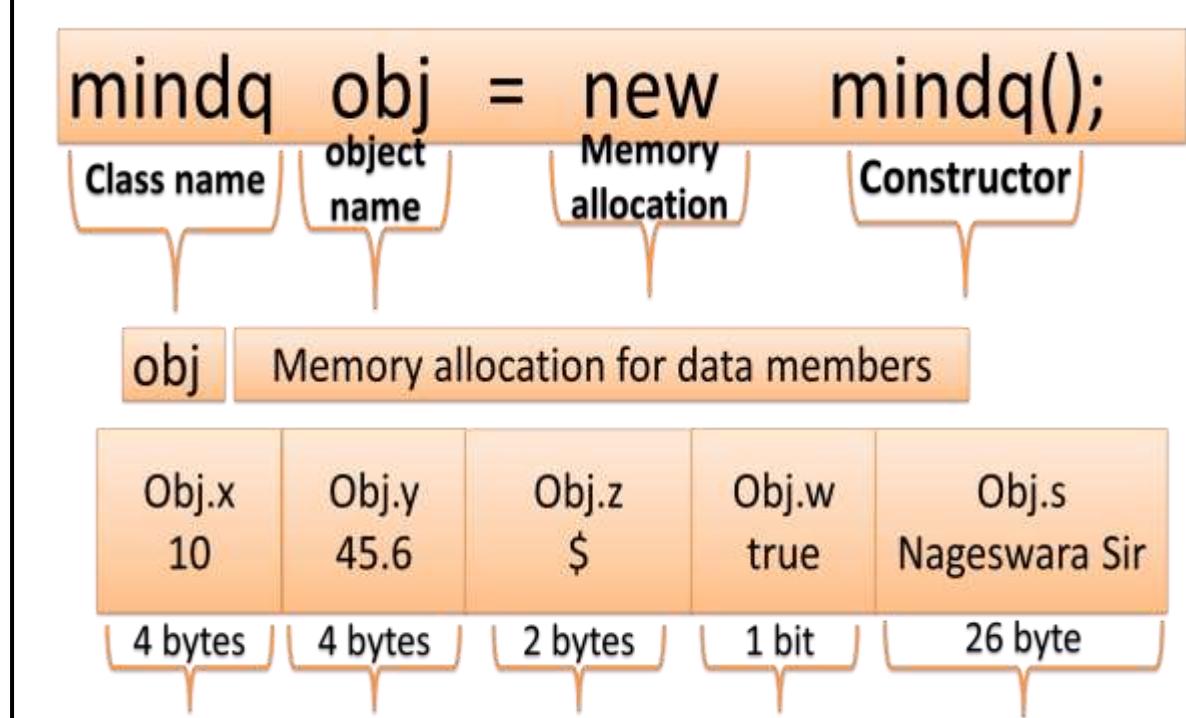
If we want to assign values to data members while creating object, we can use constructor method concepts. Here constructor method name is equal to class name & it won't return anything because it is having code to assign values to memory.

#### Example2:

```
package java1;
public class mindq
{
 int x;
 float y;
 char z;
 String s;
 boolean w;
 // Constructor method
 mindq()
 {
 x=10;
 y= (float) 45.6;
 z='$';
 s="Nageswar Sir";
 w=true;
 }
 public void display()
 {
 System.out.println(x);
 System.out.println(y);
 System.out.println(w);
 System.out.println(s);
 System.out.println(b);
 }
}
```

#### To call Constructor

```
package java1;
public class DerivedMain
{
 public static void main(String[] args)
 {
 mindq obj=new mindq();
 obj.display();
 }
}
```



a. Constructor methods can allow arguments like shown below

```
package java1;
public class mindq
{
 int x;
 float y;
 char z;
 String s;
 boolean w;
 // Constructor method
 mindq(int a, String c)
 {
 x=a;
 y= (float) 45.6;
 z='$';
 s=c;
 w=false;
 }
 public void display()
 {
 System.out.println(x);
 System.out.println(y);
 System.out.println(w);
 System.out.println(s);
 System.out.println(b);
 }
}
Main method class
package java1;
public class DerivedMain
{
 public static void main(String[] args)
 {
 Deiveddata1 obj=new Deiveddata1(10, "mahesh");
 obj.display();
 }
}
```

**obj****Memory allocation for data members**Obj.x  
56Obj.y  
45.6Obj.z  
\$Obj.w  
trueObj.s  
mahesh

4 bytes

4 bytes

2 bytes

1 bit

12 bytes

⇒ Constructor methods can support polymorphism also(Multiple constructors)

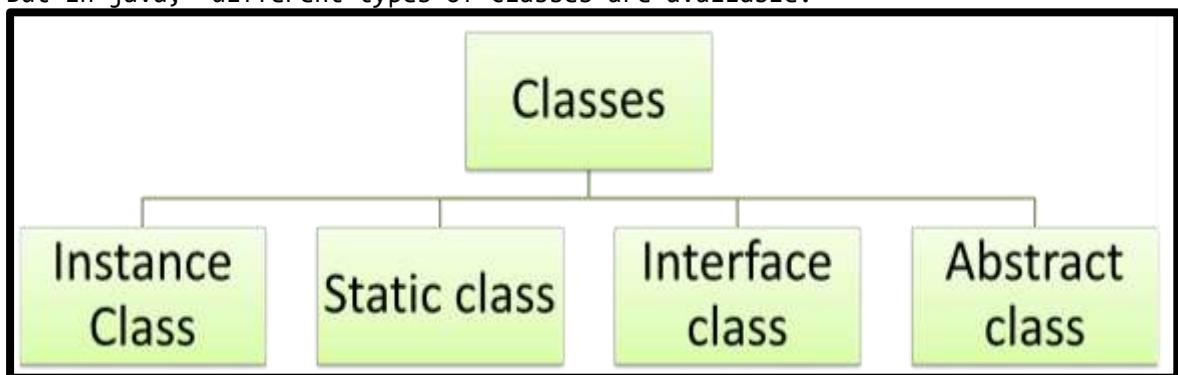
```
package java1;
public class Deiveddata1
{
 int x;
 float y;
 char w;
 String s;
 boolean b;
 // Constructor
 Deiveddata1()
 {
 x=98;
 y= (float) 45.6;
 w='$';
 s="Nageswar";
 b=false;
 }
 Deiveddata1(int a, String c)
 {
 x=a;
 y= (float) 45.6;
 w='$';
 s=c;
 b=false;
 }
 Deiveddata1(int a, String float d)
 {
 x=a;
 y= d;
 w='$';
 s=c;
 b=false;
 }
 Deiveddata1(int a, char float f)
 {
 x=a;
 y= f;
 w=c;
 s="Kalayan";
 b=false;
 }
}
```

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

Main method class
package java1;
public class DerivedMain
{
 public static void main(String[] args)
 {
 Deiveddata1 obj=new Deiveddata1(10, "mahesh");
 obj.display();
 Deiveddata1 obj1=new Deiveddata1(10, "kalayan", (float) 89.96);
 obj1.display();
 Deiveddata1 obj2=new Deiveddata1(10, '!', (float) 45.85);
 obj2.display();
 Deiveddata1 obj3=new Deiveddata1();
 obj3.display();
 }
}
```

## F. Types of classes

In general, user defined data types are classes. Every class is having data members. In methods some are constructor & some are operations methods. But in java, different types of classes are available.



### a) Instance classes :-

These classes are accessible using object. We are able to create multiple objects to one class if required.

Example:

Sikuli, Action ....Etc.

### b) Static classes :-

If any class have all data members & methods as static then that class is called as static class.

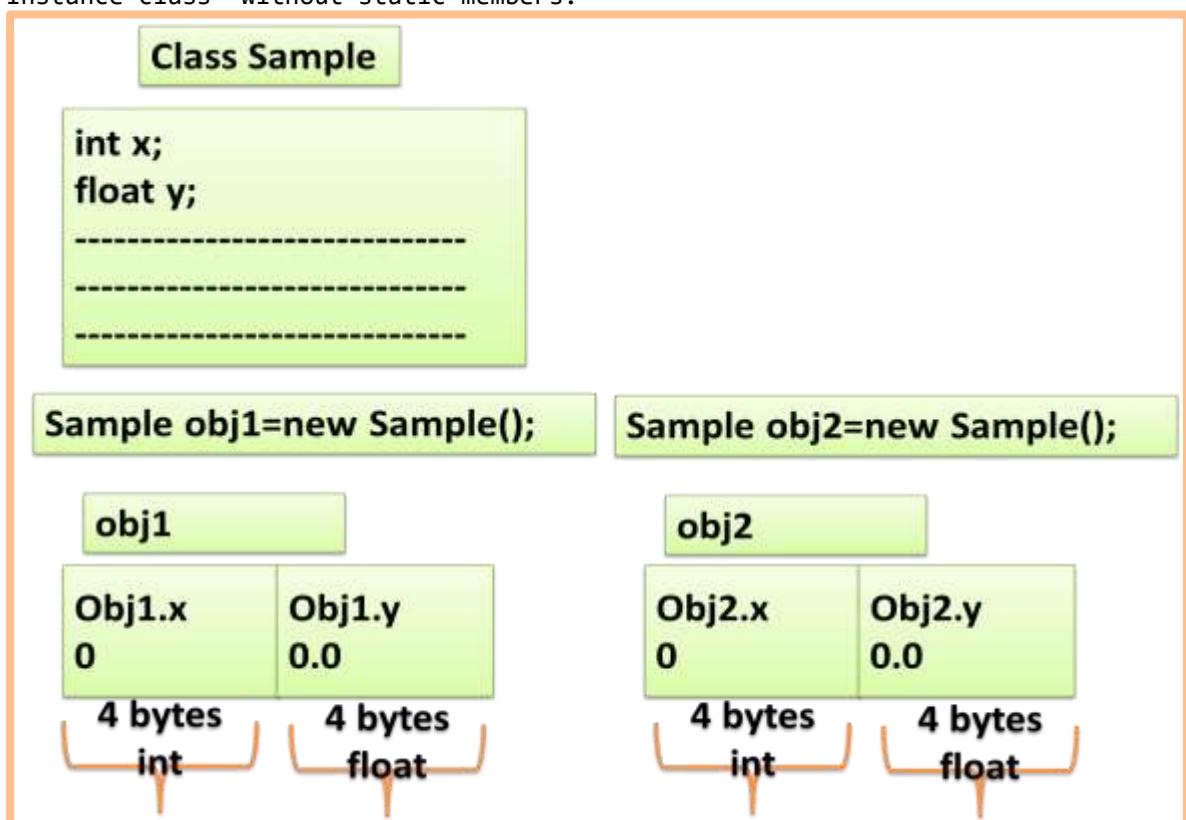
Example:

⇒ "BY" in Selenium WebDriver,  
⇒ "Keys" in Selenium WebDriver.

In general static class members are accessible by using class name instead of objects.

#### Example 1:

Instance class without static members.



#### Example 2:

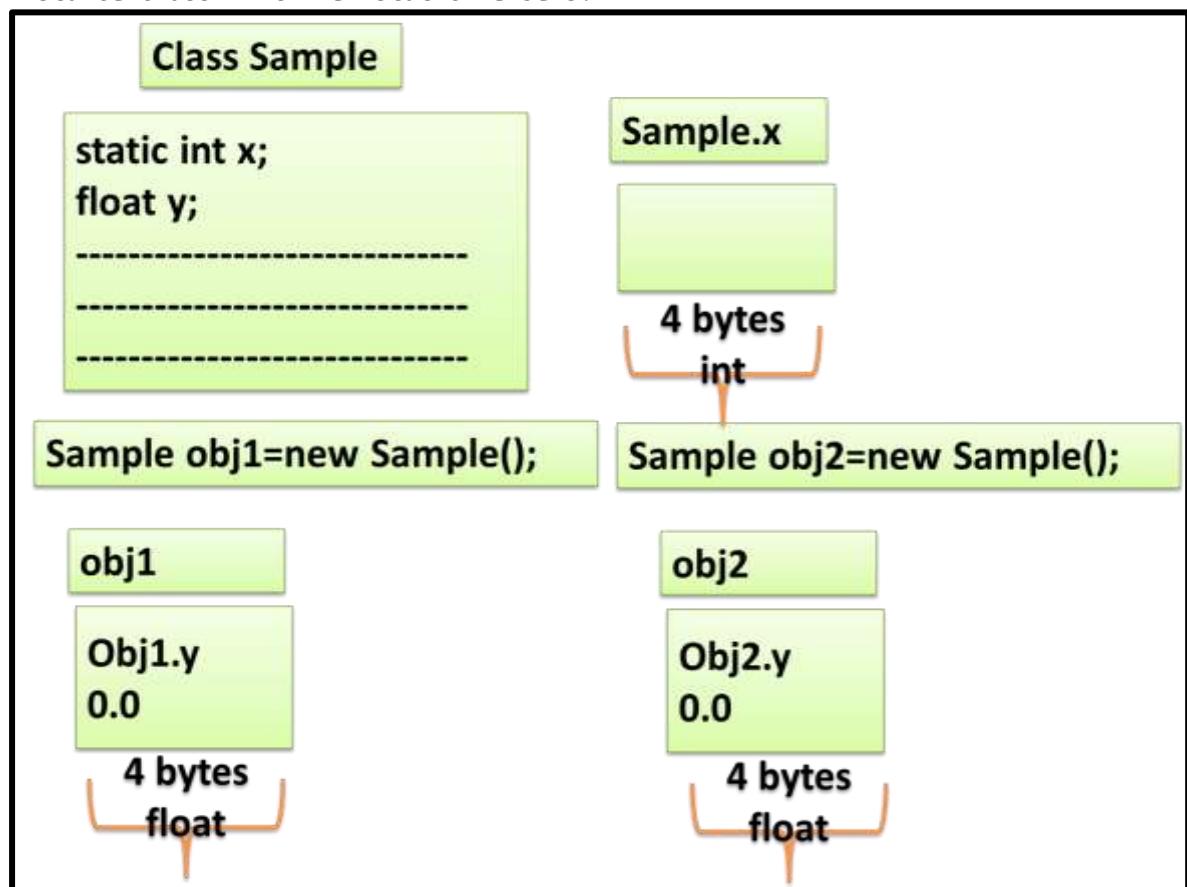
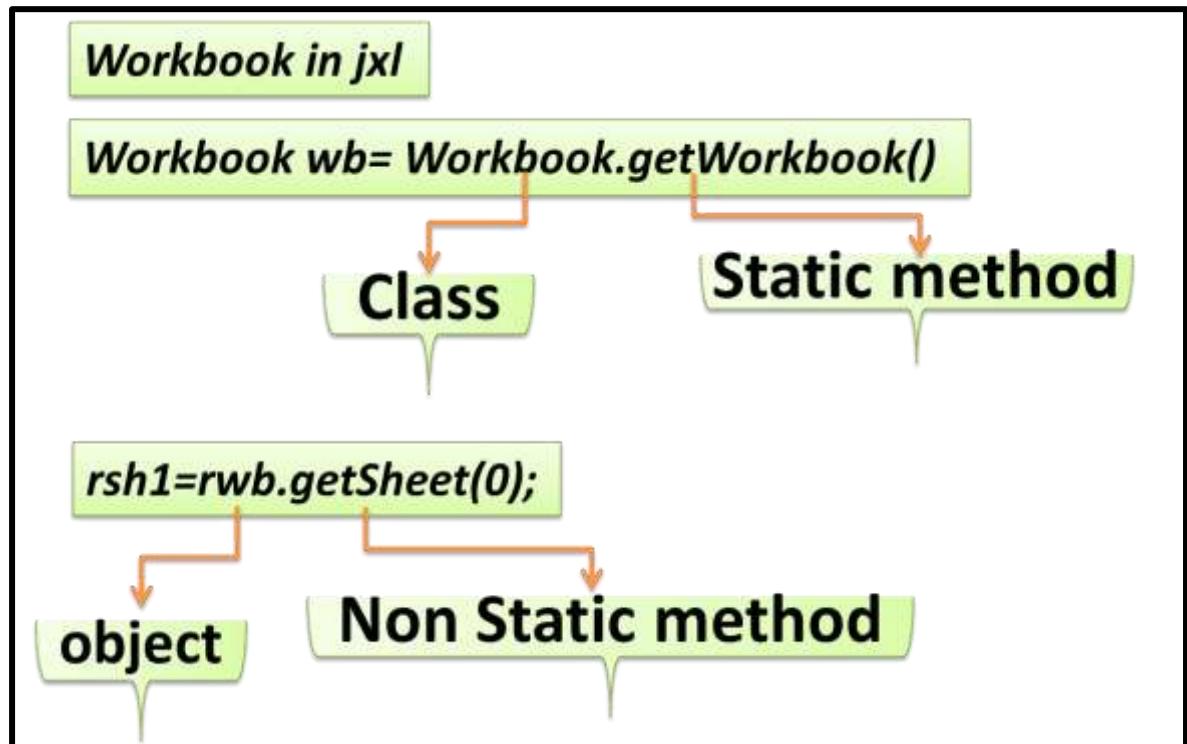
"Screens" in Sikulix".

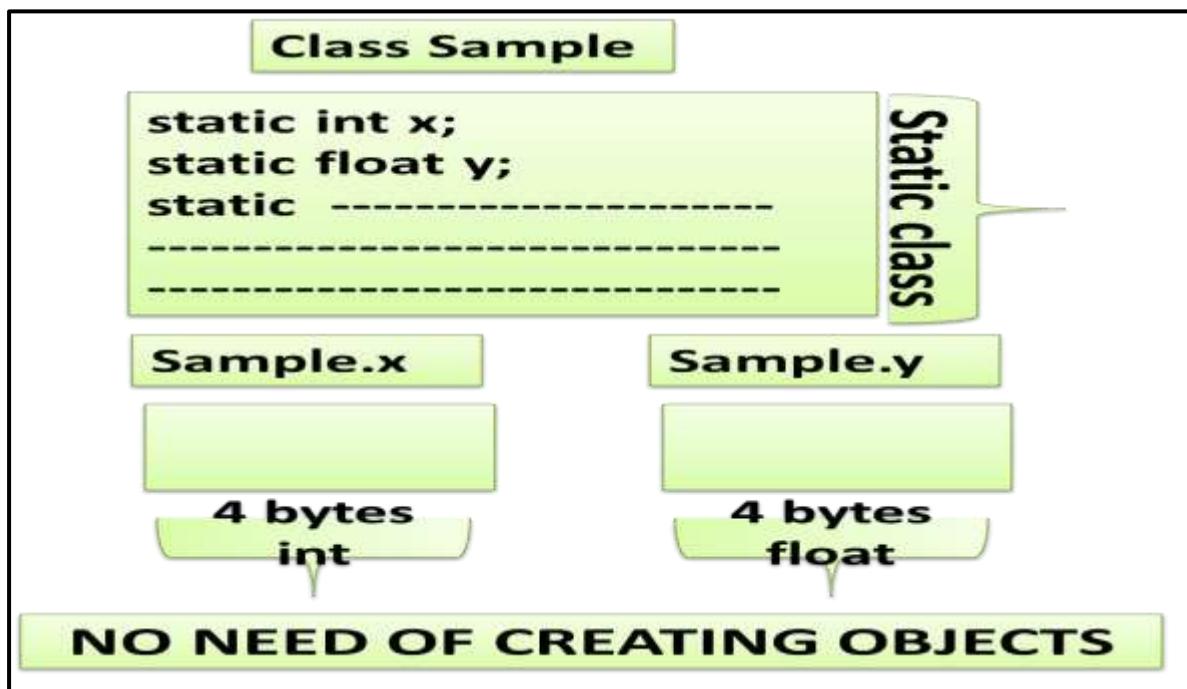
"Actions" in Selenium WebDriver.

"ChromeDriver" in Selenium WebDriver.

**Example 3:**

Instance class with few static members.

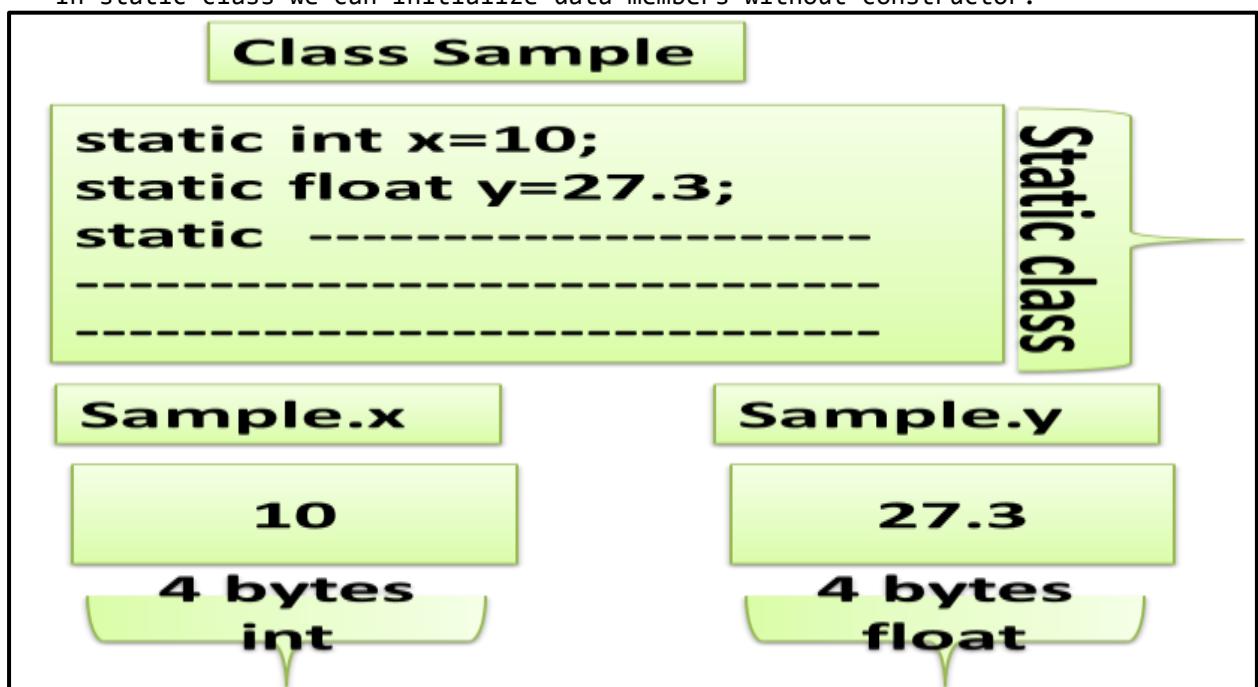
**Example4:-**

Example4:-Example

- ⇒ “BY” in Selenium webdriver.
- ⇒ “Keys” in Selenium webdriver.
- ⇒ “TimeUnit” in Selenium webdriver.
- ⇒ “Thread” in java.
- ⇒ “System” in java.
- ⇒ “Key” in Sikulix.

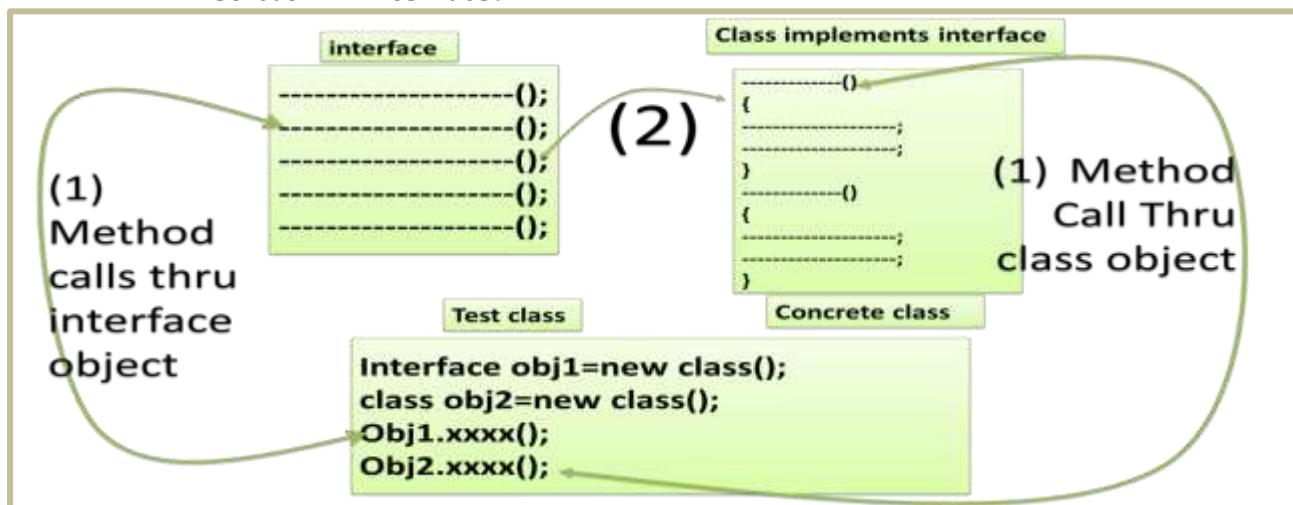
In instance class we can use constructor methods to assign values to data members while creating object.

In static class we can initialize data members without constructor.



### c) Interface classes :-

Interface is also class, but it is having methods declarations only instead of data members & data methods with bodies. Java programs can use another class to define/write bodies to those declarations methods in interface.



#### Example 1:-

##### Interface class

##### Test script

```
package interfacepack;
public interface myinterface
{
 public void mindq();
}
```

##### Concrete Class

```
package interfacepack;
public class myclass implements myinterface
{
 public void mindq()
 {
 System.out.println("hi");
 }
}
```

##### Test Class

```
package interfacepack;
public class Mainclass
{
 public static void main(String[] args)
 {
 myinterface obj1=new myclass();
 obj1.mindq(); // call method using interface's object
 myclass obj2=new myclass();
 obj2.mindq(); //call method using class's object
 }
}
```

#### Example of interface classes

“WebDriver” is interface in Selenium WebDriver jars.

“ChromeDriver”, “SafariDriver”, “FirefoxDriver”, “OperaDriver”, “InternetExploreDriver”, “RemoteDriver” (for remote execution), “AndroidDriver”, “IOSDriver” are concrete classes in selenium WebDriver Jars.

### d) Abstract classes :-

Abstract class having declarations for some methods & having declarations with bodies for other methods.

Example:-

```
Abstract Class example
package abstractpack;
public abstract class Myabstract
{
 public abstract void method1(); //Declaration without
body
 public void method2() //Declaration with body
 {
 System.out.println("Bye");
 }
}
```

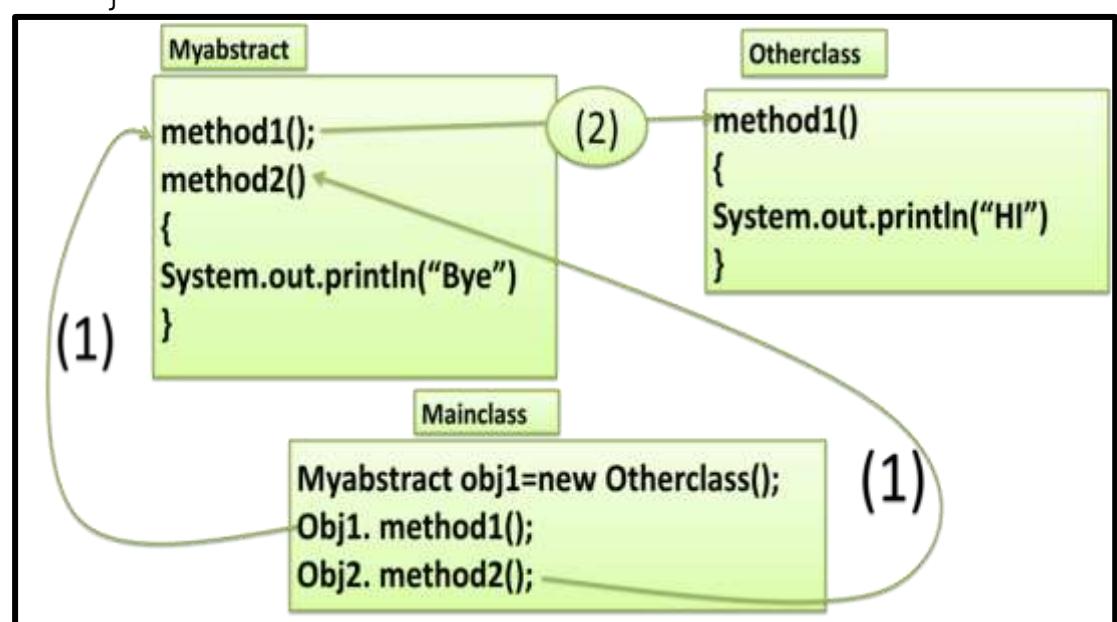
Here we need to concrete class to implements unimplemented methods in abstract class

Other Class

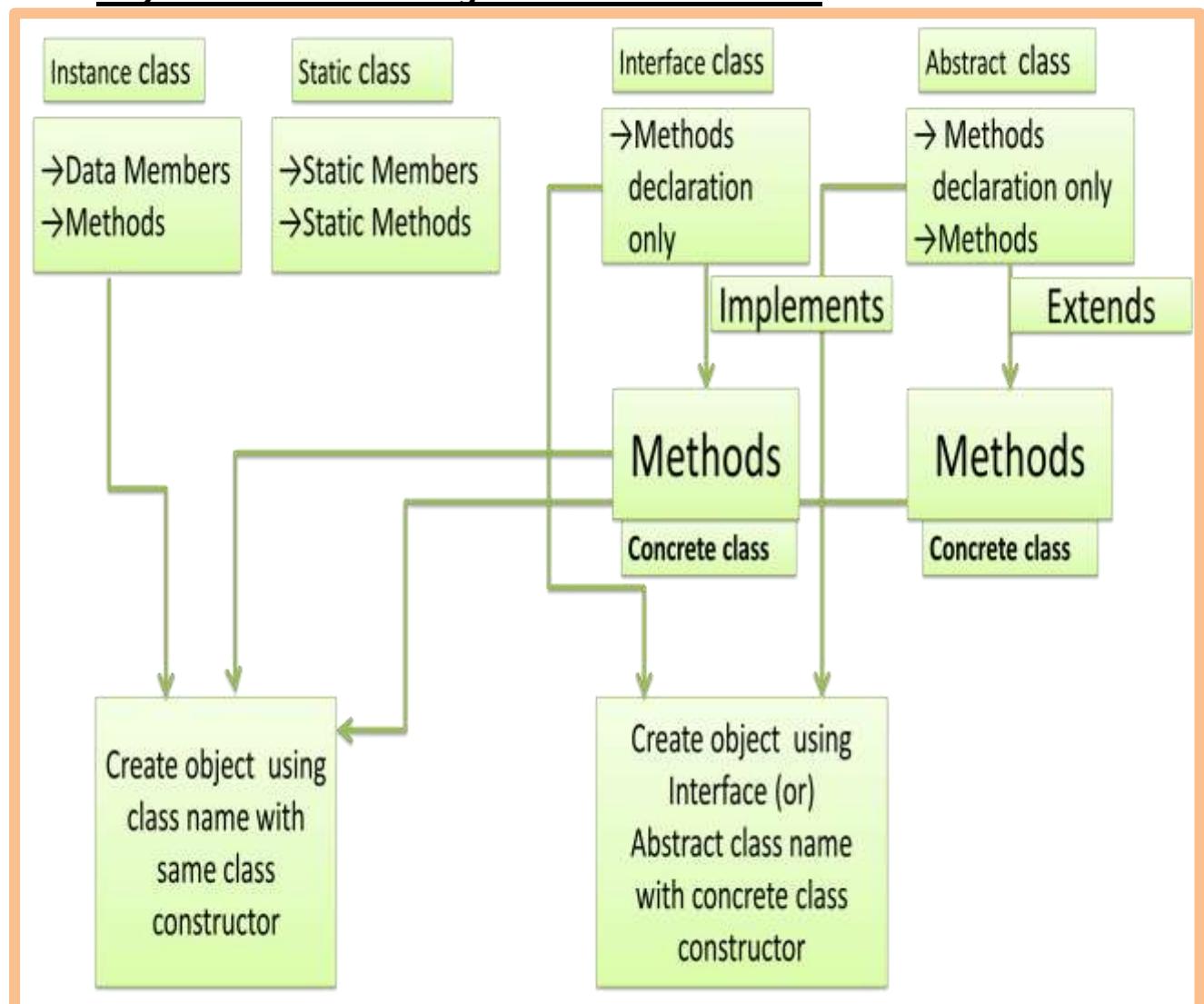
```
package abstractpack;
public class OtherClass extends Myabstract
{
 public void method1()
 {
 System.out.println("Hi");
 }
}
```

Main Class

```
package abstractpack;
public class MainClass {
 public static void main(String[] args) {
 Myabstract obj=new OtherClass();
 {
 obj.method1();
 obj.method2();
 }
 }
}
```



## G.Ways to create object to Classes:-



`Screen s = new screen();`

Instance Class/  
concrete class      object      Memory  
allocation      Constructor

`WebDriver driver = new ChromeDriver();`

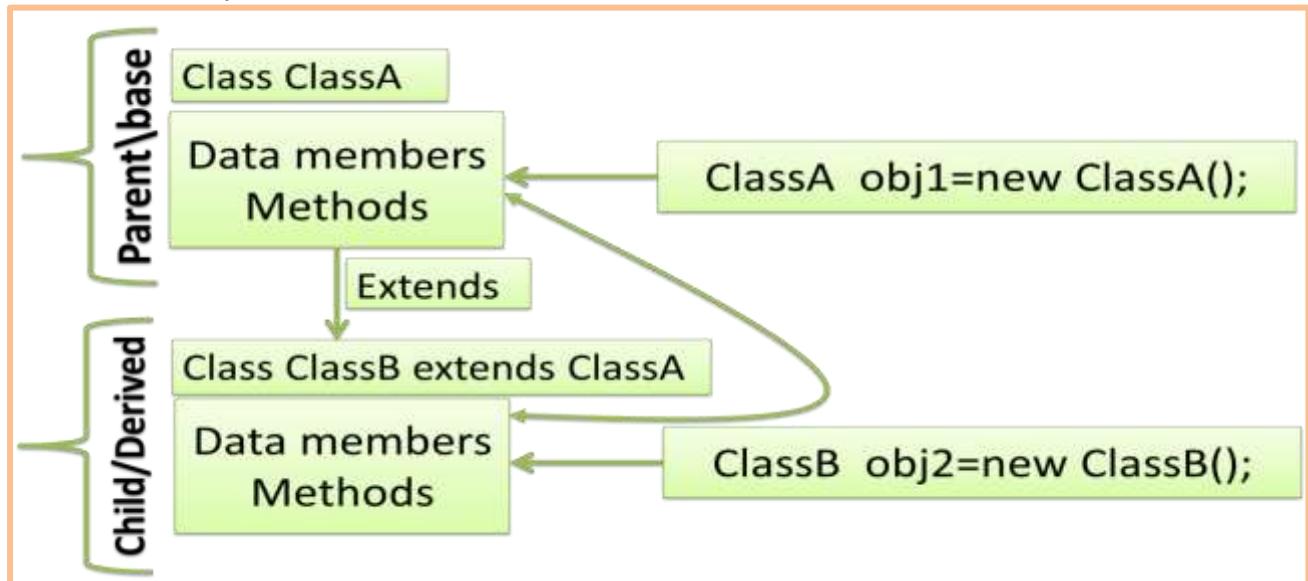
Interface Class/  
abstract class      Object      Memory  
allocation      Constructor of corresponding  
concrete class

## H. Inheritance in JAVA(is a relation)

- Like all object oriented programming languages, java can support inheritance in between instance classes.
- Inheritance is a relation & it is useful to create “is a relation” & “has a relation”.
- **“is a” relation:-** This inheritance is possible in 4 ways, Such as
  - ⇒ **Single inheritance**
  - ⇒ **Multiple inheritance**
  - ⇒ **Multilevel inheritance**
  - ⇒ **Hybrid inheritance**

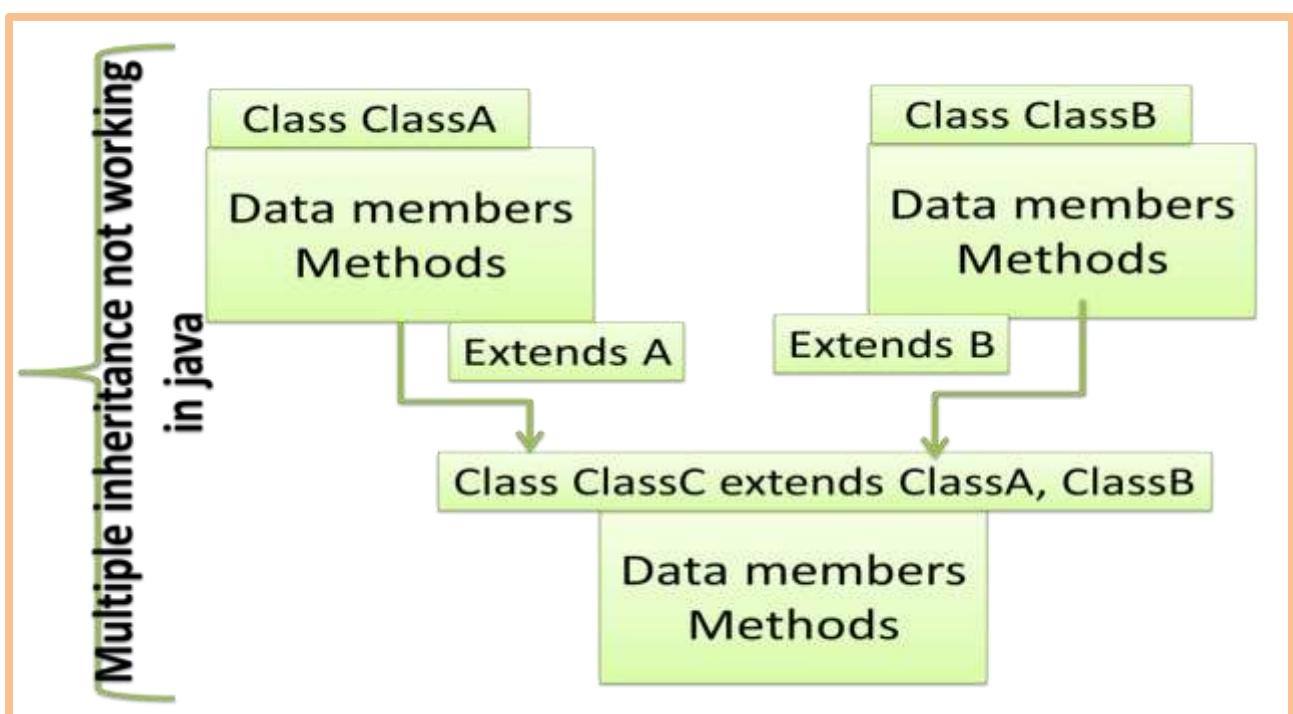
### a) Single inheritance:-

In single inheritance we can take one class as parent & another class as child, like shown below



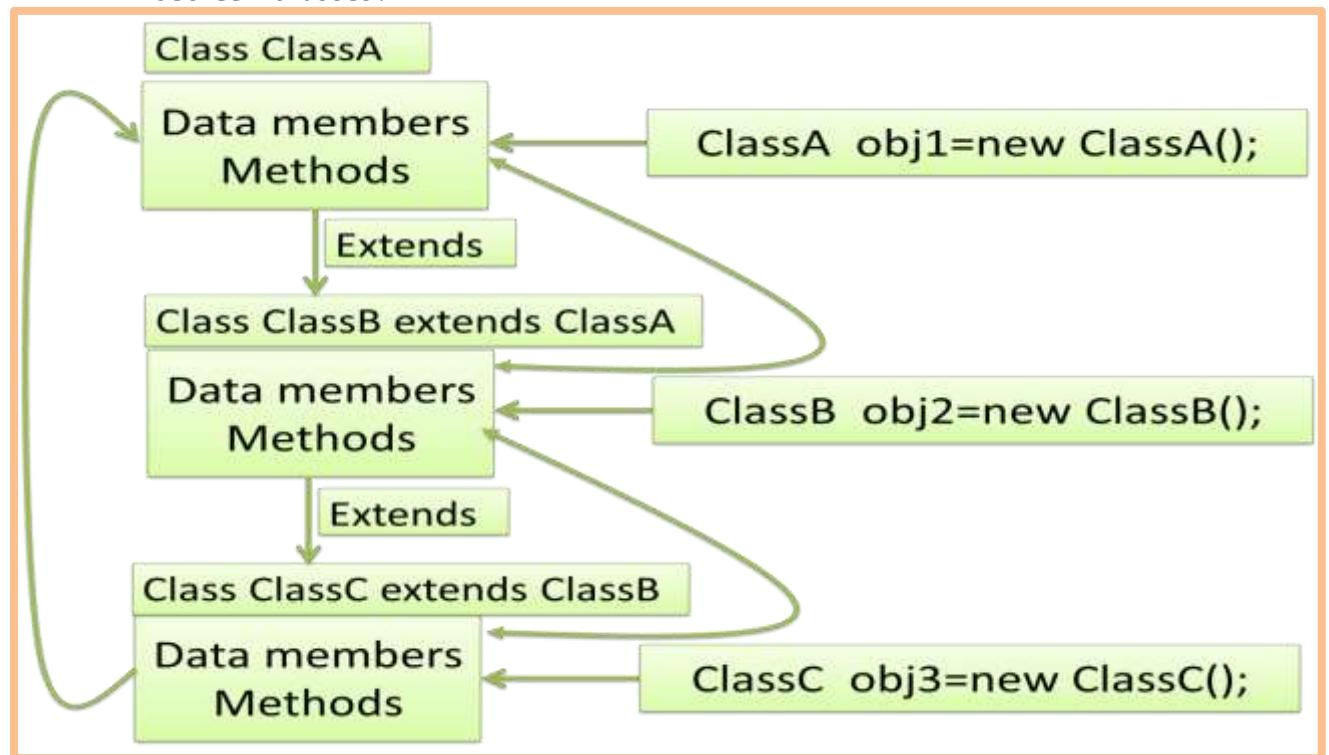
### b) Multiple inheritance:-

In multiple inheritance, we can take more than one class as parent for one child, but these concept not working in java.



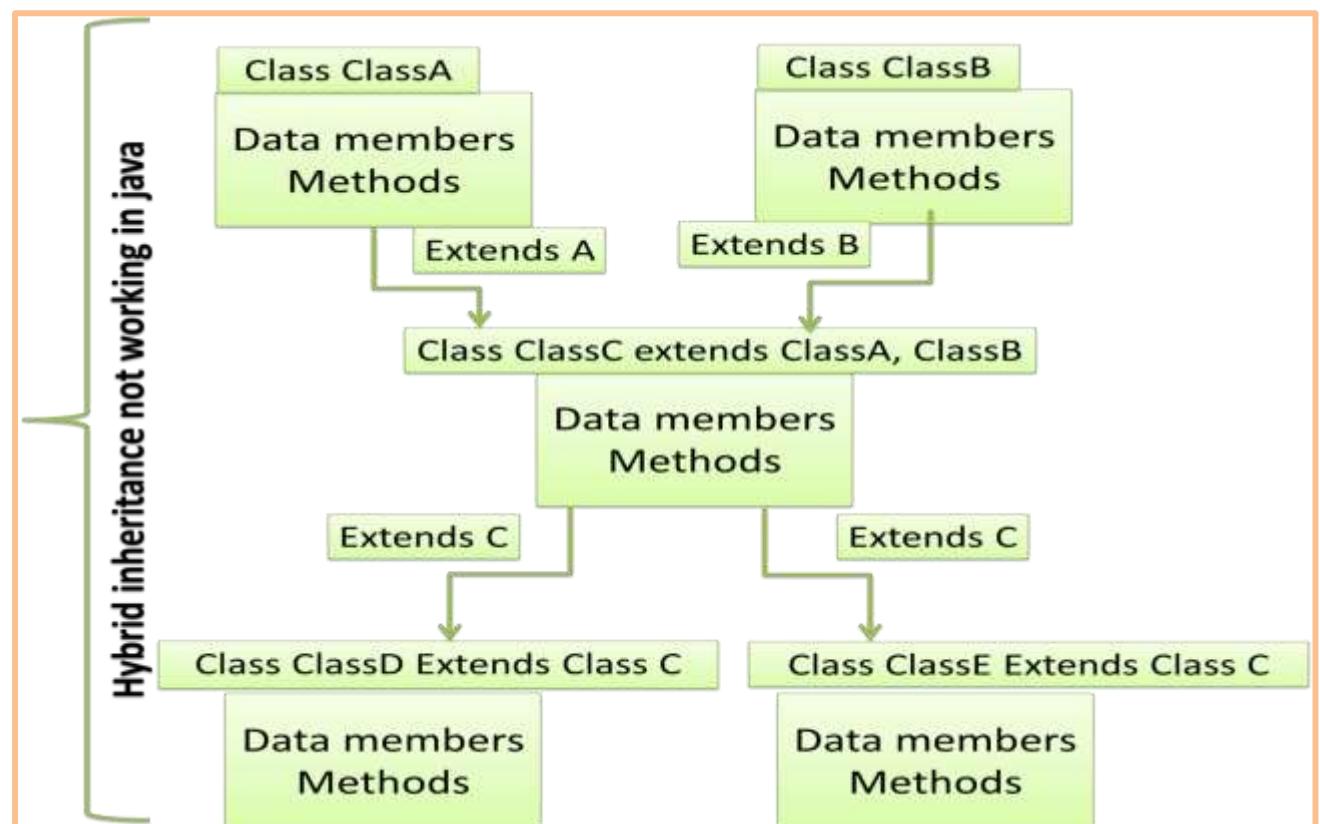
### c) Multi-level inheritance:-

In multi-level inheritance, we can take multi-level's relations in between classes.



### d) Hybrid inheritance:-

In hybrid inheritance is a combination of multiple inheritance & multi-level inheritance, but it is not working in java.



**e) "has a" relation:-**

From this relation, one class is having another class as a member.

```
//one class
package hasarelation;
public class ClassA
{
 public void method1()
 {
 System.out.println("hi");
 }
}

// other class
package hasarelation;
public class ClassB
{
 public ClassA obj;
 public void method2()
 {
 System.out.println("bye");
 }
}
// Main class
package hasarelation;
public class Mainprg
{
 public static void main(String[] args)
 {
 ClassB x=new ClassB();
 x.obj.method1();
 x.method2();
 }
}
```

## I. Operators in JAVA

Like in all programming languages, we can get different types of operators, in java language, such as unary operator, binary operator and ternary operator.

### a) Unary Operator

"-","++","--" are unary operator in java language

|                  |                                       |                    |                     |
|------------------|---------------------------------------|--------------------|---------------------|
| <b>Example 1</b> | <code>int x=10;<br/>int y=-x;</code>  | X<br><br><b>10</b> | Y<br><br><b>-10</b> |
| <b>Example 2</b> | <code>int x=10;<br/>int y=x++;</code> | X<br><br><b>11</b> | Y<br><br><b>10</b>  |
| <b>Example 3</b> | <code>int x=10;<br/>int y=++x;</code> | X<br><br><b>11</b> | Y<br><br><b>11</b>  |
| <b>Example 4</b> | <code>int x=10;<br/>int y=x--;</code> | X<br><br><b>9</b>  | Y<br><br><b>10</b>  |
| <b>Example 5</b> | <code>int x=10;<br/>int y=--x;</code> | X<br><br><b>9</b>  | Y<br><br><b>9</b>   |

### b) Binary Operator

These operators are working with two operands (data/variables)

`+, -, *, /, %` These five are used for arithmetic operations

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

`&&, ||, !` Used to combine more than 1 condition

`=` Used to assign value to variable

#### Example

`10+2=12.`

`10-2=8.`

`10*2=20.`

`10%2=0.`

`10/2=5.`

`“MindQ”+10=”mindq10”.`

`X=10` i.e. assign value to X.

`X==y` i.e. x value is comparison to y value.

### c) Ternary Operator

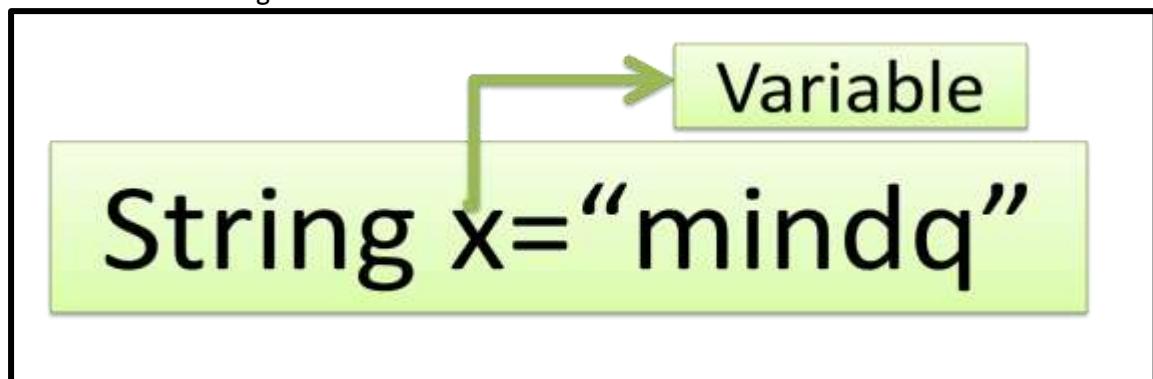
This operand can work with "3" operands.

`Z=(x>y)?x:y;`

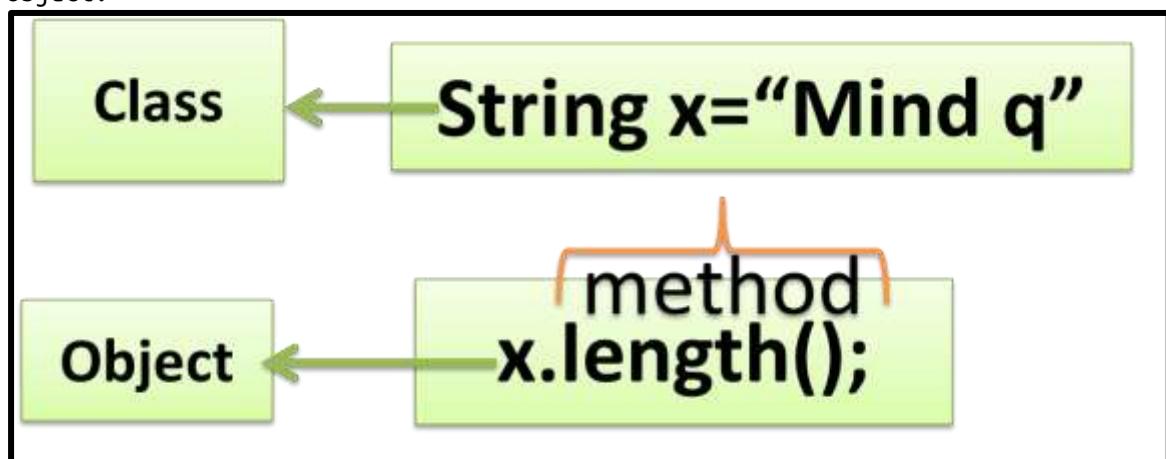
If condition is true x value is assign to z  
If condition is false y value is assign to z

### J. “String” Class methods:-

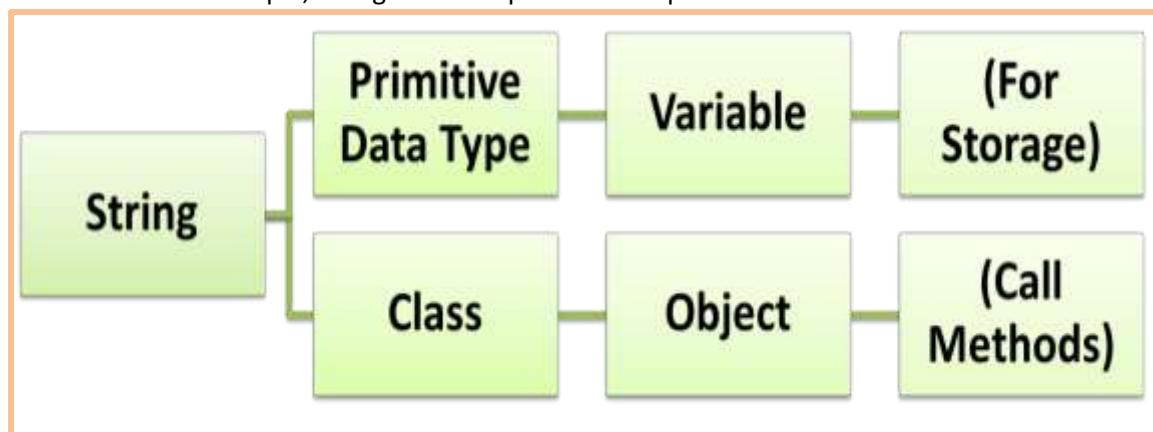
String is a derived data type in java. These data type is useful to create variable to storage.



In Java language, char s value enclosed by single quote & String value is enclosed by double quotes “String” is also working as class to create object.



From the above example, String Class can provide multiple methods like described below.



#### a) length( ):-

We can use this method to find the number of characters or positions in given string.

##### Syntax

```

String x= "mindq"
int y = x.length();
System.out.println(y);

```

Output:- 5

**b) equals( ):-**

We can use this method to compare two strings.

**Syntax**

```
String x= "mindq";
String y = "Mindq";
if(x.equals(y)) {
 System.out.println("same");
}
else
{
 System.out.println("Not same");
}
```

Output:- Not same

**c) contains( ):-**

We can use this method to check the availability of sub-string in main string.

**Syntax**

```
String x = "I am sleeping in mindq";
String y = "sleeping";
if(x.contains(y))
{
 System.out.println("Available");
}
else
{
 System.out.println("Not available");
}
```

Output:- Available

**d) equalsIgnoreCase ( ):-**

We can use this method to compare two strings without considering case.

**Syntax**

```
String x= "mindq";
String y = "Mindq";
if(x.equalsIgnoreCase(y))
{
 System.out.println("same");
}
else
{
 System.out.println("Not same");
}
```

Output:- same

**e) toLowerCase( ):-**

We can use this method to convert data or string to lowercase.

**Syntax**

```
String x = "INDIA";
String y= x.toLowerCase();
System.out.println(y);
Output:- india
```

**f) toUpperCase( ):-**

We can use this method to convert data or string to uppercase.

**Syntax**

```
String x = "INDIA";
String y = x.toUpperCase();
System.out.println(y);
Output:- INDIA
```

**g) charAt( ):-**

We can use this method to get a character from given string by specifying position, here position index start with 0(Zero).

**Example 1:-**

```
String x="INDIAN";
char y=x.charAt(0);
System.out.println(y);
```

**Output:-** I**Example2:-**

```
String x="INDIAN";
int l=x.length();
char y=x.charAt(l-1);
System.out.println(y);
```

**Output:-** N**Example 3:-**

Reverse of a Given String

```
public static void main(String[] args)
{
 String x="racecar";
 String y="";
 int l=x.length();
 for (int i=l-1;i>=0;i--)
 {
 char c=x.charAt(i);
 y=y+c;
 }
 System.out.println(y);
 if(x.equals(y))
 {
 System.out.println("palindrome");
 }
 else
 {
 System.out.println("not palindrome");
 }
}
```

**Output:-**

```
racecar
palindrome
```

**h) split( ):-**

We can use this method, to divide given string into pieces.

**Example 1:-**

```
String x="My name is Khan";
String y[] = x.split(" ");
for(String w:y)
{
 System.out.println(w);
}
```

**Output:-**

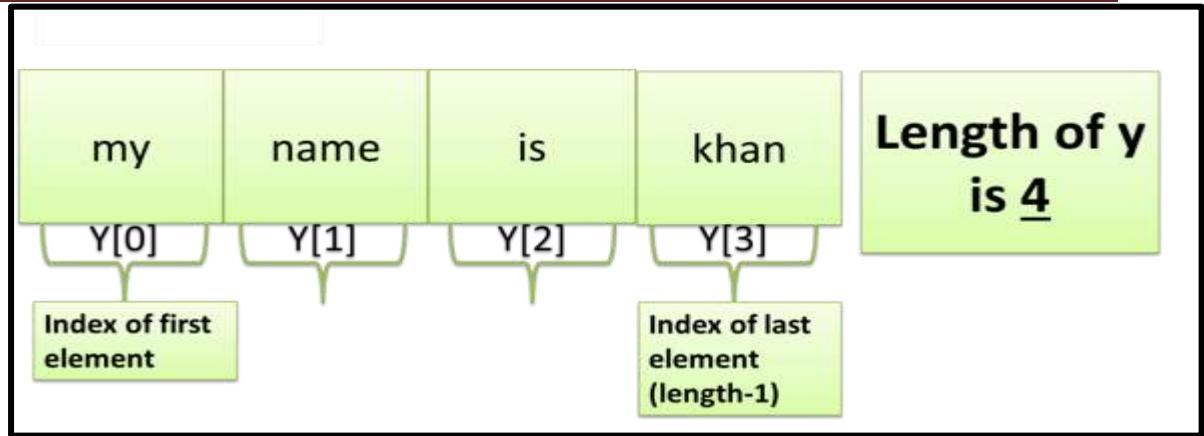
```
hi
```

```
My
```

```
name
```

```
is
```

```
Khan
```

**Note:**

array index start with 0,to get last element we need to write length-1

**Example2:-**

Find number of occurrences of "india" given.

```
String x = "My name is india in india";
String y[] = x.split(" ");
int c = 0;
for (int i = 0; i < y.length; i++)
{
 if (y[i].equalsIgnoreCase("india"))
 {
 c = c + 1;
 }
}
System.out.println(c);
```

**OUTPUT:**

2

**i) trim():-**

We can use this method to remove space at front and end of given string.

**Example:-**

```
String x = " My name is india in india ";
```

```
String y = x.trim();
```

```
System.out.println(y);
```

**OUTPUT:**

My name is india in india

**j) replace():-**

We can use this method to replace one character with another character in given string.

**Syntax 1:**

```
String x=" I am in India ";
String y=x.trim();
String z=y.replace(" ", "");
```

```
System.out.println(z);
```

**Output**

IaminIndia

**Syntax 2:**

```
String x="malayalam";
String z=x.replace("a ","*");
System.out.println(z);
```

**Output**

m\*l\*y\*l\*m

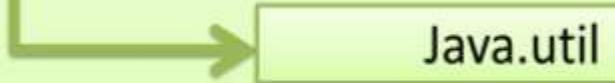
## K. Read data from keyboard:-

To read data from keyboard while running code, we can use below statements in that code



JDK insistence class

Scanner sc = new Scanner(System.in);



Java.util

String x = sc.nextLine();

Used to get string data from keyboard



Scanner sc = new Scanner(System.in);

String x = sc.nextLine();

int y=Integer.parseInt(x);

Type  
conversion

Primitive  
data type

Derived  
data type

OR

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



```
Scanner sc = new Scanner(System.in);
String x = sc.nextLine();
float y=Float.parseFloat(x); → Type
convesation
```

Primitive  
data type

Derived  
data type

OR

```
Scanner sc = new Scanner(System.in);
float x=sc.nextFloat();
```



```
Scanner sc = new Scanner(System.in);
String x = sc.nextLine();
String x = sc.nextLine();
```

```
Driver.findElement(By.xpath(x).sendKeys(y);
```

locator

parameterization

data

parameterization

## L. Type casting, Type conversation & Java Generics

⇒ Java can support type casting, to consider one type data/ object as another type

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

```

⇒ Java can support type conversation, to consider one type data into another type

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

```

⇒ Java generics concepts are used to globalize multiple types.

```
List<WebElement> l=Driver.findElements(By.tagName("a"));

```

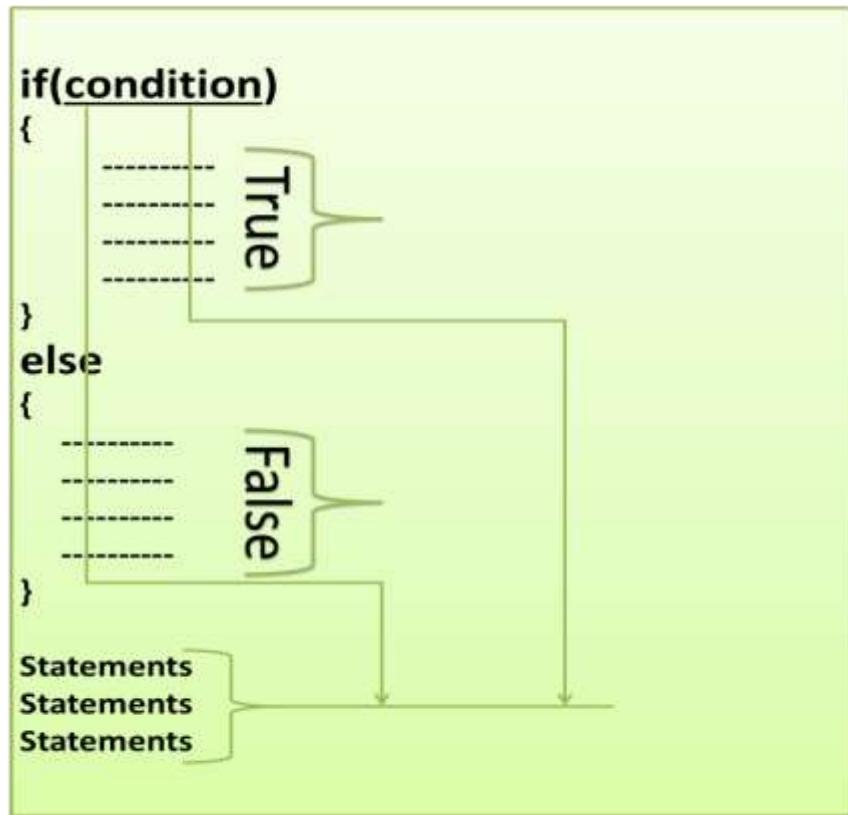
## M. Control statements in java

⇒ Like in all programming language, Java can provide different type of control statements like

- if - else
- if - else if - else
- nested if
- switch case

### a) if - else

if - else statement is used to execute a block of code depends on condition.



### Example 1

Take no from key board.

Check that number as even or odd.

#### Test Script:-

```

Scanner sc=new Scanner(System.in);
System.out.println("Enter a no");
int x=sc.nextInt();
if(x%2==0)
{
 System.out.println("Given number is Even");
}
else
{
 System.out.println("Given number is odd");
}
sc.close();

```

Output:-  
 Enter a no  
 98  
 Given number is Even

**Example 2**

```

⇒ Launch Gmail Site.
⇒ Check home page title by compare with Gmail.
⇒ Close site.
Test Script
//Launch Site
System.setProperty("webdriver.chrome.driver","d:\\ chromedriver.exe");
WebDriver driver=new ChromeDriver();
driver.get("http://www.gmail.com");
driver.manage().window().maximize();
Thread.sleep(5000);
String x=driver.getTitle();
if(x.equals("Gmail"))
{
 System.out.println("Title Test Was Passed");
}
else
{
 System.out.println("Title Test Was Failed");
}
//Close
driver.close();
Output:-
Title Test Was Passed

```

**Example 3**

- ⇒ Launch Google Site using chrome browser.
- ⇒ Enter a word & click Search.
- ⇒ Next page title contains search word title.
- ⇒ Close site.

**Test Script**

```

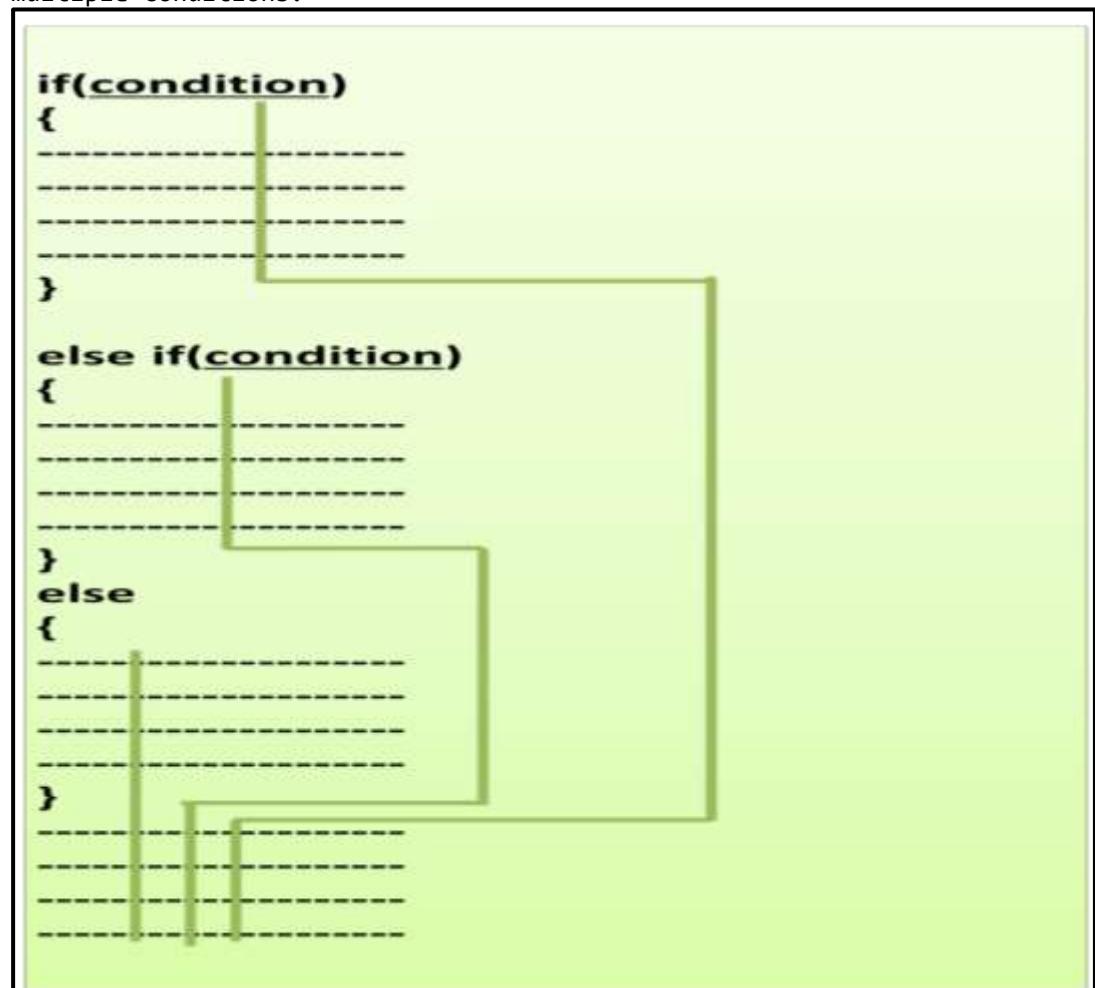
Scanner sc=new Scanner(System.in);
System.out.println("Enter a word");
String x=sc.nextLine();
//Launch Site
System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
WebDriver driver=new ChromeDriver();
driver.get("http://www.google.com");
Thread.sleep(5000);
//Search Word
driver.findElement(By.name("q")).sendKeys(x,Keys.ENTER);
driver.findElement(By.name("btnG")).click();
Thread.sleep(5000);
String t=driver.getTitle();
if(t.contains(x))
{
 System.out.println("Title Test Was Passed");
}
else
{
 System.out.println("Title Test Was Failed");
}
//Close
sc.close();
driver.close();
Output :-
Enter a word
Steve jobs
Starting ChromeDriver 2.29.461591
(62ebf098771772160f391d75e589dc567915b233) on port 30436
Only local connections are allowed.

```

```
Jul 07, 2017 7:00:32 PM org.openqa.selenium.remote.ProtocolHandshake
createSession
INFO: Detected dialect: OSS
Title Test Was Passed
```

### b) if - else if - else

we can use this concept to execute a block of code by checking multiple conditions.



From the syntax else block will be executed when all the conditions were false.

### Example 4

- ⇒ Launch way2sms Site using chrome browser.
- ⇒ Do login by entering credentials.
- ⇒ If mobile no is blank alert will be displayed.
- ⇒ If mobile is less than 10 digits alert will be displayed.
- ⇒ If mobile no is invalid “yet to register message” will be displayed.
- ⇒ If mobile no is valid & password is blank alert message will be displayed.
- ⇒ If mobile no is valid & password is invalid “Forgot password” will be displayed.
- ⇒ If mobile no is valid & password is valid “Send SMS” will be displayed.
- ⇒ Close site.

### Test Script

```

import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
public class Way2sms
{
```

```

public static void main(String[] args) throws Exception
{
 // Get a word to search from keyboard
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter the mob no");
 String m = sc.nextLine();
 System.out.println("Enter the mob no:criteria");
 String mc = sc.nextLine();
 System.out.println("Enter password");
 String p = sc.nextLine();
 System.out.println("Enter the password criteria");
 String pc = sc.nextLine();
 // launch Google site using chrome
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 WebDriver driver = new ChromeDriver();
 driver.get("http://site21.way2sms.com/content/index.html");
 Thread.sleep(5000);
 // Do login
 driver.findElement(By.name("username")).sendKeys(m);
 driver.findElement(By.name("password")).sendKeys(p);
 driver.findElement(By.id("loginBTN")).click();
 Thread.sleep(5000);
 // Testing
 if (m.equals("") && ExpectedConditions.alertIsPresent() != null)
 {
 System.out.println("Login Test Passed");
 }
 else if (m.length() < 10 && ExpectedConditions.alertIsPresent() != null)
 {
 System.out.println("Login Test Passed");
 }
 else if (p.equals("") && ExpectedConditions.alertIsPresent() != null)
 {
 System.out.println("Login Test Passed");
 }
 else if (mc.equalsIgnoreCase("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'registered
 yet')]")).isDisplayed())
 {
 System.out.println("Login Test Passed");
 }
 else if (mc.equalsIgnoreCase("valid") && pc.equalsIgnoreCase("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'Forgot
 Password')]")).isDisplayed())
 {
 System.out.println("login Test Passed");
 }
 else if (mc.equalsIgnoreCase("valid") && pc.equalsIgnoreCase("valid") &&
 driver.findElement(By.xpath("//*[text()='Send SMS']")).isDisplayed())
 {
 System.out.println("Login Test passed");
 }
 else
 {
 System.out.println("Login Test Failed");
 }
 // close site
 driver.close();
 sc.close();
}
Output
Enter the mob no
9502813512
Enter the mob no:criteria

```

```

valid
Enter password
Mahesh12
Enter the password criteria
valid
Starting ChromeDriver 2.29.461591
(62ebf098771772160f391d75e589dc567915b233) on port 3958
Only local connections are allowed.
Jul 07, 2017 8:29:17 PM org.openqa.selenium.remote.ProtocolHandshake
createSession
INFO: Detected dialect: OSS
Login Test passed

```

### c) Nested if:-

One “if” condition is inside of another “if” condition called as Nested if. (Dependency Cases).

```

if(condition)
{
 if(condition)
 {

 }

}

```

### Example 5

- ⇒ Launch Gmail Site using chrome browser.
- ⇒ Enter User ID is valid & Click Next.
  - Password page will be displayed.
  - Enter password & click next.
  - If password is valid
    - Compose page will be displayed.
  - If password is invalid
    - Error message will be displayed.
- ⇒ Enter User ID is invalid & Click Next.
  - Error message will be displayed.
- ⇒ Close site.

### Test Script

```

public static void main(String[] args) throws InterruptedException
{
// taking value from Key board
Scanner sc = new Scanner(System.in);
System.out.print("Enter User Id");
String u = sc.nextLine();
System.out.print("Enter User Id Criteria");
String uc = sc.nextLine();
String p = "", pc = "";
if (uc.equals("valid"))

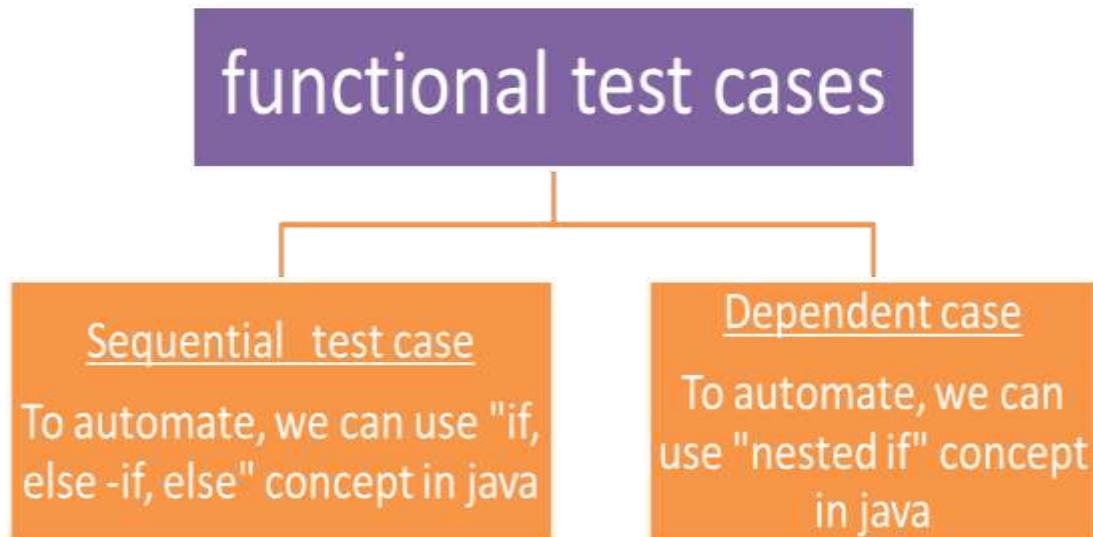
```

```
{
 System.out.print("Enter Password");
 p = sc.nextLine();
 System.out.print("Enter PasswordCriteria");
 pc = sc.nextLine();
}
// Launching Gmail
System.setProperty("webdriver.chrome.driver","d:\\mahesh\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("http://www.gmail.com/");
driver.manage().window().maximize();
driver.findElement(By.name("identifier")).sendKeys(u, Keys.ENTER);
Thread.sleep(5000);
// User Id Login Testing
try
{
 if (uc.equals("valid") &&
 driver.findElement(By.name("password")).isDisplayed())
 {
 System.out.println("User Id Test was Passed");
 // Password Testing
 driver.findElement(By.name("password")).sendKeys(p, Keys.ENTER);
 Thread.sleep(5000);
 if (pc.equals("valid") &&
 driver.findElement(By.xpath("//*[text()='COMPOSE']")).isDisplayed())
 {
 System.out.println("Password Test was Passed");
 }
 else if (pc.equals("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'Wrong password') or
 contains(text(),'Enter a password')])[2]").isDisplayed())
 {
 System.out.println("Password Test was Passed1");
 }
 else
 {
 System.out.println("Password Test was failed");
 }
 else if (uc.equals("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'Couldn') or
 contains(text(),'Enter an email')])[2]").isDisplayed())
 {
 System.out.println("User id Test was Passed2");
 }
 else
 {
 System.out.println("User Id Test was failed");
 }
}
}
catch(Exception e)
{
 System.out.println("Terminated");
}
// Close
driver.close();
sc.close();
}
```

**Note**

While automating functional test case selenium-Java, we can use "if conditions" in 2 ways

- ⇒ Sequential test cases
- ⇒ Dependent cases

**d) Switch case statement:-**

We can use this concept to execute a block of code, depends on given value instead of condition.

**Example 6**

```

public static void main(String[] args)
{
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter a no");
 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");
 }
 sc.close();
}

```

**Output**

Enter a no

4

Wednesday

In above synthetic example “break” statements is useful to terminate from current block execution.

“default” case is running when switch value doesn’t match with existing cases value.

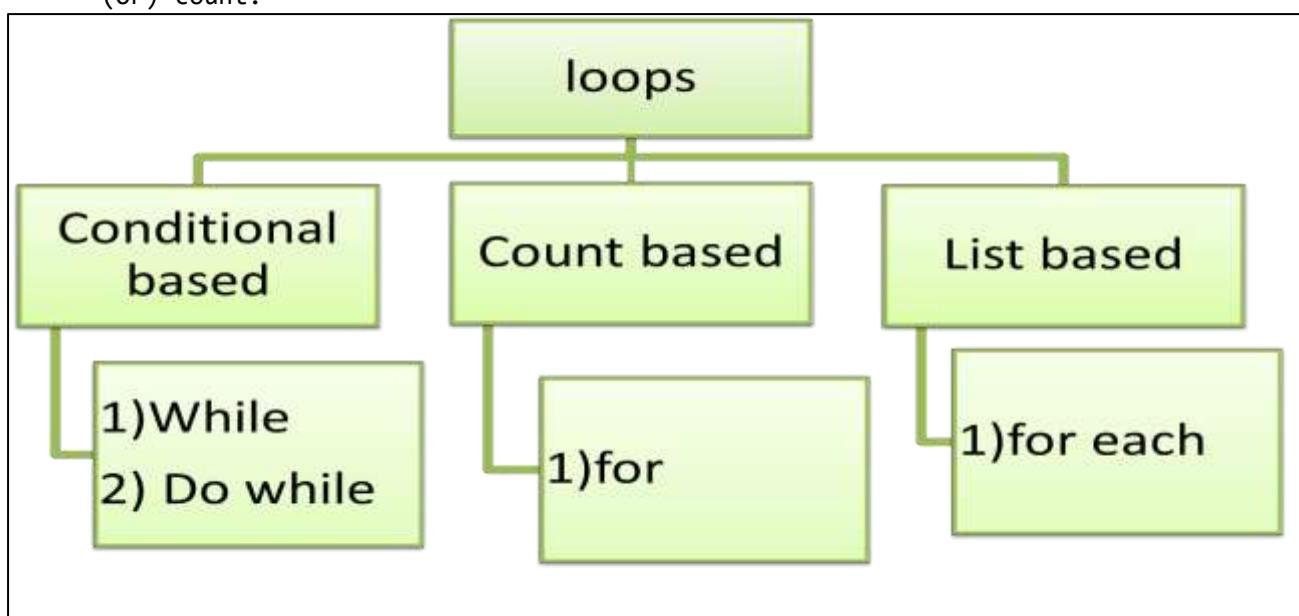
**Example 7**

```
public static void main(String[] args)
{
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter Credit Card Type");
 String cct = sc.nextLine();

 switch(cct)
 {
 case "visa":
 System.out.println("Valid");
 break;
 case "master":
 System.out.println("Valid");
 break;
 case "rupay":
 System.out.println("Valid");
 break;
 default :
 System.out.println("Wrong Day");
 }
 sc.close();
}
```

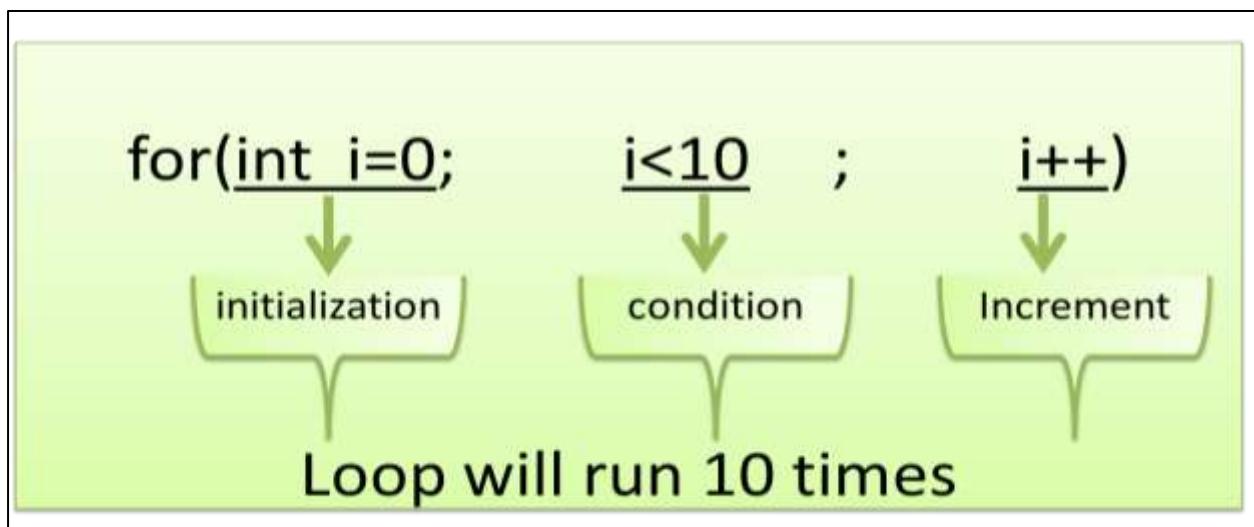
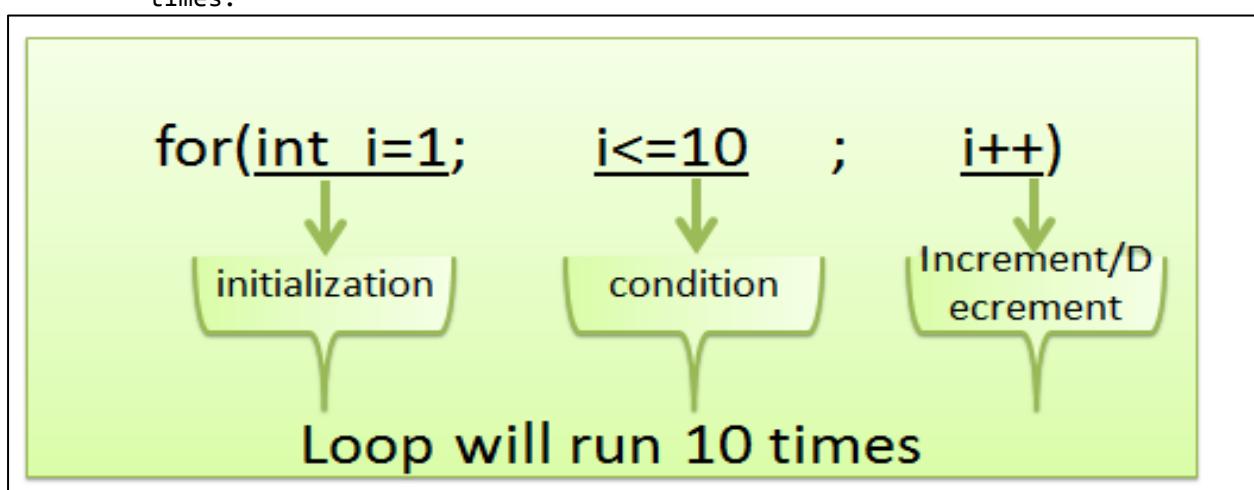
## N. Loops in java

We can use loops a block code for more than one time depends on a condition (Or) count.



### a) For Loop:-

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



**for(int i=10;      ;      )**

initialization

**i>0**

condition

**i--**)

Decrement

**Loop will run 10 times**

**for(int i=0;      ;      i++)**

initialization

**i++)**

increment

**If there is no condition in loop, Loop will run as infinite( $\infty$ )times**

```
for(; ;)
{

 if(Condition)
 {

 break;
 }

}
```

**Loop will break depends on internal condition**

### Example 1:-

Check given number is prime number or not (A number is divide number by 1 & itself)

### Test Script :-

```
public static void main(String[] args)
{
 Scanner sc=new Scanner(System.in);
 // Taking value from Key board
 System.out.println("Enter no of value");
 int x=sc.nextInt();
 int flag=0;
 for (int i=2;i<x;i++)
 {
```

```

 if(x%i==0)
 {
 flag=1;
 }
 if(flag==0)
 {
 System.out.println("Prime number :- "+ x);
 }
 else
 {
 System.out.println("Not Prime number :- "+x);
 }
 sc.close();
}

```

**Output**

Enter no of value

88

Not Prime number :- 88

**Example 2:-**

Reverse of a given String.

**Test Script :-**

```

public static void main(String[] args)
{
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter a String");
 String x=sc.nextLine();
 String y="";
 int l=x.length();
 for (int i=l-1;i>=0;i--)
 {
 char c=x.charAt(i);
 y=y+c;
 }
 System.out.println("Reverse of a Given String is :-"+y);
 sc.close();
}

```

**Output**

Enter a String

mahesh

Reverse of a Given String is :-hseha

**Example 3:-**

Display even numbers only up to given limit.

**Test Script :-**

```

public static void main(String[] args)
{
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter Limit");
 int l = sc.nextInt();
 System.out.println("No Of Even up to limit is");
 for(int i=0;i<=l;i++)
 {
 if (i % 2 == 0)
 {
 System.out.print(i + " ");
 }
 }
 sc.close();
}

```

**Output**

Enter Limit

```

38
No Of Even up to limit is
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38

```

**Example 4:-**

Display prime numbers only up to given limit.

**Test Script :-**

```

public static void main(String[] args)
{
 Scanner sc=new Scanner(System.in);
 // Taking value from Key board
 System.out.println("Enter no of prime numbers to limit");
 int l=sc.nextInt();
 System.out.println("Prime number up to limit:- ");
 for(int i=0;i<=l;i++)
 {
 int flag=0;
 for (int j=2;j<i;j++)
 {
 if(i%j==0)
 {
 flag=1;
 }
 }
 if(flag==0)
 {
 System.out.print(" "+ i);
 }
 }
 sc.close();
}

```

**Output**

Enter no of prime numbers to limit

100

Prime number up to limit:-

0 1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

**Example5**

```

*
**


```

Display above like output.

**Test Script**

```

public static void main(String[] args)
{
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter Limit");
 int l = sc.nextInt();
 for(int i=1;i<=l;i++)
 {
 for(int j=1;j<=i;j++)
 {
 System.out.print("*");
 }
 System.out.println();
 }
 sc.close();
}
```

**Example 6**

```

*

Display above like output.
Test Script
public static void main(String[] args)
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter a number");
int l=sc.nextInt();
for(int i=1;i<=l;i++)
{
 for (int j=i;j<=l;j++)
 {
 System.out.print(" ");
 }

 for(int k=1;k<(i*2);k++)
 {
 System.out.print("*");
 }
 System.out.println();
}
sc.close();
}

```

**Example 7**

Display Fibonacci Series

**Test Script**

```

public static void main(String[] args)
{
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter a limit");
 int l=sc.nextInt();
 int x=0;
 int y=1;
 int z=x+y;
 System.out.print(x+" "+y);
 for(int i=1;z<=l;i++)
 {
 System.out.print(z+" ");
 x=y;
 y=z;
 z=x+y;
 }
 sc.close();
}

```

**Output**

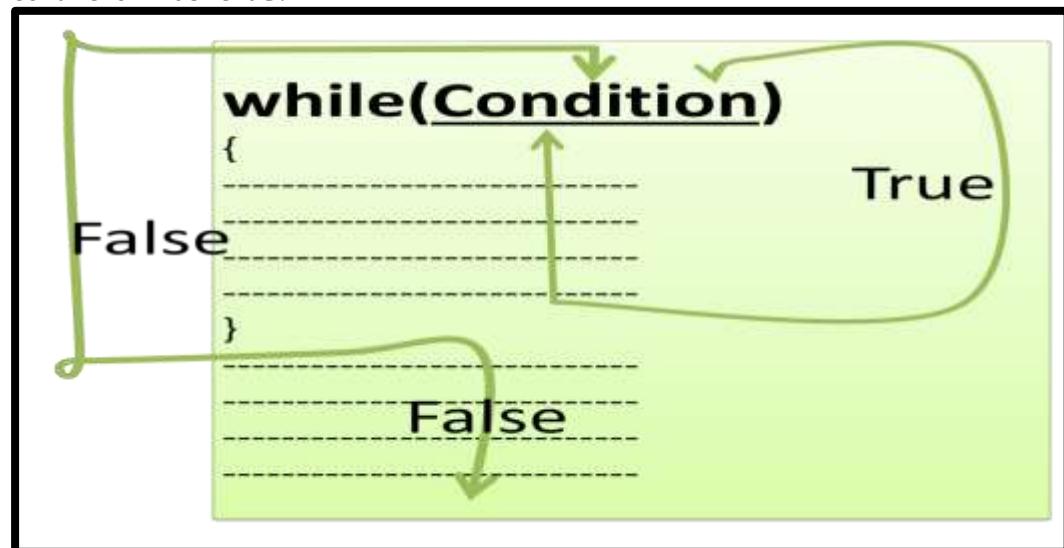
Enter a number

90

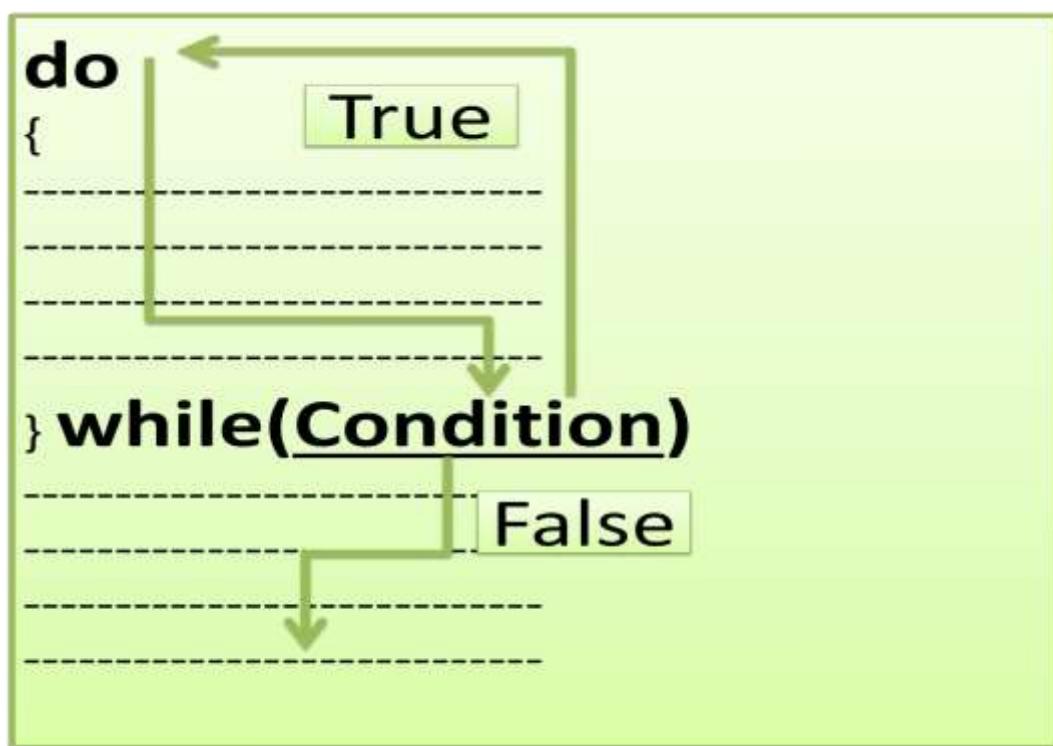
0 11 2 3 5 8 13 21 34 55 89

**b) while Loop:-**

We can use this loop to execute a block of code as long as given condition was true.

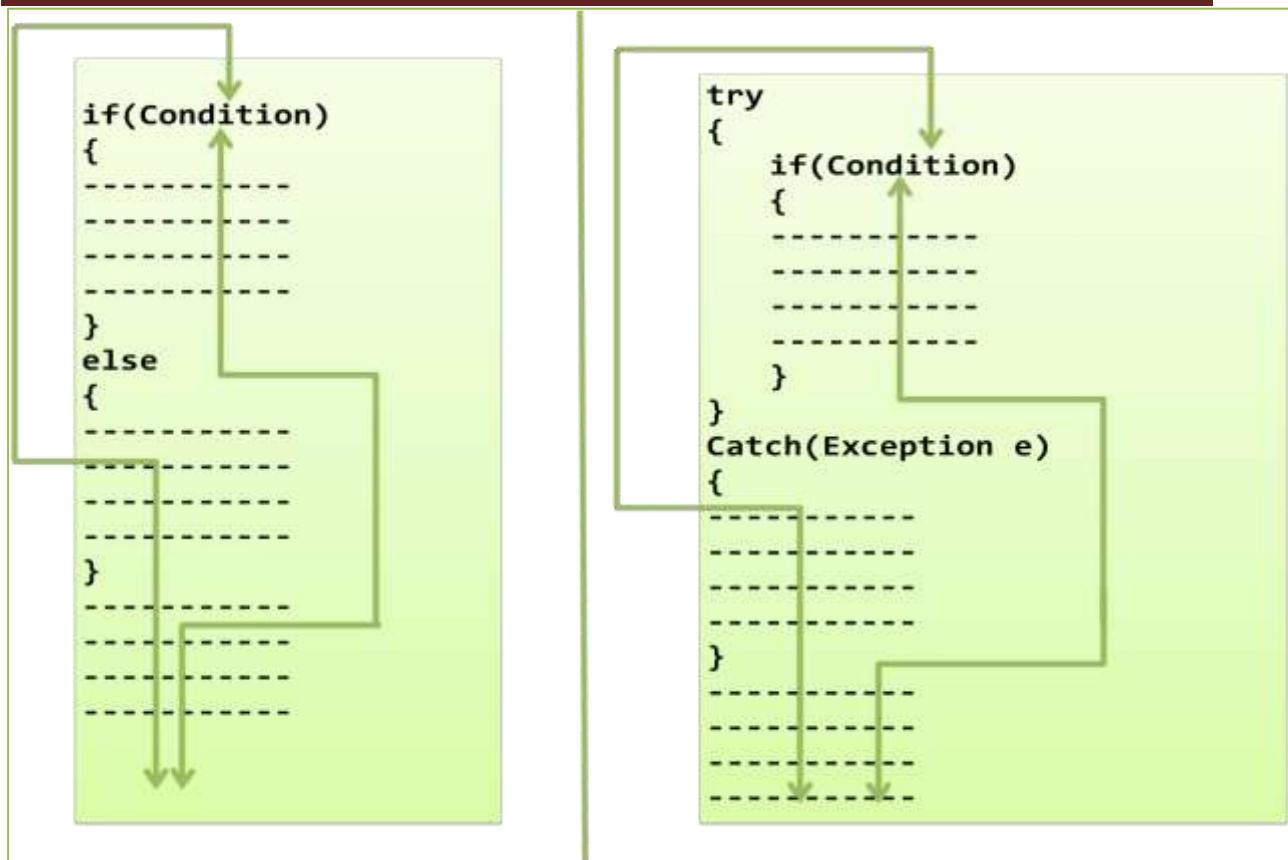
**c) do while Loop:-**

We can use this loop to execute a block code to execute for first time, it will check condition later. Here one time execution will be guaranteed.



\*\*\*  
**Note:-**

In java & java based technologies conditions are 2 types. Some conditions return `or false`, Some Conditions return `true`.  
In java language “exception class” represents any exception.

**Example 8:**

- ⇒ Launch Google site using chrome browser.
- ⇒ Enter a word to search in Google home page.
- ⇒ Count no of pages (related) with results.
- ⇒ Close site.

**Test scripts1:-**

```

package loops_1;
import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Pagination1
{
// pagination (Moving from one page to next page until last page)
public static void main(String[] args) throws Exception
{
// Get data from key board
Scanner sc=new Scanner(System.in);
System.out.println("Enter Search Key Word");
String str=sc.nextLine();
System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
WebDriver driver =new ChromeDriver();
driver.get("http://www.google.com");
driver.manage().window().maximize();
driver.findElement(By.name("q")).sendKeys(str);
driver.findElement(By.name("btnG")).click();
Thread.sleep(3000);
int c=1;
try
{
while(driver.findElement(By.xpath("//*[text()='Next']")).isDisplayed())
{
driver.findElement(By.xpath("//*[text()='Next']")).click();
}
}
}

```

```

 Thread.sleep(3000);
 c=c+1;
 }
}
catch(Exception e)
{
 System.out.println("Link Verification link completed");
}

System.out.println(c);
driver.close();
sc.close();
}
}
}

```

**Output**

```

Enter Search Key Word
kalam
Starting ChromeDriver 2.29.461591
(62ebf098771772160f391d75e589dc567915b233) on port 16319
Only local connections are allowed.
Jul 11, 2017 11:32:13 AM org.openqa.selenium.remote.ProtocolHandshake
createSession
INFO: Detected dialect: OSS
Link Verification link completed
29

```

**Test scripts2:-**

```

package loops_1;
import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Pagination2
{
// pagination (Moving from one page to next page until last page)
public static void main(String[] args) throws Exception
{
// Get data from key board
Scanner sc = new Scanner(System.in);
System.out.println("Enter Search Key Word");
String str = sc.nextLine();
System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("http://www.google.com");
driver.manage().window().maximize();
driver.findElement(By.name("q")).sendKeys(str);
driver.findElement(By.name("btnG")).click();
Thread.sleep(3000);
int c = 1;
try
{
while (2 > 1)
{
if (driver.findElement(By.xpath("//*[text()='Next']")).isDisplayed())
{
driver.findElement(By.xpath("//*[text()='Next']")).click();
Thread.sleep(3000);
c = c + 1;
}
}
}
catch (Exception e)

```

```
{
 System.out.println("pagination was completed" + c);
}
System.out.println(c);
driver.close();
sc.close();
}
}
```

**Output**

Enter Search Key Word

kalam

```
Starting ChromeDriver 2.29.461591
(62ebf098771772160f391d75e589dc567915b233) on port 16319
Only local connections are allowed.
Jul 11, 2017 11:32:13 AM org.openqa.selenium.remote.ProtocolHandshake
createSession
INFO: Detected dialect: OSS
Link Verification link completed
29
```

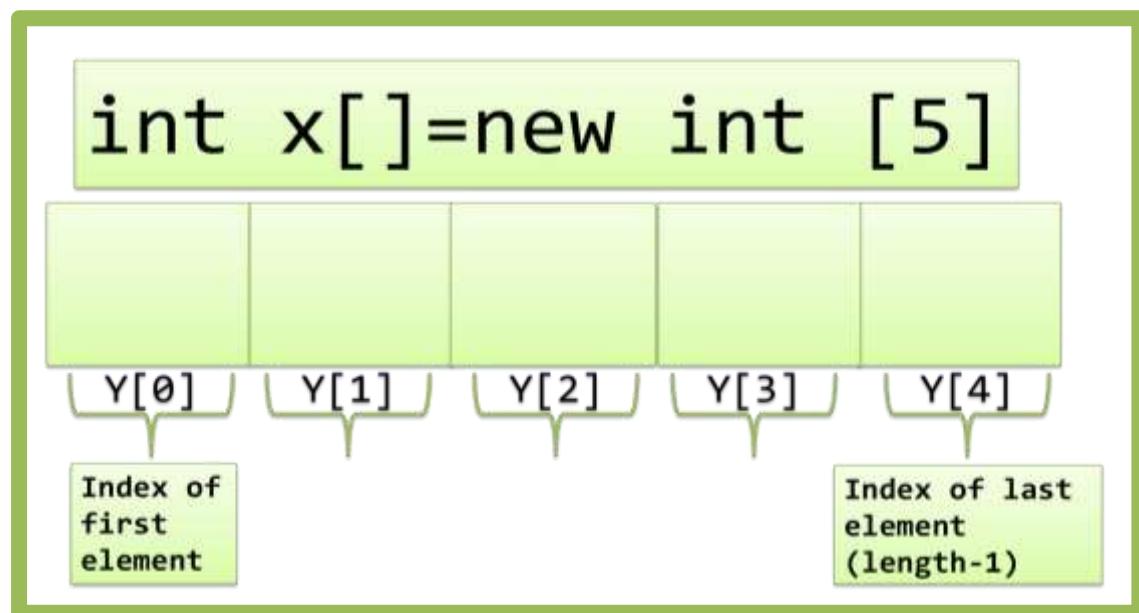
## O.Arrays In Java

Variable is used to store one value whereas arrays are used to store multiple similar values.

- Arrays are 2 types such as
  - ⇒ Static Array
  - ⇒ Dynamic Array

### ⇒ Static Array

Static array is an array to store fixed number of values.



### Example:

Write java code to create 6 numbers array to store 6 subjects' marks into that array & to calculate total 6 subjects.

### Test scripts:-

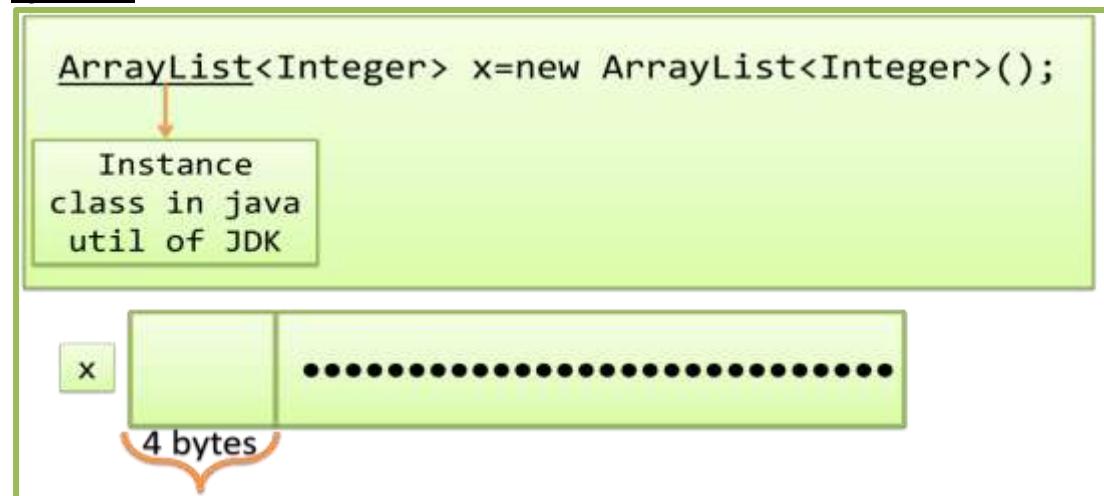
```
package array;
import java.util.Scanner;
public class Marks
{
 public static void main(String[] args)
 {
 int[] marks=new int[6];
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter 6 Subjects");
 for(int i=0;i<marks.length;i++)
 {
 marks[i]=sc.nextInt();
 }
 int tot=0;
 for(int i=0;i<marks.length;i++)
 {
 tot=tot+marks[i];
 }
 System.out.println("Marks Secured"+ tot);
 }
}
```

In static array size was fixed. due to this reason test automate will go to dynamic array is the place of static array.

## ⇒ Dynamic Array

Dynamic array is an array to store dynamic number of values (or) random number of values.

### Syntax:



### Example:

Take a dynamic array, store multiple values & calculate total of those values.

#### Test script:-

```
package array;
import java.util.ArrayList;
import java.util.Scanner;
public class MarksDynamic
{
 public static void main(String[] args)
 {
 ArrayList<Integer> marks=new ArrayList<Integer>();
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter number of subjects");
 int nos=sc.nextInt();
 System.out.println("Enter marksof each subjects");
 for(int i=0;i<nos;i++)
 {
 marks.add(sc.nextInt());
 }
 int tot=0;
 for(int i=0;i<nos;i++)
 {
 tot=tot+marks.get(i);
 }
 System.out.println("Marks Secured"+ tot);
 sc.close();
 }
}
```

#### Output

```
Enter number of subjects
7
Enter marksof each subjects
67
87
56
86
45
63
65
Marks Secured469
```

**Example:**

Take a dynamic array, store multiple values & find the maximum value & display.

**Test script:-**

```
package array;
import java.util.ArrayList;
import java.util.Scanner;

public class Max_Min_DynaArray
{
 public static void main(String[] args)
 {
 ArrayList<Integer> m=new ArrayList<Integer>();
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter number of Values");
 int nov=sc.nextInt();
 System.out.println("Enter values");
 for(int i=0;i<nov;i++)
 {
 m.add(sc.nextInt());
 }
 //Find Maximum Value
 int max=m.get(0);
 for(int i=0;i<nov;i++)
 {
 if(max<m.get(i))
 {
 max=m.get(i);
 }
 }
 System.out.println("Maximum value is :" +max);
 //Find Minimum Value
 int min=m.get(0);
 for(int i=0;i<nov;i++)
 {
 if(min>m.get(i))
 {
 min=m.get(i);
 }
 }
 System.out.println("Minimum value is :" +min);
 }
}
```

**Output**

```
Enter number of Values
7
Enter values
12132
123
1112
324
1234
908
675
Maximum value is :12132
Minimum value is :123
```

**Example:**

Sorting numbers

**Test script:-**

```
package array;
import java.util.ArrayList;
import java.util.Scanner;
public class Sorting
{
 public static void main(String[] args)
 {
 ArrayList<Integer> m = new ArrayList<Integer>();
 Scanner sc = new Scanner(System.in);
 // Store Data in array
 System.out.println("Enter no of values");
 int nov = sc.nextInt();
 System.out.println("Enter values");
 for (int i = 0; i < nov; i++)
 {
 m.add(sc.nextInt());
 }
 // to print index values
 for (int i = 0; i < nov; i++)
 {
 System.out.print(m.get(i) + " ");
 }
 System.out.println();
 // sorting
 for (int i = 0; i < nov; i++)
 {
 for (int j = 0; j < nov - 1; j++)
 {
 if (m.get(j) > m.get(j + 1))
 {
 int temp = m.get(j);
 m.set(j, m.get(j + 1));
 m.set(j + 1, temp);
 System.out.print("Index of i[" + i + "] ");
 for (int k = 0; k < nov; k++)
 {
 System.out.print(m.get(k) + " ");
 }
 System.out.println();
 }
 }
 }
 // display array
 System.out.print("data output ");
 for (int i = 0; i < nov; i++)
 {
 System.out.print(m.get(i) + " ");
 }
 sc.close();
 }
}
```

**Output**

```
Enter number of Values
6
Enter values
102
13
18
26
7
-8
102 13 18 26 7 -8
Index of i[0] 13 102 18 26 7 -8
Index of i[0] 13 18 102 26 7 -8
Index of i[0] 13 18 26 102 7 -8
Index of i[0] 13 18 26 7 102 -8
Index of i[0] 13 18 26 7 -8 102
Index of i[1] 13 18 7 26 -8 102
Index of i[1] 13 18 7 -8 26 102
Index of i[2] 13 7 18 -8 26 102
Index of i[2] 13 7 -8 18 26 102
Index of i[3] 7 13 -8 18 26 102
Index of i[3] 7 -8 13 18 26 102
Index of i[4] -8 7 13 18 26 102
data output -8 7 13 18 26 102
```

## ● Data Driven Testing

### Example:

- ⇒ Take a test data & criteria into arrays
- ⇒ Launch Gmail site
- ⇒ Do login user id using test data in arrays
- ⇒ If user id is valid
  - Password will be displayed.
  - Enter password using test data in arrays.
  - If password is valid
    - ↓ Compose page will be displayed.
  - If password is invalid
    - ↓ Error message will be displayed.
- ⇒ Enter User ID is invalid & Click Next.
  - Error message will be displayed.
- ⇒ Close site.

### Test Script:-

```

package array;
import java.util.ArrayList;
import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class GmailArrayList
{
 public static void main(String[] args) throws Exception
 {
 ArrayList<String> uids = new ArrayList<String>();
 ArrayList<String> ucs = new ArrayList<String>();
 ArrayList<String> pwds = new ArrayList<String>();
 ArrayList<String> pcs = new ArrayList<String>();
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter no of iteration");
 int noi = Integer.parseInt(sc.nextLine());
 for (int i = 0; i < noi; i++)
 {
 System.out.println("Enter Userid");
 uids.add(sc.nextLine());
 System.out.println("Enter userid criteria");
 ucs.add(sc.nextLine());
 if (ucs.get(i).equals("valid"))
 {
 System.out.println("Enter password");
 pwds.add(sc.nextLine());
 System.out.println("Enter password criteria");
 pcs.add(sc.nextLine());
 }
 else
 {
 pwds.add("");
 pcs.add("");
 }
 }
 // Data Driven Testing
 for (int i = 0; i < noi; i++)
 {
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 WebDriver driver = new ChromeDriver();
 driver.get("http://www.gmail.com");
 driver.manage().window().maximize();
 }
 }
}

```

```

Thread.sleep(5000);
// Enter user-id
driver.findElement(By.name("identifier")).sendKeys(uids.get(i),Keys.ENTER);
Thread.sleep(5000);
// User ID Testing
if (ucs.get(i).equals("valid") &&
driver.findElement(By.name("password")).isDisplayed())
{
System.out.println("userid test was passed");
// Password Testing
driver.findElement(By.name("password")).sendKeys(pwds.get(i),Keys.ENTER);
Thread.sleep(5000);
if (pcs.get(i).equals("valid") &&
driver.findElement(By.xpath("//*[text()='COMPOSE']")).isDisplayed())
{
 System.out.println("password test was passed");
}
else if (pcs.get(i).equals("invalid") &&
driver.findElement(By.xpath("(//*[contains(text(),'Wrong') or
contains(text(),'Enter a password'))][2]")).isDisplayed())
{
 System.out.println("Password Test was passed");
}
else
{
 System.out.println("password test was failed");
}
}
else if (ucs.get(i).equals("invalid") &&
driver.findElement(By.xpath("(//*[contains(text(),'Couldn') or
contains(text(),'Enter an email'))][2]")).isDisplayed())
{
 System.out.println("userid test was passed");
}
else
{
 System.out.println("userid test was failed");
}
// Close Site
driver.close();
}
sc.close();
}
}

```

**Note 1:-**

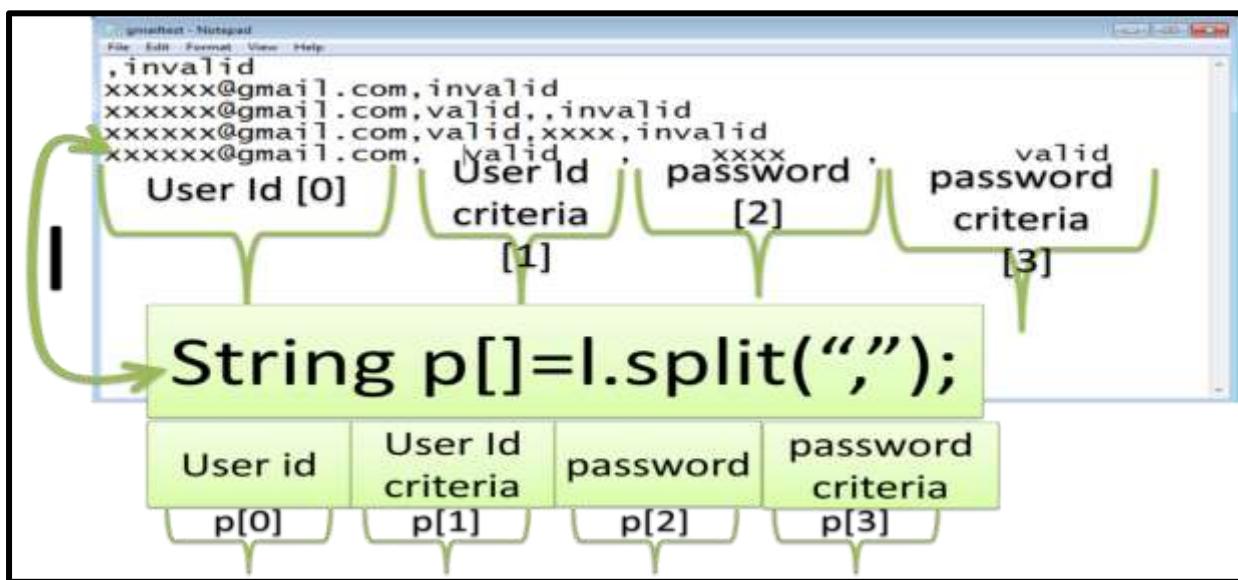
In programming language “variable” & “arrays” will be created in ram.  
 But “variable” & “arrays” life time will go to end to end of corresponding coding execution.  
 Due to this reason, test automaters (SDET) can use Files for test data, because file are permanently save in HDD.

**Note 2:-**

In above script test result will come in Eclipse IDE console. But to maintain results as permanently saved in HDD.

## P. Using text files:-

To maintain test data & test result in files we can use in java with notepad can help to us to create text files like shown below.

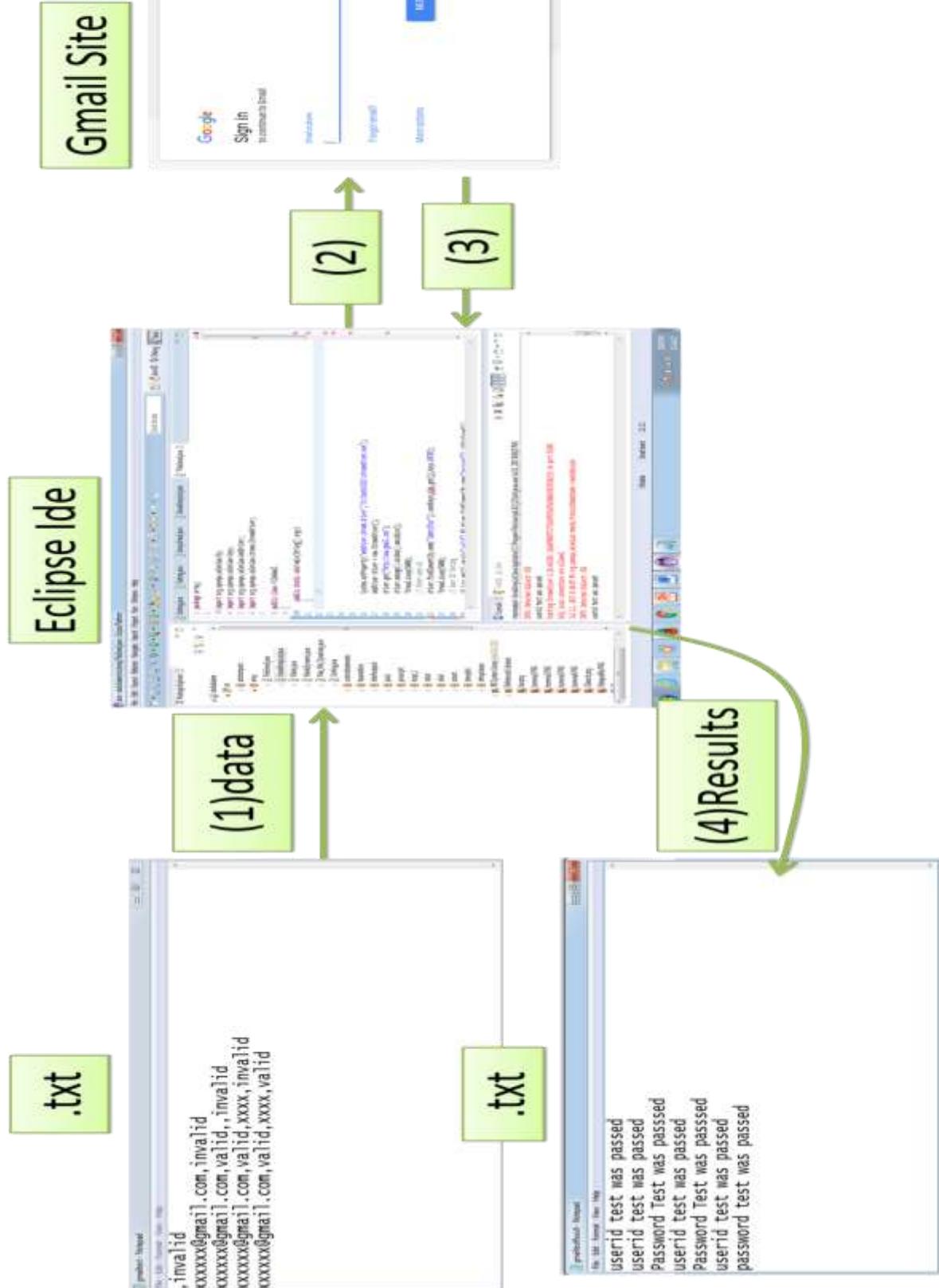


### Rules:-

- ⇒ Above like “.txt” files will be also saved as “.csv” files, because values will be separated by “, comma” in those files.
- ⇒ .txt files are sequential files, because they can allow reading (or) writing from starting only.
- ⇒ .txt files can allow either reading (or) writing at a time.
- ⇒ Don’t create new line after last line of data in text files. (to identify of last line, loop will take as new line loop will show error of values)

Example 1:-

## Data driven Frame work



**Test Script:-**

```

package array;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class FileGmail
{
 public static void main(String[] args) throws Exception
 {
 //
 File f1=new File("gmailtest.txt");
 FileReader fr=new FileReader(f1);
 BufferedReader br=new BufferedReader(fr);
 File f2=new File("gmailtestResult.txt");
 FileWriter fw=new FileWriter(f2);
 BufferedWriter bw=new BufferedWriter(fw);
 String l;
 while((l=br.readLine())!=null)
 {
 String p[]={l.split(",")};
 //Launch site
 System.setProperty("webdriver.chrome.driver","d:\\\\chromedriver.exe");
 WebDriver driver = new ChromeDriver();
 driver.get("http://www.gmail.com");
 driver.manage().window().maximize();
 Thread.sleep(5000);
 // Enter user-id
 driver.findElement(By.name("identifier")).sendKeys(p[0],Keys.ENTER);
 Thread.sleep(5000);
 // User ID Testing
 if (p[1].equals("valid") &&
 driver.findElement(By.name("password")).isDisplayed())
 {
 bw.write("userid test was passed");
 bw.newLine();
 // Password Testing

 driver.findElement(By.name("password")).sendKeys(p[2],Keys.ENTER);
 Thread.sleep(5000);
 if (p[3].equals("valid") &&
 driver.findElement(By.xpath("//*[text()='COMPOSE']")).isDisplayed())
 {
 bw.write("password test was passed");
 bw.newLine();
 }
 else if (p[3].equals("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'Wrong') or
 contains(text(),'Enter a password')])[2]").isDisplayed())
 {
 bw.write("Password Test was passed");
 bw.newLine();
 }
 }
 else
 {
 bw.write("password test was failed");
 bw.newLine();
 }
 }
 }
}

```

```

 }
 }
 else if (p[1].equals("invalid") &&
driver.findElement(By.xpath("//*[contains(text(),'Couldn') or
contains(text(),'Enter an email')])[2]").isDisplayed())
{
 bw.write("user id test was passed");
 bw.newLine();
}
else
{
 bw.write("userid test was failed");
 bw.newLine();
}

// Close Site
driver.close();
}
bw.close();
br.close();
fr.close();
fw.close();
}
}

```

**Note 1:-**

To use text files (“.txt/.csv”) for test Data Reading & test result, writing, we can use depend on few “JDK” class like Shown below.

|                   |
|-------------------|
| ⇒ File            |
| ⇒ File Reader     |
| ⇒ Buffered Reader |
| ⇒ File Writer     |
| ⇒ Buffered Writer |

Instance classes  
in “java io”  
Package in JDK.

## **Q. Extent Reports:-**

- ⇒ Available as Jars
- ⇒ Useful for test results preparation as a “html” file.
- ⇒ Test Results “html” file can maintain various colors, Pie charts ...etc.
- ⇒ We need to follow below navigation to configure, Extent Reports in tester computer.

### **Navigation:-**

- ↓ Go to [www.google.com](http://www.google.com)
- ↓ Enter “Extent Reports download” & click search.
- ↓ Go to <http://www.extentreports.com/>

**Community Edition**

- ↓ Click on “Community Edition” button.
- ↓ Click on “Java 2.41.2” button. (Stable Version).

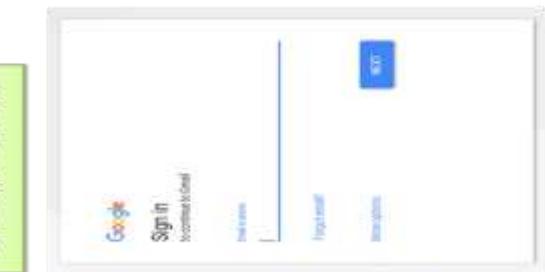
**Java 2.41.2**

- ↓ Click on “download icon” .
- ↓ Wait until that download completed.
- ↓ Paste that download in personal folder.
- ↓ Extract that download.
- ↓ Open Eclipse IDE.
- ↓ Right click on project.
- ↓ Go to properties.
- ↓ Select Java build path.
- ↓ Select libraries.
- ↓ Add External Jars.
- ↓ Browser path of lib folder inside & lib folder outside jars in extracted jars reports folder.
- ↓ Click ok

Example:

# Data driven Frame work

Gmail Site



(2)

(3)

Eclipse Ide

```
File > New > Other > Java > Java Project
Project Name: DataDrivenFramework
Next >
Finish
```

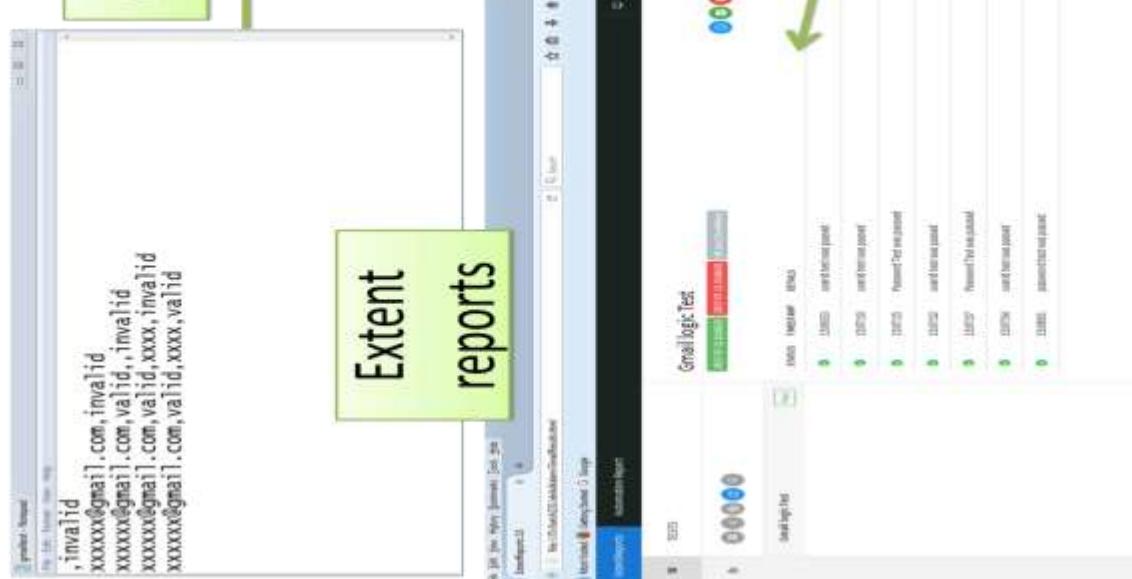
(1) data

```
invalid
invalid@gmail.com,invalid
invalid@gmail.com,valid,invalid
invalid@gmail.com,valid,xxx,invalid
invalid@gmail.com,valid,xxx,valid
```

.txt

Extent reports

(4) Results



**Test Script:-**

```

package ExtentReportFile;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import com.relevantcodes.extentreports.ExtentReports;
import com.relevantcodes.extentreports.ExtentTest;
import com.relevantcodes.extentreports.LogStatus;
public class FileExtentReport
{
 public static void main(String[] args) throws Exception
 {
 //
 File f1=new File("gmailtest.txt");
 FileReader fr=new FileReader(f1);
 BufferedReader br=new BufferedReader(fr);
 ExtentReports er=new ExtentReports("GmailResults.html");
 ExtentTest et=er.startTest("Gmail logic Test");
 WebDriver driver;
 String l;
 while((l=br.readLine())!=null)
 {
 String p[]={l.split(",")};
 //Launch site
 System.setProperty("webdriver.chrome.driver","d: \chromedriver.exe");
 driver = new ChromeDriver();
 driver.get("http://www.gmail.com");
 driver.manage().window().maximize();
 Thread.sleep(5000);
 // Enter user-id
 driver.findElement(By.name("identifier")).sendKeys(p[0],Keys.ENTER);
 Thread.sleep(5000);
 // User ID Testing
 if (p[1].equals("valid") &&
 driver.findElement(By.name("password")).isDisplayed())
 {
 et.log(LogStatus.PASS, "userid test was passed");
 // Password Testing

 driver.findElement(By.name("password")).sendKeys(p[2],Keys.ENTER);
 Thread.sleep(5000);
 if (p[3].equals("valid") &&
 driver.findElement(By.xpath("//*[text()='COMPOSE']")).isDisplayed()
 {
 et.log(LogStatus.PASS,"password test was passed");
 }
 else if (p[3].equals("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'Wrong') or
 contains(text(),'Enter a password')])[2]").isDisplayed())
 {
 et.log(LogStatus.PASS,"Password Test was passed");
 }
 else
 {
 et.log(LogStatus.FAIL,"password test was failed");
 }
 }
 }
 }
}

```

```
else if (p[1].equals("invalid") &&
driver.findElement(By.xpath("//*[contains(text(),'Couldn') or
contains(text(),'Enter an email')])[2]")).isDisplayed())
{
 et.log(LogStatus.PASS,"userid test was passed");
}
else
{
 et.log(LogStatus.FAIL,"userid test was failed");
}

// Close Site
driver.close();
}
br.close();
fr.close();
er.endTest(et);
er.flush();

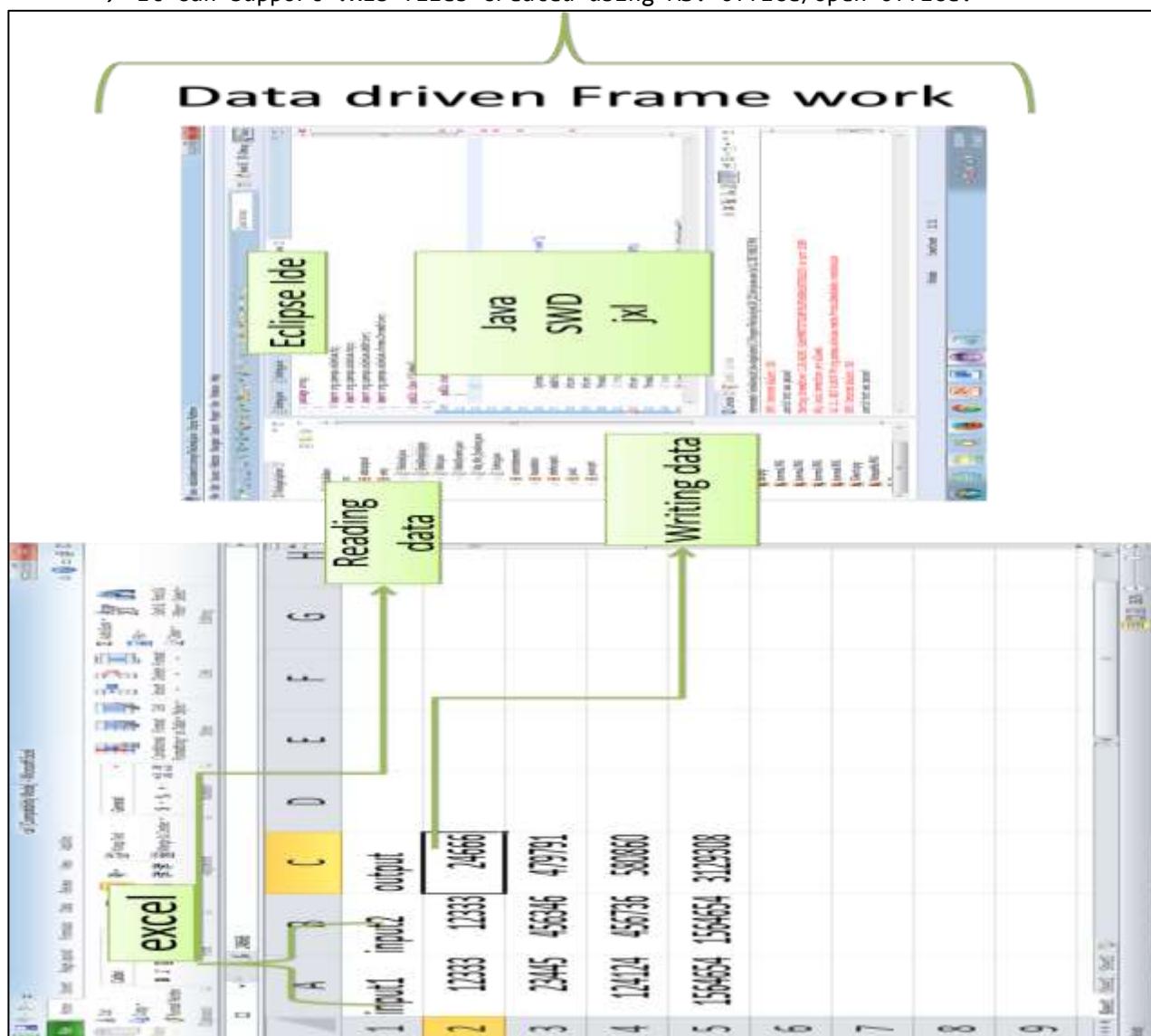
}
}
```

**Note 1:-**

Extent report jars providing some classes to make results as 'HTML' file.  
Here some classes are instance & other is static class.

## R.JXL:-

- ⇒ Developed by Apache.
- ⇒ Useful to integrated Java program with excel file(.xls)
- ⇒ It can support .xls files created using MS. Office/Open-Office.



⇒ We need follow below navigation to configure JXL in tester computer

## Navigation:-

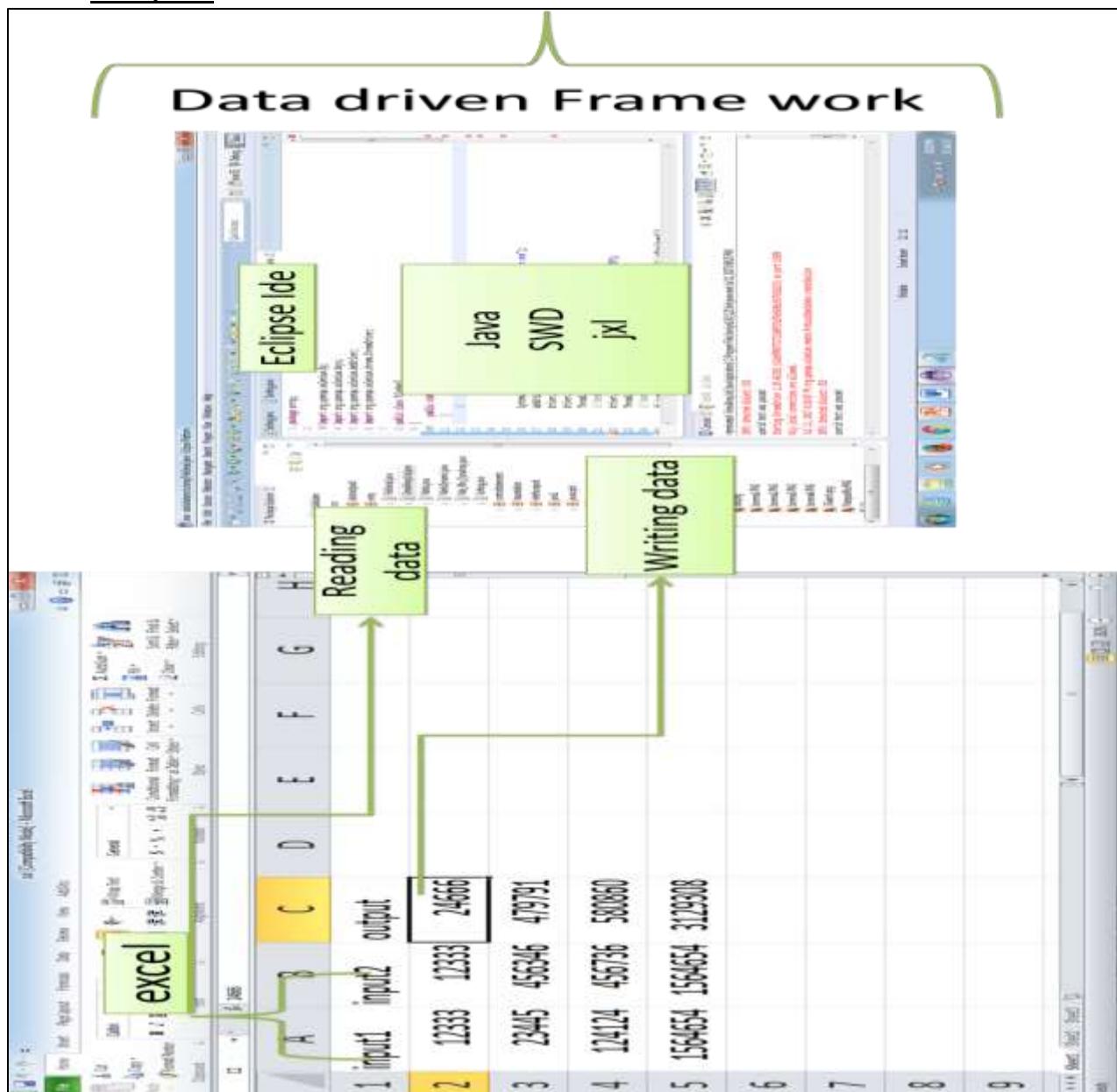
- ↓ Go to [www.google.com](http://www.google.com)
- ↓ Enter “JXL download” & click search
- ↓ Go to “<http://www.java2s.com/Code/Jar/j/Downloadjxl2612jar.htm>” site



jxl/jxl-2.6.12.jar.zip( 672 k)

- ↓ Click on [jxl/jxl-2.6.12.jar.zip\( 672 k\)](#)
- ↓ Wait until that download completed.
- ↓ Paste that download in personal folder.
- ↓ Extract that download.
- ↓ Open Eclipse IDE.
- ↓ Right click on project.
- ↓ Go to properties.
- ↓ Select Java build path.

- ↓ Select libraries.
- ↓ Add External Jars.
- ↓ Browser path of jxl jar path in personal folder.
- ↓ Click ok

**Example:**

(personal folder\sample.xls | In Microsoft Excel 2007 / 2010 & in open office, File Extension is other (xlsx,xlsm), so we need to save as to “Microsoft Excel 97-2003 Worksheet (.xls)”)

**Test Script:-**

```
package data_driven;
import java.io.File;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.Number;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
public class AddExcelData
{
```

```

public static void main(String[] args) throws Exception
{
 File f=new File("cal.xls");
 Workbook rwb= Workbook.getWorkbook(f);
 Sheet rsh1=rwb.getSheet(0); //Sheet1 because index0
 int nour=rsh1.getRows(); //Get no of used rows
 int nouc=rsh1.getColumns(); //Get no of used columns
 WritableWorkbook wwb= Workbook.createWorkbook(f, rwb);
 WritableSheet wsh=wwb.getSheet(0); // sheet1 insert data
 // data driven testing
 Label da=new Label(nouc,0,"Result");
 wsh.addCell(da);
 for(int i=1;i<nour; i++)
 {
 int x=Integer.parseInt(rsh1.getCell(0, i).getContents());
 int y=Integer.parseInt(rsh1.getCell(1, i).getContents());
 int z=x+y;
 Number n=new Number(nouc,i,z);
 wsh.addCell(n);
 }
 wwb.write(); //save
 wwb.close();
 rwb.close();
}

```

**Note 1:-**

While accessing excel file using “JXL”, we can get data as string by default.

**Note 2:-**

While using excel file in test automation, we need to follow below rules with respect to JXL

**Rules:-**

1. We need to save excel file with “.xls” file extension.
2. Remove/Delete unwanted rows and columns data
3. Create a copy file to corresponding excel file for future usage (In general, excel file can corrupt when corresponding file script was raised exception (runtime error)).
4. Close excel file on desktop before corresponding script.

**Example:**

|   | A                      | B                | C        | D                 | E |
|---|------------------------|------------------|----------|-------------------|---|
| 1 | user id                | User id criteria | password | password criteria |   |
| 2 |                        | invalid          |          |                   |   |
| 3 | maheshXXXX@gmail.com   | invalid          |          |                   |   |
| 4 | maheshsaXXXX@gmail.com | valid            |          | invalid           |   |
| 5 | maheshsaXXXX@gmail.com | valid            | fgzdfgh  | invalid           |   |
| 6 | maheshsaXXXX@gmail.com | valid            | dfaskdg  | valid             |   |
| 7 |                        |                  |          |                   |   |

**Test Script:-**

```

package data_driven;
import java.io.File;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
public class GmailExcel
{
 public static void main(String[] args) throws Exception
 {
 File f = new File("exceledata.xls");
 Workbook rwb = Workbook.getWorkbook(f);
 Sheet rsh = rwb.getSheet(0); // Sheet1 because index0
 int nour = rsh.getRows(); // Get no of used rows
 int nouc = rsh.getColumns(); // Get no of used columns
 // open same excel sheet for writing result
 WritableWorkbook wwb = Workbook.createWorkbook(f, rwb);
 WritableSheet wsh = wwb.getSheet(0); // sheet1 insert data
 // data driven testing
 Label da = new Label(nouc, 0, "Result");
 wsh.addCell(da);
 Label l1;
 for (int i = 1; i < nour; i++)
 {
 String u = rsh.getCell(0, i).getContents();
 String uc = rsh.getCell(1, i).getContents();
 String p = "";
 String pc = "";
 if (uc.equals("valid"))
 {
 p = rsh.getCell(2, i).getContents();
 pc = rsh.getCell(3, i).getContents();
 }
 System.setProperty("webdriver.chrome.driver",
 "d:\\mahesh\\chromedriver.exe");
 WebDriver driver = new ChromeDriver();
 driver.get("http://www.gmail.com");
 driver.manage().window().maximize();
 Thread.sleep(5000);
 // Enter user-id
 driver.findElement(By.name("identifier")).sendKeys(u, Keys.ENTER);
 Thread.sleep(5000);
 // User ID Testing
 try
 {
 if (uc.equals("valid") && driver.findElement(By.name("password")).isDisplayed())
 {
 // Password Testing
 driver.findElement(By.name("password")).sendKeys(p, Keys.ENTER);
 Thread.sleep(5000);
 if (pc.equals("valid") && driver.findElement(
 By.xpath("//*[text()='COMPOSE']")).isDisplayed())
 {
 l1 = new Label(nouc, i, "password test was passed");
 wsh.addCell(l1);
 }
 }
 }
 }
}

```

```

else if (pc.equals("invalid")&&
driver.findElement(By.xpath("//*[contains(text(),'Wrong') or
contains(text(),'Enter a password')])[2]").isDisplayed())
{
l1 = new Label(nouc, i, "password test was passed");
wsh.addCell(l1);
}
else
{
l1 = new Label(nouc, i, "password test was failed");
wsh.addCell(l1);
}
}
else if (uc.equals("invalid")&&
driver.findElement(By.xpath("//*[contains(text(),'Couldn') or
contains(text(),'Enter an email')])[2]").isDisplayed())
{
 l1 = new Label(nouc, i, "userid test was passed");
 wsh.addCell(l1);
}
else
{
l1 = new Label(nouc, i, "userid test was passed");
wsh.addCell(l1);
}
}
catch (Exception e)
{
l1 = new Label(nouc, i, "userid test was Terminated");
wsh.addCell(l1);
}
// Close Site
driver.close();
}
wwb.write();
wwb.close();
rwb.close();
}
}

```

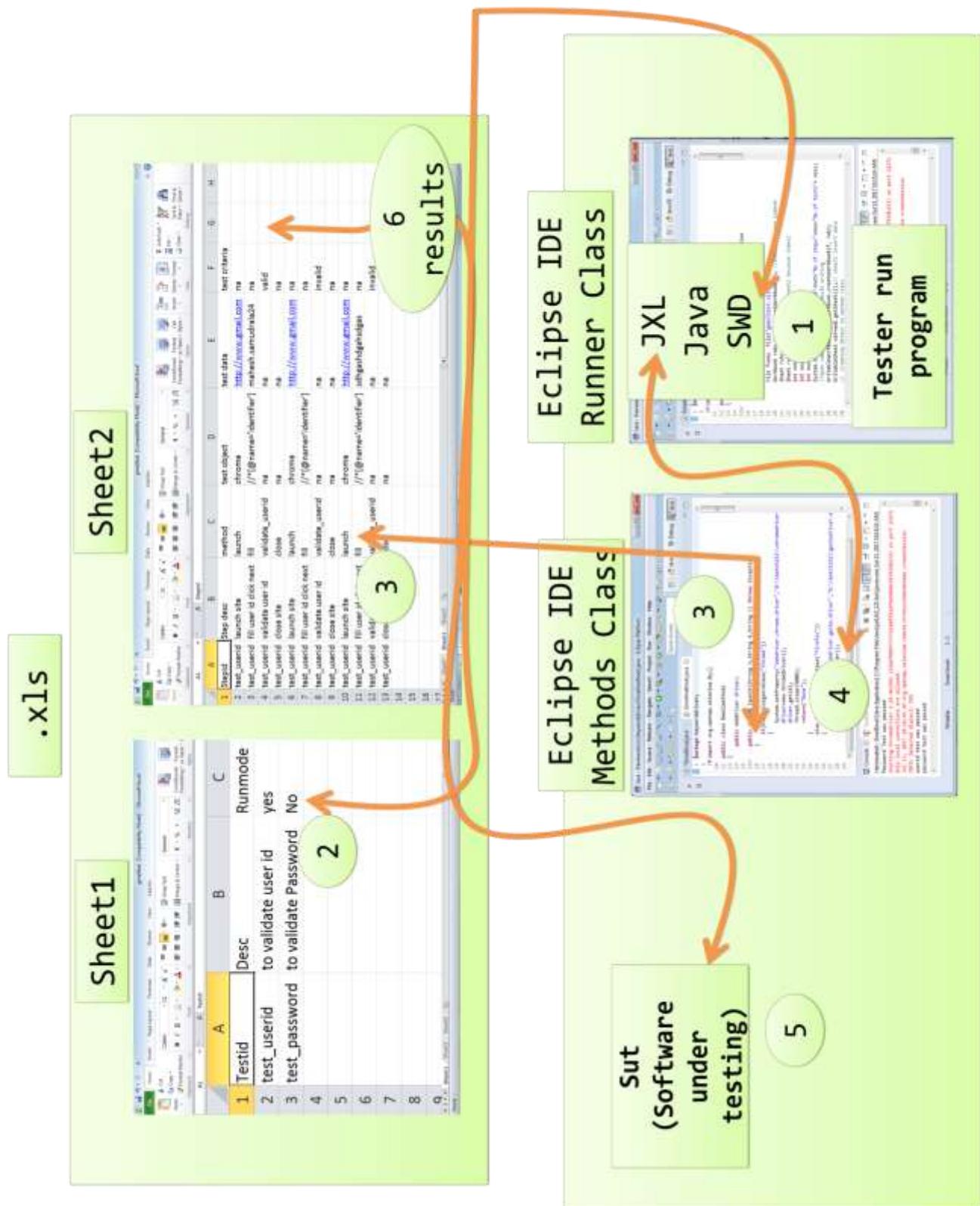
**Note 1:-**

1. While using JXL to access ‘.xls’ file for data reading & result writing, we can use below classes.

|                    |                  |
|--------------------|------------------|
| ⇒ Workbook         |                  |
| ⇒ Sheet            |                  |
| ⇒ WritableWorkbook | Instance Classes |
| ⇒ WritableSheet    | in JXL jar       |
| ⇒ Label            |                  |
| ⇒ Number           |                  |

## S.Keyword Driven Framework:-

- ⇒ Programming based framework.
- ⇒ JXL is mandatory with selenium web driver & Java because we need to write test & steps of each test in .xls file
- ⇒ We need to develop three things in this framework such as:-
  - “.xls” files tests in sheet1 & steps in sheet2.
  - Methods class which methods having selenium Web Driver automation code
  - Runner class with main() method to integrate code above .xls & methods class.



## Keyword driven framework process:-

### Step 1:-

Download and install JDK8

(create **JAVA\_HOME** environmental variables with path of JDK & extend path variable with path of JDK\bin;)

### Step 2:-

Download and launch ECLIPSE IDE(Juno Version)  
(browse a folder for work space)

### Step 3:-

Create a new Java Project with a package in Eclipse IDE(Project name (OR)  
Package name as single word without space in lower case)

### Step 4:-

Download Jars & associate to project

(set latest compiler & JRE to java project

- ⇒ Download SWD Jars.
- ⇒ Associate to java project.
- ⇒ Download Sikulix setup and install.
- ⇒ Associate Sikulix API jar to java project if required in java project if required.
- ⇒ Download & associate “JXL” jars in Java project
- ⇒ Download & associate “Extent Report” jars in Java project if required.)

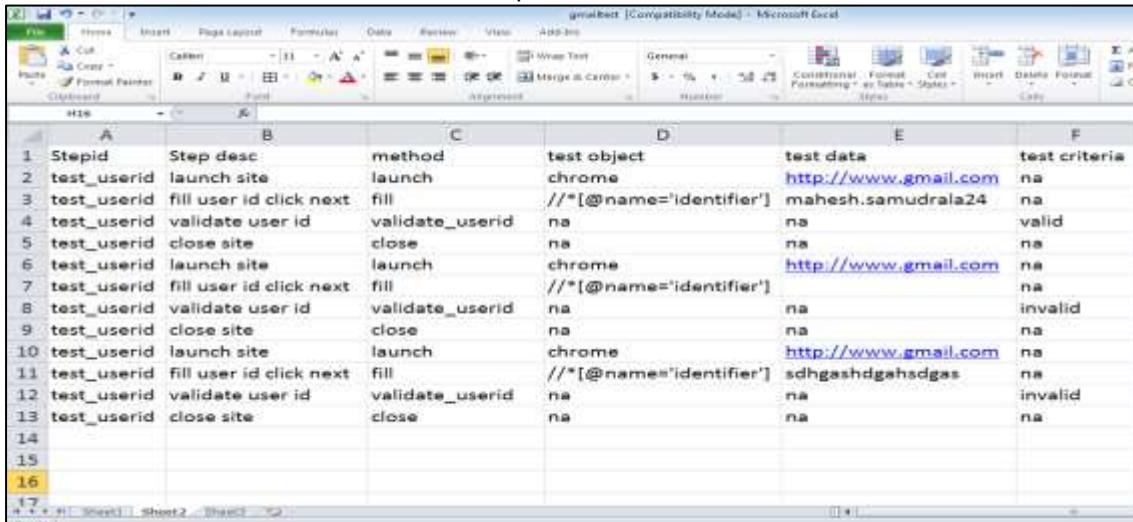
### Step 5-

Create excel file's sheet1 with tests information like shown below.

| A             | B                    | C       |
|---------------|----------------------|---------|
| Testid        | Desc                 | Runmode |
| test_userid   | to validate user id  | yes     |
| test_password | to validate Password | No      |

### Step 6-

Create excel file's sheet2 with steps information like shown below.



The screenshot shows a Microsoft Excel spreadsheet titled "Sheet2". The columns are labeled A through F. Column A contains step IDs (1 to 17). Columns B through F contain details for each step, such as step description, method, test object, test data, and test criteria. The data includes various browser actions like "launch", "fill", and "validate", along with specific URLs and identifiers.

| A  | B           | C                       | D               | E                       | F                    |
|----|-------------|-------------------------|-----------------|-------------------------|----------------------|
| 1  | Stepid      | Step desc               | method          | test object             | test criteria        |
| 2  | test_userid | launch site             | launch          | chrome                  | http://www.gmail.com |
| 3  | test_userid | fill user id click next | fill            | //*[@name='identifier'] | na                   |
| 4  | test_userid | validate user id        | validate_userid | na                      | valid                |
| 5  | test_userid | close site              | close           | na                      | na                   |
| 6  | test_userid | launch site             | launch          | chrome                  | http://www.gmail.com |
| 7  | test_userid | fill user id click next | fill            | //*[@name='identifier'] | na                   |
| 8  | test_userid | validate user id        | validate_userid | na                      | invalid              |
| 9  | test_userid | close site              | close           | na                      | na                   |
| 10 | test_userid | launch site             | launch          | chrome                  | http://www.gmail.com |
| 11 | test_userid | fill user id click next | fill            | //*[@name='identifier'] | na                   |
| 12 | test_userid | validate user id        | validate_userid | na                      | invalid              |
| 13 | test_userid | close site              | close           | na                      | na                   |
| 14 |             |                         |                 |                         |                      |
| 15 |             |                         |                 |                         |                      |
| 16 |             |                         |                 |                         |                      |
| 17 |             |                         |                 |                         |                      |

After completion of Sheet1 with tests & sheet2 with steps, we can go to save that file in project folder of work space with “.xls” extension.

### Step 7-

Develop methods with automation code

- ⇒ Right click on package
- ⇒ New
- ⇒ Class
- ⇒ Enter a name to class (Single word with Initcap Gmailmethods & deselect main() method option)
- ⇒ Click finish
- ⇒ Develop automation code in own methods like shown below.

### Example:

**Sheet1 Data:**

|   | A             | B                    | C       |
|---|---------------|----------------------|---------|
| 1 | Testid        | Desc                 | Runmode |
| 2 | test_userid   | to validate user id  | yes     |
| 3 | test_password | to validate Password | No      |
| 4 |               |                      |         |

**Sheet2 Data:**

|    | A           | B                       | C               | D                       | E                    | F             | G | H |
|----|-------------|-------------------------|-----------------|-------------------------|----------------------|---------------|---|---|
| 1  | Stepid      | Step desc               | method          | test object             | test data            | test criteria |   |   |
| 2  | test_userid | launch site             | launch          | chrome                  | http://www.gmail.com | na            |   |   |
| 3  | test_userid | fill user id click next | fill            | //*[@name='identifier'] | mahesh.samudrala24   | na            |   |   |
| 4  | test_userid | validate user id        | validate_userid | na                      | na                   | valid         |   |   |
| 5  | test_userid | close site              | close           | na                      | na                   | na            |   |   |
| 6  | test_userid | launch site             | launch          | chrome                  | http://www.gmail.com | na            |   |   |
| 7  | test_userid | fill user id click      | fill            | //*[@name='identifier'] | sdhgashdgahsdgas     | invalid       |   |   |
| 8  | test_userid | validate user id        | validate_userid | na                      | na                   | na            |   |   |
| 9  | test_userid | close site              | close           | na                      | na                   | na            |   |   |
| 10 | test_userid | launch site             | launch          | chrome                  | http://www.gmail.com | na            |   |   |
| 11 | test_userid | fill user id click next | fill            | //*[@name='identifier'] | na                   | na            |   |   |
| 12 | test_userid | validate user id        | validate_userid | na                      | na                   | na            |   |   |
| 13 | test_userid | close site              | close           | na                      | na                   | na            |   |   |
| 14 |             |                         |                 |                         |                      |               |   |   |
| 15 |             |                         |                 |                         |                      |               |   |   |
| 16 |             |                         |                 |                         |                      |               |   |   |
| 17 |             |                         |                 |                         |                      |               |   |   |

**Java Code Snippet:**

```

public String fill(String a,String b,String c) throws Exception
{
 driver.findElement(By.xpath(a)).sendKeys(b,Keys.ENTER);
 Thread.sleep(5000);
 return("Done");
}

public String validate_userid(String a,String b,String c) throws Exception
{
 try
 {
 if(c.equalsIgnoreCase("valid") && driver.findElement(By.name("password")).isDisplayed())
 {
 Thread.sleep(5000);
 return("Test Passed");
 }
 else if(c.equalsIgnoreCase("invalid") && driver.findElement(By.xpath("//*[contains(text(),'Couldn't') or contains(text(),'Enter an email'))][2]")).isDisplayed())
 {
 Thread.sleep(5000);
 return("Test Passed");
 }
 }
}

```

**SWD  
Sikulix  
Robot**

**Test Script:-**

```
package keyworddriven;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.opera.OperaDriver;

public class Gmailmethods
{
 public WebDriver driver;
 public String launch(String o, String d, String c) throws Exception
 {
 // Cross Browser
 if (o.equalsIgnoreCase("Chrome"))
 {
 System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
 driver = new ChromeDriver();
 driver.get(d);
 Thread.sleep(5000);
 return ("Done");
 }
 else if (o.equalsIgnoreCase("Firefox"))
 {
 System.setProperty("webdriver.gecko.driver", "d:\\geckodriver.exe");
 driver = new FirefoxDriver();
 driver.get(d);
 Thread.sleep(5000);
 return ("Done");
 }
 else if (o.equalsIgnoreCase("opera"))
 {
 System.setProperty("webdriver.opera.driver", "d:\\operadriver.exe");
 driver = new OperaDriver();
 driver.get(d);
 Thread.sleep(5000);
 return ("Done");
 }
 else if (o.equalsIgnoreCase("i.e."))
 {
 System.setProperty("webdriver.ie.driver", "d:\\IEDriverServer.exe");
 driver = new InternetExplorerDriver();
 driver.get(d);
 Thread.sleep(5000);
 return ("Done");
 }
 else
 {
 System.exit(0);
 return ("wrong browser");
 }
 }

 public String fill(String o, String d, String c) throws Exception
 {
 driver.findElement(By.xpath(o)).sendKeys(d, Keys.ENTER);
 Thread.sleep(5000);
 return ("Done");
 }
}
```

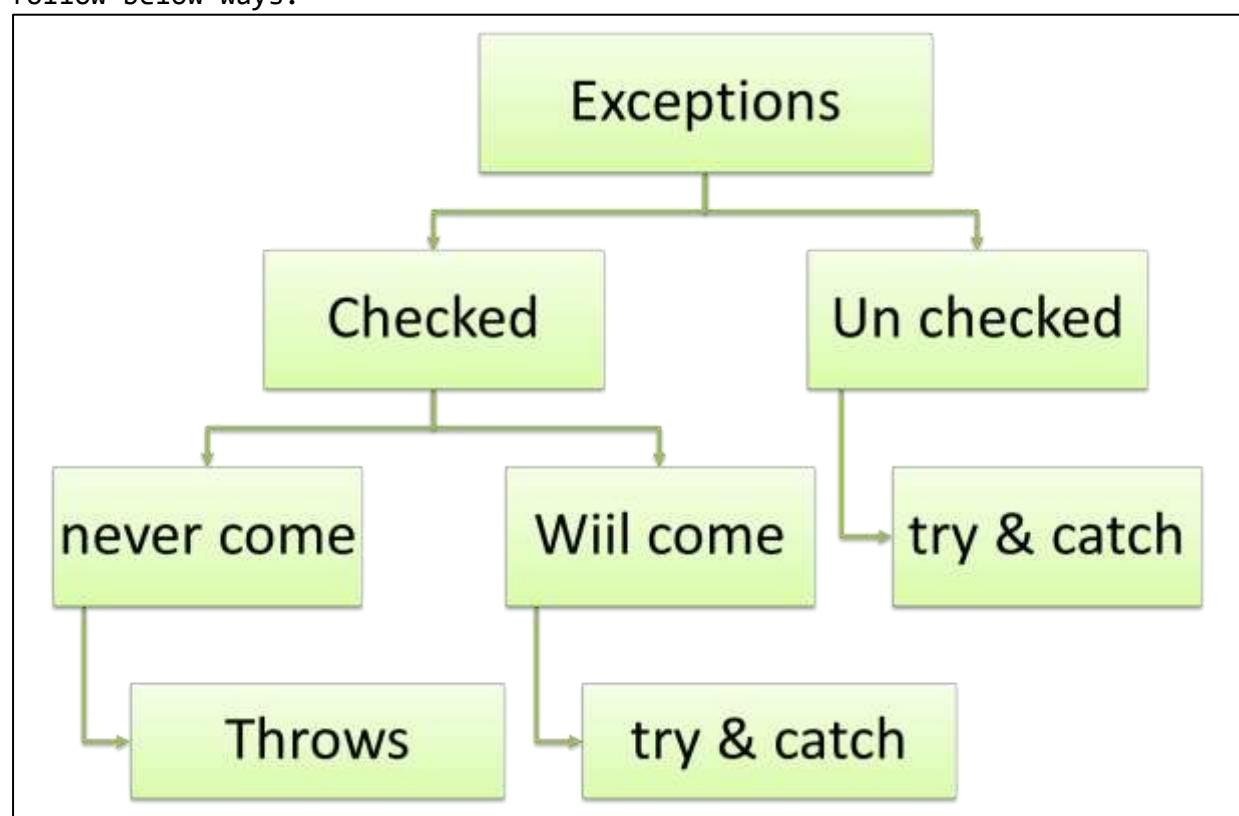
```

public String validate_userid(String o, String d, String c) throws Exception {
{
try {
if (c.equalsIgnoreCase("valid") &&
driver.findElement(By.name("password")).isDisplayed()) {
Thread.sleep(5000);
return ("Test Passed");
}
else if (c.equalsIgnoreCase("invalid") &&
driver.findElement(By.xpath("//*[contains(text(),'Couldn't') or
contains(text(),'Enter an email')])[2]").isDisplayed()) {
Thread.sleep(5000);
return ("Test Passed");
}
else {
Thread.sleep(5000);
return ("Test Failed");
}
}
catch (Exception e) {
return ("Testing was terminated");
}
}
public String close(String o, String d, String c) throws Exception {
driver.close();
Thread.sleep(5000);
return ("Done");
}
}
}

```

**Note 1:-**

While writing automation code (SWD + Sikulix + Java Robot) in methods, we can face exceptions. Every exception is runtime error but Eclipse IDE can suggest some exceptions to us while writing automation code called Checked Exceptions. Some exceptions are not suggested by eclipse called as unchecked Exceptions. To handle checked & unchecked Exceptions we can follow below ways.



**Step 8-****Develop runner class**

Runner class is useful to integrate excel file Sheet1 & Sheet2 with methods class() methods.

Here we need to use JXL with Java in runner classes.

- ↓ Right click on package.
- ↓ New Class.
- ↓ Enter a name to class(Single word starts with upper case)
- ↓ Select main() method option.
- ↓ Click finish.
- ↓ Write JXL Java code shown below.

**Test Script:-**

```
package keyworddriven;

import java.io.File;
import java.lang.reflect.Method;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;

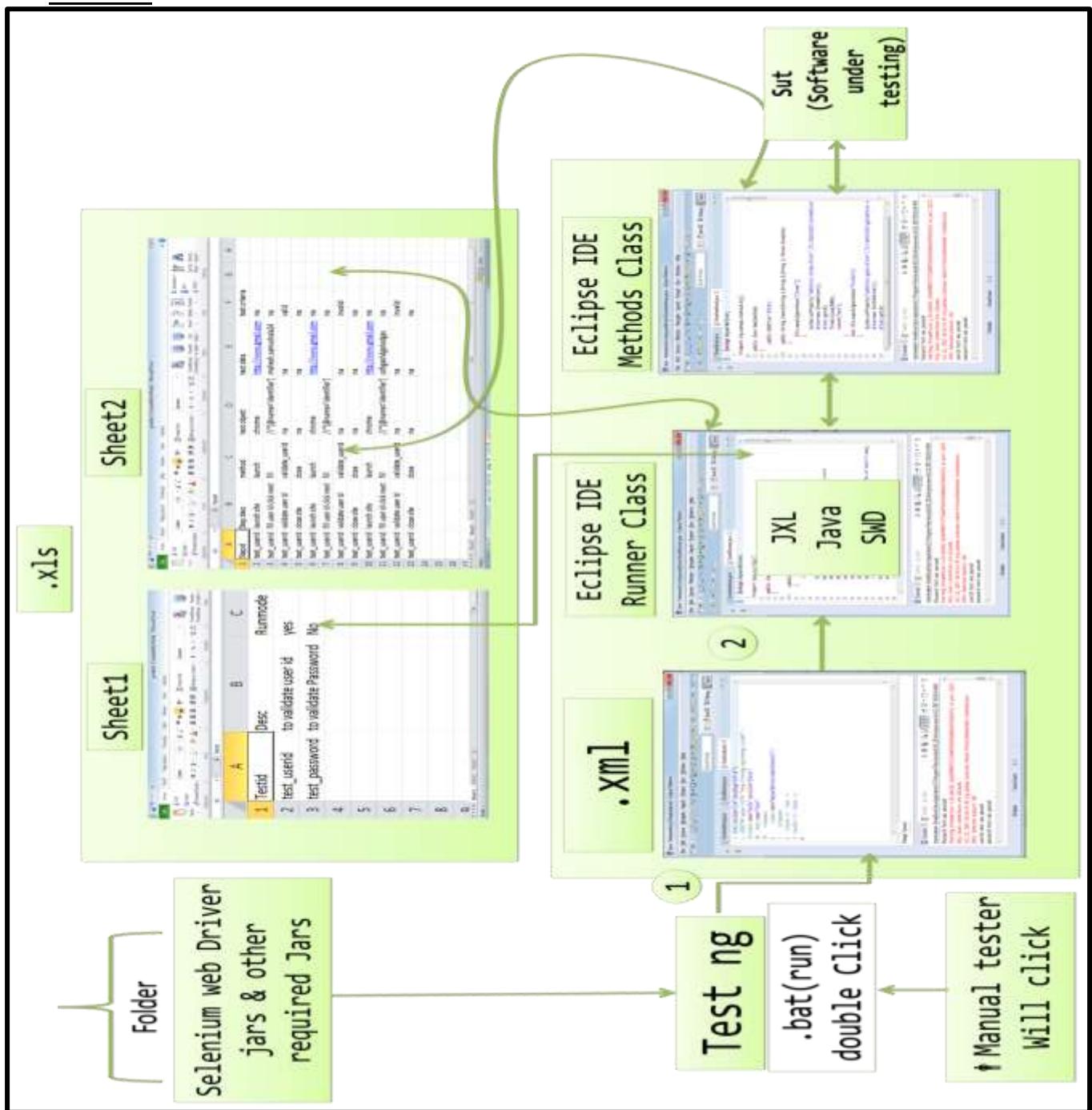
public class GmailRunner
{
 public static void main(String[] args) throws Exception
 {
 // open excel file for data reading
 File f = new File("gmailtest.xls");
 Workbook rwb = Workbook.getWorkbook(f); // Sheet1 because index0
 Sheet rsh1 = rwb.getSheet(0);
 Sheet rsh2 = rwb.getSheet(1); // Sheet2 because index2
 int not = rsh1.getRows();
 int nos = rsh2.getRows();
 int noc = rsh2.getColumns();
 System.out.println("No of tests" + not + "No of steps" + nos + "No of tests" + nos);
 // Open same Excel file for result writing
 WritableWorkbook wwb = Workbook.createWorkbook(f, rwb);
 WritableSheet wsh = wwb.getSheet(1); // sheet1 insert data
 // Creating Object to method class
 Gmailmethods gm = new Gmailmethods();
 Method m[] = gm.getClass().getMethods();
 // Keyword driver start with data driven
 // 0th row having name of columns in sheet1
 for (int i = 1; i < not; i++)
 {
 String tid = rsh1.getCell(0, i).getContents();
 String mode = rsh1.getCell(2, i).getContents();
 if (mode.equalsIgnoreCase("yes"))
 {
 // 0th row having name of columns in sheet2
 for (int j = 1; j < nos; j++)
 {
 String sid = rsh2.getCell(0, j).getContents();

 if (tid.equalsIgnoreCase(sid))
 {
 String mn = rsh2.getCell(2, j).getContents();
 String o = rsh2.getCell(3, j).getContents();
 String d = rsh2.getCell(4, j).getContents();
 String c = rsh2.getCell(5, j).getContents();
 }
 }
 }
 }
 }
}
```

```
System.out.println(mn);
System.out.println(o);
System.out.println(d);
System.out.println(c);
 for (int k = 0; k < m.length; k++)
 {
 if (mn.equalsIgnoreCase(m[k].getName()))
 {
 String r = (String) m[k].invoke(gm, o, d, c);
 Label l = new Label(noc, j, r);
 wsh.addCell(l);
 }
 }
}
}
wwb.write();
wwb.close();
rwb.close();
}
}
```

**Note 1:-**

“Gmail methods” classes have methods. We collected those on methods by using an array of “method” class in “java.lang.reflect” package of JDK.

Note 2:-

The above diagram was developed by Test Automater, we need to implement “Test Ng” process to execute framework without involvement of Eclipse IDE. Here, we need to change runner class like shown below

```
public class GmailRunner1
{
 @Test
 public void method() throws Exception
 {
 File f=new File("gmailtest.xls");
 Workbook rwb= Workbook.getWorkbook(f); //Sheet1 because index0
 Sheet rsh1=rwb.getSheet(0);
 Sheet rsh2=rwb.getSheet(1); //Sheet2 because index2
 int not=rsh1.getRows();
 int nos=rsh2.getRows();
 }
}
```

In above Example runner class code, “@Test” annotations are related “Test NG” plugin of Eclipse IDE. We need to follow below navigation to “Test NG” plugin to eclipse.

- ↓ Open Eclipse IDE
- ↓ Help Menu
- ↓ Eclipse market place
- ↓ Enter “Test NG” in find box under search box
- ↓ Click install for “Test NG” for Eclipse.
- ↓ Uncheck other plugin Except “Test NG”.
- ↓ Click Next.
- ↓ Select Accept agreement.
- ↓ until Finish.
- ↓ Restart Eclipse after installation
- ↓ Go TO runner class
- ↓ Replace main() method with @Test annotated method
- ↓ Right Click on corresponding project
- ↓ Select properties
- ↓ Select Java Bulid Path
- ↓ Select Libraries
- ↓ Select Add Libraries
- ↓ Select Test NG
- ↓ Click Finish
- ↓ **import** Test NG (org.testng.annotations.Test) library in corresponding method() Class.
- ↓ Run Runner Class.

#### Note 3:-

When we added “Test NG” Plugin to Eclipse IDE, we are able to “XML” Suite File by calling Runner Class.

- ↓ Right Click on corresponding project.
- ↓ Select “Test NG”.
- ↓ Create “Test NG”.
- ↓ Enter a name to suite file with .xml Extension.
- ↓ Click Finish.
- ↓ Delete Newly created class
- ↓ Open Suite File.
- ↓ Specify Runner Class Name

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Suite" parallel="false">
 <test name="Test">
 <classes>
 <class name="keyworddriven.GmailRunner1"/>
 </classes>
 </test> <!-- Test -->
</suite> <!-- Suite -->
```

#### Note 4:-

“Test NG” can allow as to run suite file thru a command batch command file. Here we need to collect all jars related project in a personal folder (SWD Jars, JXL Jar, Test NG Jar, Path of personal folder\org.testng.eclipse\lib and other required jars) & then we need to create a batch file like shown below.

```

mahesh.bat - Notepad
File Edit Format View Help
d:
Path of project
cd D:\batch232\framework\
Path of jars folder
java -cp "D:\batch232\jars*;
Path of classes (methods& runner)
D:\batch232\framework\bin"
Path of xml suite file
org.testng.TestNG D:\batch232\framework\src\Gmailsuite.xml

```

Save as batch file “Gmailbatch.bat” on desktop.

We need to save above like notepad file with filename.bat as extension.

#### **Note 5:-**

Sometimes manual tester request test Automater to run automation script's at specific time automatically, for this periodic execution. We can use continues integration tool like “Jenkins”

- ↓ Go to [www.google.com](http://www.google.com) site.
- ↓ Enter Jenkins download.
- ↓ Go to <https://jenkins.io/download/> site.
- ↓ Click on windows.

Windows

- ↓ Wait until download finish.
- ↓ Paste that download in personal folder.
- ↓ Extract that download in personal folder.
- ↓ Double click on Jenkins set up.
- ↓ Click next until finish.
- ↓ Open Jenkins page using browser
- ↓ <http://localhost:8080>
- ↓ Login Jenkins page by giving userid as admin & password as in (C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword)
- ↓ Click login
- ↓ Create New item
- ↓ Enter a name to project
- ↓ Select free style project option
- ↓ Click Ok
- ↓ Select  bulid periodically check option
- ↓ MINUTES Minutes in one hour (0-59)
- ↓ HOURS Hours in one day (0-23)
- ↓ DAYMONTH Day in a month (1-31)
- ↓ MONTH Month in a year (1-12)
- ↓ DAYWEEK Day of the week (0-7) where 0 and 7 are Sunday
- ↓ MINUTE (0-59), HOUR (0-23), DAY (1-31), MONTH (1-12), DAY OF THE WEEK (0-7)

(\*\* \* \* \* \* \*)

\*indicates every week/day/month/time(HH:MM)

(minutes	hours	day	month	weekday
(0-59	0-23	1-31	1 -12	0-7 )

Example:-

45 11 \* \* \* → every day 11:45 AM

45 \* \* \* \* → every hour 45 minute day, month, year

45 11 1 \* \* → every 1st month at 11:45am

- ↓ Click add build setup
- ↓ Select Execute window batch command option
- ↓ Specify path of batch file

Example:-



- ↓ Click Save
- ↓ Close Jenkins page

Note 6:-

If we want to run batch file using Jenkins now, we need follow below navigation

- ↓ Launch Jenkins page using browser  
(<http://localhost:8080>) ↪
- ↓ Do login with credentials & click ok.

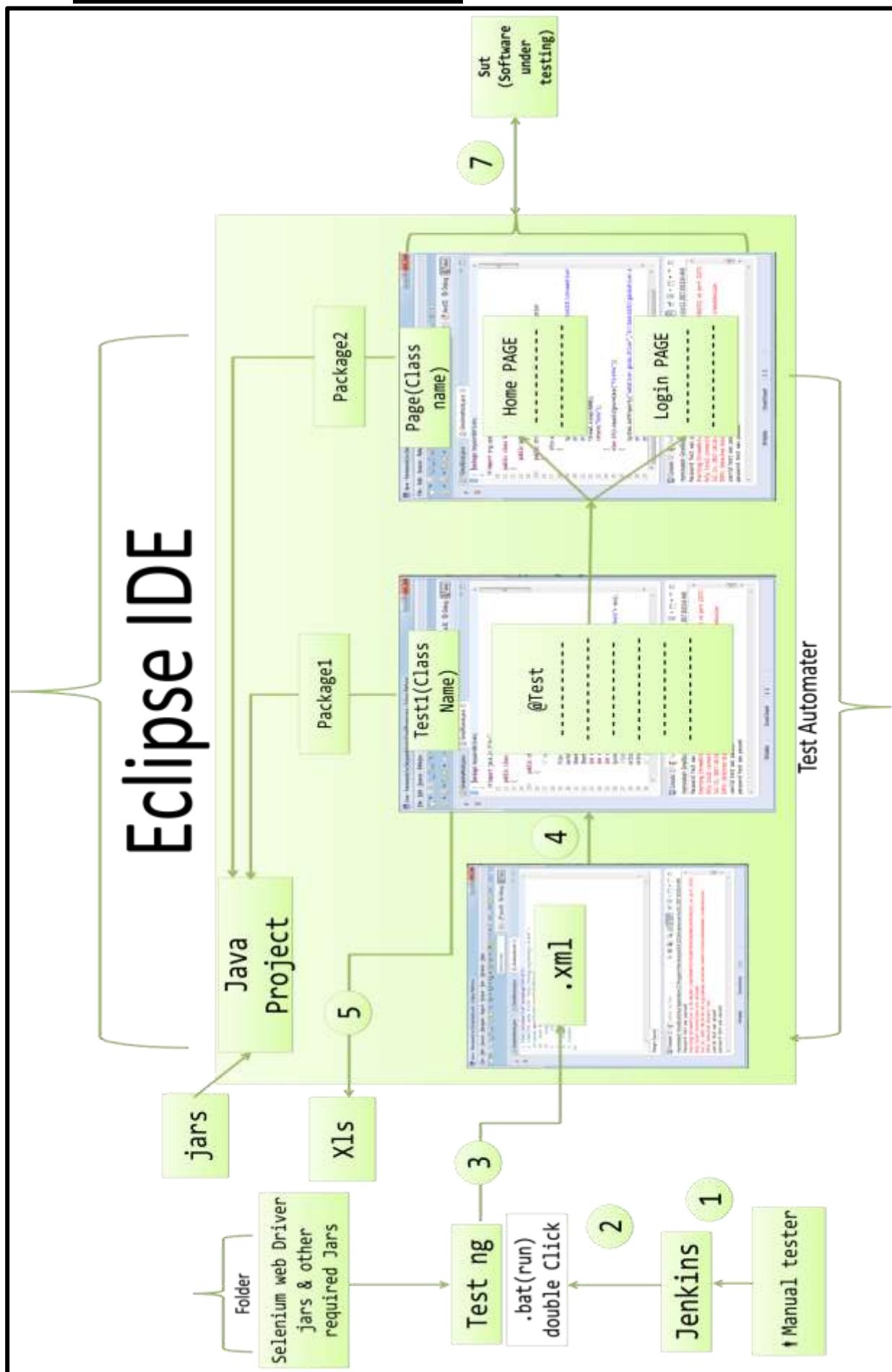
- ↓ Click item list icon of Jenkins
- ↓ Click All
- ↓ Select project
- ↓ Click OK.

**Note 7:-**

If we want to change existing project batch file path/Schedule time, we need to follow below navigation.

- ↓ Launch Jenkins page using browser  
(<http://localhost:8080>)
- ↓ Do login with credentials & click ok.
- ↓ Click item list icon of Jenkins
- ↓ Click All
- ↓ Go to Existing Project.
- ↓ Click on configure
- ↓ Change Time Schedule/bulid path (batch file path) if required.
- ↓ Click save.

## T.POM (Page Object Model)



**Step 1:-**

Download and install JDK8

(create JAVA\_HOME environmental variables with path of JDK & extend path variable with path of JDK\bin;)

**Step 2:-**

Download and launch ECLIPSE IDE(Juno Version)  
(browse a folder for work space)

**Step 3:-**

Create a new Java Project in Eclipse IDE (Project name as single word without space in lower case)

**Step 4:-**

Download Jars & associate to project  
(set latest compiler & JRE to java project

- ⇒ Download SWD Jars.
- ⇒ Associate to java project.
- ⇒ Download Sikulix setup and install.
- ⇒ Associate Sikulix API jar to java project if required in java project if required.
- ⇒ Download & associate “JXL” jars in Java project
- ⇒ Download & associate “Extent Report” jars in Java project if required.)

**Step 5:-**

Create two packages, named as “tests” & “pages”

**Step 6:-**

Create page classes in page package like shown below.  
(do not select main() in these page package)

```
//Login Page
package page;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;

public class Homepage
{
 public WebDriver driver;
 public By uid=By.name("identifier");
 public By uiderror=By.xpath("//*[contains(text(),'Couldn't find element') or
 contains(text(),'Enter an email'))][2]");

 public Homepage(WebDriver x)
 {
 this.driver=x;
 }

 public void UidFill(String x)
 {
 driver.findElement(uid).sendKeys(x,Keys.ENTER);
 }
}
```

```
//Login Page
package page;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;

public class Loginpage
{
 public WebDriver driver;
 public By pwd=By.name("password");
 public By pwderror=By.xpath("//*[contains(text(),'Wrong') or
 contains(text(),'Enter a password')])[2]");

 public Loginpage(WebDriver x)
 {
 this.driver=x;
 }

 public void PasswordFill(String x)
 {
 driver.findElement(pwd).sendKeys(x,Keys.ENTER);
 }

}
```

**Note:-**

"this" is built-in-object in java language. It is always working as object to current class.

**Step 7:-**

After completion of page class's creation with locator & operational methods, we need to concentrate on test classes creation.

Every test classes is responsible for test data reading & calling pages classes methods, validate output & send results to file.

**Navigation:-**

- ↓ Open Eclipse IDE
- ↓ Help Menu
- ↓ Eclipse market place
- ↓ Enter "Test NG" in find box under search box
- ↓ Click install for "Test NG" for Eclipse.
- ↓ Uncheck other plugin Except "Test NG".
- ↓ Click Next.
- ↓ Select Accept agreement.
- ↓ until Finish.
- ↓ Restart Eclipse after installation.
- ↓ Right click on test package in project.
- ↓ New
- ↓ Class
- ↓ Enter a name to test class
- ↓ Type @Test annotated method in that class like shown below

	A	B	C	D	E
1	user id	User id criteria	password	password criteria	
2		invalid			
3	<a href="mailto:maheshXXXX@gmail.com">maheshXXXX@gmail.com</a>	invalid			
4	<a href="mailto:maheshsaXXXX@gmail.com">maheshsaXXXX@gmail.com</a>	valid		invalid	
5	<a href="mailto:maheshsaXXXX@gmail.com">maheshsaXXXX@gmail.com</a>	valid	fgzdfgh	invalid	
6	<a href="mailto:maheshsaXXXX@gmail.com">maheshsaXXXX@gmail.com</a>	valid	dfaskdg	valid	
7					

```
//test package
package test;

import java.io.File;

import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

import page.Homepage;
import page.Loginpage;
public class test1
{
@Test
public void method() throws Exception
{
File f = new File("exceledata.xls");
Workbook rwb = Workbook.getWorkbook(f);
Sheet rsh = rwb.getSheet(0); // Sheet1 because index0
int nour = rsh.getRows(); // Get no of used rows
int nouc = rsh.getColumns(); // Get no of used columns
// open same excel sheet for writing result
WritableWorkbook wwb = Workbook.createWorkbook(f, rwb);
WritableSheet wsh = wwb.getSheet(0); // sheet1 insert data
Label da = new Label(nouc, 0, "Result");
wsh.addCell(da);
Label l1;
//pom with data driven testing
//0th row have names of columns
for (int i = 1; i < nour; i++)
{
 String u = rsh.getCell(0, i).getContents();
 String uc = rsh.getCell(1, i).getContents();
 String p = "";
 String pc = "";
 if (uc.equals("valid"))
 l1 = new Label(p, pc);
 else
 l1 = new Label("invalid", "invalid");
 wsh.addCell(l1);
}
}
}
```

```
{
 p = rsh.getCell(2, i).getContents();
 pc = rsh.getCell(3, i).getContents();
}
System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("http://www.gmail.com");
driver.manage().window().maximize();
Thread.sleep(5000);
Homepage HP =new Homepage(driver);
Loginpage LP=new Loginpage(driver);
HP.UidFill(u);
Thread.sleep(10000);
try
{
if (uc.equals("valid") && driver.findElement(LP.pwd).isDisplayed())
{
l1 = new Label(nouc, i, "test was passed");
wsh.addCell(l1);
// Password Testing
LP.PasswordFill(p);
Thread.sleep(10000);
//
if (pc.equals("valid") &&
driver.findElement(By.xpath("//*[text()='COMPOSE']")).isDisplayed())
{
l1 = new Label(nouc, i, "password test was passed");
wsh.addCell(l1);
}
else if (pc.equals("invalid") && driver.findElement(LP.pwderror).isDisplayed())
{
l1 = new Label(nouc, i, "password test was failed");
wsh.addCell(l1);
}
}
else if (uc.equals("invalid") && driver.findElement(HP.uiderror).isDisplayed())
{
l1 = new Label(nouc, i, "userid test was passed");
wsh.addCell(l1);
}
else
{
l1 = new Label(nouc, i, "userid test was failed");
wsh.addCell(l1);
}
}
catch (Exception e)
{
 l1 = new Label(nouc, i, "userid test was Terminated");
 wsh.addCell(l1);
}
// Close Site
driver.close();
}
wwb.write();
wwb.close();
rwb.close();
}
}
```

**Step 8:-**

Creating Suite file, after completion of test classes preparation, with “@Test” annotations method we need to create Test NG suite file for further execution.

**Navigation:-**

- ↓ Right click on test  package
- ↓ Select Test NG
- ↓ Create Test NG class
- ↓ Enter a name suite file with .xml extension
- ↓ Click finish
- ↓ Delete new test class file(.java newly created)
- ↓ Click on suite file
- ↓ Specify test class names like shown below

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="false">
 <test name="Test">
 <classes>
 <class name="test.Test1"/>
 </classes>
 </test> <!-- Test -->
</suite> <!-- Suite -->
```

- ↓ Run suite file.

**Step 9:-****Creating batch file**

Test NG allow us to run suite file through a batch command like shown below.



```
d:
cd D:\batch232\pomdatadriven
java -cp "D:\batch232\jars*;D:\batch232\pomdatadriven\bin"
org.testng.TestNG D:\batch232\pomdatadriven\src\pomtest.xml
```

(most of the test automaters can save above batch file on desktop.)

**Step 10:-**

- ↓ Schedule batch file execution in Jenkins.
- ↓ Go to [www.google.com](http://www.google.com) site.
- ↓ Enter Jenkins download.
- ↓ Go to <https://jenkins.io/download/> site.
- ↓ Click on windows.



- ↓ Wait until download finish.
- ↓ Paste that download in personal folder.
- ↓ Extract that download in personal folder.
- ↓ Double click on Jenkins set up.
- ↓ Click next until finish.
- ↓ Open Jenkins page using browser
- ↓ <http://localhost:8080>

- ↓ Login Jenkins page by giving userid as admin & password as in (C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword)
- ↓ Click login
- ↓ Create New item
- ↓ Enter a name to project
- ↓ Select free style project option
- ↓ Click OK
- ↓ Select  build periodically check option

**45 11 \* \* \* → every day 11:45 AM**

**45 \* \* \* \* → every hour 45 minute day, month, year**

**45 11 1 \* \* → every 1st month at 11:45am**

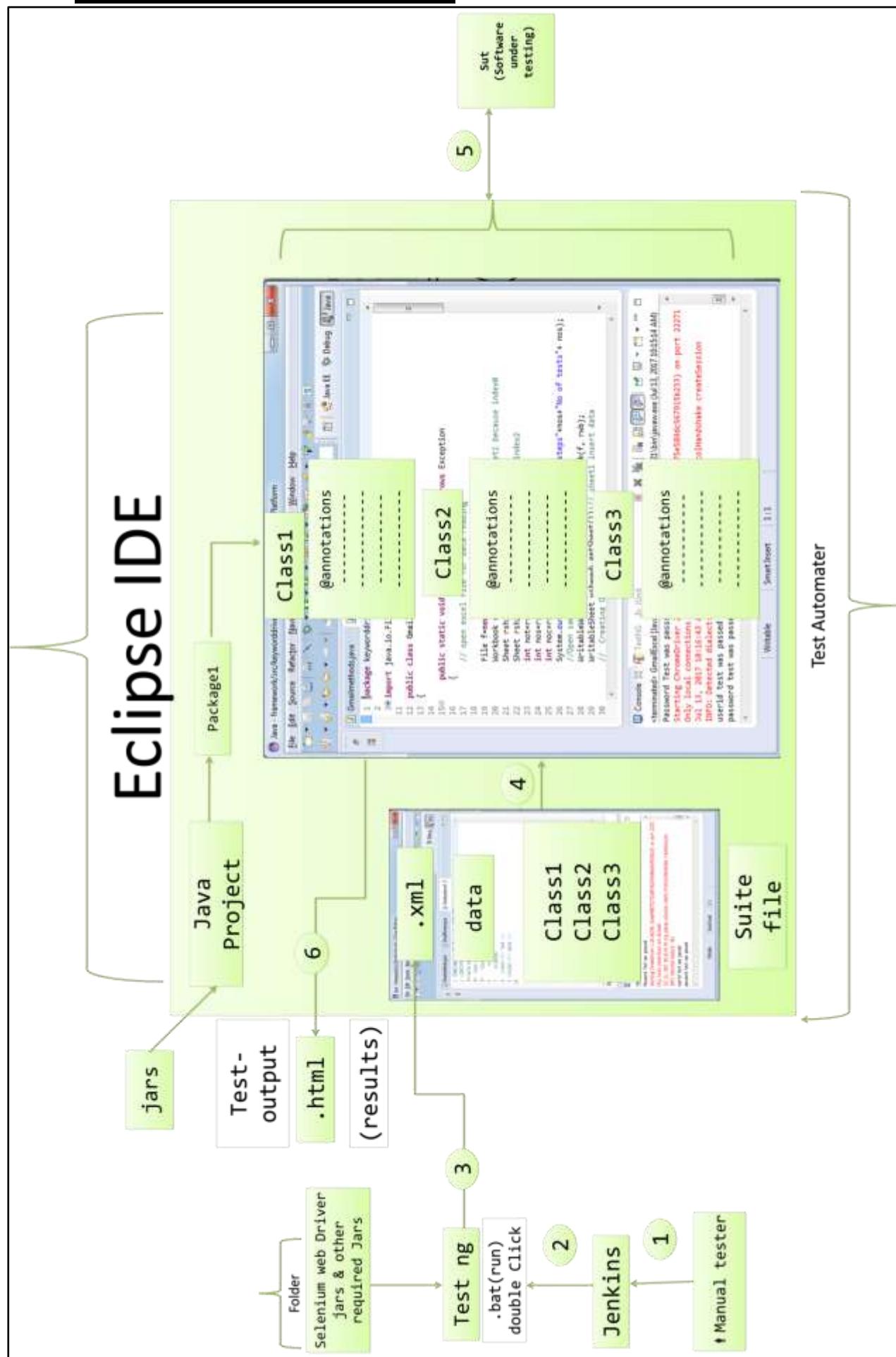
- ↓ Click add build setup
- ↓ Select Execute window batch command option
- ↓ Specify path of batch file

#### Example:-



- ↓ Click Save
- ↓ Close Jenkins page

## U. Module-driven framework



**Step 1:-**

Download and install JDK8

(create JAVA\_HOME environmental variables with path of JDK & extend path variable with path of JDK\bin;)

**Step 2:-**

Download and launch ECLIPSE IDE(Juno Version)  
(browse a folder for work space)

**Step 3:-**

Create a new Java Project in Eclipse IDE(Project name as single word without space in lower case)

**Step 4:-**

Download Jars & associate to project

(Set latest compiler & JRE to java project

- ⇒ Download SWD Jars.
- ⇒ Associate to java project.
- ⇒ Download Sikulix setup and install.
- ⇒ Associate Sikulix API jar to java project if required in java project if required.
- ⇒ Install Test NG plugin to Eclipse IDE if required.
- ⇒ ...etc

**Step 5:-**

- ↓ Create package in java project
- ↓ Create class in package
- ↓ Create annotated methods in each class like shown below

**Test scripts:-**

```
package gmailpack;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.Test;

public class Test1
{
 public WebDriver driver;

 @Test(priority=0)
 public void launch() throws Exception
 {

 System.setProperty("webdriver.chrome.driver","d:\\chromedriver.exe");
 driver = new ChromeDriver();
 driver.get("http://www.gmail.com");
 driver.manage().window().maximize();
 Thread.sleep(5000);
 }

 @Test(priority=1)
 public void userid_test(String u, String c) throws Exception
 {
 driver.findElement(By.name("identifier")).sendKeys(u,Keys.ENTER);
 Thread.sleep(5000);
 }
}
```

```

try
{
 if (c.equalsIgnoreCase("valid") &&
 driver.findElement(By.name("password")).isDisplayed())
 {
 Thread.sleep(5000);
 Assert.assertTrue(true);
 }
 else if (c.equalsIgnoreCase("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'Couldn') or
 contains(text(),'Enter an email')][2]")).isDisplayed())
 {
 Thread.sleep(5000);
 Assert.assertTrue(true);
 }
 else
 {
 Thread.sleep(5000);
 Assert.fail();
 }
}
catch (Exception e)
{
 Assert.assertTrue(false, "Terminated");
}
}

@Test(priority=2)
public void close()
{
 driver.close();
}
}

```

**Note 1:-**

- In above test NG based we used “@Test” & “@parameters” annotations for methods.
- If we are interested, we able to use other annotations like shown below:-

**@BeforeSuite :-**

The annotated method will be run before all tests in suite have run.

or

(The annotated method will be run only once before all tests in this suite have run.)

**@AfterSuite :-**

The annotated method will be run after all tests in suite have run.

or

The annotated method will be run only once after all tests in this suite have run.

**@BeforeTest :-**

The annotated method will be run before any test method belonging to the classes inside the <test> tag is run.

**@AfterTest :-**

The annotated method will be run after all the test methods belonging to the classes inside the <test> tag have run.

**@BeforeGroup :-**

The annotated method will be run before all methods in group have run.

**@AfterGroup :-**

The annotated method will be run after all methods in group have run.

**@BeforeClass :-**

The annotated method will be run before the first test method in current class is invoked

**@AfterClass :-**

The annotated method will be run after all test methods in current class is have been run.

**@BeforeMethod :-**

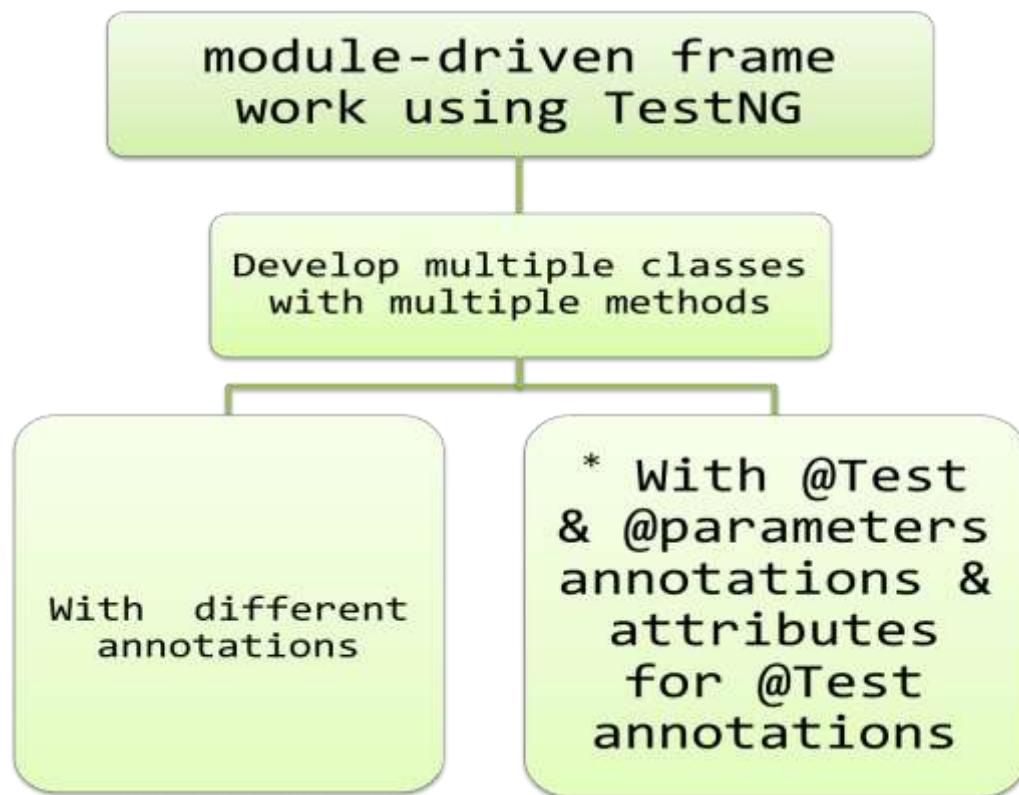
The annotated method will be run before each test method

**@AfterMethod :-**

The annotated method will be run after each test method

**Note 2:-**

⇒ In general we can use “@Test” & “@Parameters” annotations for methods in classes instead of all annotations. But to execute “@Test” & “@Parameters” annotates in order, we can use attributes of @Test annotations.

**@Test:-**

This annotation can allow below attributes to schedule methods for execution in classes.

**alwaysRun :-**

If set to true, this test method will always be run even if it depends on a method that failed.

```
@Test(alwaysRun = True)
Public void method()
{

}
```

**dependsOnGroups :-**

This attributes method will run when given group methods are passed.

**`@Test(groups={"A"})`**



**`@Test(dependsOnGroups={"A"})`**


**Example:-**

```
import org.testng.Assert;
import org.testng.annotations.Test;
public class DependencyTestUsingAnnotation {
 String message = "mahesh";
 MessageUtil messageUtil = new MessageUtil(message);

 @Test(groups = { "init" })
 public void testPrintMessage() {
 System.out.println("Inside testPrintMessage()");
 message = "mahesh";
 Assert.assertEquals(message, messageUtil.printMessage());
 }

 @Test(dependsOnGroups = { "init.*" })
 public void testSalutationMessage() {
 System.out.println("Inside testSalutationMessage()");
 message = "Hi!" + "mahesh";
 Assert.assertEquals(message, messageUtil.salutationMessage());
 }

 @Test(groups = { "init" })
 public void initEnvironmentTest() {
 System.out.println("This is initEnvironmentTest");
 }
}
```

**dependsOnMethods :-**

This attribute method will run when given methods are passed.

```
@Test
```

```
Public void method1()
```

```
{
```

```

```

```

```

```
}
```

```
@Test(dependsOnMethods={ 'method1' })
```

```
Public void method2()
```

```
{
```

```

```

```

```

```
}
```

**Example:-**

```
import org.testng.Assert;
import org.testng.annotations.Test;

public class DependencyTestUsingAnnotation {
 String message = "Mahesh";
 MessageUtil messageUtil = new MessageUtil(message);

 @Test
 public void testPrintMessage() {
 System.out.println("Inside testPrintMessage()");
 message = "Mahesh";
 Assert.assertEquals(message, messageUtil.printMessage());
 }

 @Test(dependsOnMethods = { "initEnvironmentTest" })
 public void testSalutationMessage() {
 System.out.println("Inside testSalutationMessage()");
 message = "Hi!" + "Mahesh";
 Assert.assertEquals(message, messageUtil.salutationMessage());
 }

 @Test
 public void initEnvironmentTest() {
 System.out.println("This is initEnvironmentTest");
 }
}
```

**enabled:-**

This attribute is useful to activate & deactivate a method from execution.

```
@Test(enabled=false)// skip method from execution
Public void method1()
{

}
```

**Example:-**

```
import org.testng.Assert;
import org.testng.annotations.Test;

public class IgnoreTest {
 String message = "Mahesh";
 MessageUtil messageUtil = new MessageUtil(message);

 @Test(enabled = false)
 public void testPrintMessage() {
 System.out.println("Inside testPrintMessage()");
 message = "Mahesh";
 Assert.assertEquals(message, messageUtil.printMessage());
 }

 @Test
 public void testSalutationMessage() {
 System.out.println("Inside testSalutationMessage()");
 message = "Hi!" + "Mahesh";
 Assert.assertEquals(message, messageUtil.salutationMessage());
 }
}
```

**expectedExceptions:-**

The list of exceptions that a test is expected to throw. If no exception or a different exception is thrown, this test will be marked a failure.

```
@Test(expectedExceptions={"No such Element"})
```

```
Public void method1()
{

}
```

**groups:-**

This attribute is used to make multiple methods as a group.

```
@Test(group={"A"})
Public void method1()
{

}

@Test(group={"A", "B"})
Public void method2()
{

}

@Test(group={"B"})
Public void method3()
{

```

**Example:-**

```
import org.testng.Assert;
import org.testng.annotations.Test;
public class GroupTestExample {
 String message = "@gmail.com";
 MessageUtil messageUtil = new MessageUtil(message);

 @Test(groups = { "functest", "checkintest" })
 public void testPrintMessage() {
 System.out.println("Inside testPrintMessage()");
 message = "@gmail.com";
 Assert.assertEquals(message, messageUtil.printMessage());
 }

 @Test(groups = { "checkintest" })
 public void testSalutationMessage() {
 System.out.println("Inside testSalutationMessage()");
 message = "mahesh" + ".@gmail.com";
 Assert.assertEquals(message, messageUtil.salutationMessage());
 }

 @Test(groups = { "functest" })
 public void testingExitMessage() {
 System.out.println("Inside testExitMessage()");
 message = "samu" + "mahesh" + ".@gmail.com";
 Assert.assertEquals(message, messageUtil.exitMessage());
 }
}
```

**invocationCount:-**

The number of times this method should be invoked.

```
@Test(invocationCount = 10) // Method will run 10 times
```

```
Public void method1()
{

}
```

**Example:-**

```
public class TestNgTest
{
 @Test(invocationCount = 10)
 public void test1ShouldFail() throws InterruptedException
 {
 Thread.sleep(2000);
 }
}
```

**invocationTimeOut:-**

We can use this attribute along with invocation count, attribute to give time limit for execution.

```
@Test(invocationTimeOut = 5000, invocationCount = 10)
// Method will run 10 time, Time out is in millseconds
```

```
Public void method1()
{

}
```

**Example:-**

```
public class TestNgTest
{
 @Test(invocationTimeOut = 10, invocationCount = 10)
 public void test1ShouldFail() throws InterruptedException
```

```

{
 Thread.sleep(2000);
}

@Test(invocationTimeOut = 100, invocationCount = 10)
public void test2ShouldFail() throws InterruptedException
{
 Thread.sleep(2000);
}

@Test(invocationTimeOut = 500, invocationCount = 10)
public void test3ShouldFail() throws InterruptedException
{
 Thread.sleep(2000);
}
}

```

**priority:-**

we can use this attribute to schedule/orderize methods in execution.

`@Test(priority=0)// here priority index starts with zero`

```

public void method1()
{

}

```

`@Test(priority=1)`

```

public void method2()
{

}

```

**Example:-**

```

public class Test1
{
 public WebDriver driver;

 @Test(priority=0)
 public void launch() throws Exception
 {
 System.setProperty("webdriver.chrome.driver","d: \chromedriver.exe");
 driver = new ChromeDriver();
 driver.get("http://www.gmail.com");
 driver.manage().window().maximize();
 Thread.sleep(5000);
 }
}

```

```

 }

 @Test(priority=1)
 public void userid_test(String u,String c) throws Exception
 {
 driver.findElement(By.name("identifier")).sendKeys(u,Keys.ENTER);
 Thread.sleep(5000);
 if (c.equalsIgnoreCase("valid") &&
 driver.findElement(By.name("password")).isDisplayed())
 {
 Thread.sleep(5000);
 Assert.assertTrue(true);
 }
 else if (c.equalsIgnoreCase("invalid") &&
 driver.findElement(By.xpath("//*[contains(text(),'Couldn't') or
 contains(text(),'Enter an email')][2]")).isDisplayed())
 {
 Thread.sleep(5000);
 Assert.assertTrue(true);
 }
 else
 {
 Thread.sleep(5000);
 Assert.fail();
 }
 }

 @Test(priority=2)
 public void close()
 {
 driver.close();
 }
}

```

**timeOut:-**

we can use this attributes to specify time limit for one time execution in millseconds.

```

@Test(priority=0, timeOut=5000)
// here priority index starts with zero, timeOut in Mill sec

```

```

public void method1()
{

}

```

**Example:-**

```

import org.testng.annotations.*;
public class TestNGTimeTest
{
 @Test(timeOut = 1000)
 public void waitLongTime() throws Exception {
 Thread.sleep(1001);
 }
}

```

**Note 3:-**

To use suite file data in method body, we can use “@parameters” or “@DataProvider”

**Step 6:-**

Developing Suite File(.xml File)

To create xml file to Run Class, we need to follow below

**Navigation:-**

- ↓ Right click on package in corresponding project
- ↓ Select Test NG
- ↓ Select Create Test NG class
- ↓ Enter a name suite file with .xml extension
- ↓ Click finish
- ↓ Delete new test class file(test class were already created with annotated method).
- ↓ Extend suite file code like shown below.

```

<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="false">

 <test name="iteration1">
 <parameter name="uid" value="" />
 <parameter name="criteria" value="invalid" />
 <classes>
 <class name="gmailpack.Test1" />
 </classes>
 </test> <!-- Test -->

 <test name="iteration2">
 <parameter name="uid" value="xxxxxxxx" />
 <parameter name="criteria" value="invalid" />
 <classes>
 <class name="gmailpack.Test1" />
 </classes>
 </test> <!-- Test -->

 <test name="iteration3">
 <parameter name="uid" value="xxxxxxxx" />
 <parameter name="criteria" value="valid" />
 <classes>
 <class name="gmailpack.Test1" />
 </classes>
 </test> <!-- Test -->
</suite> <!-- Suite -->

```

**Note 1:-**

From the above suite file <suite> tag is having multiple <test> tag is having multiple <parameter> & <classes> tag. <classes> tag is having multiple <class> tag. <class> tag is specified a class name, which class consists multiple annotated method

**Note 2:-**

Test NG based module driven framework can allow us to run multiple test classes as parallel for it we need to perform changes in suite file like shown below :-

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="tests" thread-count="3">
<!-- parallel="tests"→ to run <test> tag parallelly, thread-count="3" to
run number of test tags -->

<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="classes" thread-count="3">
<!-- parallel="classes"→ to run <classes> tag parallelly, thread-count="3"
to run number of test classes -->

<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="methods" thread-count="3">
<!-- parallel="methods"→ to run <methods>tag parallelly, thread-count="3"
to run number of test methods -->
```

By practically methods class will not work in parallel  
Ex:- Login & Logout will not work at a time.

**Step 7:-****Creating batch file**

```
d:
cd D:\batch232\moduledriven
java -cp "D:\batch232\jars*;D:\batch232\moduledriven\bin"
org.testng.TestNG D:\batch232\moduledriven\src\mdfgmail.xml
```

**Note 1:-**

In Test NG based module driven framework, we can result as html file in test-output folder of project.

D:\batch232\moduledriven\test-output\index.html  
**Path of project folder**

Folder created automatically

Latest Result

D:\batch232\moduledriven\test-output\old\index.html  
Path of project folder

Folder created automatically

Previous Output\Result

#### Step 8:-

- ↓ Schedule batch file execution in Jenkins.
  - ↓ Go to [www.google.com](http://www.google.com) site.
  - ↓ Enter Jenkins download.
  - ↓ Go to <https://jenkins.io/download/> site.
  - ↓ Click on windows.
- Windows
- ↓ Wait until download finish.
  - ↓ Paste that download in personal folder.
  - ↓ Extract that download in personal folder.
  - ↓ Double click on Jenkins set up.
  - ↓ Click next until finish.
  - ↓ Open Jenkins page using browser
  - ↓ <http://localhost:8080>
  - ↓ Login Jenkins page by giving userid as admin & password as in (C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword)
  - ↓ Click login
  - ↓ Create New item
  - ↓ Enter a name to project
  - ↓ Select free style project option
  - ↓ Click Ok
  - ↓ Select  bulid periodically check option

45 11 \* \* \* → every day 11:45 AM

45 \* \* \* \* → every hour 45 minute day, month, year

45 11 1 \* \* → every 1st month at 11:45am

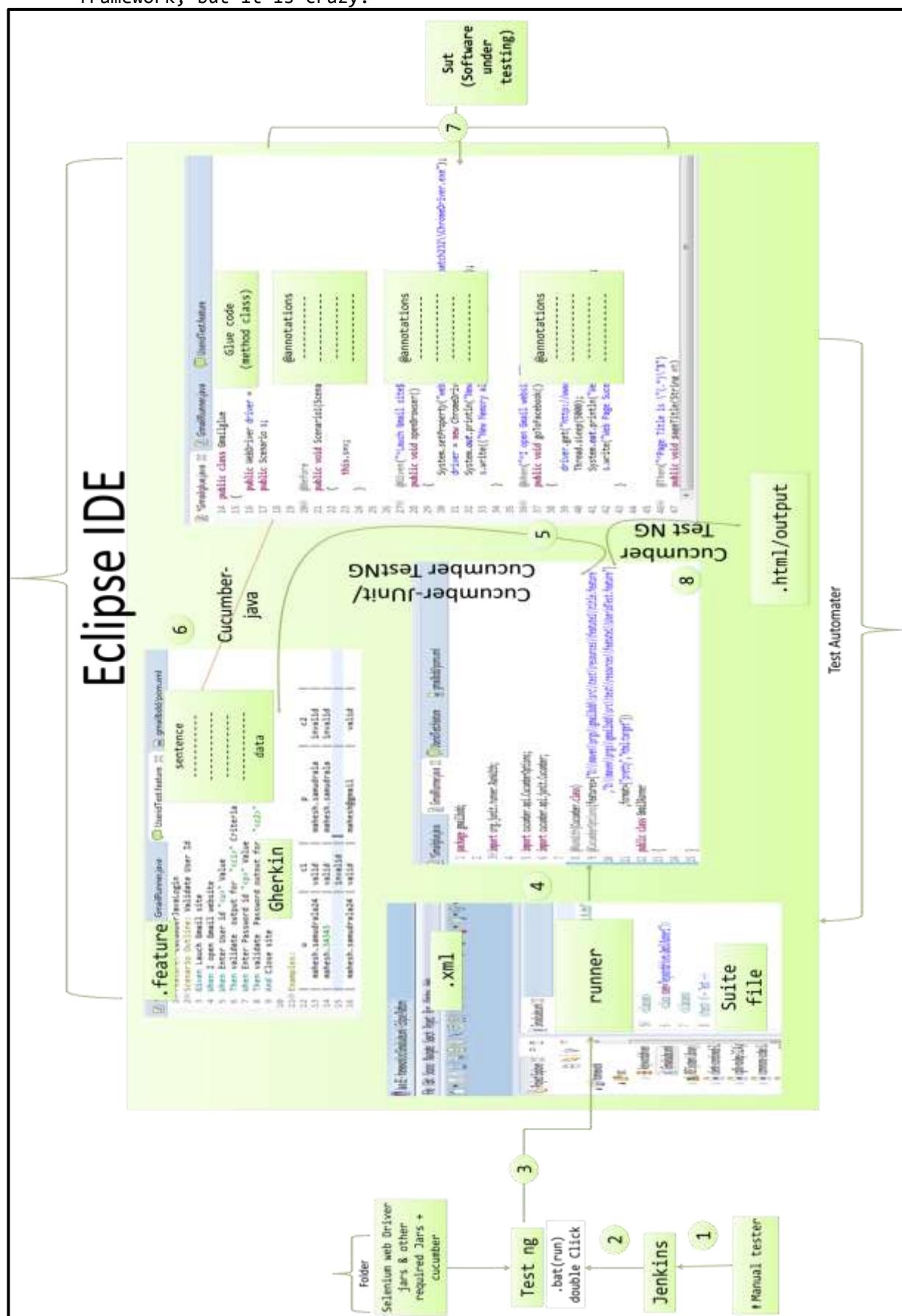
- ↓ Select Click add build setup
- ↓ Select Execute window batch command option
- ↓ Specify path of batch file

**Example:-**

- ↓ Click Save
- ↓ Close Jenkins page

## V. BDD framework using cucumber:-

BDD stands for Behavior Driven Development, it is best readymade framework for test automaters, this frame work was derived from keyword driven framework, but it is crazy.



**Step 1:-**

Download and install JDK8

(create **JAVA\_HOME** environmental variables with path of JDK & extend path variable with path of JDK\bin;)

**Step 2:-**

Download and launch ECLIPSE IDE(Juno Version)

(Browse a folder for work space)

**Step 3:-**

- ↓ Install maven plugin to Eclipse IDE
- ↓ Open Eclipse IDE
- ↓ Help MENU
- ↓ Install new software
- ↓ Click add
- ↓ Enter name as “M2Eclipse” & Enter location as  
[“<http://download.eclipse.org/technology/m2e/milestones/1.0>”](http://download.eclipse.org/technology/m2e/milestones/1.0)
- ↓ Click ok
- ↓ Select  maven integration for eclipse
- ↓ Click next
- ↓ Accept License Agreement
- ↓ Click Finish
- ↓ Restart Eclipse IDE After Completion of Installation
- ↓ File Menu
- ↓ New
- ↓ Other
- ↓ Maven Project
- ↓ Click Next
- ↓ Select  create a simple project option
- ↓ Click Next
- ↓ Enter company name as Group Id- mindq
- ↓ Enter project name as Artifact Id- gmailbdd
- ↓ Click Finish

**Note 1:-**

In general, maven project can provide 4 folders for package & classes.

- ⇒ src/main/java.
- ⇒ src/main/resources.
- ⇒ src/test/java.
- ⇒ src/test/resources.

**Note 2:-**

Every maven project can provided pom.xml file. In this file we can add required jars dependencies code like shown below

```

<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>gmail</groupId>
 <artifactId>gmailbdd</artifactId>
 <version>0.0.1-SNAPSHOT</version>
 <dependencies>
 <dependency>
 <groupId>org.seleniumhq.selenium</groupId>
 <artifactId>selenium-java</artifactId>

```

```

 <version>3.4.0</version>
 </dependency>
 <dependency>
 <groupId>info.cukes</groupId>
 <artifactId>cucumber-java</artifactId>
 <version>1.1.5</version>
 <scope>test</scope>
 </dependency>
 <dependency>
 <groupId>info.cukes</groupId>
 <artifactId>cucumber-junit</artifactId>
 <version>1.1.5</version>
 <scope>test</scope>
 </dependency>

 <dependency>
 <groupId>junit</groupId>
 <artifactId>junit</artifactId>
 <version>4.11</version>
 <scope>test</scope>
 </dependency>

</dependencies>

</project>

```

**Note 3:-**

When we can save pom.xml, corresponding maven project can get ALL jars related dependencies automatically corresponding download jars are available in “**maven dependencies**” option in corresponding maven project.

**Note 4:-**

- ↓ We need to set java compiler & JRE to latest to corresponding maven project
- ↓ Right click on maven project
- ↓ Set compiler to latest & JRE to latest

**Step 4:-**

- ↓ Launch the *Eclipse IDE* and from Help menu, click “**Install New Software**”.
- ↓ You will see a dialog window, click “**Add**” button.
- ↓ Type name as you wish, let’s take “**Cucumber**” and type “<http://cucumber.github.com/cucumber-eclipse/update-site>” as location.
- ↓ Click **OK**.
- ↓ You come back to the previous window but this time you must see **Cucumber Eclipse Plugin** option in the available software list. Just **Check** the box and press “**Next**” button.
- ↓ Click on **Next**.
- ↓ Click “**I accept the terms of the license agreement**” then click **Finish**.
- ↓ Let it install, it will take few seconds to complete.
- ↓ You may or may not encounter a Security warning, if in case you do just click **OK**.
- ↓ You are all done now, just Click **Yes**.
- ↓ Restart Eclipse IDE

**Step 5:-**

In BDD frame work, we need to prepare feature file with Gherkin sentences to define test scenario & cases. Gherkin language is providing keywords like shown below & it's

Gherkin Keyword	Description
<b>Feature:</b>	Module name in corresponding sprint/software
<b>Scenario:</b>	Test scenario sentence with validate/verify/check
<b>Scenario Outline:</b>	Test scenario sentence, but it can be validates for multiple data(data-driven)
<b>Examples:</b>	To provide data for variables in table format using pipe symbol
<b>Given</b>	To write precondition //operation
<b>When</b>	To write key operation (corresponding operation) //operation
<b>Then</b>	To write observation/validation/assertion/Verification/test case
<b>And</b>	To execute Given, When, Then steps
<b>But</b>	To execute Given, When, Then steps
#	Comments
“”	To mark data in sentences
“< >	To mark variable sentences
(pipe)	To write data for variables in “Example:” tag
@	To write annotations /tag /labels

To create Gherkin based feature files we need to follow below navigation

- ⇒ Go to src/test/resources folder in maven project
- ⇒ Right click on it
- ⇒ New
- ⇒ Package
- ⇒ Enter a name as single word in lower case
- ⇒ Click ok
- ⇒ Right click on package
- ⇒ New
- ⇒ File
- ⇒ Enter file name with .feature extension
- ⇒ Click ok
- ⇒ Write Gherkin sentences in that feature file like shown below

**Example:-**

Title\_test.feature

**Test Script:-**

```
Feature: CucumberJavaLogin
Scenario: Validate Page Title
Given Launch Gmail site
When I open Gmail website
Then Page Title is "Gmail"
And Close site
```

**Example:-**

```
UserId_test.feature
```

**Test Script:-**

```
Feature: CucumberJavaLogin
Scenario Outline: Validate User Id
Given Launch Gmail site
When I open Gmail website
When Enter User id "<u>" Value
Then validate output for "<c1>" Criteria
When Enter Password id "<p>" Value
Then validate Password output for "<c2>" Criteria
And Close site
```

**Examples:**

u	c1	p	c2
xxxxxxxxxx	valid	xxxxxxxxxx	invalid
xxxxxxxxxx	valid	xxxxxxxxxx	invalid
	invalid		
xxxxxxxxxx	valid	xxxxxxxxxx	valid

**Note 1:-**

⇒ We need create & save feature file in src/test/resources folder package in corresponding maven project

**Note 2:-**

One feature file is related one feature (modules) only, but they are having one or more scenario (or) scenario outline.

**Note 3:-**

Every Scenario (or) Scenario Outline will be defined with multiple sentences by using Given, When, Then, And, But, Examples:, .....etc keywords in Gherkin language.

**Step 6:-**

Developing Glue Code (method's class)

- ↓ Go to src/test/java in maven project
- ↓ right click on it
- ↓ new
- ↓ package
- ↓ Enter a name  
(Single word as lower case)
- ↓ Click OK
- ↓ Right click on that package
- ↓ New
- ↓ Class without main method()
- ↓ Enter a name  
(Single word but start with upper cases)
- ↓ Click OK
- ↓ Develop Glue code with respect to sentence's in feature file like shown below:-

**Test Script:-**

```
package gmailbdd;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
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;
public class Gmailglue
{
 public WebDriver driver = null;
 public Scenario s;
 @Before
 public void Scenario01(Scenario x)
 {
 this.s = x;
 }
 @Given("^Lauch Gmail site$")
 public void openBrowser() throws Exception
 {
 System.setProperty("webdriver.chrome.driver", "D:\\ChromeDriver.exe");
 driver = new ChromeDriver();
 }

 @When("^I open Gmail website$")
 public void gmailsite() throws Exception
 {
 driver.get("http://www.gmail.com");
 Thread.sleep(5000);
 s.write("Web Page Sucessfull Launched");
 }
 @Then("^Page Title is \"(.*)\"$")
 public void pageTitle(String et)
 {
 String at = driver.getTitle();
 if (at.equalsIgnoreCase(et))
 {
 s.write("Page Title is Gmail Passed");
 }
 else
 {
 s.write("Page Title is Gmail Failed");
 }
 }
 @And("^Close site$")
 public void closesite()
 {
 driver.close();
 s.write("Close site");
 }
 @When("^Enter User id \"(.*)\" Value$")
 public void user_Id(String uid) throws Exception
 {
 driver.findElement(By.name("identifier")).sendKeys(uid, Keys.ENTER);
 Thread.sleep(5000);
 }
 @Then("^validate output for \"(.*)\" Criteria$")
 public void criterzia(String c)
 {
 try
 {
 if (c.equalsIgnoreCase("valid") &&
 driver.findElement(By.name("password")).isDisplayed())
 {
 s.write("Userid Test passed");
 }
 else if (c.equalsIgnoreCase("invalid") && driver

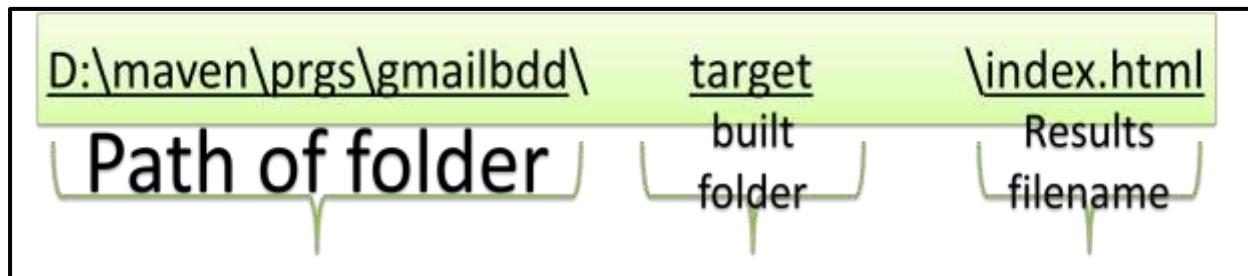
```

```
.findElement(By.xpath("//*[contains(text(),'Could') or contains(text(),'Enter an Email')])[2]")).isDisplayed())
{
 s.write("Userid Test passed");
}
else
{
 s.write("Userid Test failed");
}
}
catch (Exception e)
{
 s.write("Test Terminated");
}
}
@When("^Enter Password id \"(.*)\" Value")
public void password(String pd) throws Exception
{
try
{
if (driver.findElement(By.name("password")).isDisplayed())
{
driver.findElement(By.name("password")).sendKeys(pd, Keys.ENTER);
s.write("User id valid");
Thread.sleep(5000);
}
else
{
 System.out.println("User id invalid");
 s.write("User id invalid");
}
}
catch (Exception e)
{
 s.write("User id terminated");
}
}
@Then("^validate Password output for \"(.*)\" Criteria")
public void passwordcritertia(String c) throws Exception
{
try
{
if (c.equalsIgnoreCase("valid") &&
driver.findElement(By.xpath("//*[text()='COMPOSE']")).isDisplayed())
{
 s.write("password Test passed");
 Thread.sleep(5000);
}
else if (c.equalsIgnoreCase("invalid") &&
driver.findElement(By.xpath("//*[contains(text(),'Wrong') or
contains(text(),'Enter a password'))][2]")).isDisplayed())
{
 s.write("password Test passed");
}
else
{
 s.write("Password Test");
}
}
catch (Exception e)
{
 s.write("Test Terminated");
}
}
```

}

**Note:-**

- Cucumber-java jar file(Maven project was downloaded this jar automatically via pom.xml) is providing “Scenario” class & automatically @Before, @After, @Given, @When, @Then, @And, @But ..etc
- Scenario class object is useful to generate results that HTML file

**Step 7:-**

- ⇒ After completion of feature file creation in package of `src/test/resources` folder & corresponding glue code class creation in package of `src/test/java` in maven project, we need to develop a JUnit runner class in `src/test/java` folder.

**Navigation:-**

- ↓ Go to `src/test/java`.
- ↓ Right click on internal package.
- ↓ New
- ↓ class without main() method.
- ↓ Enter a name (single word & start with upper case).
- ↓ Click OK
- ↓ Implement annotations to that class like shown below.

**Test Script:-**

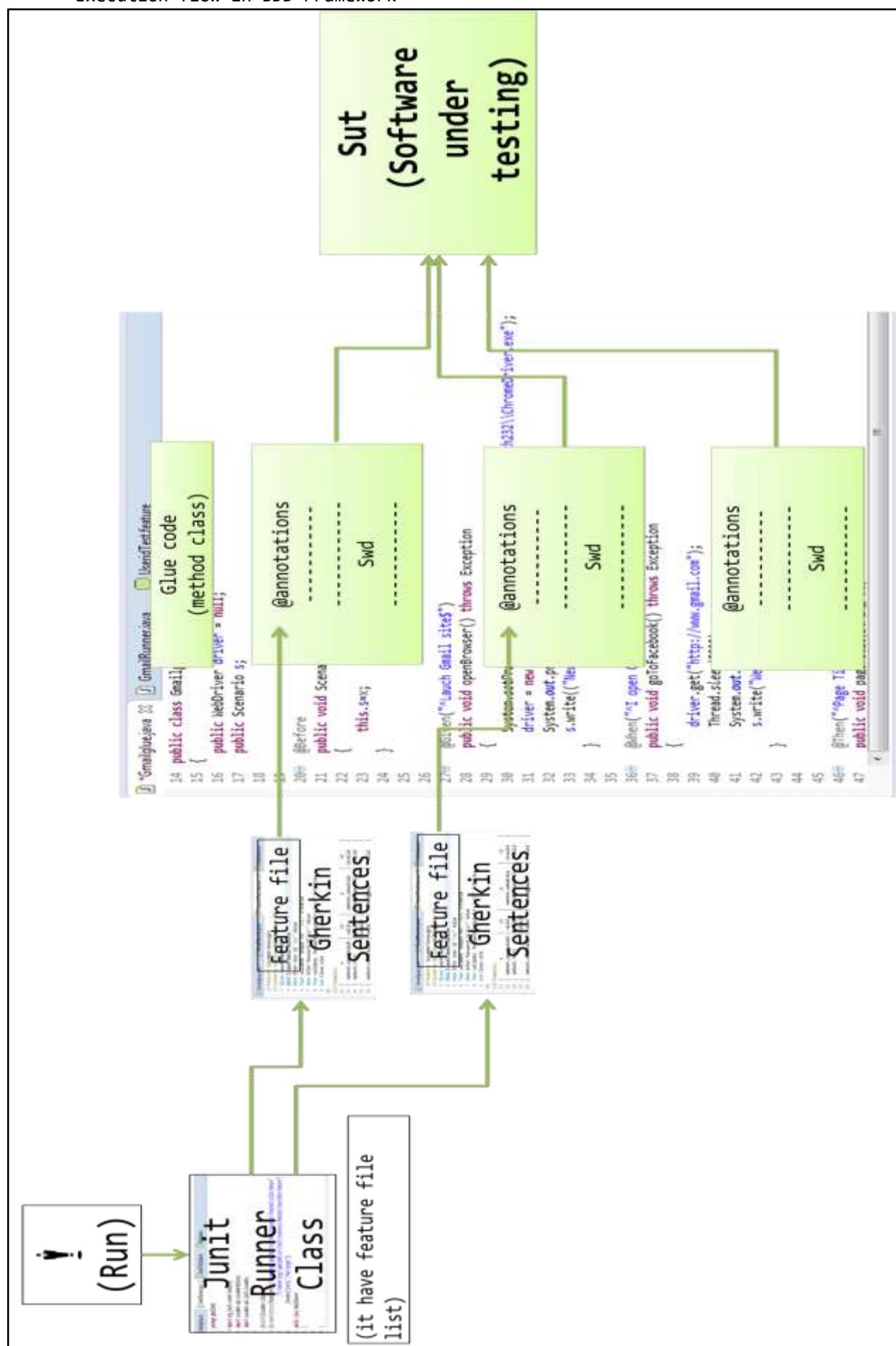
```

package gmailbdd;
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;
@RunWith(Cucumber.class)
@CucumberOptions(features={"D:\\maven\\prgs\\gmailbdd\\src\\test\\resources\\feature1\\title.feature",
"D:\\maven\\prgs\\gmailbdd\\src\\test\\resources\\feature1\\UseridTest.feature"})
,format={"pretty","html:target"})
public class GmailRunner
{
}
@RunWith → Junit annotation
@CucumberOptions → Cucumber annotations

```

**Note 1:-**

Execution flow in BDD Framework



**Note 2:-**

For JUnit runner class, we need two annotations like @RunWith, @CucumberOptions, but @CucumberOptions annotations can allow attributes with values.

- Features :- To give path of features files {1,}
- Monochrome :- To remove funny symbols in Eclipse console output
- Tags:- To filter scenario in feature during Execution

**Example:-**

The diagram illustrates the execution flow of Cucumber annotations across three Java files: `GmailRunner.java`, `UserTestFeature.java`, and `GmailFeature.java`.

**GmailRunner.java:**

```

1 package gmail;
2
3 import org.junit.runner.RunWith;
4 import cucumber.api.CucumberOptions;
5 import cucumber.api.junit.Cucumber;
6
7 @RunWith(Cucumber.class)
8 @CucumberOptions(features = {"D:\\selen\\lps\\gmai\\src\\test\\resources\\feature\\title.feature",
9 "D:\\selen\\lps\\gmai\\src\\test\\resources\\feature\\UserTest.feature"},
10 monochrome = true, format = {'pretty', 'html:target'},
11 tags = {"@title"})
12
13 public class GmailRunner {
14
15 }

```

**UserTestFeature.java:**

```

1 package com;
2
3 Feature: CucumberJavaLogin
4 Scenario: Validate Page Title
5 Given I open Gmail website
6 Then Page Title is "Gmail"
7 And Close site

```

**GmailFeature.java:**

```

1 package com;
2
3 Feature: CucumberJavaLogin
4 Scenario: Validate Page Title
5 Given Launch Gmail site
6 When I open Gmail website
7 Then Page Title is "Gmail"
8 And Close site

```

**Annotations:**

- title**: Annotates the `UserTestFeature.java` file.
- Matchining annotation**: Annotates the `GmailFeature.java` file.
- Matchining annotation**: Annotates the `UserTestFeature.java` file.

**Execution Flow:**

An orange arrow points from the `title` annotation in `UserTestFeature.java` to the `tags = {"@title"}` section in the `GmailRunner.java` code. This indicates that the `UserTestFeature` will be executed when the `@title` tag is present in the feature file.

A green box labeled "Running only matching tags only" highlights the `tags = {"@title"}` section in the `GmailRunner.java` code.

**Matching annotation**

```

1 @Title
2 @Feature: CucumberJavaLogin
3 @Scenario: Validate Page Title
4 Given Launch Gmail site
5 When I open Gmail website
6 Then Page Title is "Gmail"
7 And Close site

```

```

1 package gmail;
2
3 import org.junit.runner.RunWith;
4 import cucumber.api.CucumberOptions;
5 import cucumber.api.junit.Cucumber;
6
7 @RunWith(Cucumber.class)
8 @CucumberOptions(features = { "D:\\maven\\proj\\gmail\\src\\test\\resources\\features\\title.feature",
9 "D:\\maven\\proj\\gmail\\src\\test\\resources\\features\\title.feature" },
10 monochrome = true, format = {"pretty", "html:target"},
11 tags = {"@title" })
12
13 public class GmailRunner
14 {
15 }
16
17

```

Here '@' means  
matching @  
annotations except all

**Matching annotation**

```

1 @Launch
2 @Feature: CucumberJavaLogin
3 @Scenario Outline: Validate User Id
4 Given Launch Gmail site
5 When I open Gmail website
6 When Enter User Id "<u>" Value
7 Then validate output for "<c1>" Criteria
8 When Enter Password id "<p>" Value
9 Then validate Password output for "<c2>" Criteria
10 And Close site
11
12 @ Examples:
13 | u | c1 | p | c2 |
14 | mahesh.samudrala24 | valid | mahesh.samudrala | invalid |
15 | mahesh.34345 | valid | mahesh.samudrala | invalid |
16 | mahesh.samudrala24 | valid | mahesh@gmail | valid |
17

```

Matching annotation

**Format**

To specify results folder

**Example:-**

```
format = {"pretty", "html:target" }
```

↓  
To remove  
funny symbols  
in html file

↓  
To remove  
funny symbols  
in html file

**Step 8:-**

Create TestNG Runner Class(If required)

In rare cases in BDD framework, we are able to use TestNG Runner Class instead of Junit Runner Classes. For it we need to perform two changes.

**Change 1:-**

ADD Cucumber-TestNG dependency & TestNG dependency to pom.xml in corresponding in maven project.

```
// cucumber-TestNG dependency
<dependency>
 <groupId>info.cukes</groupId>
 <artifactId>cucumber-testng</artifactId>
 <version>1.1.5</version>
</dependency>

//TestNG dependency
<dependency>
 <groupId>org.testng</groupId>
 <artifactId>testng</artifactId>
 <version>6.8</version>
 <scope>test</scope>
</dependency>
```

And then we need to save pom.xml to get jars for maven project.

**Change 2:-**

- ↓ Go to src/test/java.
- ↓ Right click on internal package.
- ↓ New
- ↓ class without main() method.
- ↓ Enter a name (single word & start with upper case).
- ↓ Click OK
- ↓ add annotations to that class like shown below.

**Test Script:-**

```
package gmailbdd;
import cucumber.api.CucumberOptions;
import cucumber.api.testng.AbstractTestNGCucumberTests;
public class Cumbrt_TestNG
{
 //@RunWith(Cucumber.class)
 @CucumberOptions(features = {
 "D:\\maven\\prgs\\gmailbdd\\src\\test\\resources\\feature1\\title.feature",
 "D:\\maven\\prgs\\gmailbdd\\src\\test\\resources\\feature1\\UseridTest.feature" },
 monochrome = true, format = {"pretty", "html:target" },
 tags = {"~@title" })
```

```
public class GmailRunner extends AbstractTestNGCucumberTests
{
}
```

**Note 1:-**

When TestNG Runner Class is ready, we are able to create “.xml” suite file.

- ↓ Right click on package in corresponding project
- ↓ Select Test NG
- ↓ Select Create Test NG class
- ↓ Enter a name suite file with .xml extension
- ↓ Click finish
- ↓ Delete new test class file(test class were already created with annotated method).
- ↓ Enter TestNG class namely in suite file.

**Example:-**

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="false">
 <test name="Test">
 <classes>
 <class name="gmailbdd.Cucumbrt_TestNG"/>
 </classes>
 </test> <!-- Test -->
</suite> <!-- Suite -->
```

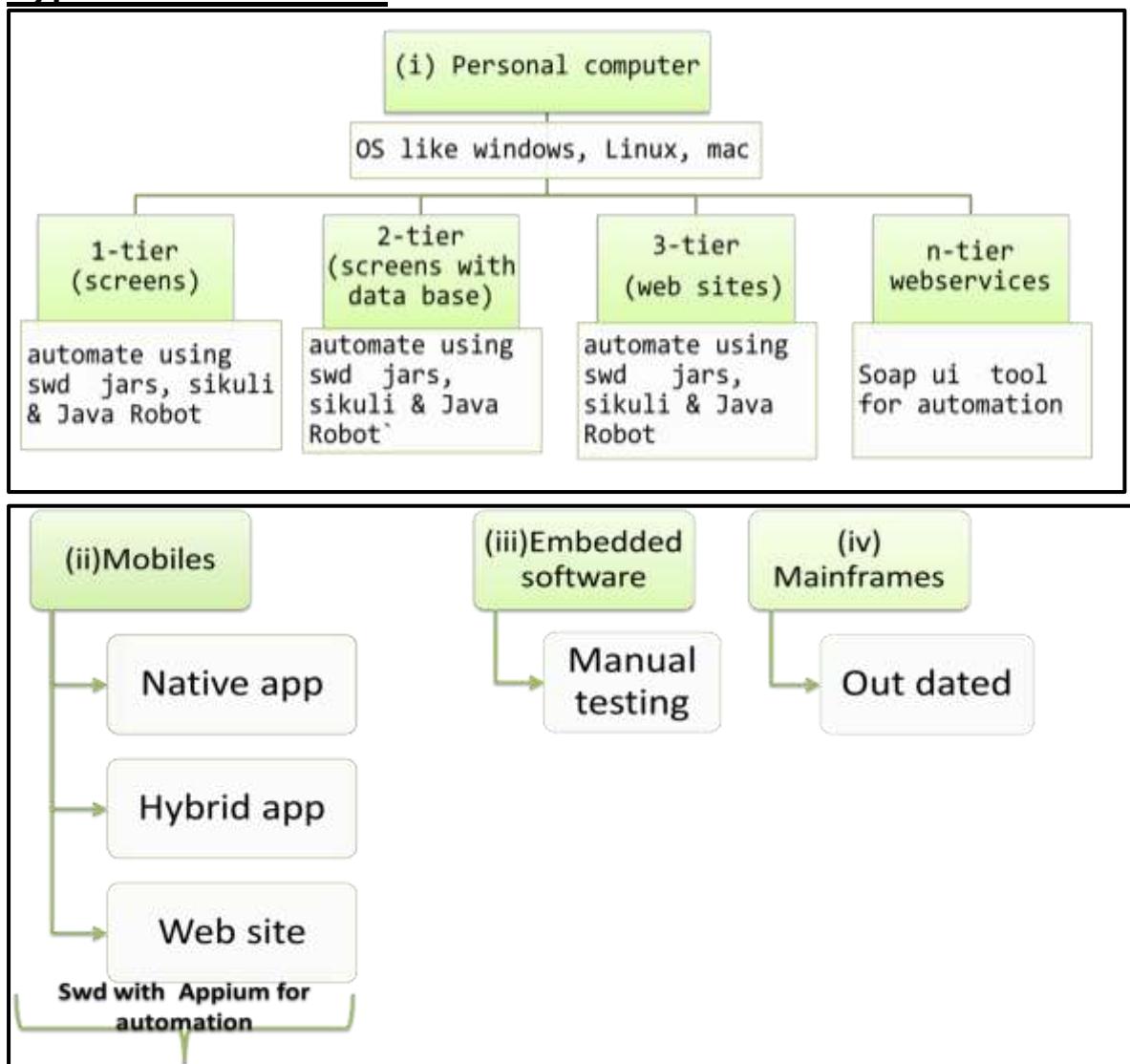
- ↓ Save file name and extend with .xml file

**Note 2:-**

To run above like suite file without involvement of Eclipse IDE. We need to create batch file (.bat file) with testing command & we need to associate this “.bat” file to “Jenkins” for periodic execution & continuous integration.

## VI. APPIUM

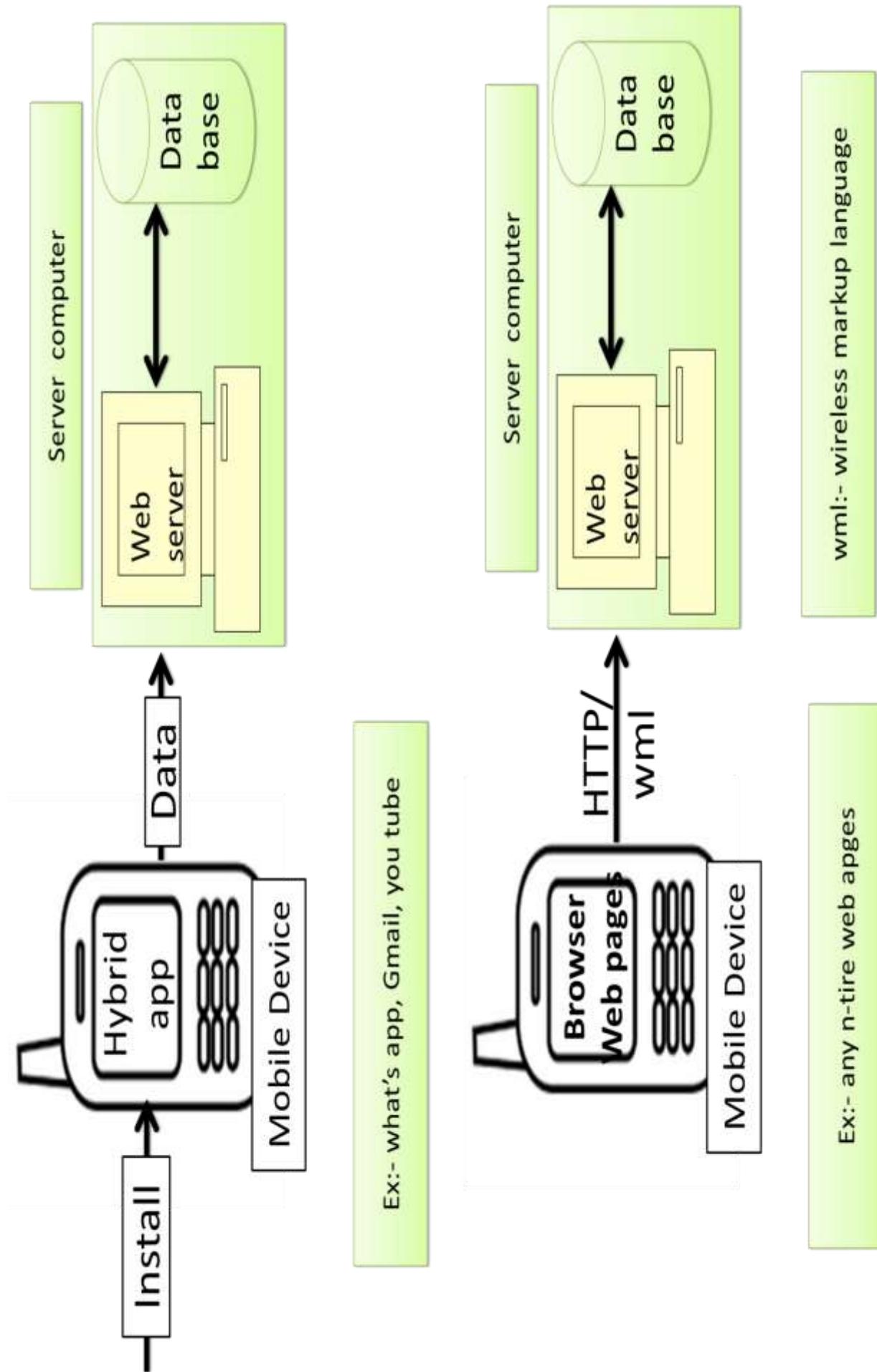
### A. Types of software



### B. Types of mobile software



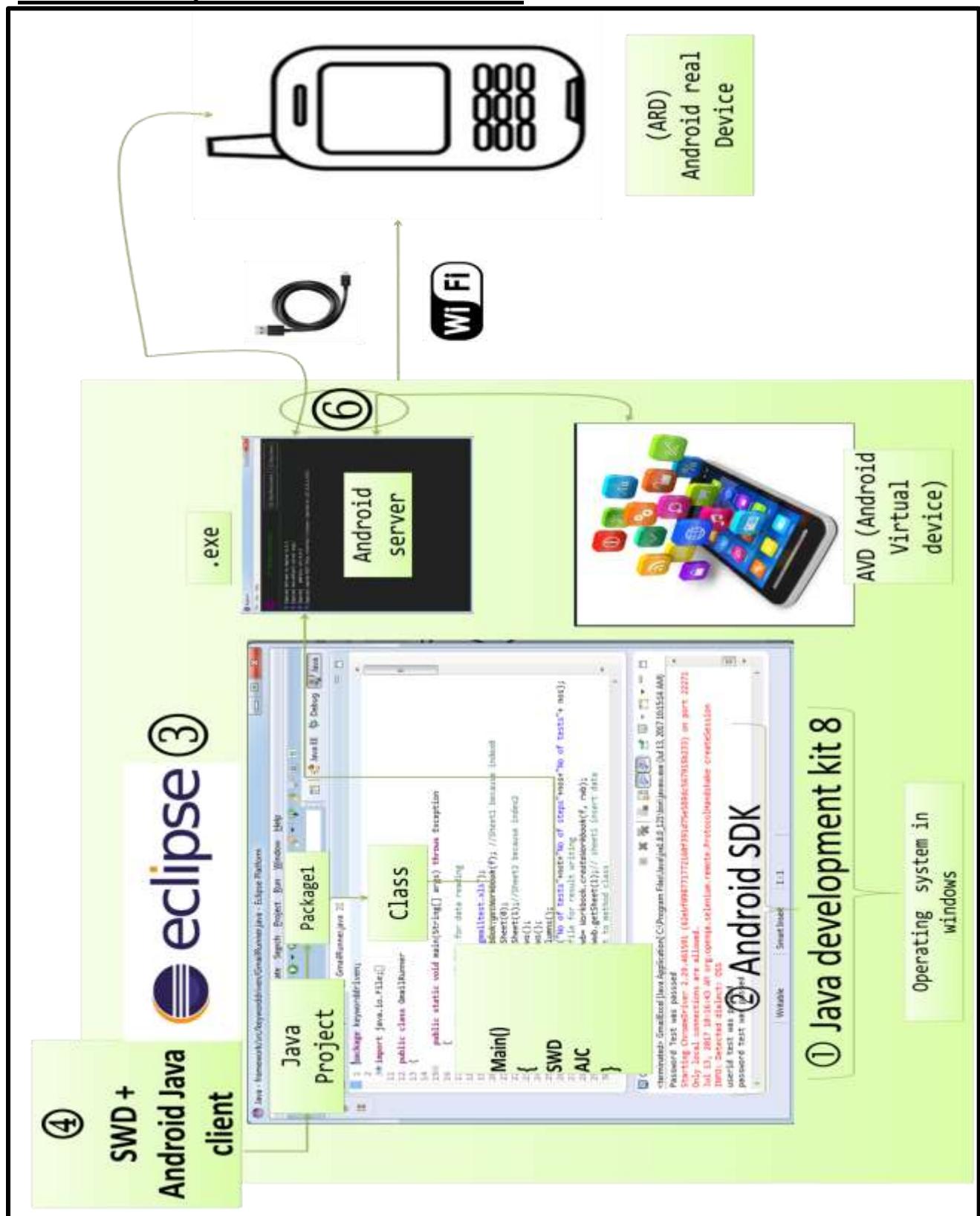
Ex:- calendar, calculator,  
save contacts ...etc



## C. About Appium :-

- Developed by Dan Cuellar
- Available as “Jar” & “.exe”
- “.exe” Appium Server & “.jar” is Appium Client Jar
- To automate apps (native & Hybrid) & site in Android & IOS platform
- To integrate with selenium Web Driver, because we need to use SWD Jars classes objects in mobile app test scripts along with Appium Java-client jar classes objects.

### i). Tester computer OS is windows



AVD → Android Virtual Device  
 ARD → Android Real Device  
 AJC → Appium Java Client JAR  
 AS → Appium server  
 SDK → Software Development Kit  
 JP → Java Project

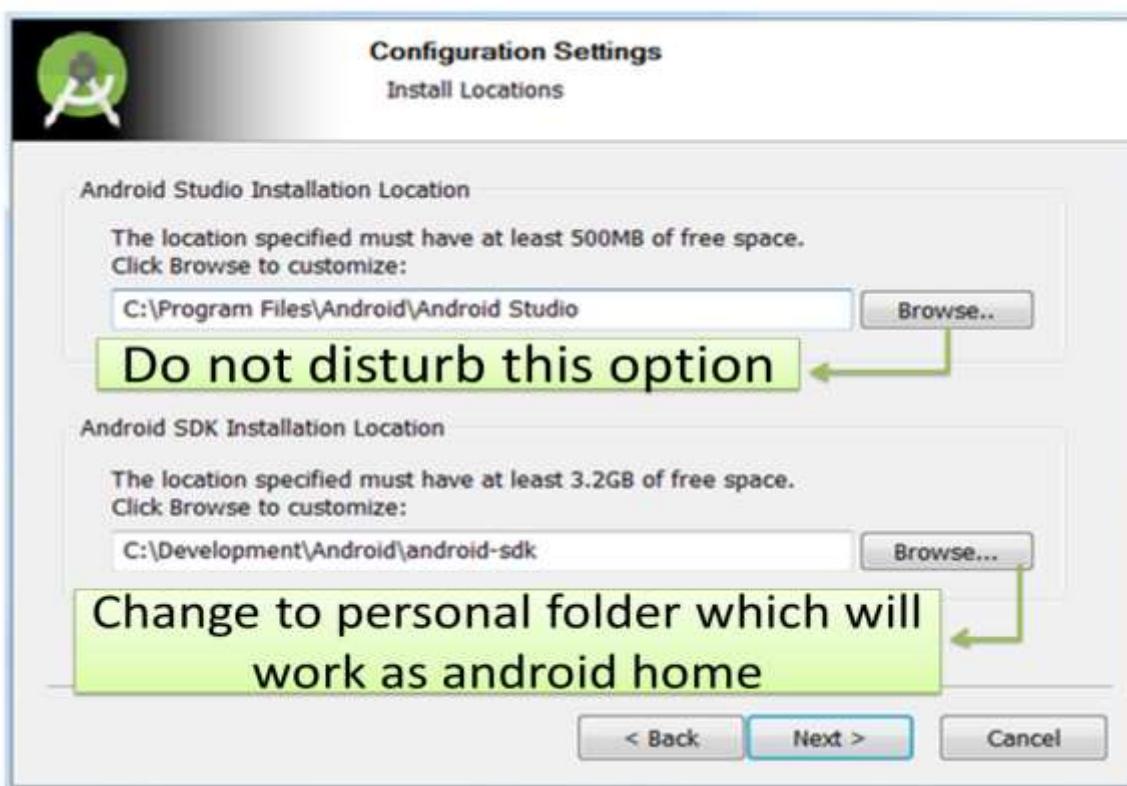
### **a) Configure Mobile app test setup**

#### **Step 1:- Download and install JDK8**

(create JAVA\_HOME environmental variables with path of JDK & extend path variable with path of JDK\bin;)

#### **Step 2:- Download & install Android SDK**

- ↓ Go to [www.google.com](http://www.google.com) site
- ↓ Enter android SDK bundle 145(version) for windows
- ↓ Click search
- ↓ Go to tools.android.com site
- ↓ Click on link for “windows bundle with sdk”
- ↓ Wait until complete download(1.6GB)
- ↓ Paste that download in personal folder
- ↓ Double click on that download to start installation
- ↓ Click “Run” to initiate installation
- ↓ Click Next
- ↓ Browse any empty personal as Android Home folder



- ↓ Click Next Until Finish(Folder Size will be app 11.6 GB after completion of software)

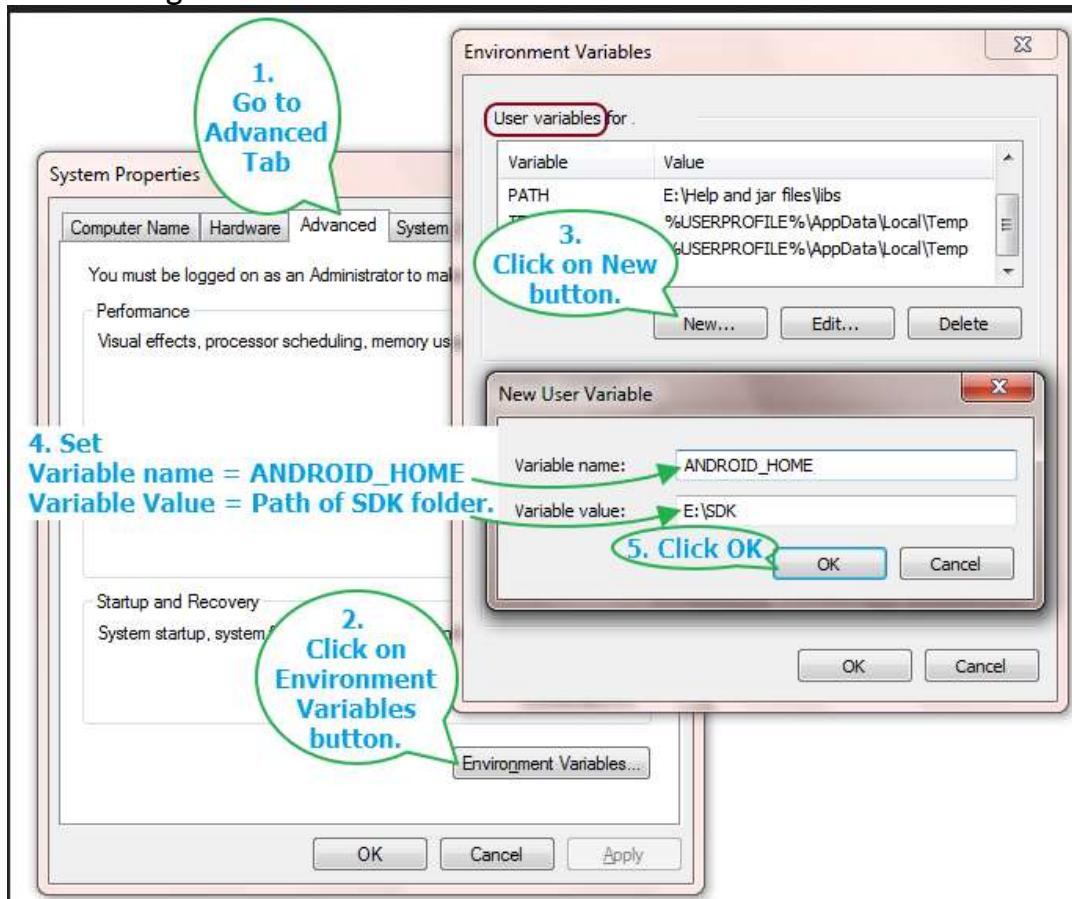
#### **Note 1:-**

After completion of Android SDK installation, we need to observe corresponding folder which consist some sub folder like **AVD Manager.exe** and **SDK Manager.EXE**

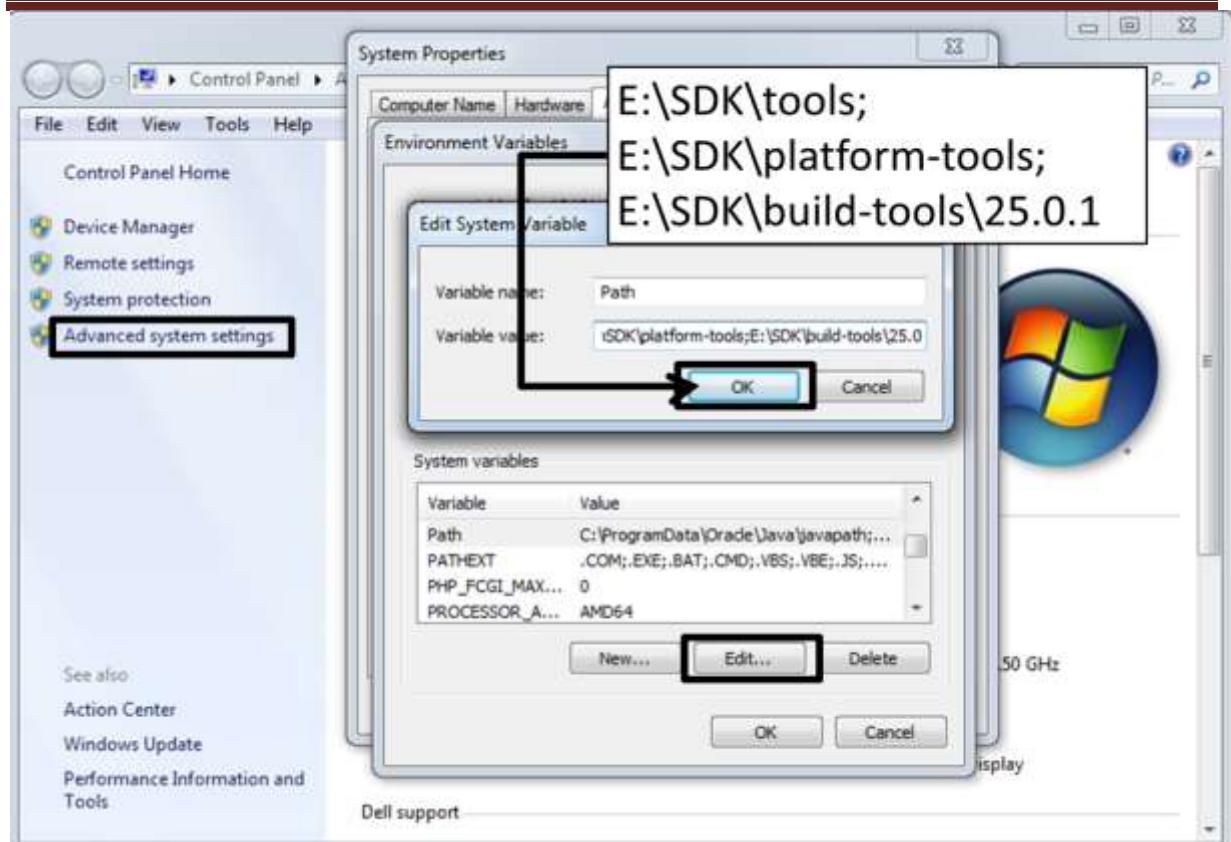
**Note 2:-**

After getting completion of complete installation, we need to create **ANDROID\_HOME** in Environment variable like shown below

- ⇒ Copy path of personal folder which have **Android SDK Folder** & software
- ⇒ Right-click My Computer on the desktop and select ‘Properties’.
- ⇒ Click the ‘Advanced System settings’ button
- ⇒ To open the System properties window
- ⇒ Under the ‘Advanced’ tab, select ‘Environment Variables’
- ⇒ Click on New button under User Variable table. It will open New User Variable dialog.
- ⇒ Set **Variable Name = ANDROID\_HOME** and **Variable value = E:\SDK** (Path of SDK folder). Path can be different for you as per your SDK folder location as described in previous post.
- ⇒ Click on OK button to close New User Variable dialog as shown in below image.



- ⇒ And look for the Variable ‘Path’ in the System variables window.
- ⇒ After pressing Edit, scroll to the end of ‘Variable value:’ and add the full path to the directory to the end of the path, separated by a semicolon from the previous path. In the above example, you would add as shown below:



- ⇒ Click Ok.
- ⇒ Close Computer properties window.

### Note 3:-

To know the correctness of android SDK installation and configuration, we can run a command at command prompt.

```
C:\Users\APARNA>adb devices
List of devices attached
* daemon not running. starting it now at tcp:5037 *
* daemon started successfully *
emulator-5554 device
```

```
C:\Windows\system32\cmd.exe
C:\Users\APARNA>adb
Android Debug Bridge version 1.0.39
Revision 3db08f2c6889-android
Installed as D:\Appium\Soft\platform-tools\adb.exe

global options:
-a listen on all network interfaces, not just localhost
-d use USB device <error if multiple devices connected>
-e use TCP/IP device <error if multiple TCP/IP devices available>
-s SERIAL use device with given serial number <overrides $ANDROID_SERIAL>
-p PRODUCT name or path ('angler'/'out/target/product/angler');
 default $ANDROID_PRODUCT_OUT
-H name of adb server host [default=localhost]
-P port of adb server [default=5037]
-L SOCKET listen on given socket for adb server [default=tcp:localhost:5037]

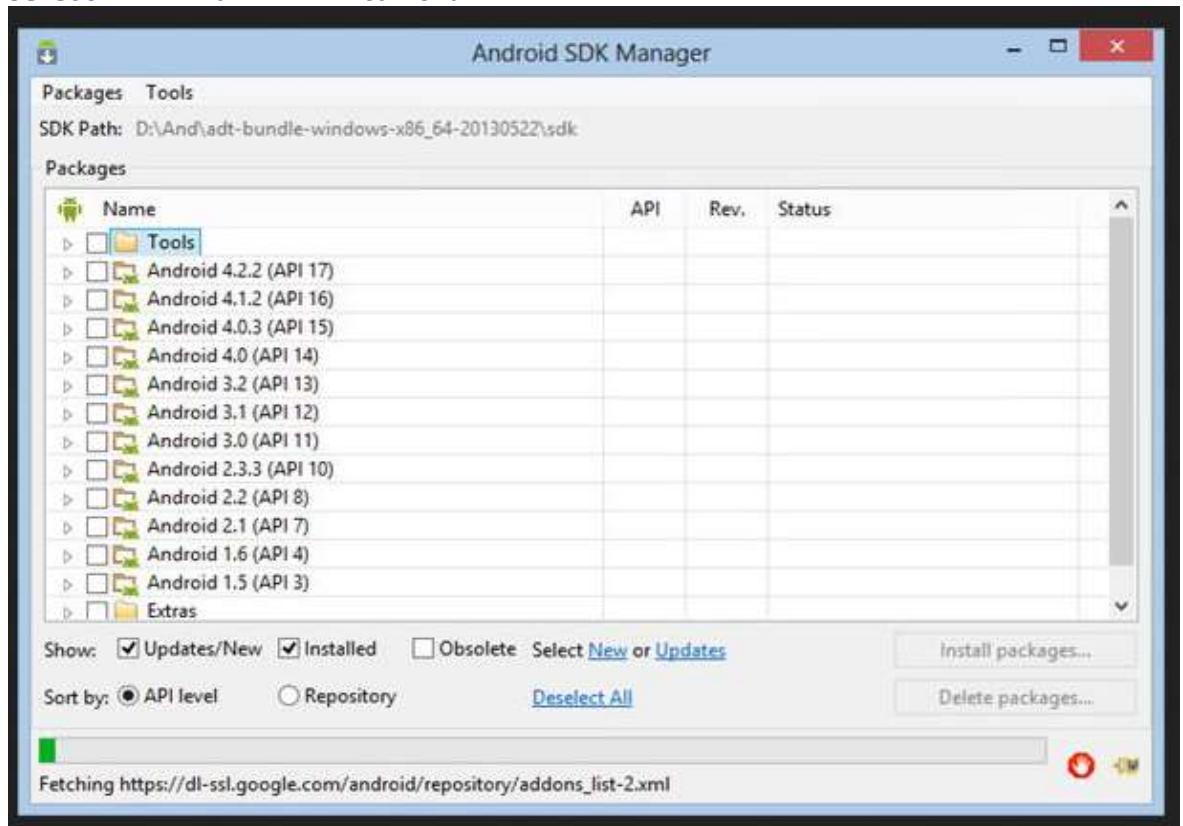
general commands:
devices [-l] list connected devices (-l for long output)
help show this help message
version show version num

networking:
connect HOST[:PORT] connect to a device via TCP/IP [default port=5555]
```

**Note 4:-**

To work with various version of android we need to install corresponding **Android API Levels** (Application programming Interface Levels) 17 Or 17 to 25.

- ↓ Open Android Home Folder
- ↓ Double Click on SDK Manager
- ↓ Select API 17 Or API 17 to 25 all

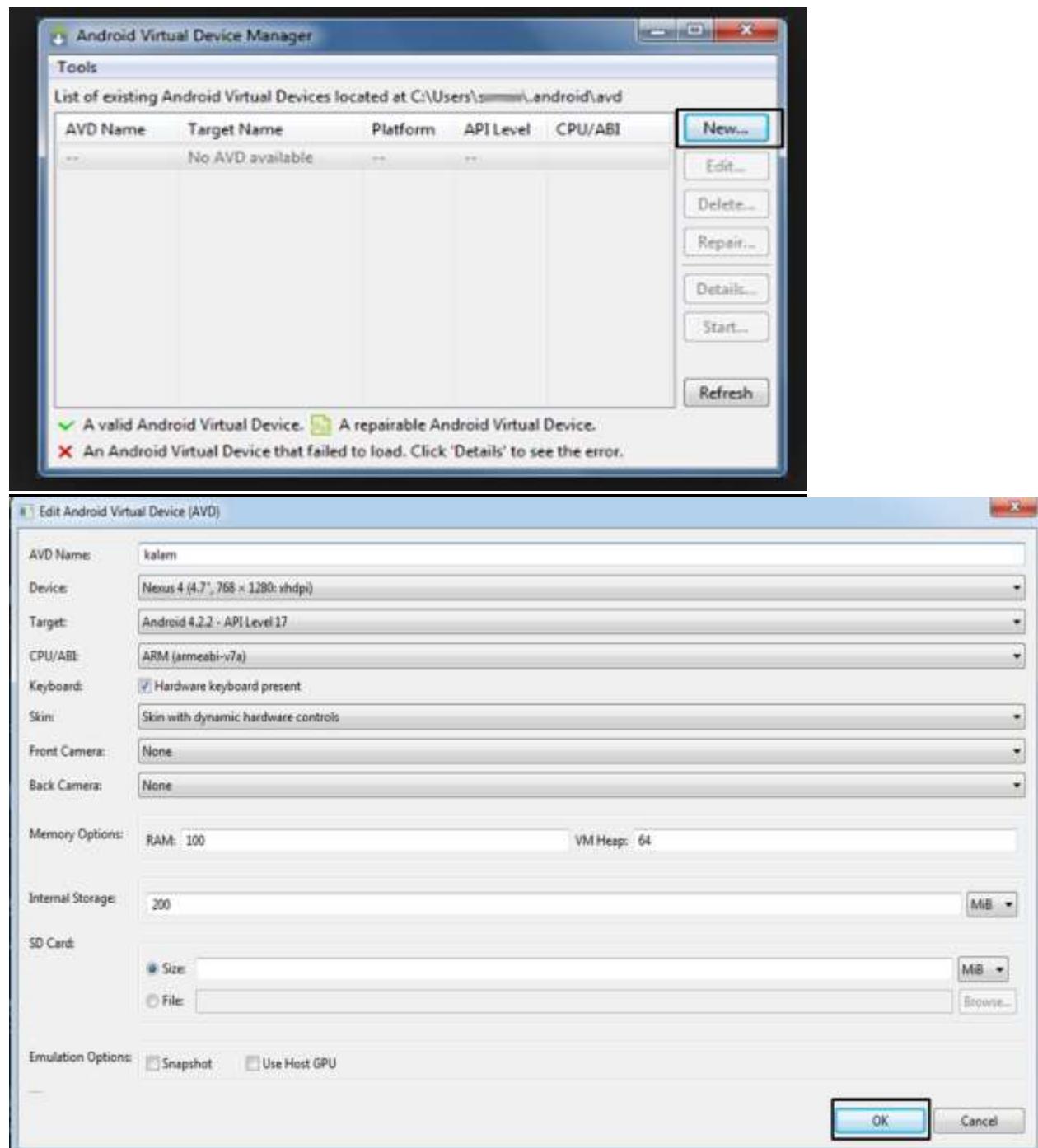


- ↓ Click on install packages
- ↓ Wait until complete installation
- ↓ Close SDK Manager Window.

**Step 3:- AVD(Android Virtual Device) Creation:**

AVD stands for android virtual device, we are able to create AVD's by following below navigation in tester computer

- ↓ Open Android Home Folder
- ↓ Double Click On AVD Manager
- ↓ Click Create
- ↓ Enter a name to AVD & Select least configurations
- ↓ Click Ok
- ↓ Click Start
- ↓ Click Launch to get AVD
- ↓ Close AVD Manager

Example

#### Step 4:- Connect real device to tester computer.

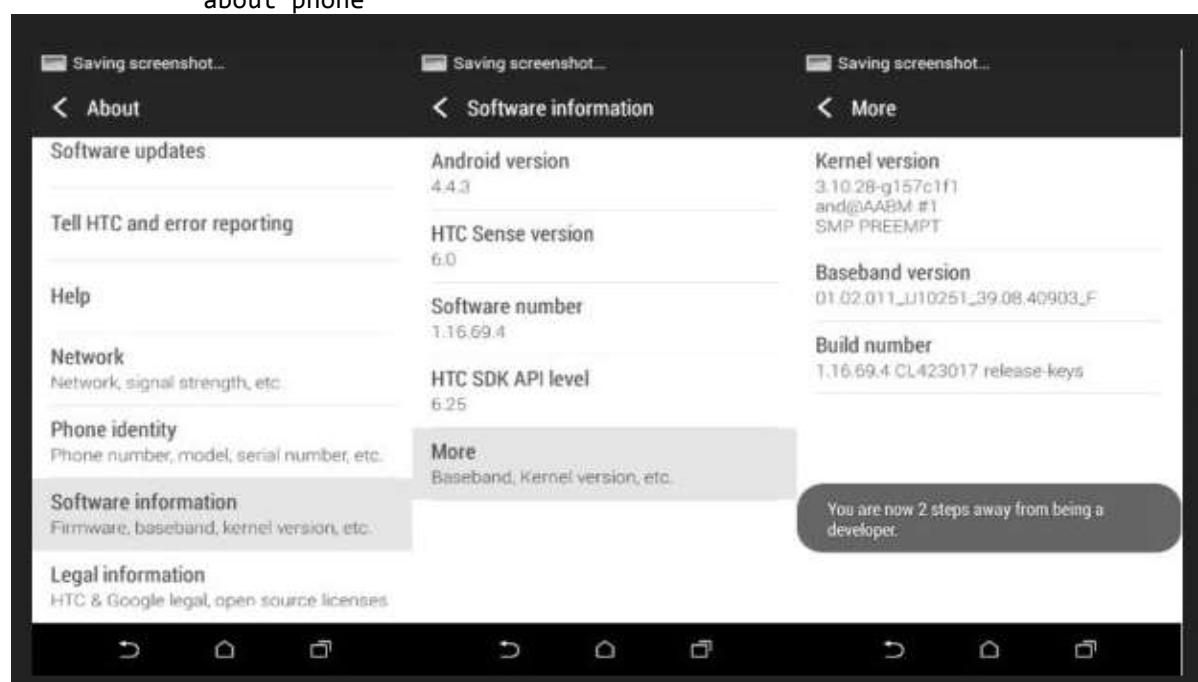
ARD(Android Real Device)

ARD stands for Android Real Device (Real Android Mobile)

- ↓ Open Settings in ARD
- ↓ Developer Options
- ↓ Stay Awake
- ↓ Select USB debugging



- ↓ Back To Home
- ↓ (if we didn't get developer options in settings, we need go to about phone)



- ↓ Click 6 times on build version (or) imui no)

- ↓ Connect corresponding ARD to tester computer using USB Data cable
- ↓ Wait until “Your device is ready to use” message.



- ↓ (If your device system is unable to recognize download corresponding ARD driver, we can install [Pdanet+ Driver](#) software)
- ↓ Open command prompt & run below command
- ↓ adb device ↵

### **Step 5:- adb & aapt commands .**

adb stands for android debug bridge.

adb commands are useful to connected AVD & ARD.

#### **1. adb devices ↵**

```
C:\Users\APARNA>adb devices
List of devices attached
* daemon not running. starting it now at tcp:5037 *
* daemon started successfully *
emulator-5554 device
```

This command shows how many AVD & ARD device connected to tester computer

#### **2. adb -s device name shell pm list packages -f ↵**

sys:- adb -s emulator-5555 shell pm list packages -f

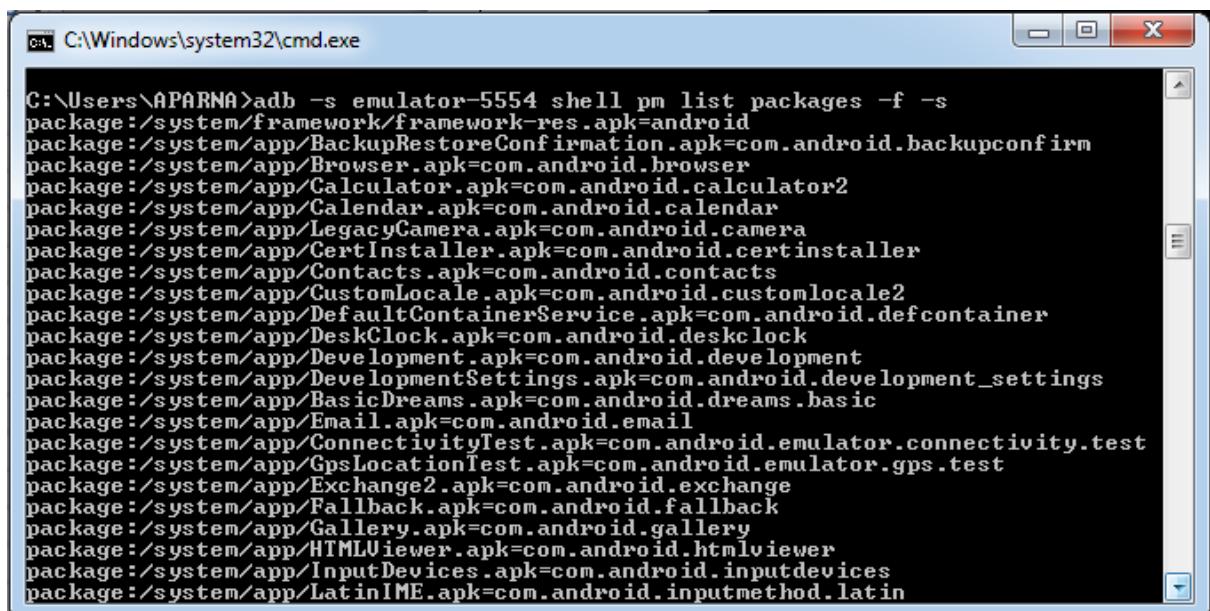
To get all system , 3<sup>rd</sup> party apps and get path of apps also

```
C:\ C:\Windows\system32\cmd.exe
C:\Users\APARNA>adb -s emulator-5554 shell pm list packages -f
package:/system/framework/framework-res.apk=android
package:/system/app/BackupRestoreConfirmation.apk=com.android.backupconfirm
package:/system/app/Browser.apk=com.android.browser
package:/system/app/Calculator.apk=com.android.calculator2
package:/system/app/Calendar.apk=com.android.calendar
package:/system/app/LegacyCamera.apk=com.android.camera
package:/system/app/CertInstaller.apk=com.android.certinstaller
package:/system/app/Contacts.apk=com.android.contacts
package:/system/app/CustomLocale.apk=com.android.customlocale2
package:/system/app/DefaultContainerService.apk=com.android.defcontainer
package:/system/app/DesktopClock.apk=com.android.deskclock
package:/system/app/Development.apk=com.android.development
package:/system/app/DevelopmentSettings.apk=com.android.development_settings
package:/system/app/BasicDreams.apk=com.android.dreams.basic
package:/system/app/Email.apk=com.android.email
package:/system/app/ConnectivityTest.apk=com.android.emulator.connectivity.test
package:/system/app/GpsLocationTest.apk=com.android.emulator.gps.test
package:/system/app/Exchange2.apk=com.android.exchange
package:/system/app/Fallback.apk=com.android.fallback
package:/system/app/Gallery.apk=com.android.gallery
package:/data/app/GestureBuilder.apk=com.android.gesture.builder
package:/system/app/HTMLViewer.apk=com.android.htmlviewer
package:/system/app/InputDevices.apk=com.android.inputdevices
```

3. adb -s device name shell pm list packages -f -s ↵

sys:- adb -s emulator-5555 shell pm list packages -f -s

To get all system apps only and get path of apps.



```
C:\Users\APARNA>adb -s emulator-5554 shell pm list packages -f -s
package:/system/framework/framework-res.apk=android
package:/system/app/BackupRestoreConfirmation.apk=com.android.backupconfirm
package:/system/app/Browser.apk=com.android.browser
package:/system/app/Calculator.apk=com.android.calculator2
package:/system/app/Calendar.apk=com.android.calendar
package:/system/app/LegacyCamera.apk=com.android.camera
package:/system/app/CertInstaller.apk=com.android.certinstaller
package:/system/app/Contacts.apk=com.android.contacts
package:/system/app/CustomLocale.apk=com.android.customlocale2
package:/system/app/DefaultContainerService.apk=com.android.defcontainer
package:/system/app/DesktopClock.apk=com.android.deskclock
package:/system/app/Development.apk=com.android.development
package:/system/app/DevelopmentSettings.apk=com.android.development_settings
package:/system/app/BasicDreams.apk=com.android.dreams.basic
package:/system/app/Email.apk=com.android.email
package:/system/app/ConnectivityTest.apk=com.android.emulator.connectivity.test
package:/system/app/GpsLocationTest.apk=com.android.emulator.gps.test
package:/system/app/Exchange2.apk=com.android.exchange
package:/system/app/Fallback.apk=com.android.fallback
package:/system/app/Gallery.apk=com.android.gallery
package:/system/app/HTMLViewer.apk=com.android.htmlviewer
package:/system/app/InputDevices.apk=com.android.inputdevices
package:/system/app/LatinIME.apk=com.android.inputmethod.latin
```

4. adb -s device name shell pm list packages -f -3 ↵

sys:- adb -s emulator-5555 shell pm list packages -f -3

To get all 3<sup>rd</sup> party apps only & and get path of apps also.

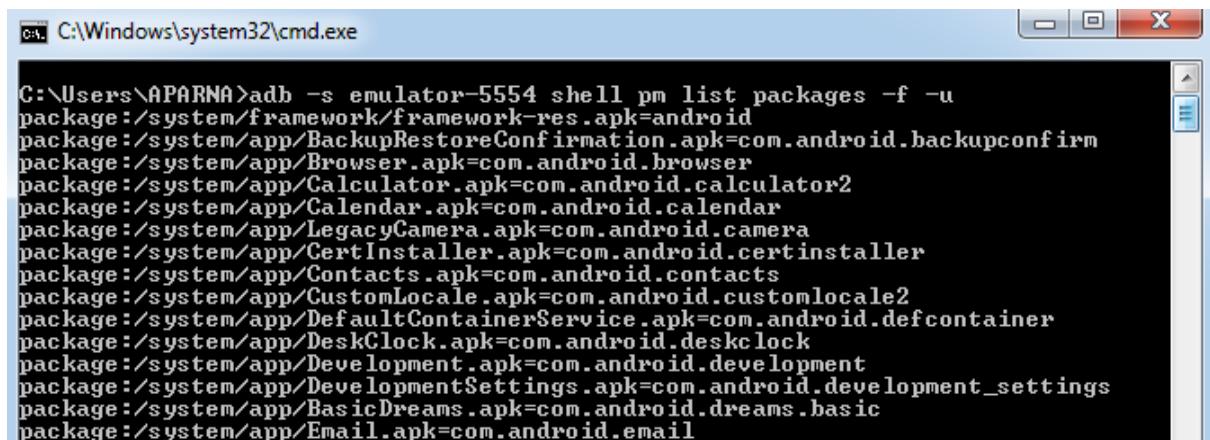


```
C:\Users\APARNA>adb -s emulator-5554 shell pm list packages -f -3
package:/data/app/GestureBuilder.apk=com.android.gesture.builder
package:/data/app/SmokeTestApp.apk=com.android.smoketest
package:/data/app/SmokeTest.apk=com.android.smoketest.tests
package:/data/app/WidgetPreview.apk=com.android.widgetpreview
package:/data/app/ApiDemos.apk=com.example.android.apis
package:/data/app/CubeLiveWallpapers.apk=com.example.android.livecubes
package:/data/app/SoftKeyboard.apk=com.example.android.softkeyboard
package:/data/app/io.appium.settings-1.apk=io.appium.settings
package:/data/app/io.appium.unlock-1.apk=io.appium.unlock
```

5. adb -s device name shell pm list packages -f -u ↵

sys:- adb -s emulator-5555 shell pm list packages -f -u

To get all uninstall apps from the time of mobile activation

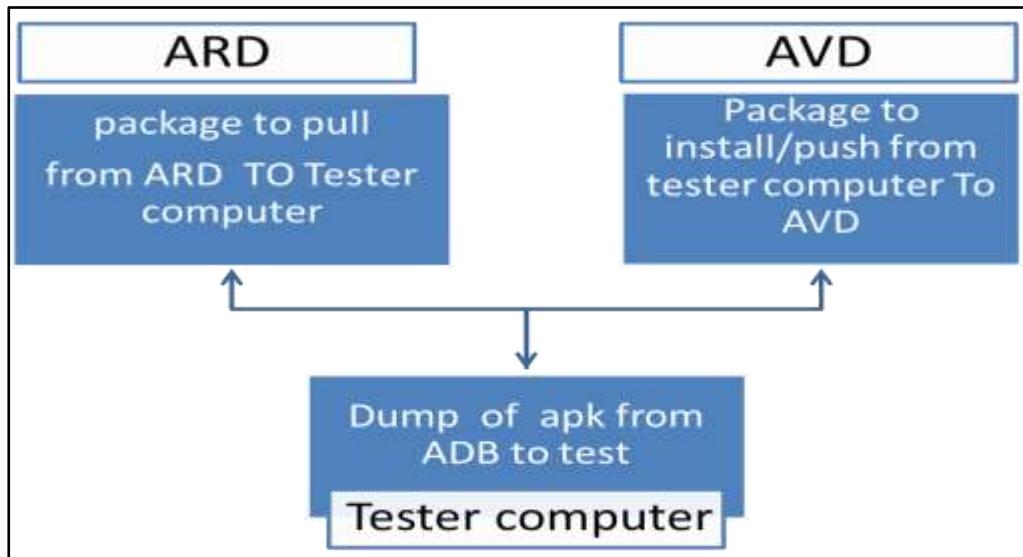


```
C:\Users\APARNA>adb -s emulator-5554 shell pm list packages -f -u
package:/system/framework/framework-res.apk=android
package:/system/app/BackupRestoreConfirmation.apk=com.android.backupconfirm
package:/system/app/Browser.apk=com.android.browser
package:/system/app/Calculator.apk=com.android.calculator2
package:/system/app/Calendar.apk=com.android.calendar
package:/system/app/LegacyCamera.apk=com.android.camera
package:/system/app/CertInstaller.apk=com.android.certinstaller
package:/system/app/Contacts.apk=com.android.contacts
package:/system/app/CustomLocale.apk=com.android.customlocale2
package:/system/app/DefaultContainerService.apk=com.android.defcontainer
package:/system/app/DesktopClock.apk=com.android.deskclock
package:/system/app/Development.apk=com.android.development
package:/system/app/DevelopmentSettings.apk=com.android.development_settings
package:/system/app/BasicDreams.apk=com.android.dreams.basic
package:/system/app/Email.apk=com.android.email
```

6. adb -s device name pull path of app.apk ←  
 sys:- adb -s FYTOOBWORCEAZLK7 pull /data/app/com.whatsapp-1/base.apk  
 to get installed app in ARD to tester computer  
 C:\Users\APARNA>adb -s FYTOOBWORCEAZLK7 pull /data/app/com.whatsapp-1/base.apk  
 [ 58% ] /data/app/com.whatsapp-1/base.apk  
 After completion of 100% pull from ARD  
 C:\Users\APARNA>adb -s FYTOOBWORCEAZLK7 pull /data/app/com.whatsapp-1/base.apk  
 /data/app/com.whatsapp-1/base.apk: 1 f.... 2.5 MB/s <30407204 bytes in 11.822s>

7. adb -s device name install of apk in tester computer ←  
 sys:- adb -s emulator-5554 install base.apk  
 To install apk from test computer into ARD/AVD

```
C:\Users\APARNA>adb -s emulator-5554 install base.apk
base.apk: 1 file pushed. 3.0 MB/s <30407204 bytes in 9.666s>
pkg: /data/local/tmp/base.apk
Success
```



To pull package from real device to tester computer & from tester computer to AVD

A screenshot of a Windows command prompt window titled "cmd.exe". The window shows a list of installed packages on an Android device. The text output is as follows:

```

C:\Windows\system32\cmd.exe
Path of package name
package
name
package:/data/app/com.whatsapp-1/base.apk=com.whatsapp
package:/data/app/com.indanegas-1/base.apk=com.indanegas
package:/data/app/com.domobile.applock-1/base.apk=com.domobile.applock
package:/data/app/in.fourthlion.ccd.mobileapp-1/base.apk=in.fourthlion.ccd.mobileapp
package:/data/app/com.appxy.tinyscanner-2/base.apk=com.appxy.tinyscanner
package:/data/app/com.freeit.java-1/base.apk=com.freeit.java
package:/data/app/in.amazon.nShop.android.shopping-1/base.apk=in.amazon.nShop.android.shopping
package:/data/app/com.tencent.mm-2/base.apk=com.tencent.mm
package:/data/app/com.flipkart.android-2/base.apk=com.flipkart.android
package:/preload/vendor/app/com.guvera.android.apk=com.guvera.android
package:/data/app/in.startv.hotstar-2/base.apk=in.startv.hotstar
package:/data/app/com.olacabs.customer-1/base.apk=com.olacabs.customer
package:/data/app/com.bigmarket.mobileapp-1/base.apk=com.bigmarket.mobileapp
package:/data/app/com.google.zxing.client.android-1/base.apk=com.google.zxing.client.android

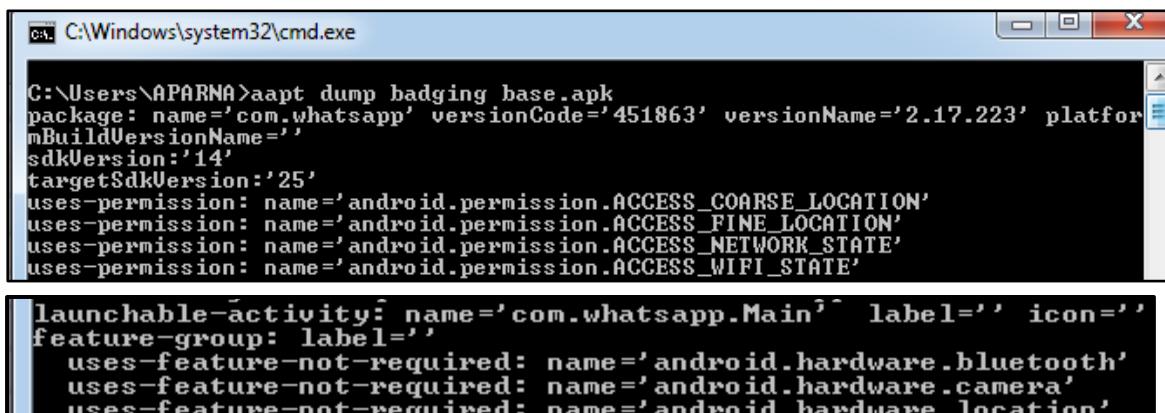
```

8. adb -s device name uninstall path of package name ARD/AVD ←  
Sys:- adb -s emulator-5554 uninstall com.whatsapp  
 To uninstall specific app from ARD/AVD

```
C:\Users\APARNA>adb -s emulator-5554 uninstall com.whatsapp
Success
C:\Users\APARNA>
```

- aapt dump badging path of apk in computer ↵**  
Sys:- aapt dump badging base.apk

To get app package name & launching activity name in AVD/ARD.



```
C:\Windows\system32\cmd.exe
C:\Users\APARNA>aapt dump badging base.apk
package: name='com.whatsapp' versionCode='451863' versionName='2.17.223' platformBuildVersionName=''
sdkVersion:'14'
targetSdkVersion:'25'
uses-permission: name='android.permission.ACCESS_COARSE_LOCATION'
uses-permission: name='android.permission.ACCESS_FINE_LOCATION'
uses-permission: name='android.permission.ACCESS_NETWORK_STATE'
uses-permission: name='android.permission.ACCESS_WIFI_STATE'

launchable-activity: name='com.whatsapp.Main' label='' icon=''
feature-group: label=''
 uses-feature-not-required: name='android.hardware.bluetooth'
 uses-feature-not-required: name='android.hardware.camera'
 uses-feature-not-required: name='android.hardware.location'
```

#### App in mobile(AVD/ARD)

- adb -s device name pull path of app.apk ↵**  
sys:- adb -s FYTOOBWORCEAZLK7 pull /data/app/com.whatsapp-1/base.apk  
 To get installed app in ARD to tester computer
- aapt dump badging path of apk in computer ↵**  
Sys:- aapt dump badging base.apk  
 To get app package name & launching activity name in AVD/ARD.

#### App in Computer

- aapt dump badging path of apk in computer ↵**  
Sys:- aapt dump badging base.apk  
 To get app package name & launching activity name in AVD/ARD.
- adb -s device name install of apk in tester computer ↵**  
sys:- adb -s emulator-5554 install base.apk  
To install apk from test computer into ARD/AVD

#### Note 1:-

In android, every app is called as package. Every package is having one Or more classes with methods. Here every package file extension is “.apk” (android package kit).

#### Note 2:-

While working with android app (native\hybrid), we can use “adb”, “aapt” & “uiautomatorviewer” like commands at cmd prompt to get apps information before going to testing.

- ⇒ Here “adb” stands for android debug bridge
- ⇒ “aapt” stands for android asset packaging tool
- Here “uiautomatorviewer” is useful to get details at element app screen

Android Application	Purpose of test Automation
<b>Adb</b>	⇒ To get apps/package list ⇒ To install app/package to mobiles ⇒ To uninstall app/package from mobiles ⇒ To pull app/package .apk from mobiles to computer ⇒ ...etc
<b>Aapt</b>	⇒ To get specified app package name & launching activity name ⇒ ...etc
<b>uiautomatorviewer</b>	⇒ To get details of elements corresponding app screens ⇒ ..etc

**Note 3:-**

In AVD (OR) ARD apps are called as packages. In general, developers can release app as “.apk” file to testers.

So testers can install that “.apk” into corresponding AVD/ARD using below cmd  
adb -s device name install of apk in tester computer ↵

**Step 6:-**

- ↓ Download and launch ECLIPSE IDE(Juno Version).
- ↓ Browse a folder for work space.
- ↓ Create a new Java Project with a package in Eclipse IDE.  
(Project name (OR) Package name as single word without space in lower case)
- ↓ Create a class with main() method.
- ↓ Download & associate “SWD jar” & “Appium Java client 4.0” to java project

**Step 7:-**

Download & install Appium server

After completion of JDK & Android SDK configuration & installation.

Download & configure Eclipse IDE, Java project creation & required Jar associations & AVD OR ARD connection with Appium.

We can follow below navigation to download & install Appium server.exe

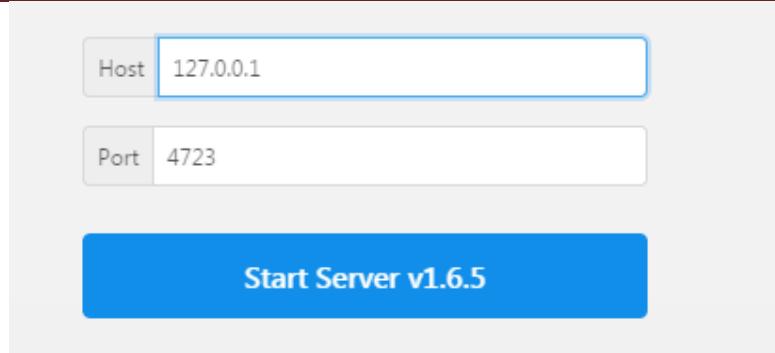
**Navigation:-**

- ↓ Go to [www.google.com](http://www.google.com) site.
- ↓ Enter Appium server download.
- ↓ Go to Appium.io site.
- ↓ Click on Appium-desktop OSX, Window & Linux link
- ↓ Click on appium-desktop-Setup-1.1.0-beta.4.exe
- ↓ Start download & wait until download completed
- ↓ Paste that download in personal folder
- ↓ Double click on that download
- ↓ Click next until finish
- ↓ Observe Appium shortcut on desktop

**Note 1:-**

We need start Appium server daily before going to mobile test automation for it.

- ↓ We need to double click on Appium desktop,
- ↓ Click start server button
- ↓ Observer i.p. address :- 127.0.0.1
- ↓ Port:- 4723
- ↓ Full address :127.0.0.1:4723

**Note 2:-**

When we get any problem with Appium server, we need to install ".net framework 4.0"

**Step 8:-****Developing Scripts**

After completion Entire set up building , we need to start scripts writing in main() method of class in package in project.

While writing scripts we need to get regarding mobile (AVD & ARD)

**Example 1:-**

- ↓ Launch calculator app in AVD.  
(get avd device details from adb devices cmd)
- ↓ (App package name & activity name from aapt cmd)
- ↓ Click clear button
- ↓ Enter input1
- ↓ Click +
- ↓ Enter input1
- ↓ Click =
- ↓ Get output display
- ↓ Close app

**Test Script:**

```
package app;
import io.appium.java_client.android.AndroidDriver;
import java.net.URL;
import org.openqa.selenium.By;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.remote.DesiredCapabilities;
public class MobileAppCal
{
 public static void main(String[] args) throws Exception
 {
 DesiredCapabilities d=new DesiredCapabilities();
 d.setCapability(CapabilityType.BROWSER_NAME, "null");
 d.setCapability("deviceName", "emulator-5554");
 d.setCapability("platformName", "Android");
 d.setCapability("platformVersion", "4.2.2");
 d.setCapability("appPackage", "com.android.calculator2");
 d.setCapability("appActivity", "com.android.calculator2.Calculator");
 URL u =new URL("http://127.0.0.1:4723/wd/hub");
 AndroidDriver driver =new AndroidDriver(u,d);
 driver.findElement(By.xpath("//*[@text='DELETE'][@index='0']]")).click();
 Thread.sleep(5000);
 driver.findElement(By.xpath("//*[@text='9'][@index='2']]")).click();
 Thread.sleep(5000);
```

```

driver.findElement(By.xpath("//*[@text='9'][@index='2']]")).click();
Thread.sleep(5000);
driver.findElement(By.xpath("//*[@content-desc='plus'][@index='3']]")).click();
Thread.sleep(5000);
driver.findElement(By.xpath("//*[@text='9'][@index='2']]")).click();
Thread.sleep(5000);
driver.findElement(By.xpath("//*[@text='9'][@index='2']]")).click();
Thread.sleep(5000);
String y=driver.findElement(By.xpath("//*[@class='android.widget.EditText']"
[@index='0']")).getText();
System.out.println(y);
driver.closeApp();
}
}

```

**Example 2:-**

- ↓ Launch what's app in ARD.  
(get ARD device details from adb devices cmd)  
(App package name & activity name from aapt cmd)

```
C:\Users\APARNA>aapt dump badging base.apk
package: name='com.whatsapp' versionCode='451900' versionName='2.17.254' platformBuildVersionName=''
sdkVersion:'15'
targetSdkVersion:'25'
uses-permission: name='android.permission.ACCESS_COARSE_LOCATION'
uses-permission: name='android.permission.ACCESS_FINE_LOCATION'
uses-permission: name='android.permission.ACCESS_NETWORK_STATE'
uses-permission: name='android.permission.ACCESS_WIFI_STATE'
```



- ↓ Search conversation list any name like(Sis, Amma, Nana)
- ↓ Click on searched Conversation contact
- ↓ Enter Message
- ↓ Click Send button
- ↓ Close App

**Test Script:**

```

package app;
import io.appium.java_client.android.AndroidDriver;
import java.net.URL;
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.remote.DesiredCapabilities;
public class WhatsApp_1
{
public static void main(String[] args) throws Exception
{
DesiredCapabilities d = new DesiredCapabilities();
d.setCapability(CapabilityType.BROWSER_NAME, "null");
d.setCapability("deviceName", "192.168.0.104:1232");
d.setCapability("platformName", "Android");
}
}
```

```
d.setCapability("platformVersion", "6.0");
d.setCapability("appPackage", "com.whatsapp");
d.setCapability("appActivity", "com.whatsapp.Main");
URL u = new URL("http://127.0.0.1:4723/wd/hub");
AndroidDriver driver = new AndroidDriver(u, d);
Thread.sleep(5000);
Dimension size;
size = driver.manage().window().getSize();
System.out.println(size);
// Find swipe start and end point from screen's width and height.
// Find starty point which is at bottom side of screen.
int starty = (int) (size.getHeight() * 0.80);
// Find endy point which is at top side of screen.
int endy = (int) (size.getHeight() * 0.20);
// Find horizontal point where you want to swipe. It is in middle of
// screen width.
int startx = size.getWidth() / 2;
System.out.println("starty=" + starty + ",endy = " + endy+ ",startx = " + startx);
while (2 > 1)
{
try
{
// searching conversation list in chat tab & click on it
driver.findElement(By.xpath("//*[contains(@text,'Sis')]/parent::*")).click();
Thread.sleep(5000);
//Send message
driver.findElement(By.xpath("//*[@resource-
id='com.whatsapp:id/entry']")).sendKeys("Hi Sis");
Thread.sleep(5000);
// click on send button
driver.findElement(By.xpath("//*[@resource-id='com.whatsapp:id/send']")).click();
Thread.sleep(5000);
//close app
driver.closeApp();
Thread.sleep(5000);
break;
}
catch (Exception e)
{
/* Swipe from Bottom to Top.(because in screen if name not exist it will scroll
bottom to Top */
driver.swipe(startx, starty, startx, endy, 3000);
Thread.sleep(2000);
}
}
}
}
```

**Example 3:-**

- ⇒ Launch gmail Site using ARD
- Get ARD details using adb devices
- ⇒ Login with credential's
- ⇒ Close site

**Test Script:**

```
package app;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.remote.MobileCapabilityType;
import java.net.URL;
import java.util.Scanner;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
```

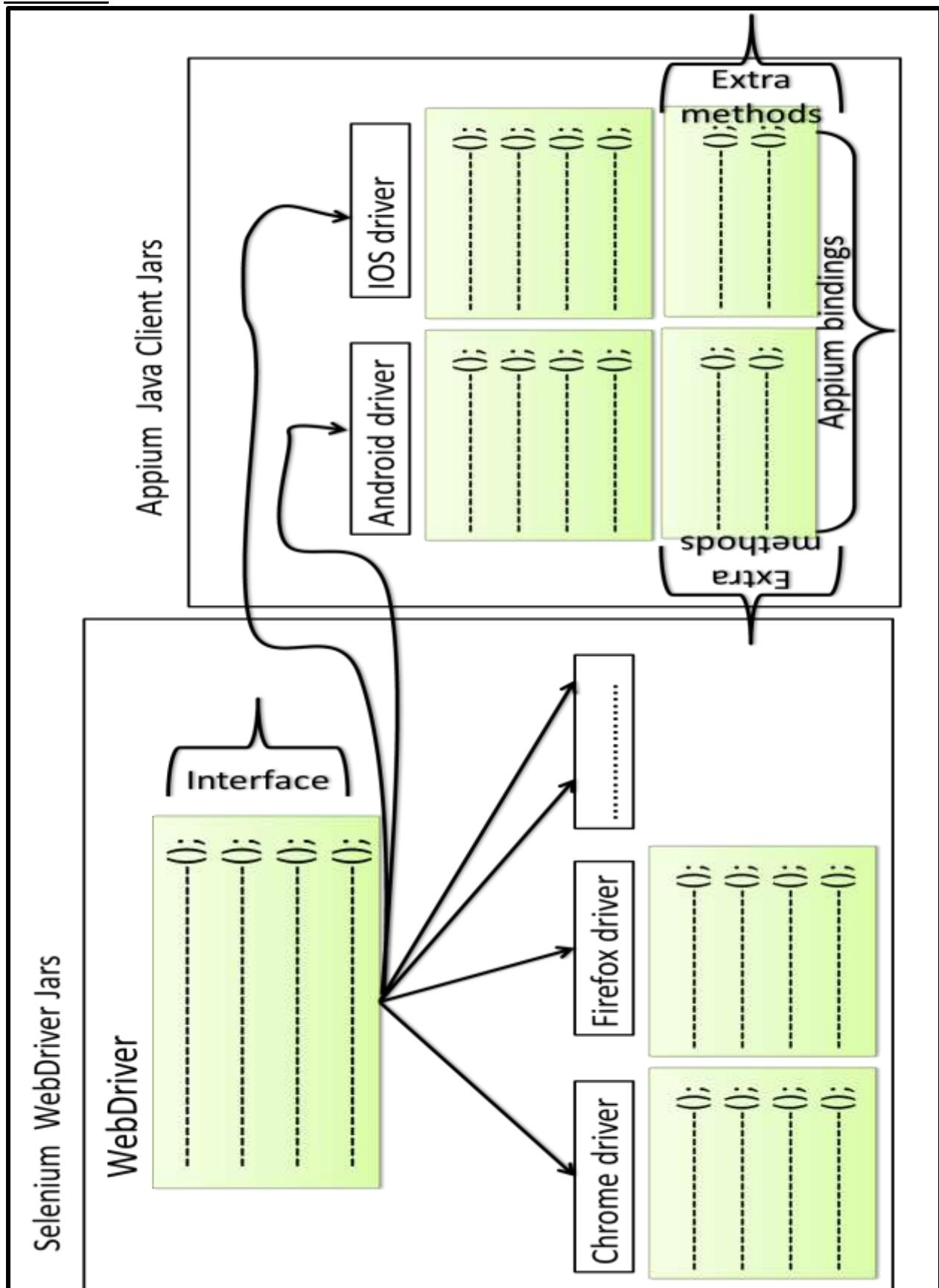
```

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
public class GmailUsingARD
{
public static void main(String[] args) throws Exception
{
WebDriver driver;
Scanner sc=new Scanner(System.in);
System.out.println("Enter Mobile|PC");
String device=sc.nextLine();
if(device.equalsIgnoreCase("device"))
{
DesiredCapabilities d = new DesiredCapabilities();
d.setCapability(MobileCapabilityType.BROWSER_NAME, "CHROME");
d.setCapability(MobileCapabilityType.DEVICE_NAME, "FYTOOBWORCEAZLK7");
d.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
d.setCapability(MobileCapabilityType.PLATFORM_VERSION, "6.0");
URL u = new URL("http://127.0.0.1:4723/wd/hub");
driver = new AndroidDriver(u, d);
Thread.sleep(5000);
}
else
{
System.setProperty("webdriver.chrome.driver", "d:\\chromedriver.exe");
driver =new ChromeDriver();
}
driver.get("http://www.gmail.com");
Thread.sleep(5000);
driver.findElement(By.id("identifierId")).sendKeys("xxxx@gmail.com",Keys.ENTER);
Thread.sleep(1000);
driver.findElement(By.name("password")).sendKeys("xxxx",Keys.ENTER);
Thread.sleep(2000);
driver.close();
}
}

```

**Note 1:-**

APP Test Automation	Site Test Automation
Get .apk file path of app from developer	Get url from developers
Run “aapt” command to get “app package name” & “app activity name for that “.apk”	No need of app package & app activity
Install “.apk” into AVD/ARD	No need of install but browser need EX:- Chrome
Develop test scripts by giving “device name”, “platform Name”, “App package”, “App Activity”. Here BROWSER_NAME is null	Develop test scripts by giving “device name”, “platform Name”, “App package”, “App Activity”. BROWSER_NAME as CHROME
Create Object to “ANDROID Driver” class in Appium Java Client Jar	Create Object “Web Driver” interface in SWD Jars, but constructor methods is related to “Android Driver” class in Appium Java Client JAR
UiAutomatorviewer is helpful to get details of elements in app screen in AVD/ARD	Open site in computer & firebug/Ospy like inspector to get details for web elements in pages

Note 2:-Note 3:-

"Android driver" & "iOS Driver" class are having bodies to method declaration in "**selenium webdriver**" interface but these classes having extra methods also called as "**Appium Bindings**".

**1. CloseApp():-**

we can use this method to close active app in mobile device.

**Syntax:-**

```
driver.closeApp();
```

**2. rotate():-**

we can use this method to rotate device screen landscape (or) portrait.

**Syntax:-**

```
driver.rotate(ScreenOrientation.LANDSCAPE);
```

```
Thread.sleep(2000);
```

```
driver.rotate(ScreenOrientation.PORTRAIT);
```

**3. pinch():-**

we can use this method to get long press effect on device screen.

**Syntax:-**

```
driver.pinch(driver.findElement(By.xpath(locator)));
```

OR

```
int x=driver.findElement(By.xpath(locator")).getLocation().getX();
```

```
int y=driver.findElement(By.xpath(locator")).getLocation().getY();
```

```
driver.pinch(x, y);
```

**4. hideKeyboard():-**

we can use this method to hide keyboard when it was opened in active app screen.

**Syntax:-**

```
Thread.sleep(5000);
```

```
driver.hideKeyboard();
```

**5. openNotifications():-**

we can use this method to open notification screens in device

**Syntax:-**

```
driver.openNotifications();
```

**6. pressKeyCode():-**

we can use this method to get effect of android defaults keys like HOME, SETTINGS, MENU, BACK, ....etc

**Syntax:-**

```
driver.pressKeyCode(AndroidKeyCode.HOME);
```

```
driver.pressKeyCode(AndroidKeyCode.MENU);
```

```
driver.pressKeyCode(AndroidKeyCode.SETTINGS);
```

```
driver.pressKeyCode(AndroidKeyCode.BACK);
```

// AndroidKeyCode is a static class in Appium Java Client.

**7. zoom():-**

we can use this method to zoom to specify element in mobile app screen

**Syntax:-**

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

```
driver.zoom(e);
```

or

```
int x=e.getLocation().getX();
```

```
int y=e.getLocation().getY();
```

```
driver.zoom(x,y);
```

**8. context():-**

we can use this method to change mobile app screen as window to get width & height.

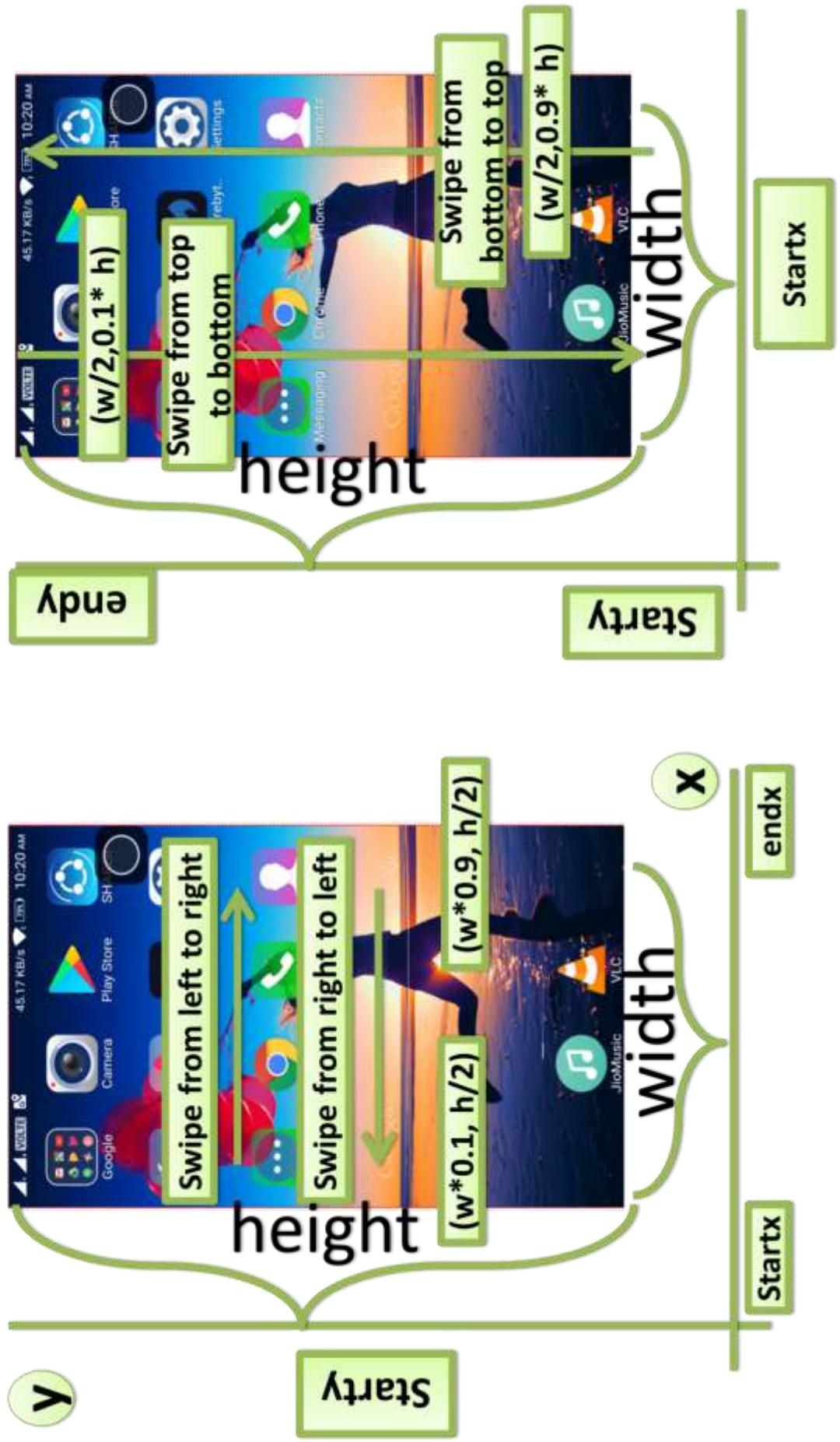
**Syntax:-**

```
driver.context("NATIVE_APP");
```

### 9. swipe():-

we can use this method to swipe effect on mobile screen.

**Mobile Screen either bottom to top, top to bottom Or right to left, left to right**



**Syntax:-**

```
//Find swipe start and end point from screen's width and height.

//Find startx point which is at right side of screen.
int startx = (int) (size.width * 0.9);

//Find endx point which is at left side of screen.
int endx = (int) (size.width * 0.1);

//Find vertical point where you wants to swipe. It is in middle of screen height.
int starty = size.height / 2;
System.out.println("startx = " + startx + " ,endx = " + endx + " , starty = " +
starty);

//Swipe from Right to Left.
driver.swipe(startx, starty, endx, starty, 5000);
Thread.sleep(2000);

//Swipe from Left to Right.
driver.swipe(endx, starty, startx, starty, 5000);
Thread.sleep(2000);

//Find swipe start and end point from screen's width and height.

//Find starty point which is at bottom side of screen.
int starty = (int) (size.height * 0.80);

//Find endy point which is at top side of screen.
int endy = (int) (size.height * 0.20);

//Find horizontal point where you wants to swipe. It is in middle of screen width.
int startx = size.width / 2;

System.out.println("starty = " + starty + " ,endy = " + endy + " , startx = " +
startx);

//Swipe from Bottom to Top.
driver.swipe(startx, starty, startx, endy, 3000);
Thread.sleep(2000);

//Swipe from Top to Bottom.
driver.swipe(startx, endy, startx, starty, 3000);
Thread.sleep(2000);
```

## ii). Tester computer OS is MAC (APPLE):-



**Note 1:-**

While working in **windows** platform for android apps testing (or) while working in **mac** platform for android apps & ios apps testing, we are able to connect real mobile devices to tester computer via Wi-Fi also by following below navigation.

1. Select “stay awake” & “USB debugging” in developer options” in settings of ARD then Connect that ARD to computer via USB.
2. Make sure device and computer are connected to the same Wi-Fi.
3. Run below command to reset a port no to ARD make it work over tcpip:  
`adb tcpip 5555`
4. Disconnect ARD from tester computer  
Get IP address of your phone (“**Settings**” -> **Wifi** -> “**Your connected network**” -> **Your IP address**”)
5. Run this command to connect adb to ARD over Wi-Fi using IP address:  
`adb connect <your phones ip address>`  
ex:- `adb connect 192.168.0.111:5555`
6. Verify, that adb works remotely through below comand  
`adb devices`
7. Run tests over Wi-Fi!





## VII. SOAP UI

### (Service Oriented Application Programming User Interface)

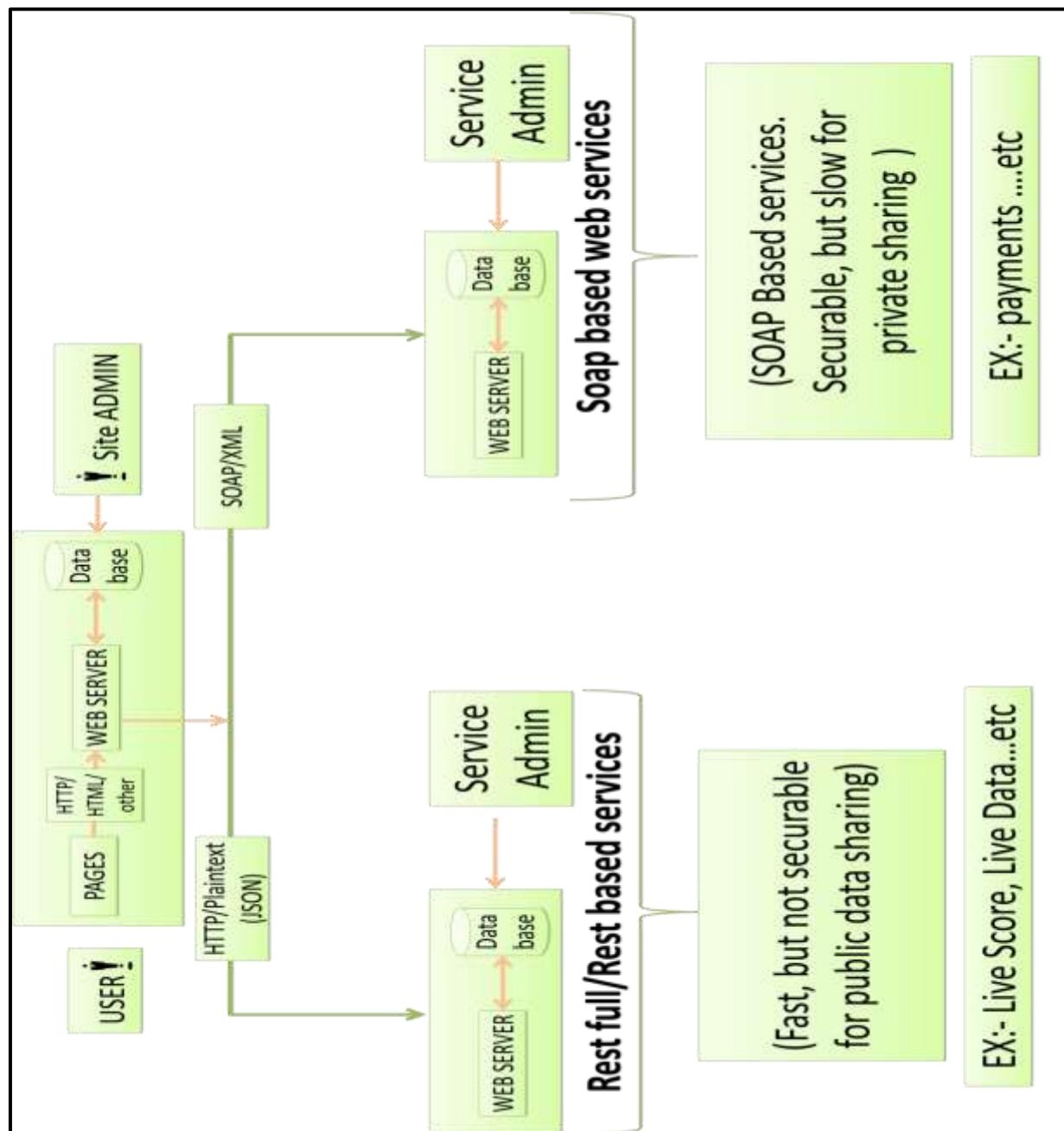
- ⇒ Open Source Tool
- ⇒ Available as licensed, but called as “SOAP UI PRO”
- ⇒ Developed by smart bear.
- ⇒ Available as a tool (So we need for IDEs).
- ⇒ To automate web services (“service oriented applications”).
- ⇒ GUI Testing means best for functionality for webpages and app screens.
- ⇒ API Testing means test the functionality for services in server computer.
- ⇒ DB testing means test the correctness of data base in server computer

### A. Types of Web Services:-

Web services is a n-tier software. It is not having front-end because it can respond to other websites (or) mobile Apps by sharing data.

Web services are two types:-

- i). Soap based
- ii). Rest based



WS:- web server.  
 DB:-data base.  
 HTTP:-Hyper Text Transfer Protocol.  
 HTML:-Hyper Text Markup Language.  
 SOAP:-Simple object Access Protocol.  
 JSON:-Java Script Object Notation  
 XML:- Extensible Markup Language  
 REST:- Representational state transfer

## B. Soap Based Services v/s Restful Services:-

Soap based services	Rest Based
⇒ Soap based services are securable (Use Encryption/Decryption)	⇒ Rest based services are not securable (No encryption/Decryption)
⇒ Slow in performance	⇒ Fast in performance
⇒ Suitable for private data transfer	⇒ Suitable for public data transfer
⇒ Services address is called as WSDL(Web services description language)	⇒ Services address is called as WADL (Web application description language)
⇒ Every services is having one/more functionality	⇒ Every services is having one functionality
⇒ Access thru soap protocol	⇒ Access thru HTTP
⇒ Language is xml for request & response	⇒ Language is plain text(JSON) for request & response
⇒ Used by web sites (OR) Mobile Hybrid Apps	⇒ Used by websites(Or) Mobile Hybrid Apps

## C. Configure SOAP UI TOOL in Tester Computer:-

- ↓ Go to [www.google.com](http://www.google.com) site.
- ↓ Enter SOAPUI TOOL Download.
- ↓ Click Search
- ↓ Go to official site <https://www.soapui.org/downloads/soapui.html>
- ↓ Select SoapUI Open Source



- ↓ Click On
- ↓ Wait until download completed.
- ↓ Paste that download in personal folder.
- ↓ Double click on that download.
- ↓ Click next until finish



- ↓ Observe SoapUI logo shortcut on desktop.

## D. Example WSDL's & WADL's:-

### WSDL:-

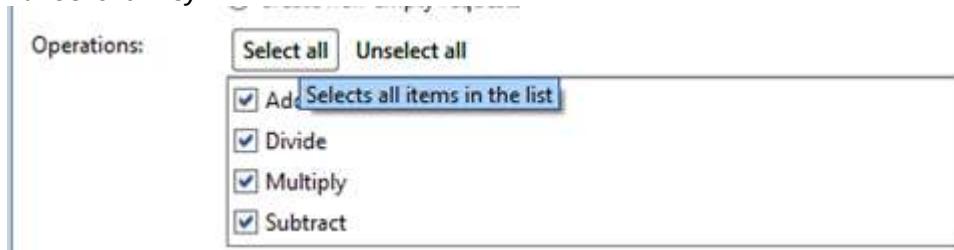
- ⇒ [www.webservicex.com/globalweather.asmx?WSDL](http://www.webservicex.com/globalweather.asmx?WSDL)
- ⇒ <http://www.thomas-bayer.com/axis2/services/BLZService?wsdl>
- ⇒ <http://wsf.cdyne.com/WeatherWS/Weather.asmx?WSDL>
- ⇒ <http://www.webservicex.net/geoipservice.asmx?WSDL>
- ⇒ <http://www.dneonline.com/calculator.asmx?wsdl>
- ⇒ <https://www.w3schools.com/xml/tempconvert.asmx?wsdl>
- ⇒ <http://currencyconverter.kowabunga.net/converter.asmx?wsdl>
- ⇒ <http://www.webservicex.com/country.asmx?wsdl>
- ⇒ <http://freewebservicesx.com/live-sensex-bse-index.asmx?WSDL>

**WADL:-**

- ⇒ <http://maps.googleapis.com/maps/api/geocode/json?address=xxxx&sensor=true>
- ⇒ [http://www.herongyang.com/Service/Hello\\_REST.wadl](http://www.herongyang.com/Service/Hello_REST.wadl)

**E. Soap based test automation****Step 1:- Collect service information from developers****Example:-**

- ⇒ WSDL → <http://www.dneonline.com/calculator.asmx?wsdl>
- ⇒ Functionality

**Note:-**

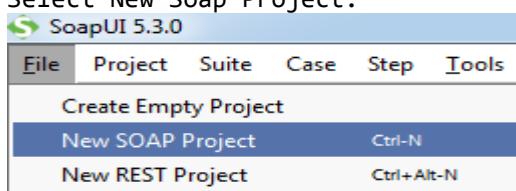
We are able to get above like information by WSDL path in Browser information

**Step 2:- Create Project in SoapUI Tool**

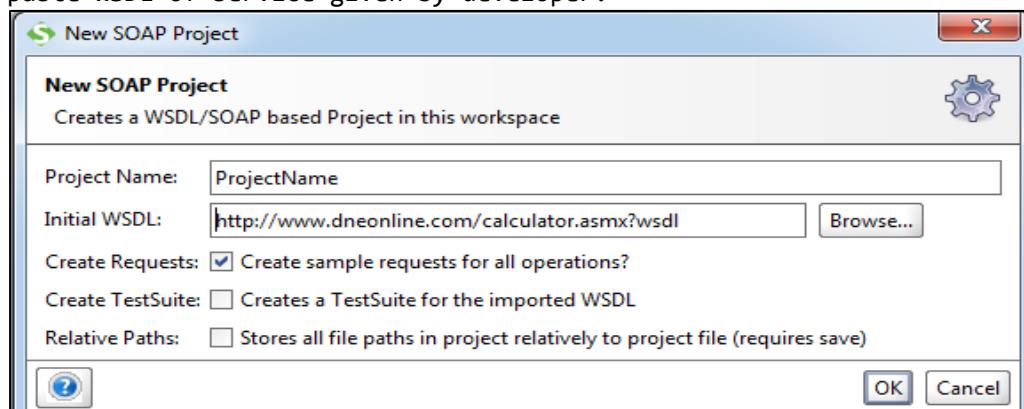
- ↓ Launch SoapUI Tool by double click on icon.



- ↓ Go to File menu.
- ↓ Select New Workspace.
- ↓ Enter a name to workspace.
- ↓ Click Ok.
- ↓ Browser path of personal folder.
- ↓ Click Save.
- ↓ Right Click on workspace.
- ↓ Select New Soap Project.



- ↓ Enter a name to project.
- ↓ paste WSDL of service given by developer.



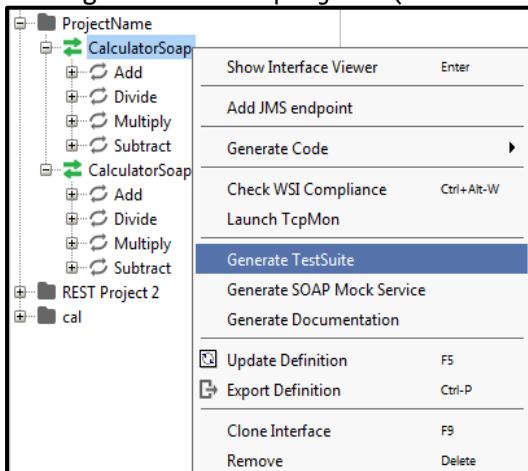
- ↓ Click ok
- ↓ Remove Soap2 (2<sup>nd</sup> of soap)
- ↓ Right Click on CalculatorSoap12 & Select Remove
- ↓ Click yes



### Step 3:- Create Test Cases:

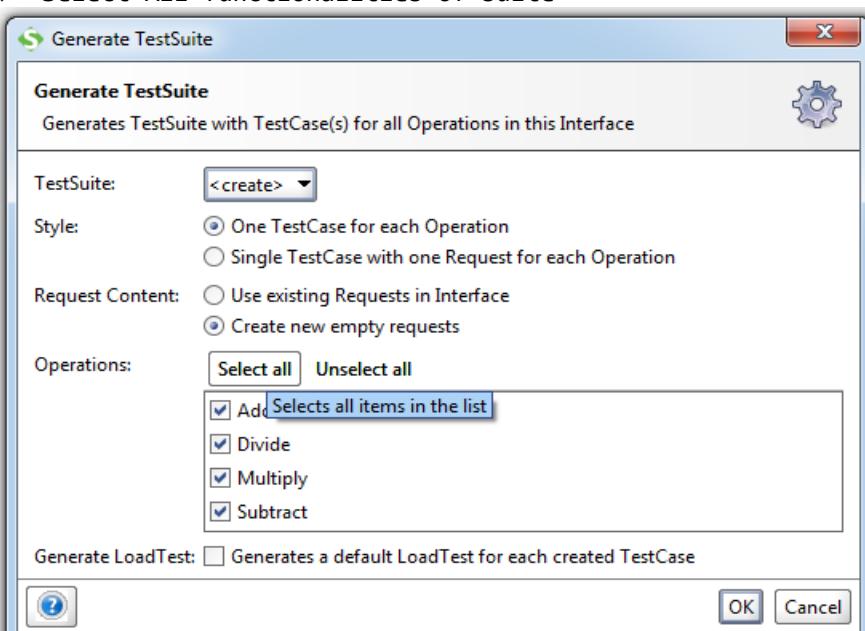
After Completion of project creation with wsdl, we can get list of functionalities in that service. But we need to create test cases to test functionalities by following below navigation

↓ Right Click on project (CalculatorSoap)



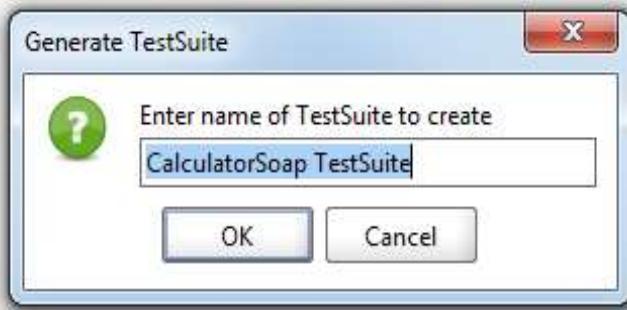
↓ Select Generate TestSuite

↓ Select All functionalities of suite



↓ Click OK

↓ Enter a name of TestSuite

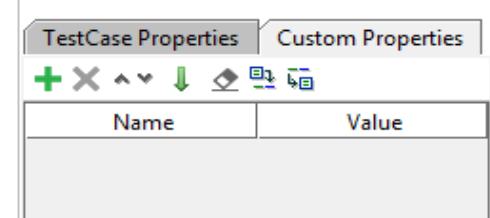
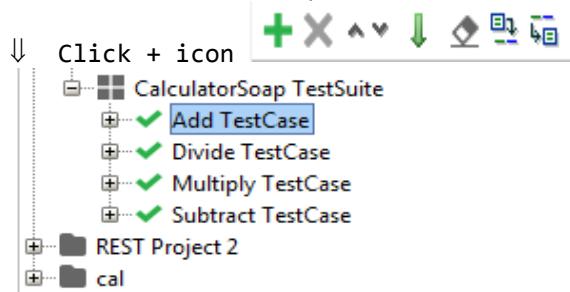


↓ Click OK

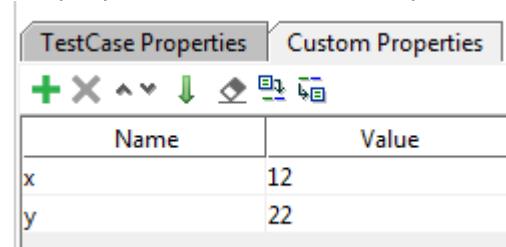
#### **Step 4:- parameterize Request:**

↓ Click on corresponding Test case Name in suite.

↓ Select Custom Properties



↓ Enter a name to property with default value, click add icon to create More properties in Custom Properties

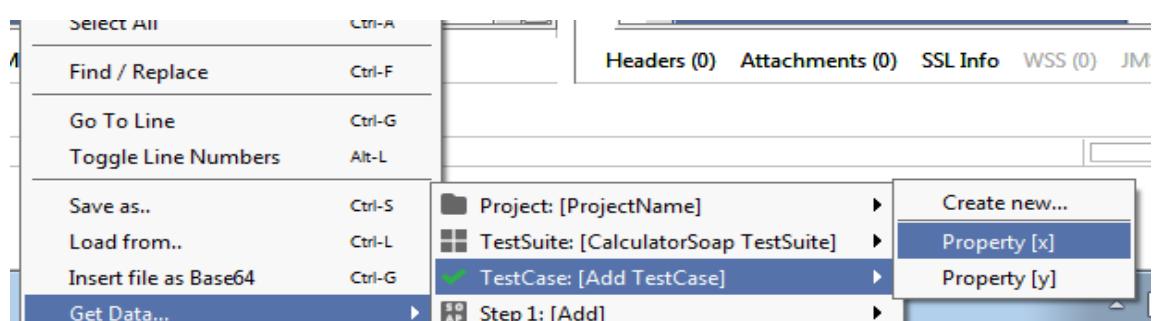


↓ Double Click on corresponding test case

↓ Open xml request in steps in corresponding case

↓ Remove ? in xml request

↓ Right Click & Select Get Data



- ↓ Select Test Case & Select Property
- ↓ Follow above navigation to replace ? with properties in xml request
 

```
<tem:Add>
 <tem:intA>#${TestCase#x}</tem:intA>
 <tem:intB>#${TestCase#y}</tem:intB>
</tem:Add>
```

### **Step 5:- Insert Assertion:**

After completion of request parameterization, we need to insert assertions for that request .Assertion can take expected value. While running request Assertion can compare expected value with actual value in response.

- ↓ Click On Assertion

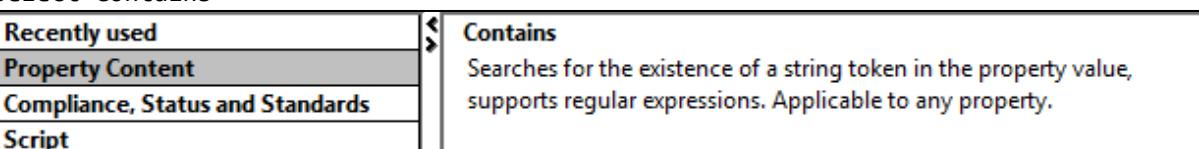


- ↓ Click ADD(+) icon



- ↓ Select Property content

- ↓ Select Contains



- ↓ Click Add



- ↓ Enter Expected Value

- ↓ Click Ok



Follow above navigation to create more assertions if needed

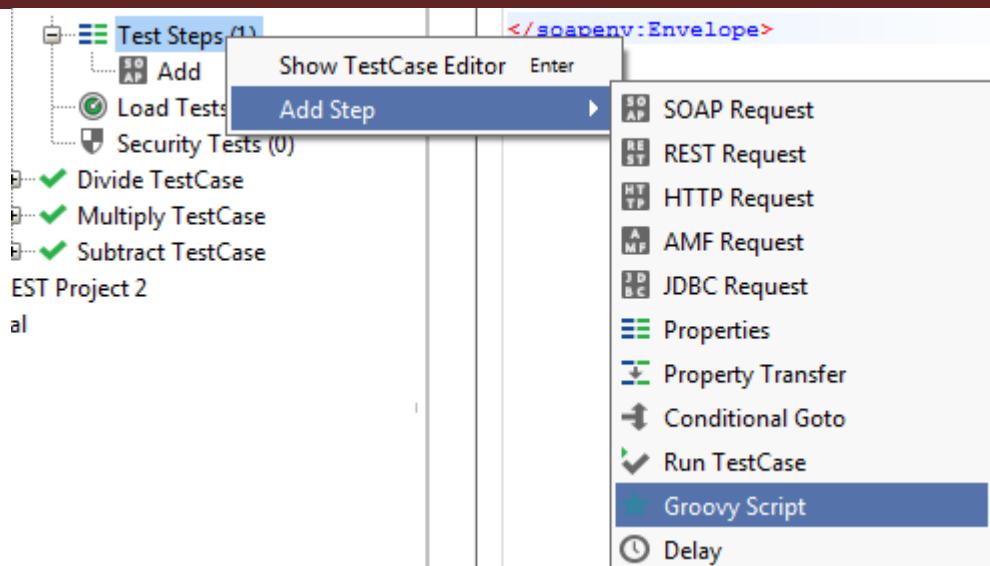
### **Step 6:- Insert Groovy Script:**

SOAPUI tool can follow us to run request on service in two ways such as click

- \* ★ **Groovy Script** run icon manually in request window (Triangle ▶ symbol) and
- \* ★ **Groovy Script** run groovy script code.

To write groovy Script code for any request, we need to follow below navigation

- ↓ Right Click on corresponding test steps
- ↓ Select Add Step
- ↓ Select Groovy Script with Java Code For corresponding Services operation (or) functionality testing



- ↓ Type Groovy Code
- ↓ Click Run icon
- ↓ Analysis Result
- ⇒ In general Groovy language is object based language. This language is having built in objects like:
  - testRunner
  - context
  - testStepContext
  - log
  - ...etc
- ⇒ Support new object creation also using "Classes"
- ⇒ Able to integrated with core-java & external jars like JXL

### Example 1:-

**Groovy Script**

```

1 // GroovyScript
2 def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context);
3 def tc=testRunner.testCase;
4 def ts=tc.getTestStepByName("Add");
5 testRunner=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestCaseRunner(tc,null);
6 testStepContext=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestStepContext(ts);
7 tc.setProperty("Response",testStepContext);
8 ts.run(testStepContext);
9 def res=ts.getTestStepResult();
10 def holder=groovyUtils.getXmlHolder(res.responseContent);
11 def output=holder.value["/*:AddResult"];
12 def outputText=output.text;
13 log.info(outputText);
14 //java
15 if(outputText.equals("1444"))
16 {
17 log.info("Test Passed");
18 }
19 else
20 {
21 log.info("Test failed");
22 }

```

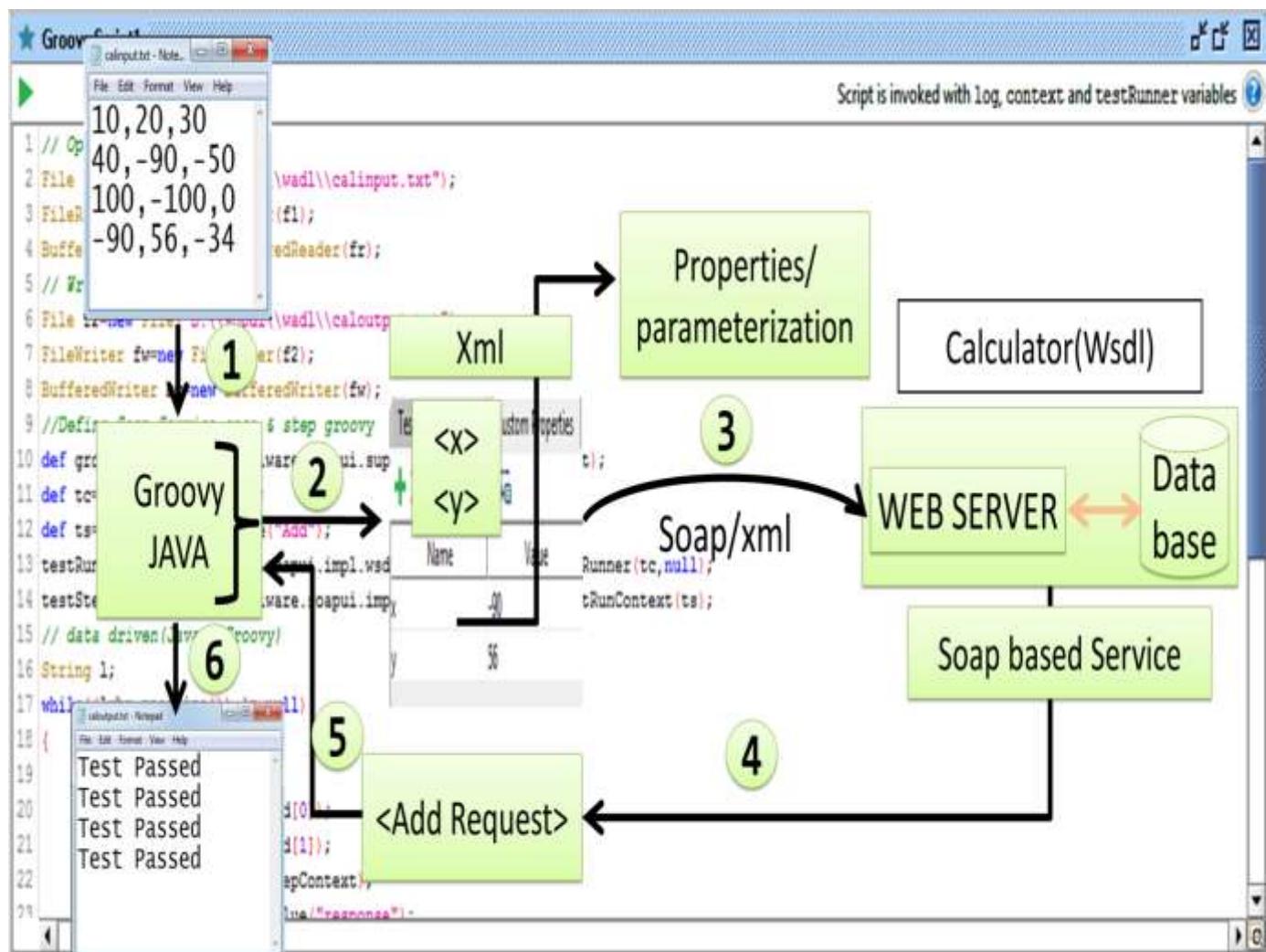
The diagram illustrates the interaction of a Soap-based service. It shows a 'Groovy JAVA' code block containing Groovy script. An 'Xml Request' block is shown being sent to a 'WEB SERVER' block. The 'WEB SERVER' block is connected to a 'Data base' block via a double-headed arrow, indicating a bidirectional relationship. The 'Soap based Service' is represented by the 'WEB SERVER' block.

**Test Script:**

```
// GroovyScript
def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context);
def tc=testRunner.testCase;
def ts=tc.getTestStepByName("Add");
testRunner=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestCaseRunner(tc,null);
testStepContext=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestRunContext(ts);
tc.setPropertyValue("x","13434");
tc.setPropertyValue("y","222");
ts.run(testRunner,testStepContext);
def res=ts.getPropertyValue("response");
def holder=groovyUtils.getXmlHolder("$res");
def output=holder.getNodeValue("//*:AddResult");
log.info(output);
//java
if(output.equals("1444"))
{
 log.info("Test Passed");
}
else
{
 log.info("Test failed");
}
```

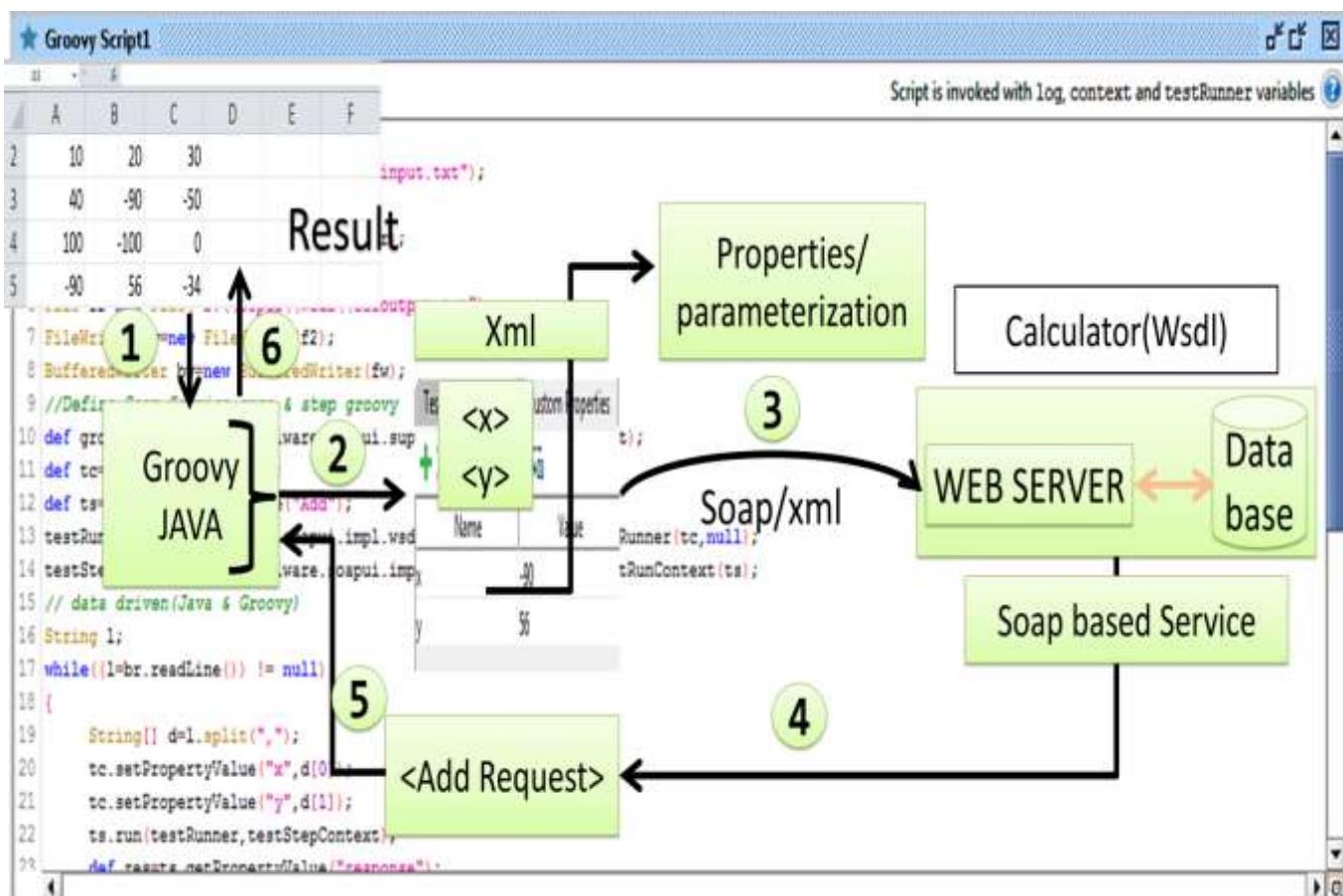
**Example 2:-**

- ⇒ Connect to calculator services using wsdl.
- ⇒ Parametric “add” request with test case level properties.  
Ex:-x,y
- ⇒ Validate response output value.



**Test Script:**

```
// Open File
File f1=new File("D:\\sopui\\wadl\\calinput.txt");
FileReader fr=new FileReader(f1);
BufferedReader br=new BufferedReader(fr);
// Write data File
File f2=new File("D:\\sopui\\wadl\\caloutput.txt");
FileWriter fw=new FileWriter(f2);
BufferedWriter bw=new BufferedWriter(fw);
//Define Soap Service case & step groovy
def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context);
def tc=testRunner.testCase;
def ts=tc.getTestStepByName("Add");
testRunner=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestCaseRunner(tc,null);
testStepContext=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestRunContext(ts);
// data driven(Java & Groovy)
String l;
while((l=br.readLine()) != null)
{
 String[] d=l.split(",");
 tc.setPropertyValue("x",d[0]);
 tc.setPropertyValue("y",d[1]);
 ts.run(testRunner,testStepContext);
 def res=ts.getPropertyValue("response");
 def holder=groovyUtils.getXmlHolder("$res");
 def output=holder.getNodeValue("//*:AddResult");
 //java
 if(output.equals(d[2]))
 {
 bw.write("Test Passed");
 bw.newLine();
 }
 else
 {
 bw.write("Test Failed");
 bw.newLine();
 }
}
br.close();
fr.close();
bw.close();
fw.close();
```

Example 3:-

In general, soapui tool can support java language & groovy script by default. If we want to use any external jar like Jxl We need to follow below navigation:-

- ↓ Close soapui tool
- ↓ Download Jxl jar (Java2s.com site)
- ↓ Paste that download in personal folder
- ↓ Go to C:\Program Files (x86)\SmartBear\SoapUI-5.3.0\lib
- ↓ Paste that Jar in that folder
- ↓ Launch soapui tool

Test Script:

```

import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;

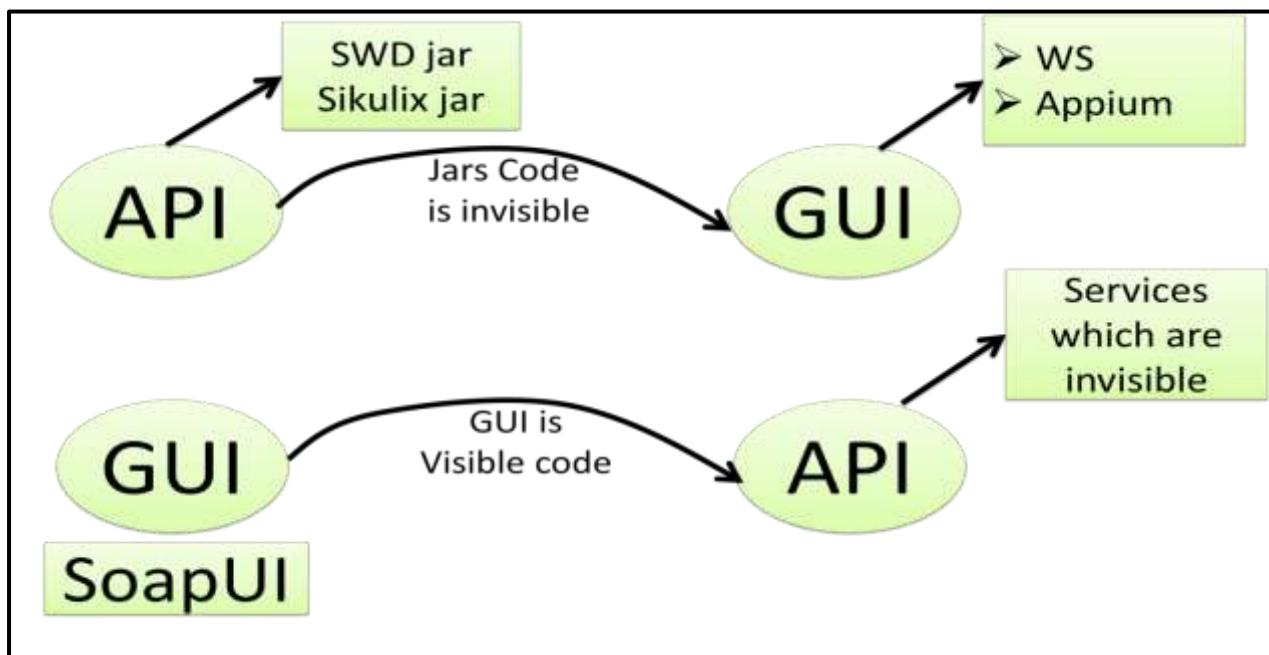
File f = new File("D:\\sopui\\wadl\\Cal1.xls");
Workbook rwb = Workbook.getWorkbook(f);
Sheet rsh = rwb.getSheet(0); // Sheet1 because index0
int nour = rsh.getRows(); // Get no of used rows
int nouc = rsh.getColumns(); // Get no of used columns
// open same excel sheet for writing result
WritableWorkbook wwb = Workbook.createWorkbook(f, rwb);
WritableSheet wsh = wwb.getSheet(0); // sheet1 insert data
//Define Soap Service case & step groovy
def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context);
def tc=testRunner.testCase;
def ts=tc.getTestStepByName("Add");
testRunner=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestCaseRunner(tc,null);
testStepContext=new com.eviware.soapui.impl.wsdl.testcase.WsdlTestRunContext(ts);
// data driven(Java & Groovy)
for(int i=1;i<nour;i++)

```

```

{
String a = rsh.getCell(0, i).getContents();
String b = rsh.getCell(1, i).getContents();
 tc.setPropertyValue("x",a);
 tc.setPropertyValue("y",b);
 ts.run(testRunner,testStepContext);
def res=ts.getPropertyValue("response");
def holder=groovyUtils.getXmlHolder("$res");
def output=holder.getNodeValue("//*:AddResult");
//java
if(output.equals(rsh.getCell(2, i).getContents()))
{
 Label l=new Label(nouc,i,"Test Passed");
 wsh.addCell(l);
}
else
{
 Label l=new Label(nouc,i,"Test Failed");
 wsh.addCell(l);
}
wwb.write();
wwb.close();
rwb.close();
}

```

Note1

GUI	Open Sources	Jars API
⇒ Web Pages	⇒ Selenium web driver	
⇒ Screens	⇒ Appium Java Client	
⇒ Mobiles Apps	⇒ Sikuli ⇒ Java	

API	Open Sources	Executable file
⇒ Soap based	⇒ SoapUI Tool	
⇒ Rest full services		

## F. Restable Service automation

### Step 1:- Collect service information from developers

Example:-

- ⇒ WADL → <http://maps.googleapis.com/maps/api/geocode/json?address=xxxx&sensor=true>
- ⇒ <http://maps.googleapis.com/maps/api/geocode/json?> → this is service path
- ⇒ address=xxxx&sensor=true → it is properties

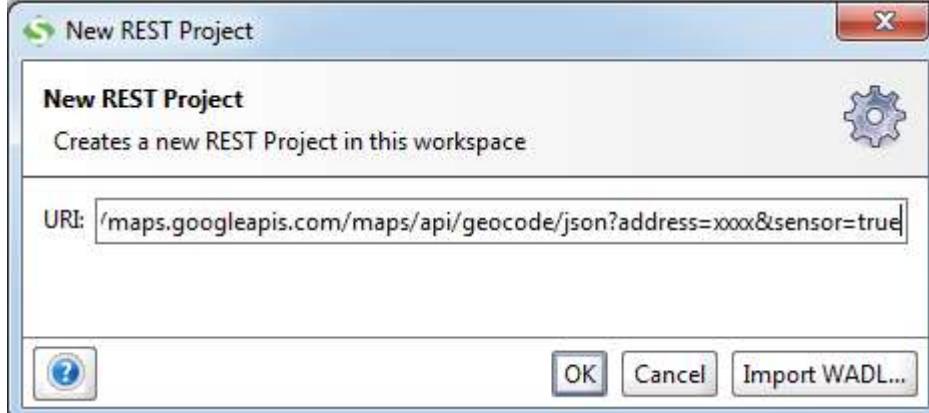
### Step 2:- Create Project

- ⇒ Right Click on workspace

New REST Project      Ctrl+Alt+N

- ⇒ Select New Rest Project

- ⇒ Paste that WADL Path

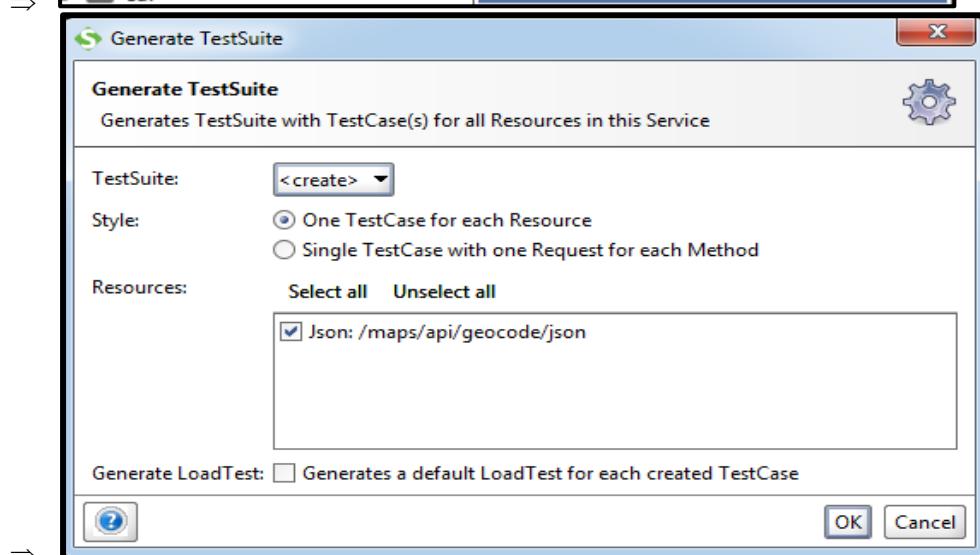


- ⇒ Click Ok

### Step 3:- Create Test Suite

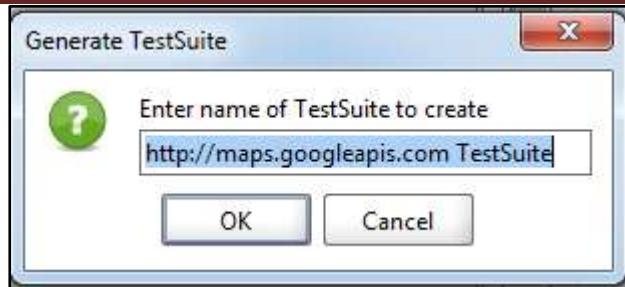
- ⇒ Right click on Project

- ⇒ Select Generate New Suite

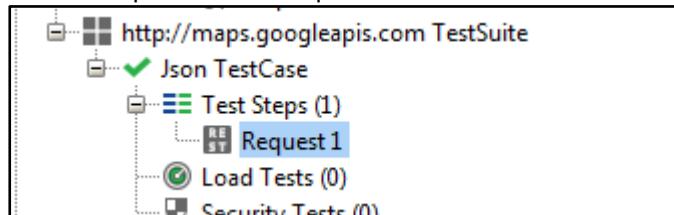


- ⇒ Click Ok

- ⇒ Enter a name to Suite



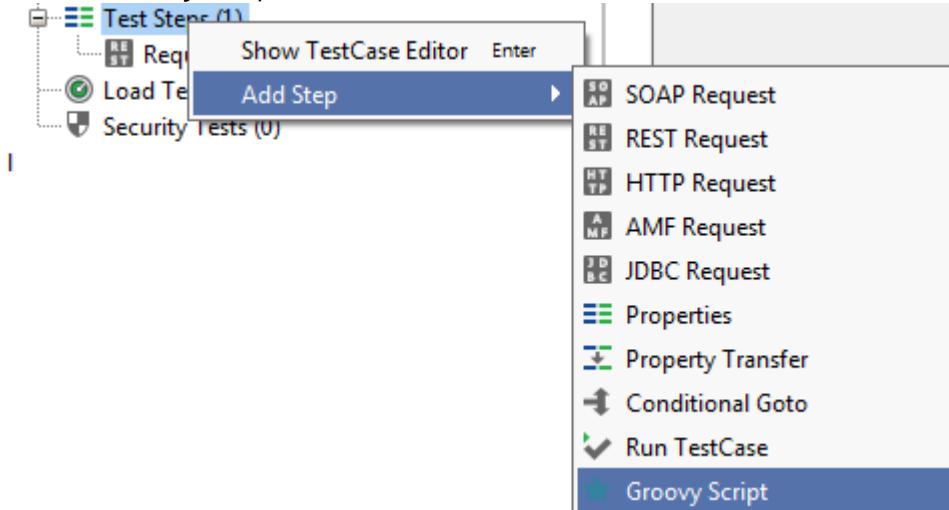
- ⇒ Click Ok
- ⇒ Go to request in "Steps" in "case" in "Suite"



- ⇒ Provide data to request properties (Properties are in built)

Name	Value	Style	Level
address	Guntur	QUERY	RESOURCE
sensor	true	QUERY	RESOURCE

- ⇒ Right Click On Test Steps
- ⇒ Select Add Step
- ⇒ Select GroovyScript

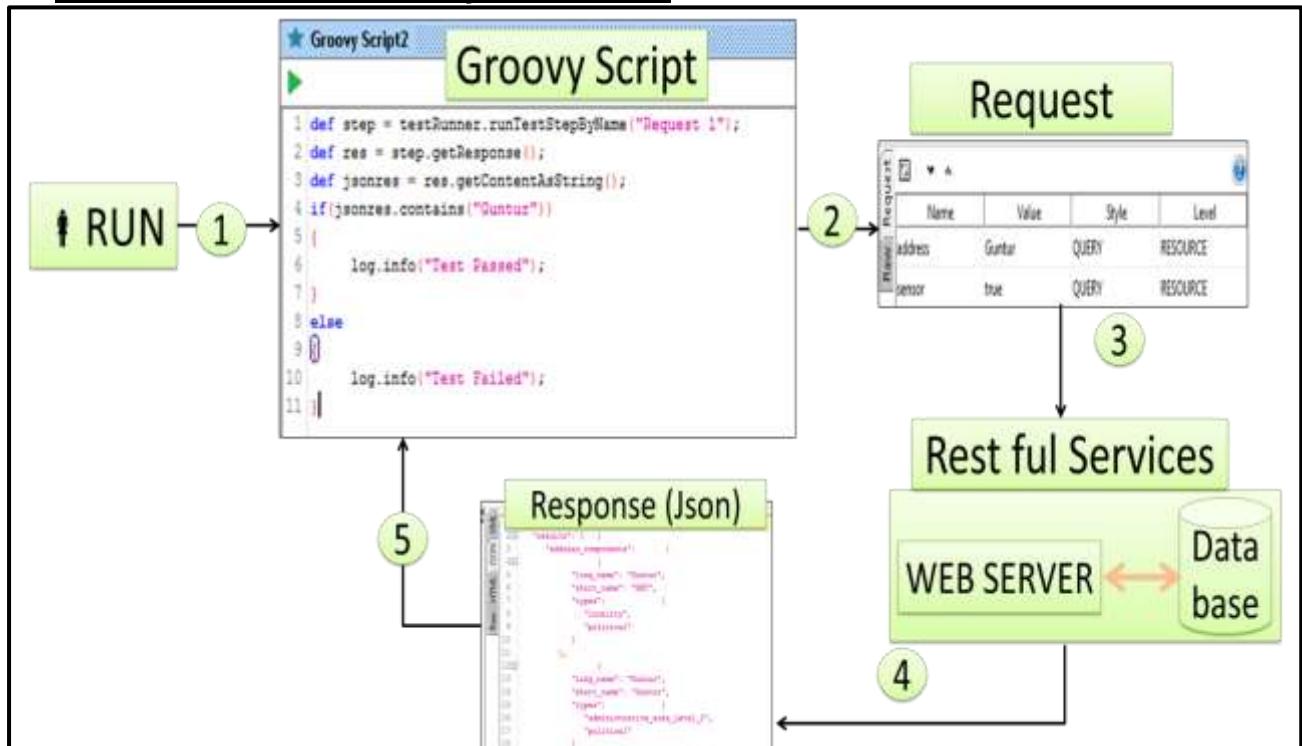


- ⇒ Enter a name to Groovy Script
- ⇒ Click Ok



- ⇒ Write GroovyScript

## Step 4:- Write Groovy Script:-



```

def step = testRunner.runTestStepByName("Request 1");
def res = step.getResponse();
def jsonres = res.getContentAsString();
if(jsonres.contains("Guntur"))
{
 log.info("Test Passed");
}
else
{
 log.info("Test Failed");
}

```