Shree

04/12/2017

SELENIUM

Important Manual Topics :-

1) Regression Testing
   - Unit regression
   - Regional regression
   - Full regression testing

2) Software Testing Life Cycle :- STLC

3) Defect Life Cycle

4) Severity & priority

5) Compatablity Testing

6) Usablity Testing

7) performance Testing

8) Exploratory Testing & Adhoc Testing :-
   No Test case no automation.

9) Manual Test case

Important Java Topics :-

1) class
2) Object
3) Members (Static & Non static)
4) OOP's Concepts
   - ① Encapsulation ② Abstraction ③ Inheritance
   - ④ polymorphism
     - ⓐ Static compiletime polymorphism
     - ⓑ Dynamic Runtime polymorphism.

__/__/____

5) Condition Statements (if Statements)
6) looping Statements (For loop)
7) Exception Handling (try, catch, block)
   [Compile Time Exception - Which is checked exception]
   [Run Time Exception - Which is unchecked exception]
8) Collections - (Set & List)

CODE OPTIMIZATION 05/12/2017

The process of reducing number of statements but still getting same output by performing same action is called as code optimization.

To optimize a code following 2 conditions should be satisfied.

(1) There must be a Common Variable between the two statements.

(2) There must be an assignment operator for that Common Variable in 1st Statement.

Example program:-

```
int i = 10;
int j = 20;
int k = i+j;
Sopln (k);
```

⇒ or

```
int i = 10;
int j = 20;
Sopln (i+j);
```

⇒ or

```
int i = 10;
Sopln (i+20);
```

⇓ or

```
SOPln (10+20);
```

2nd Example program:-

```
String s = "QSP";
int l = s.length();
Sopln (l);
```

⇒

```
String s = "QSP";
SOPln (s.length());
```

⇓

```
Sopln ("QSP".length());
```

__/__/____

3rd Example program :-

```
class A
{
String s = "Qsp";
}
A a1 = new A();
String v = a1.s;
int L = v.length();
Sopln(L);
```

or ⟹

```
A a1 = new A();
Sopln(a1.s.length());
```

or ⟹

```
A a1 = new A();
String v = a1.s;
Sopln(v.length());
```

or ⟹

```
Sopln(new A().s.length());
```

4th Example program :-

```
class Dog
{
Static String name = "Koriya";
}
```

⟹

```
String v = Dog.name;
Sopln(v.length());
```

⟹

```
String v = Dog.name;
int L = v.length();
Sopln(L);
```

5th Example program :-

```
Class A
{
Void test(int i)
{
Sopln(i);
}
Void test(String s)
{
Sopln(s);
```

```
Class B
{
Static A a1 = new A();
}

A x1 = B.a1
x1.test(123);
B.a1.test(123);
B.a1.test("abc");
```

__/__/____

In the example B is a class & a1 is static reference Variable of type A. Test Method is not static Overloaded method present in A class.

6th Example program:-

```
class printStream
{
void println (int i)
{
SOPln (i);
}
Void println (String s)
{
SOPln (s);
}

class System
{
Static printstream out = new printstream ();
}
printstream x1 = System.out
        x1 = println(123);

System.out.println (114);
System.out.println ("Sai");
```

__/__/____

Explain System.out.println();

① System is a builtin java class (Final class)

② Out is a public static final reference variable of type Printstream.

③ Println is non static over loaded method of printstream class.

METHODS

06/12/2017

Code Optimization

Example program

```
Class A
{
String getValue()
{
String s = "abc";
return s;
}
}
Mainclass
{
PsvM (String ... args)
A a1 = new A();
String v = a1.getvalue();
int l = v.length();
Sopln (l);
      "Or"
A a1 = new A();
int l = a1.getvalue().length();
Sop (l);
```

__/__/____

"METHOD CHAINING"

process of calling multiple methods one after the other in the same statement is called as method chaining.

In the above example or previous program length method is present inside String object which is returned by getValue method.

Example No 2°-

```
V. method1().method2().method3();
```

In the above example method3() is present inside the object which is returned by method2() & method2() is present inside the object which is returned by method1() & so on.

Example No 3 ⇒ Realtime Example

1) `driver.navigate().back();` ⇒ previous page action
2) `driver.manage().Window().maximize();`
   ⇒ To maximize the window.
3) `driver.findElement(b).getlocation().getx();`

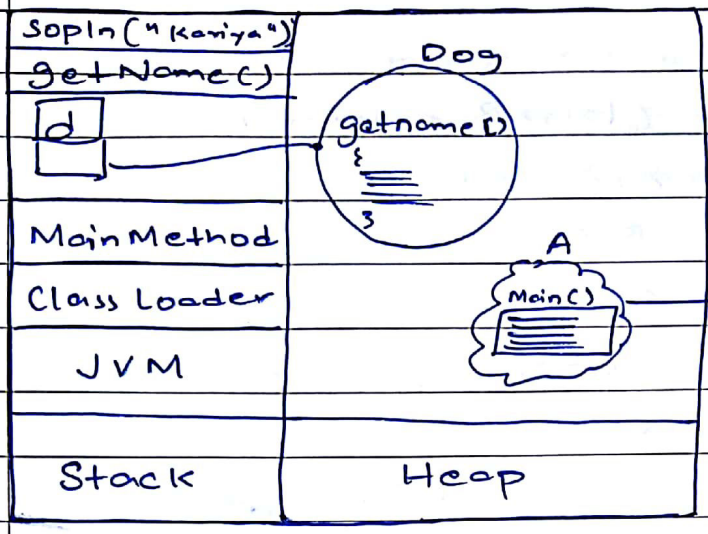
Example program for Memory Allocation

```
class A
{
PsvM (String ... args)
{
Dog d = new Dog();
d.getName();
}
}
```

```
class Dog
{
Void getName();
{
Sopln ("Kariya");
}
}
```

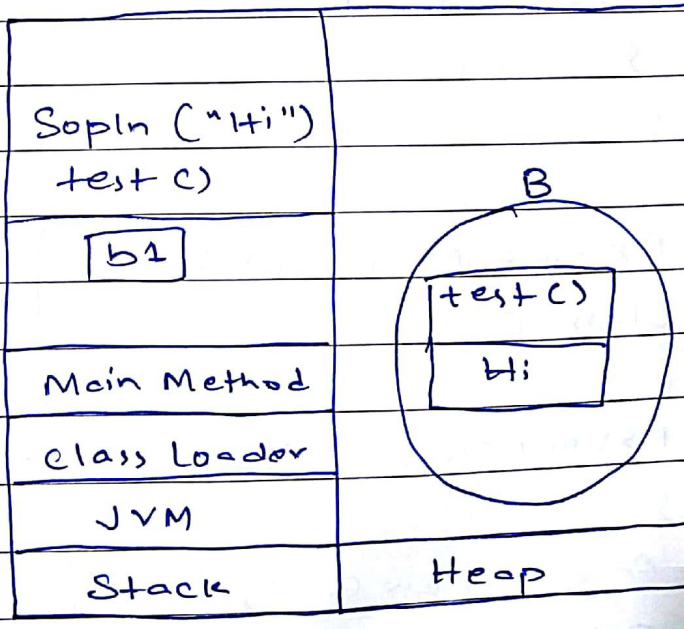
__/__/____

First in Last out



Example program for Inheritance :-

```
Class A {
A (){ Void test C){
        SOPln ("Hi");
      }
   }
Class B extends A {
 {
   B C){
   Super C);
 }
 Class C
 {
 Psvm (String ... args)
 B b1 = new BC);
  b1 .test C);
 }
}
```



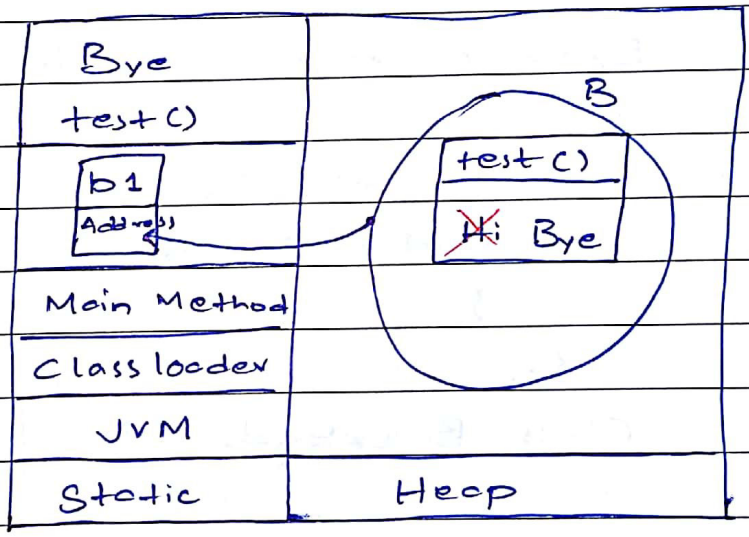
__/__/____

Example program For

```
Class A
{
Void test()
{
SOPln("Hi");
}
A()
{
}
```

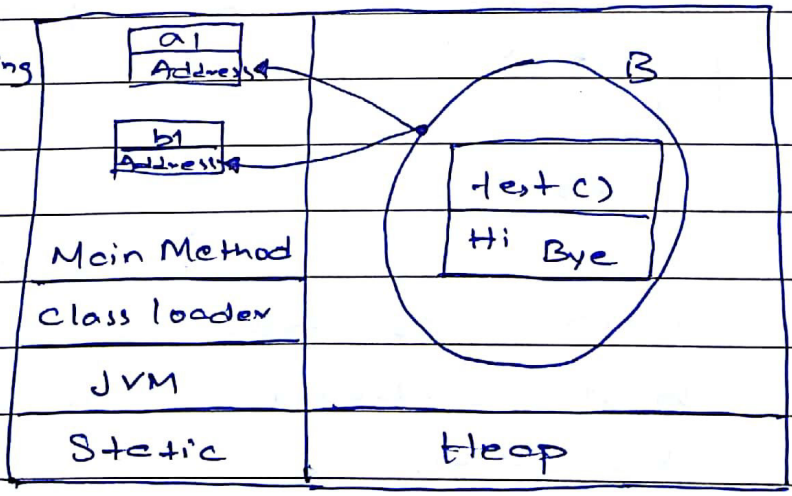
```
Class B extends A
{
B()
{
Super();
}
Void test()
{
SOPln("Bye");
}
}
```

```
Class C
{
PSVM(String ... args)
{
B b1 = new B();
b1.test();
}
}
```



Example program for Overriding

```
class C
{
PSVM(String ... args)
{
A a1 = new B();
a1.test();
}
}
```



__/__/____

NOTE :-

After upcasting if we call a method which is present in both parent & child it always executes method present in child class (overriden Method)

8/12/2017

COMPILE TIME POLYMORPHISM OR STATIC POLYMORPHISM.

1) While Compiling the java program compiler checks the Syntax of each Statement, if everything is correct it will also check for calling Statements.

2) IF any statement is calling a method it will link the Calling Statement with body of the method this is called as Compile time polymorphism (or) Static polymorphism (or) Static binding etc

3) Method overloading is an example for this.

Example

```
class A {
    void test()
    SOpln ("Dinga");
    }
    void test (String s)
    SOPln (s);
    }
}
```

```
Class B {
void test B (A a1)
{
a1.test();
}
}
```

Early Binding.

UP CASTING USING INTERFACE :-

```
interface Remote {
 void SwitchonOff();
 void changechannel();
 }
Class LCDTV implements Remote
{
Public void SwitchonOff
{
SOPln("ON OFF LCD");
```

__/__/____

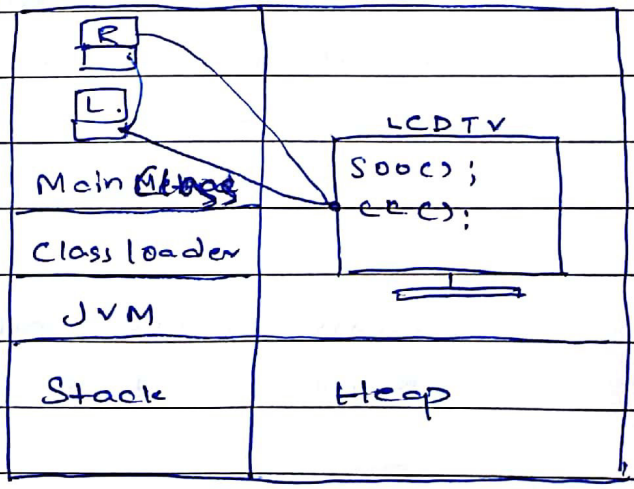
```
Public Void changechannel()
{
 SOPln(" Change channel LCD");
}
Class LEDTV implements Remote
{
Public void SwitchONOFF()
{
SOPln(" SwitchTV ON OFF");
}
Public Void changechannel()
{
SOPln(" change channel LED");
}
```

NOTE :- ① We can create reference variable for interface, Abstract class & Concrete class. But we can create object for only concrete class.

② Using reference variable of interface we can call any of the method present in interface but it always executes overriden method.

Example :-

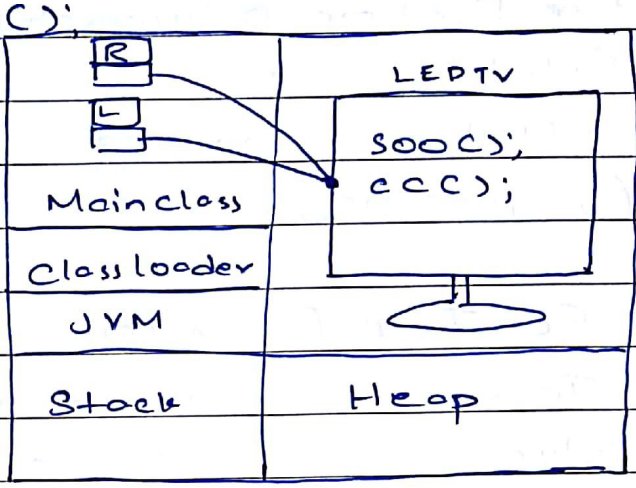
```
LCDTV l = new LCDTV();
Remote r;
r = l;
r = SwitchonOFF();
```



__/__/____

```
LEDTV L = new LEDTV();
Remote r;
r = L;
r = SwithonOFF();

Remote r1 = new LEDTV();
    r1.SOOC()
    r1.CCC()
```



O/P ⇒ ON/OFF LEDTV changechannel LED TV

NOTE:-

In the reference variable r we can store the object of any of its child class such as LCDTV, LEDTV Etc....

Example program:-

```
Class Customer
{ void TVDemo (Remote r)
  {
     r.SOSOFFC();
     r.CC C();
  }
}
```

We can Compile the above class when we compile it Compiler can only check the syntax but it cannot link Calling Statement (r.SONSOFF();) with method body, becouse it can be any of the method present in the child class.

The method binding will be done by JVM during runtime hence it is called as Late binding (or) Runtime polymorphism (or) Dynamic polymorphism.

__/__/____

~~Example Program:~~

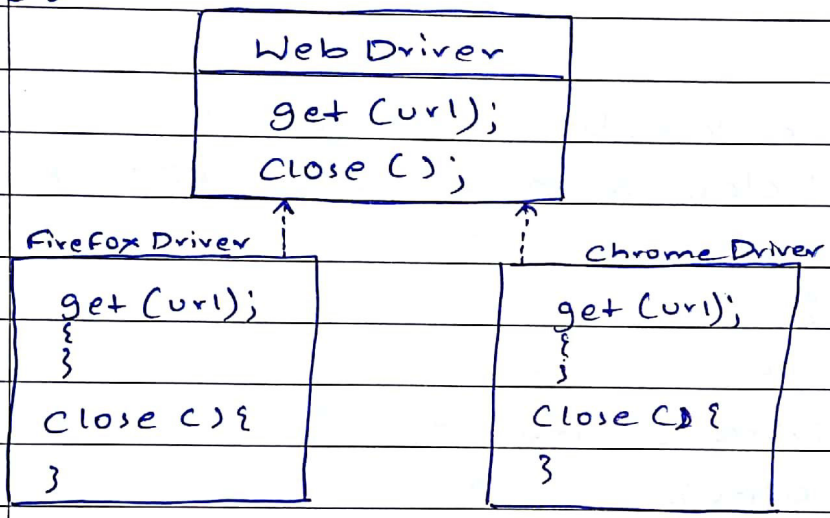
For Runtime polymorphism following steps are Mandatory

1. Inheritance
2. Method Overriding
3. Up casting.

RUN TIME POLYMORPHISM IN SELENIUM :-

1) To run the Script on any browser we use runtime polymorphism Concept in selenium.

Example :-




```
Void testScript 1 ( Fire fox driver f)
{
F.get (url);
F.close ();
}
```

Test Script 1 works only on firefox browser

```
Void testScript 1 ( chrome driver c)
{
F.get (url);
F.close ();
}
```

Test script 2 works only on chrome browser.

__/__/____

AUTOMATION

```
Void testscript3 (WebDriver d)
{
 d.get (url);
 d.close ();
}
```

Test Script 3 works on any browser.

11/12/17

"AUTOMATION"

Performing any task using a system or a tool is called as Automation.

Example :- Washing Machine, Dish Washer, Mixer, Grinder. etc...

IT Example :- EMail, Naukri.com, Conference Call.

Advantages Of Automation :-

1) Saves time Coz it is faster
2) Reduces effort
3) Multitasking.
4) Multipurpose & reusable.
5) It is Accurate & Accuracy percentage is high.
6) Machines are restless.

Disadvantages of Automation :-

1) Initial investment is high. (Because of this reason we go for automation for long term projects).
2) It requires Constant maintainence
3) Requires additional Skillset.

Q) What is Automation tool give on example ?

Ans Automation tool is a Software which is used for automation.

Example :- Example for. Test Automation tool :-

open source →
1. Selenium
2. J Meter
8. Sikuli
9. AutoIT

Paid software
3. QTP - (UFT)
4. Load Runner.
5. RFT (Rational Functional Tester by IBM)
6. Silk Test
7. Silk Performer
10. Monkey Runner

SELENIUM

__/__/____

"SELENIUM"

Selenium is a open source web application automation tool.
It can be downloaded from : http://www.seleniumhq.org/download/
look for : Selenium Standalone Server
Click on : Download version 3.8.1
Filename : Selenium-Server-Standalone-3.8.1.Jar

Third party Browser Drivers
Click on the link next to Google Chrome Driver → 2.33
Click on chromedriver_win32.zip.

12/12/2017

What are the Components of Selenium or
What are the flavours of Selenium?

ANS

Selenium Core	Used for Web
Selenium IDE	
Selenium RC (Remote Control)	
Selenium WD (Web Driver)	
Selenium Grid	
Selendroid	For Mobile applications
Appium	
Winium	used for windows Applications

What are the languages Supported by Selenium?

ANS
1) Java
2) C# - C Sharp
3) Ruby
4) Python
5) Java Script (Node JS)
6) Pearl
7) PHP
8) Haskell
9) Objective-C
10) Java Script
11) R
12) Dart
13) Tcl
14) Elixir

__/__/____

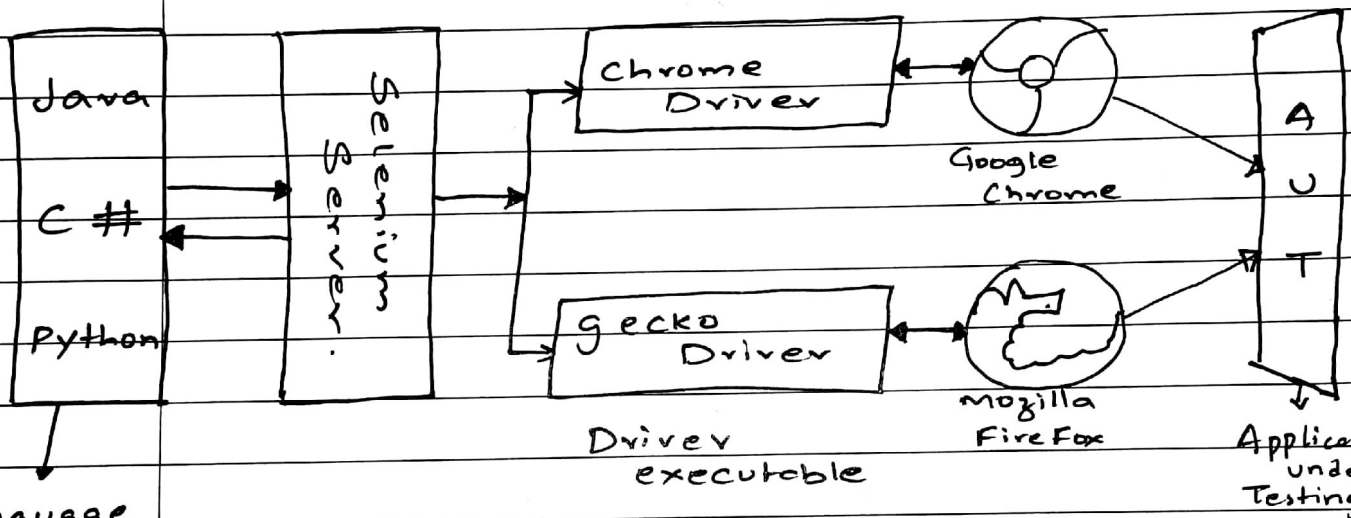
Q) Which operating System (OS) is supported by Selenium?

ANS ⇒ It Supports all the operating system

Q) Which is the operating System (OS) is not Supported by Selenium?

ANS ⇒ UNIX is the OS not supported by Selenium, because it is a charecter based interface

"HIGH LEVEL ARICHITECTURE OF SELENIUM"



Language Binding or Client Binding

* Selenium Supports many coding languages Such as Java, C#, python etc.....
* These longuage Specific selenium Softwares are called as Longuage Bindings or Client Bindings.
* These Client Bindings Communiter with Selenium Server
* Selenium Server performs the action on the browser with the help of browser Speific driver executables Such as Chrome driver for google chrome browser gecko driver for Mozilla five Fox browser etc.....
* Selenium uses a protocol called "JSON" (Java Script Object Notation) Wire protocol.

__/__/____

To install & configure Selenium we need Client Bindings Selenium Server & driver executable. But Selenium Server contains java client bindings also, hence we need only 2 files ie:-

① Selenium Server & ② Driver Executable.

"STEPS TO INSTALL SELENIUM"

"REQUIRED SOFTWARES"

⇒ JDK 1.8 (installed)

⇒ Eclipse ID (installed)

⇒ Selenium Server Standalone .Jar (Download)

⇒ Chromdriver (Download)

Step 1) Unzip Chrome driver file.

Step 2) In Eclipse go to to the required java project which is present undr package explorer.
(Windows → Show view → package Explorer)

Step 3) Right click on the Java project go to new Select Folder. Specify the name as driver & Click Finish.

Step 4) Copy paste chrome driver.exe into this Folder.
(Do not build to path the exe file)

Step 5) Create another folder with the name jar & then Copy paste Selenium Server Standalone .jar into this folder

Step 6) Right click on the Selenium jar file go to build path & select add to build.

Step 7) Create a class with main method & write a Code as shown below & execute.

__/__/____

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

public class Demo
{
Public static void main (String [] args)
{ // Set the path of driver executable
 String key = "Webdriver.chrome.driver";
 String value = "./driver/chromedriver.exe";
 System.setproperty (key, value);
 // opens the chrome browser
 ChromeDriver driver = new ChromeDriver ();
 // closes the browser
 driver.close ();
 }
 }
```

Scanned by CamScanner

14/12/2017

Example program :- Code for FireFox Browser.

```
package qsp;
import org.openqa.selenium.firefox.FirefoxDriver;
public class DemoA
{
Static
{
String key = "webdriver.gecko.driver";
String value = "./driver/geckodriver.exe";
System.SetProperty (key, value);
}
Psvm (String ... args)
{
FireFox Driver driver = new FirefoxDriver();
}
}
```

NOTE :- 1) We should set the path of driver executable before opening the browser

2) To set the path of driver executable we use System.SetProperty

3) We can use System.SetProperty inside Static block so that it gets executed first & then browser is launched.

__/__/____

Steps to See the Source code of Selenium Class :-

* We Should know the FQCN (Fully Qualified Class Name)
  example :- Org.openqa.selenium.FireFox.FireFoxDriver;

1) Go to download page of Selenium
   ie Seleniumhq.org/download
2) Click on Source code link.
3) Click on Java
4) Click on Client
5) Click on Src
6) Click on org/openqa/selenium
7) Click on Firefox
8) Click on FireFoxDriver.java

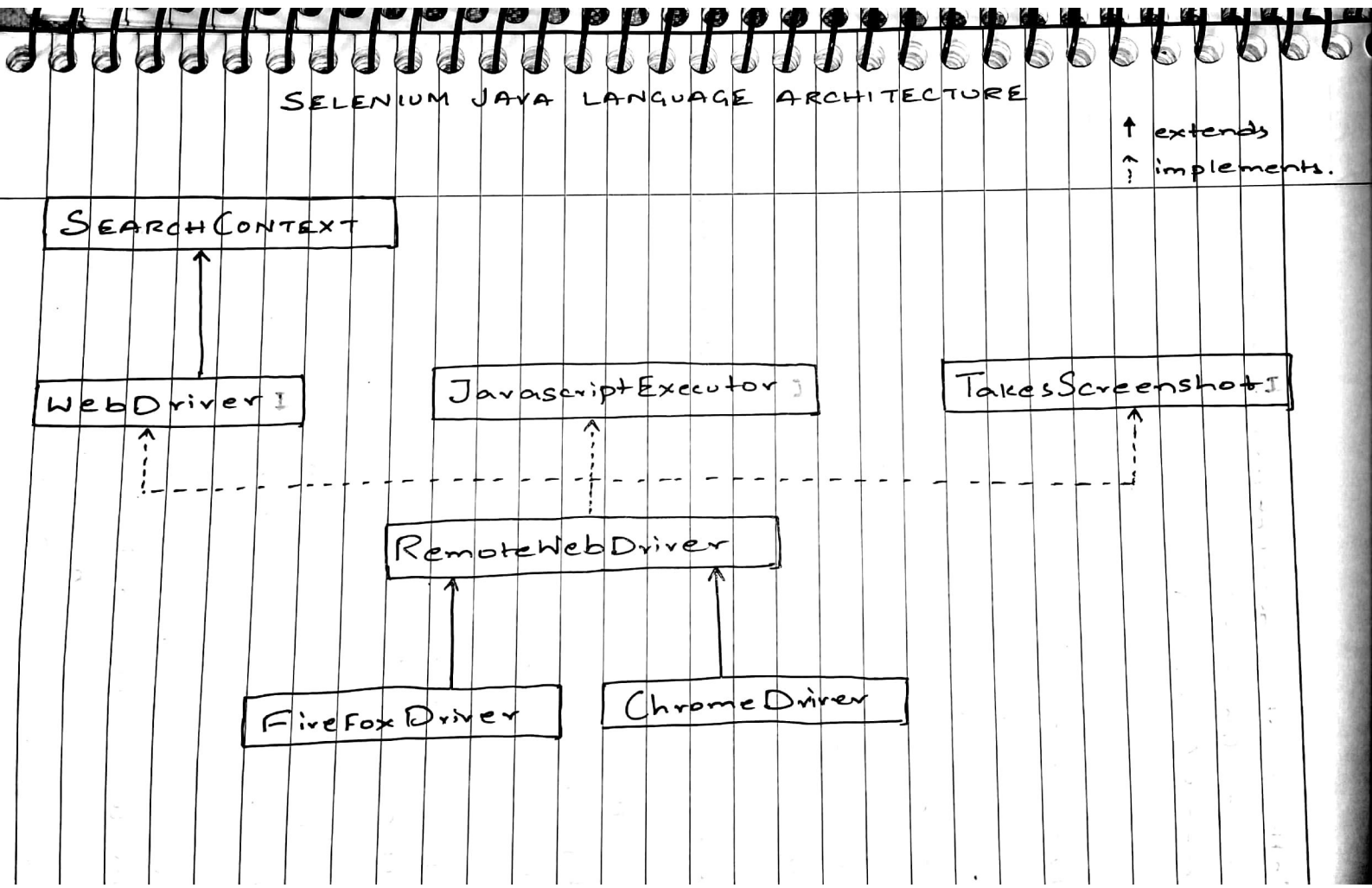
Q) Explain the architecture of Selenium Java Language binding ?

In Selenium for every browser we have respective classes such as ChromeDriver for google chrome browser, FireFox Driver for mozilla FireFox browser, InternetExplorer Driver for IE browser etc.....
All these browser classes extends from a Super class called remote web Driver.
Remote web driver implements multiple Interphases ..(13) such as WebDriver, JavaScriptExecutor & TakesScreenshot etc
WebDriver extends from SearchContext interface.

Q) Hiding Methods of Object class in eclipse?

Ans! Go to Windows → preferences → Java → Appreance → TypeFilters → Click Add → Type! Java.Lang.Object → Click on ok & OK.

SELENIUM JAVA LANGUAGE ARCHITECTURE



__/__/____

Methods of SearchContext interFace

1) Find Element (By arg);
2) Find Elements (By arg);

15/12/2017

Types Of Application

1) Stand Alone Application
   - calculator, Notepad, Ms paint
2) Client Server Application
   - yahoo Messenger, Skype, Outlook etc
3) Web Application
   - Facebook, Gmail, Naukri.com

Different Ways Of Accessing Web Application

1) With internet Connection
   - open the browser & Enter the url
     https://demo.actitime.com.
2) Without Internet Connection
   - open the browser & Enter the below mentioned url
     http://localhost/Login.do

NOTE:- Before using the above url we should have installed actitime application.
It can be downloaded from the website:
https://www.actitime.com/download.php.

3) Without internet & without installation
   - 1) Open the notepad type the following code Save it as login.html in "D" drive
     2) Double click on the html file.
        ie file:///D:/Login.html

__/__/____

WEB ELEMENT

Example Html code :-

<title> QSP </title>

UN : <input type = "text" >

PW : <input type = "password" >

<input type = "button" value = "Login" >

Methods of SearchContext interface

→ findElement(By arg 0) : WebElement.

→ findElements(By arg 0) : List (WebElement)

Methods of JavascriptExecutor

→ etecute AsyncScript C)

→ execute Script+C)

Methods of TakesScreenshot

→ getScreenshotAs (output Type <x> arg 0) : x

Methods of WebDriver

- close C):
- get (string arg 0) : void
- getCurrenturl C) : String
- getpageSource C) : String
- get Title C) : String
- manage C) : options
- navigate C) : Navigation
- quit C) : void
- SwitchTo C) : TargetLocator - WebDriver.
- get WindowHandle C)
- get WindowHandles C)

18/12/2017

Calling static method of abstract class & sending its output as argument for another method.

```
Class A
{
void testA(String s)
{
SOP(s);
}
}

Abstract class B
{
static String testB()
{
return "QSP";
}
}

Main class
{
PSVM(String... args)
{
String V = B.testB();
A a1 = new A();
a1.testA();
}
}
```

(or)

```
A a1 = new A();
a1.testA(B.testB());
```

"WEB ELEMENT"

1) Before performing any action such as typing, clicking, Selecting etc we should first find the web element.
2) Anything present on the web page is called as Webelement
   ex:- Text box, button, Link, image, radio button, check box, table etc
3) These web elements are created using html ("HyPer text mark up language).
4) To see the Source code of element, right click on the element in web page & select "Inspect element" or "inspect" It will display the Source code of selected element.
5) In any html code we can find tag, attribute & text.

__/__/____

Sample html code :-

```
<html>
    <head>
        <title> Qspiders </title>
    </head>
    <body>
        <input type = "text" value = "admin">
    </body>
</html>
```

Example for tag

<html> , <head> , <title> , <body> , <input , a

Example for attribute

type = "text" , Value = "admin" , name = "remember" value = "on" , id = "keep...." , title = "Do...."

Example for Text

Qspiders ; actiTIME

"LOCATOR"

1) Locators are used to search the element in selenium locators are the static method of "By" class

2) In selenium there are 8 type of locators such as

ⓐ tagName ⇒ Superfast

ⓑ id

ⓒ name

ⓓ className

ⓔ LinkText

ⓕ partialLinkText

ⓖ cssSelector

ⓗ xpath ⇒ Superslow .

__/__/____

Sample Code :-

```
<a id="a1" name="n1" class="c1"
href="http://localhost/login.do">QSpiders</a>
```

Selenium code to click on the link present in the Sample Web page

```
public class DemoA
{
Static{
System.setproperty("webdriver.chrome.driver",
                   "./driver/chromedriver.exe");
}
psvm(String ... args)
{
// open the browser
WebDriver driver = new ChromeDriver();
// Enter the url
driver.get("File:///C:/users/QSP/Desktop/Sample.html");
// Find the element with tag 'a'
WebElement element = driver.FindElement(By.tagName("a"));
// Click on that element
element.click();
}
}
```

19/12/2017

Generally we write optimized code as shown below :-

```
driver.findElement(By.tagName("a")).click();
```

Meaning :-

"In the browser find the Element By using tag name 'a' & click on it"

Example for using the locator id :-

```
driver.findElement(By.id("a1")).click();
```

Example for using the locator name :-

```
driver.findElement(By.name("n1")).click();
```

Example for using the locator class Name :-

```
driver.findElement(By.class("c1")).click();
```

NOTE :- In html attribute is class whereas in selenium the locator is By.className();

Example for using LinkText :-

```
driver.findElement(By.linkText("QSpiders")).click();
```

NOTE :- Both link text & partiallinkText can be used only to locate the link. (tag of the Element should be a)
If we try to used it on any other type of element we get : NosuchElementException.

Example for using partialLinkText :-

```
driver.findElement(By.partialLinkText("Spi")).click();
```

NOTE :- partialLinkText is used when the text of the link is partially changing or very Lengthy.

Example :- <a .....> Inbox (3)

```
driver.findElement(By.partialLinkText("Inbox")).click();
```

__/__/____

NOTE :- If the Specified locator is matching with multiple elements then find element method will return the address of 1st matching element

Example Web page :-

```
UN: <input type = "text">
PW: <input type = "password">
```

```
// open the browser
WebDriver driver = new ChromeDriver();
// Enter the URL
driver.get("file:///E:/sample.html");
// Enter 'UserA' in UN text box
driver.findElement(By.tagName("input")).sendKeys("UserA");
// Enter 'QSP' in PW text box ?
driver.findElement(By.tagName("input")).sendKeys("QSP");
```

NOTE :- When we run the above code it will enter the password QSP in the Username textbox itself because to locate the password field we cannot use id, name or className because it is not given by the developer.

We cannot use linkText & partialLinkText because it is not a link.

We can use tagName but it is duplicate with UserName field.

FindElement returns the address of 1st matching element when the locator value is duplicate.

In the above example it returns the address of username text box.

__/__/____

When we cannot use 1st 6 type of locators we can go for cssSelector

20/12/2017

"CSS SELECTOR"

1) CSS Stands for ⇒ Cascading Style Sheets
2) Css Selector is a type of locator, it is a static method present in 'By' class.
3) The Syntax of css Selector is tag [AN = 'AV']
   ie tag [Attribute Name = 'Attribute Value']

Example :-

input [type = 'text'] → UN
input [type = 'password'] → PW

Using CSS Selector in Selenium

```
driver.FindElement(By.cssSelector("input[type='text']")).SendKeys("User A");
driver.FindElement(By.cssSelector("input[type='password']")).SendKeys("QSP");
```

NOTE :- If there is any mistake in the Syntax of cssSelector then we get InvalidSelectorException.

Steps to Check the cssSelector in the browser :-

1) Open the required page in chrome browser
2) Right click anywhere on the page & select inspect (F12)
3) press ctrl + F
4) Type the cssExpression ie ⇒ <input[type='text']>

__/__/____

Steps to check the cssSelector in FireFox :-

1) Open the required page in firefox browser
2) Select inspect element by right clicking on the page
3) press ctrl + F
4) Type css Expression & press Enter

IMPORTANT NOTE :-

In cssSelector # represents id , . represents class.

Examples For cssExpressions :-

Sample.html

```
<a id="a1" name="n1" class="c1"
href="http://localhost/login.do">QSpiders</a>
```

Syntax

```
tag [AN='AV']
```

Ex :-

1) a[id='a1'] or a#a1
2) a[name='n1']
3) a[class='c1'] or a.c1
4) a[href='http://localhost/login.do']

Css Selector DO NOT support text :(

__/__/____

"xPath"

It is the path of the element in html tree.

Example Sample Web page :-

```
<html>
  <body>
    FN: <input type = "text">
    LN: <input type = "text">
  </body>
</html>
```

NOTE :- In xpath . present at the beginning represents current web page

/ represents child Element.

It is not mandatory to use . in xpath.

Example :-

```
driver.FindElement(By.xpath(".//html/body/input"
    )).SendKeys("bhanu");
driver.FindElement(By.xpath("//html/body/input[2]
    ")).sendKeys("prakash");
```

A D ⇒ /html/body/div[1]/input | /html/body/div[2]/input[2]

21/12/2017

Open Laptop Create Demo X1.html on Desktop with following Code

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

Write xpath for elements

xpath
/html/body/div/input
/html/body/div[1]/input

Matching Elements :-

A B C D	⇒
A B	⇒
A	⇒ /html/body/div[1]/input[1]
B	⇒ /html/body/div[1]/input[2]
C D	⇒ /html/body/div[2]/input
C	⇒ /html/body/div[2]/input[1]
D	⇒ /html/body/div[2]/input[2]
A C	⇒ /html/body/div/input[1]
B D	⇒ /html/body/div/input[2]
A D	⇒ ~~/html/body/div[1]/input[2]~~
B C	

__/__/____

"ABSOLUTE XPATH"

1) Writing the Complete path of the element (from html till the element) is called an Absolute xpath.
2) In xpath we can use index which starts from 1
3) Index will become 2 if there is on another element under the same parent with the same tag. (sibling)

NOTE:- To join multible xpath expressions we use pipline Symbol.

To check xpath expression in firefox browser we should install addons.

Example:- In firefox go to tools, addons, extensions. Search for try xpath, Click install button of try xpath Click add Click ok & restart the ff browser. After restarting click on try xpath icon (TX) write the expression press enter. It will highlight the matching elements.

"RELATIVE XPATH"

Using absolute xpath on realtime application will be very difficult because absolute path will be very lengthy In order to reduce the length of xpath expression we use relative xpath.

In relative xpath we use descendent. the shortcut is "//"

Relative xpath	Matching Elements
//input	A B C D
//div[1]/input	A B
//div[1]/input[1]	A
//div[1]/input[2]	B
//div[2]/input	C D

__/__/____

Relative x path	Matching Elements
//div[2]/input[1]	C
//div[2]/input[2]	D
[redacted]//input[1]	A C
//div/input[2]	B D
//div[1]/input[1]|//div[2]/input[2]	A B
//div[2]/input[2]|//div[2]/input[1]	B C
	A B C

/ represents childelement

// represents descendent element

Write a xpath which matches with all the images present on the application?

Ans //img

Write a xpath which matches with all the links present on the webpage?

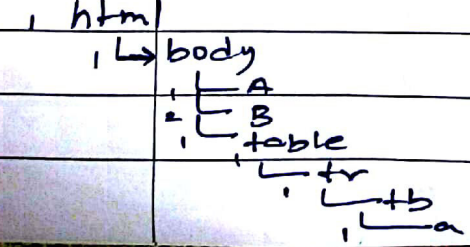
Ans //a

Write a xpath which matches with all the links & images present on the webpage?

Ans //a | //img

What is the difference between //a & //table //a?

//a matches with all the links whereas //table //a matches with all the links which are present in all the table



__/__/____

What is the difference between //img & //div //img ?

Ans //img ⇒ all the images

//div //img ⇒ all the images in division.

What is the difference between //a & //a[1] ?

Ans //a ⇒ all the links

//a[1] ⇒ Matches with all the 1st link in current page

What are the types of xpath available ?

Ans In xpath there are only 2 types :-

absolute xpath & relative xpath.

22/12/2017

Xpath by Attribute :-

While writing xpath expression we can include attribute instead of index. Because if the position of the element changes index also changes.

To Specify the attribute we should use following Syntax

⇒ tag [@AttributeName = 'AttributeValue']

Example For textbox 'B'

// input [@value='B']

Q) Write xpath in browser & check how many matches & mention what are the matches.

html code :-

```
A1 <input type = "text" Value = "A">
B1 <input type = "text" Value = "B">
A2 <input type = "text" Value = "A">
B2 <input type = "text" Value = "B">
C1 <input type = "Checkbox" checked>
```

Xpath	No	Matching Elements
// input	5	A1 B1 A2 B2 C1
// input[@type='text']	2	A1 B1
// input[@type='button']	2	A2 B2
// input[@ ='checkbox']	1	C1
// input[@value='A']	2	A1 A2
// input[@Value='B']	2	B1 B2
// input[@type='text' and @value='A']	1	A1
// input[@type='text' or @value='A']	3	A1 B1 A2
// input[not(@type='checkbox')]	4	A1 B1 A2 B2

__/__/____

Q) How do you include multiple attribute in the Single xpath expression.

Ans By using and & or.

Example For xpath by Attribute for the elements which are present in Login page of actitime.

Elements	xpath
Clock img	//img[@height='98']
plz identify	//td[@id='headerContainer']
UN	//input[@id='username']
PW	//input[@name='pwd']
CheckBox	//input[@name='remember']
View Licence	//a[@id='licenseLink']

Example for xpath by text :-

If there is no attribute or if the attribute is matching with multiple elements in such cases we can identify the element using its text.

The syntax is :- tag[text()='textvalue']

Example :-

Elements	xpath
plz identify	//td[text()='please identify yourself']
Login	//div[text()='Login']
View License	//a[text()='View License']
actiTIME inc	//a[text()='actiTIME Inc.']

xpath By Contains

If attribute value or text value is partially changing then we can identify using contains function.

There are 2 Syntax :-

① tag[contains(@AN,'PAV')]

__/__/____

(2) tag [contains (text(), 'PTV')]

Example :-

Elements	x path
clock	//img [contains (@src,'timer')]
acHTIME	//img [contains (@src,'Logo')]
actiTIME 2017.4	//nobr [contains (text(),'actiTIME')]

How do you handle dynamically without using partial link text?

Ans Using Contains() Function of xpath

Example :- <a> Inbox (18)

//a [contains (text(), 'Inbox')]

Assignment :-

Q) Write xpath for element present in login page of gmail?

27/12/2017

"XPath AXES".

1) Navigating from one element to another element is called as traversing.

2) In order to navigate we use xpath axes

3) Following are the important xpath axes :-

1. child → /
2. descendent → //
3. parent → /..
4. ancestor
5. Following-sibling
6. preceding-sibling

Syntax :-

/axes::tag[Option]

NOTE :- Option can be index, attribute, text, Contains etc....

Example program

```
<html>
  <body>
  <div>
   <span>
      <select size=5>
    <option> Idly </option>
    <option> Vada </option>
    <option> Poori </option>
    <option> Dosa </option>
    <option> pongal </option>
    </select>
</span>
</div>
</body>
</html>
```

__/__/____

html → /child::html/child::body (or) /html/body

↳ body

↳ div ← /descendant::div (or) //div

↳ Span ← //Select/parent::span (or) //Select/..

↳ Select

//Select/ancestor::div

- option Idly
- option Vada
- option poori
- option Dosa
- option pongal

Following sibling :-

//option[text()='poori']/following-sibling::option

//option[text()='poori']/following-sibling::option[1]

//option[text()='poori']/following-sibling::option[2]

preceding sibling :-

//option[text()='poori']/preceding-sibling::option

//option[text()='poori']/preceding-sibling::option[1]

//option[text()='poori']/preceding-sibling::option[2]

↳ Select

- option Idly ← //option[.='poori']/p-s::option
- option Vada ←
- option poori ← //option[.='poori']/f-s::option.
- option Dosa ←
- option pongal