Table of Contents

[Explain about your roles and responsibilities 3](#_Toc74548182)

[What are the different ways of identifying objects in selenium? 4](#_Toc74548183)

[What are the different waits supported by selenium 4](#_Toc74548184)

[Different Types of Navigation commands 4](#_Toc74548185)

[Difference between driver.close() and driver.quit()? 5](#_Toc74548186)

[How to scroll down a page using JavaScript? 5](#_Toc74548187)

[Handle Mouse Actions 5](#_Toc74548188)

[Handle Keyboard Actions 6](#_Toc74548189)

[Handle WebTables 6](#_Toc74548190)

[Taking Screenshot 6](#_Toc74548191)

[Difference between findElement and findElements 6](#_Toc74548192)

[Difference between Relative and Absolute Xpath 6](#_Toc74548193)

[How to move from one element to another in selenium 7](#_Toc74548194)

[How to handle webelemets with dynamic behavior? 7](#_Toc74548195)

[How to handle the dropdowns in selenium 7](#_Toc74548196)

[Different types of assertions in TestNG 8](#_Toc74548197)

[Extracting the list of webElements 8](#_Toc74548198)

[isDisplayed, IsEnabled, IsSelected 8](#_Toc74548199)

[Switching between Frames using SwitchTo() 9](#_Toc74548200)

[Switching between Windows/Tabs using selenium 9](#_Toc74548201)

[JavaScript Alerts 9](#_Toc74548202)

[Handle Jquery Sliders 10](#_Toc74548203)

[Handle Jquery Calenders 10](#_Toc74548204)

[Using AShot API to take Element and Page screenshot 10](#_Toc74548205)

[Different types of Exceptions in Selenium 10](#_Toc74548206)

[How to handle the exceptions? 10](#_Toc74548207)

[TestNG annotations 11](#_Toc74548208)

[TestNG Execution order 11](#_Toc74548209)

[Explain TestNG.xml syntax 11](#_Toc74548210)

[Importance of Data Providers in TestNG 12](#_Toc74548211)

[Importance of Parameters in TestNG 12](#_Toc74548212)

[Importance of Groups in TestNG 12](#_Toc74548213)

[How to run parallel execution with TestNG 13](#_Toc74548214)

[Importance of Listeners in TestNG 14](#_Toc74548215)

[How to handle failure e test cases in your framework 14](#_Toc74548216)

[What are the different Maven commands 14](#_Toc74548217)

[How to run the test using Maven 15](#_Toc74548218)

[Importance of Maven in building project 15](#_Toc74548219)

[What are the different plugins you used in Maven and usage of it? 15](#_Toc74548220)

[What are the OOPS concepts? 15](#_Toc74548221)

[What are the different collections you used? 15](#_Toc74548222)

[What is Map? What are the different classes in Map Interface? 15](#_Toc74548223)

[Difference between Abstraction and Interface? 15](#_Toc74548224)

[What is Inheritence? How do you handle it? Where it is used? 16](#_Toc74548225)

[Example of Interface you used in Selenium? 17](#_Toc74548226)

[Different types of inheritance in Java 17](#_Toc74548227)

[Importance of static keyword in Java, where do you use it ? 18](#_Toc74548228)

[What are the different methods available in HashMap? 18](#_Toc74548229)

[What is the difference between FileInputStream and BufferedReader 18](#_Toc74548230)

[How do you read data from Excel files? Define the syntax 18](#_Toc74548231)

[How to extract records from SQL DB? Explain the logic? 18](#_Toc74548232)

[What are the different String methods you used? Explain few of them? 18](#_Toc74548233)

[What is the difference between Super and This keyword 18](#_Toc74548234)

[Why should Selenium be selected as a testing tool for web applications or systems? 19](#_Toc74548235)

[What are the disadvantages of using Selenium as a testing tool? 19](#_Toc74548236)

[What is meant by Selenium Suite and what are its different components? 19](#_Toc74548237)

[What is meant by Selenese? Explain different types of Selenium commands 20](#_Toc74548238)

[State the major difference between “assert” and “verify” commands in Selenium. 20](#_Toc74548239)

[What is meant by an exception test in Selenium? 20](#_Toc74548240)

[What is meant by XPath in Selenium. Explain XPath Absolute and XPath Relative. 20](#_Toc74548241)

[Can selenium be used to launch web browsers? 20](#_Toc74548242)

[With the help of code snippets, explain how we can create right-click and mouse hover actions in Selenium. 21](#_Toc74548243)

[Can we handle a windows-based pop-up in Selenium, and if not, then what are the alternatives? 21](#_Toc74548244)

[What is Object Repository ? 21](#_Toc74548245)

[Explain how you can find broken images in a page using Selenium Web driver ? 21](#_Toc74548246)

[Explain how you can handle colors in web driver? 21](#_Toc74548247)

[*Using web driver how you can store a value which is text box?* 22](#_Toc74548248)

[Explain how you can switch between frames? 22](#_Toc74548249)

[How can you find if an element is displayed on the screen? 22](#_Toc74548250)

[What type of test cases not to be automated? 22](#_Toc74548251)

[What are the types of WebDriver APIs available in Selenium? 23](#_Toc74548252)

[Which WebDriver implementation claims to be the fastest? 23](#_Toc74548253)

[What is the difference between “/” and “//” 23](#_Toc74548254)

[What is the difference between setSpeed () and sleep () methods? 23](#_Toc74548255)

[We do create a reference variable ‘driver’ of type WebDriver as shown below. What is the purpose of doing this way? 23](#_Toc74548256)

[How to handle hidden elements in Selenium WebDriver? 24](#_Toc74548257)

[How do you read test data from excels? 24](#_Toc74548258)

[Is it possible to automate the captcha using Selenium? 24](#_Toc74548259)

[How can you use the Recovery Scenario in Selenium WebDriver? 24](#_Toc74548260)

[How to connect a Database in selenium? 24](#_Toc74548261)

[What is desired capabilities? 25](#_Toc74548262)

# Explain about your roles and responsibilities

Hi. My Name is XXXXX . I am having 8+ years of work experience in testing. I worked on Manual and Automation testing during these years. Started with a manual tester role, I moved to Automation testing using Selenium. We used Maven for the Project management and TestNG for the test case management. Our framework is a BDD framework using cucumber. We create different features files based on the test scenarios and write step definitions for them. Using the extent reports, we are generating the reports. We are using Bitbucket for the source code management and Source Tree for doing the Git Operations. Good knowledge on Agile methodology and ceremonies involved. We used ALM for the test case management and JIRA for tracking issues, stories, efforts etc. I have experience in developing the framework from scratch using BDD or TDD. We are using Jenkins for creating the jobs for running our scripts and also there are around 300 sanity test cases which are scheduled to run every day during the night time and reports the defect if any of the functionality is broken. I am good team player who looks over their deliverables and give my helping hand to resolve their blockers.

Coming to current project, It is XXXXXXXXX, I joined this organization on XXXXX .

XXXXXXXXXXXX

Here the client developed a framework on top of selenium using their own wrapper classes. It’s a kind of Data Driven Framework where the test data is maintained in the separate excel sheets and provide it to test case. Here the page objects are storing the properties file and read them during the run time which helps in creating the dynamic xpath’s using the string manipulation techniques. Created separate TestNG.xml for the each module and stored in resources folder under suites. We are maintaining different packages like ComponenetHelpers, Utils, Base, TestScripts, Constants, UserDefinedExceptions, Runner packages in our current project.

# What are the different ways of identifying objects in selenium?

1. ID
2. Xpath
3. ClassName
4. Name
5. CSSSelector
6. LinkText
7. PartialLinkText
8. TagName

Most commonly, we used xpath and ID for the object identification

# What are the different waits supported by selenium

1. **Implicit wait** - Implicit wait commands Selenium to wait for a certain amount of time before throwing a “No such element” exception.

Example (if asked)

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

1. **Explicit wait** - Explicit wait is used to tell the Web Driver to wait for certain conditions before throwing an "ElementNotVisibleException" exception.

Example(if asked)

WebDriverWait wait = new WebDriverWait(driver, 30);

1. **Fluent wait** - It is used to tell the web driver to wait for a condition, as well as the frequency with which we want to check the condition before throwing an "ElementNotVisibleException" exception.

Example(if asked)

Wait wait = new FluentWait(WebDriver reference).withTimeout(timeout, SECONDS).pollingEvery(timeout, SECONDS).ignoring(Exception.class);

# Different Types of Navigation commands

We have 4 different navigation commands like,

1. driver.navigate().to("https://www.ebay.in/"); - Navigates to the provided URL
2. driver.navigate().refresh(); - This method refreshes the current page
3. driver.navigate().forward(); - This method does the same operation as clicking on the Forward Button of any browser. It neither accepts nor returns anything.
4. driver.navigate().back(); - This method does the same operation as clicking on the Back Button of any browser. It neither accepts nor returns anything.

# Difference between driver.close() and driver.quit()?

1. driver.close()

This command closes the browser’s current window. If multiple windows are open, the current window of focus will be closed.

1. driver.quit()

When quit() is called on the driver instance and there are one or more browser windows open, it closes all the open browser windows.

# How to scroll down a page using JavaScript?

To scrolling a webpage horizontally or vertically, we use Java Script methods available in Document.

window.ScrollTo(X axis coordinates like 200, Y axis coordinates 1000)

window.ScrollTo(0, 1000)

window.ScrollTo(0, -1000) : to take up back to top

**Syntax**:

  JavascriptExecutor js = (JavascriptExecutor) driver

 js.executeScript("window.scrollBy(0,1000)");

# Handle Mouse Actions

In order to deal with mouse actions, we have to use Actions class

Actions act= new Actions(driver);

Different mouse actions available are:

1. ContextClick() : It is used for performing right click
2. DragAndDrop(src,target) : It is used to drag from src and drop it to target positions.
3. MoveToElement(src) : It is used to hover on any WebElement
4. ClickAndHold() : Its going to press the left mouse button and hold it until we send the release action.
5. Release() : Its used to release the left mouse button that was pressed by the ClickAndHold() method

For all the above actions, we need to use Build() to build the actions followed by Perform() to perform those actions.

# Handle Keyboard Actions

In order to deal with Keyboard actions, we have to use Actions class

Actions act= new Actions(driver);

Different keyboard actions available are:

1. KeyDown() : It is used to press any key on the keyboard
2. KeyUp() : It is called to release the key that was pressed earlier.
3. sendKeys() : It is used to send any string along with Keys actions.

Example:

//To Press ctrl + t

//act.KeyDown(Keys.LeftControl)

// .SendKeys("t")

// .KeyUp(Keys.LeftControl)

// .Build()

// .Perform();

# Handle WebTables

Using the Table, TR, TD tags we can handle the table elements.

Example: **//table[@class='dataTable']/tbody/tr/td[4]**

We can do indexing if the table rows are fixed like tr[2] and td[3]. If it’s a dynamic table, read the table and store in a list and using for loop, iterate through the values in the table and do the validations.

# Taking Screenshot

TakeScreenshot interface can be used to take screenshots in WebDriver.

getScreenshotAs() method can be used to save the screenshot

Syntax: File scrFile = ((TakeScreenshot)driver).getScreenshotAs(outputType.FILE);

# Difference between findElement and findElements

* **findElement:** A command used to uniquely identify a web element within the web page.

**Example:** driver.findElement(By.id(“id of the webelement”));

* **findElements**: A command used to identify a list of web elements within the web page.

Example:

List<WebElement>elementsList= driver.findElements(By.id(“id of the webelement”));

SYSO(elementsList.size());

# Difference between Relative and Absolute Xpath

Single backslash: We define it for absolute xpath. When we are writing xpaths, if you want to start our xpath from the root level i.e., from html tag. We will use single backslash tag

Example: /html/head/body/div[2]/input/…..

Double backslash: It is used for relative xpath. The relative xpath starts by referring to the element that we want to identify and not from the root node

Example: //input[@id=’testexampeid’]

# How to move from one element to another in selenium

We have the axes concept in xpath to move from one element to another. When there is no unique identifier present for an element, using the parents reference we can move to child. Similar fashion, when there is no unique identifier present for the parent, we can child reference to move back to parent object.

**Examples:**

Following-sibling

Following

Parent

Child

Preceding-sibling

Ancestor

Descendent

# How to handle webelemets with dynamic behavior?

We have the concept in xpath to handle dynamic change of WebElements by using

1. Contains
2. Starts-with
3. Ends-with

**Examples:**

1. driver.findElement(By.xpath(“//div[contains(text(),’randomn text’)]
2. driver.findElement(By.xpath(“//div[contains(@id,’randomn text’)]
3. driver.findElement(By.xpath(“//div[contains(@class,’randomn text’)]
4. driver.findElement(By.xpath(“//div[starts-with(text(),’randomn text’)]
5. driver.findElement(By.xpath(“//div[ends-with(text(),’randomn text’)]

# How to handle the dropdowns in selenium

We have the select class to perform drop down operation if the tag starts with Select in the html DOM.

**Syntax**:

Select sel = new Select(driver.findElement(By.xpath(“xpath of the element”));

Sel.selectByIndex(provides the index in integer);

Sel.SelectByValue(“get the value from the DOM”);

Sel.SelectByVisibleText(“get the visible text from the DOM”);

# Different types of assertions in TestNG

Hard Assert: Execution will stop and testcase will fail when the assertion is failed.

Soft Assert: Execution will continue even though the assertion is failed. But finally in the report, it shows as failed.

* assertEquals
* assertNotEquals
* assertTrue
* assertFalse
* assertNull
* assertNotNull

Example:

Assert assert= new Assert();

String actual= driver.findElement(By.id(“id of the text”)).getText();

assert.assertEquals(actual,”Hi Mahesh’’);

Assert.assertTrue(condition);

SoftAssert assert = new SoftAssert();

2. assertTrue(driver.findElement(By.id(“id of the checkbox”)).isSelected())

# Extracting the list of webElements

We have the findElements method in selenium to extract the list of elements which are having unique behavior. For Example, in order to get the elements which contains the tagname as input, the example is given below

Driver.findElements(By.tagName(“input”));

Driver.findElements(By.tagName(“a”));

# isDisplayed, IsEnabled, IsSelected

isDisplayed() : The isDisplayed method in Selenium verifies if a certain element is present and displayed. If the element is displayed, then the value returned is true. If not, then the value returned is false.

boolean eleSelected= driver.findElement(By.xpath("xpath")).isDisplayed();

IsEnabled() : This method verifies if an element is enabled. If the element is enabled, it returns a true value. If not, it returns a false value.

boolean eleEnabled= driver.findElement(By.xpath("xpath")).isEnabled();

IsSelected(): The isSelected() method checks that if an element is selected on the web page or not. It returns a boolean value (true) if selected, else false for deselected. It can be executed only on a radio button, checkbox, and so on.

boolean eleSelected =driver.findElement(By.locatorType(“path”)).isSelected();

# Switching between Frames using SwitchTo()

When the webpage is splitted into multiple frames, inorder to perform operation on the webelement present in the frame, we need to switch to frame.

* **By Index**
* **By Name or Id**
* **By Web Element**

Driver.switchTo().frame(1) // 1 – index of the frame when there are 2 frames, index starts with 0

Driver.switchTo().frame(“a077aa5e”) // id of the frame

Driver.switchTo().frame(“name of the frame”) // name of the frame

Driver.switchTo().frame(driver.findElement(By.id(“id of the element”));

# Switching between Windows/Tabs using selenium

When there are multiple windows opened, we need to use SwitchTo() command for moving between windows/tabs.

Driver.getWindowHandles() method returns me list of windows opened. After that, we need to use switchTo command to move to our desired window.

Driver.switchTo().window(0 or 1 ) -> index of the window starts with 0.

Driver.switchTo().window(“windowName” )

# JavaScript Alerts

We are having different methods available for handling the alerts.

1) void dismiss() // To click on the 'Cancel' button of the alert.

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

2) void accept() // To click on the 'OK' button of the alert.

driver.switchTo().alert().accept();

3) String getText() // To capture the alert message.

driver.switchTo().alert().getText();

4) void sendKeys(String stringToSend) // To send some data to alert box.

driver.switchTo().alert().sendKeys("Text");

# Handle Jquery Sliders

In order to deal with Keyboard actions, we have to use Actions class

Actions act= new Actions(driver);

WebElement ele = object of the slider;

Act.dragAndDropBy(ele,X offset, Y Offset).build().perform();

# Handle Jquery Calenders

# Using AShot API to take Element and Page screenshot

# Different types of Exceptions in Selenium

1. StaleElementReferenceException

When the element that you are trying to perform an operation on is no longer present on the page or DOM has been refreshed then this exception is thrown.

1. NoSuchElementException

The reason is your element is no accessible to your code. This occurs because you have given an incorrect locator, element is inside the frame or the element is not loaded

1. ElementNotVisibleException:

This exception occurs when an element is present in the DOM but it is not visible on the web page.

1. NoAlertPresentException

This occurs when you are trying to accept, dismiss an alert or read text from an alert that is not yet displayed.

# How to handle the exceptions?

There are instances in Selenium when you intend your code to do something, but it does not do as expected. They are called exceptions.

When an exception occurs, your program flow disrupts, and it can halt your execution. A programmer should always handle checked exceptions. To handle exceptions, you should understand what this exception is and how to handle/resolve it.

**Exception Handling:**

Using the try catch blocks for unknown exceptions or Throws keyword if the exception known to us.

try{

public void testMethod(){

//desired code to perform

}

}catch(Exception e){

//Logic to implement the failure mechanism

}

# TestNG annotations

@BeforeSuite

@BeforeTest

@BeforeClass

@BeforeMethod

@Test

@AfterMethod

@AfterClass

@AfterTest

@AfterSuite

@Parameters

@DataProvider

# TestNG Execution order

When there are two test methods in a package, the execution order is like below.

BeforeSuite -> Before Test -> BeforeClass -> BeforeMethod ->Test 1 ->AfterMethod -> Before Method -> Test 2 -> After Method -> After Class -> After Test -> After Suite

# Explain TestNG.xml syntax

<Suite>

<Test name=”sample test” />

<Classes>

<Class Name=”com.api.name.testclass”/>

</Classes>

</Test>

</Suite>

# Importance of Data Providers in TestNG

The DataProviders in TestNG are another way to pass the parameters in the test function, the other one being TestNG parameters

@DataProvider (name = “name\_of\_dataprovider”)

public Object[][] dpMethod() {

return new Object [][] { values}

}

@Test (dataProvider = "dpMethod ")

public void myTest (int a, int b, int result) {

int sum = a + b;

Assert.assertEquals(result, sum);

}

# Importance of Parameters in TestNG

<suite name=”test suite” >

<parameter name=”browser” value=”chrome” />

<parameter name=”searchKey” value=”India” />

<test name=”testingWebSite”>

<classes>

<class name=”com.cucumber.test.sampletc1” />

<class name

</classes>

</test>

</suite>

In the test case, inorder to use the parameters

@Test

@Parameters(“browser”, “searchKey”)

Public void testSeach(String authorKey, String SearchKey){

}

# Importance of Groups in TestNG

TestNG Groups allow you to perform groupings of different test methods. Grouping of test methods is required when you want to access the test methods of different classes.

Example

@Test(groups= {"SmokeTest"})

public void WebLoginPersonalLoan()

{

System.out.println("Web Login Personal Loan");

}

TestNG.xml notation

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="test\_suite">

<groups>

<run>

<include name="SmokeTest"/>

</run>

</groups>

<test name="Personal Loan">

<classes>

<class name="com.javatpoint.Personal\_loan"/>

</classes>

</test> <!-- Test -->

# How to run parallel execution with TestNG

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="TestSuite" **thread-count="3" parallel="methods"** >

<test name="testGuru">

<classes>

<class name="TestGuru99MultipleSession">

</class>

</classes>

</test>

</suite>

OR

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >

<suite name = "Parallel Testing Suite">

   <test name = "Parallel Tests" **parallel = "classes" thread-count = "2**">

      <classes>

         <class name = "ChromeTest" />

         <class name = "FirefoxTest" />

      </classes>

   </test>

</suite>

OR

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >

<suite name = "Parallel Testing Suite" **parallel = "tests" thread-count = "2">**

<test name = "Parallel Tests1">

<classes>

<class name = "ChromeTest" />

</classes>

</test>

<test name = "Parallel Tests2">

<classes>

<class name = "ChromeTest" />

</classes>

</test>

</suite>

# Importance of Listeners in TestNG

TestNG Listeners in Selenium WebDriver are modules that listens to certain events and keep track of test execution while performing some action at every stage of test execution.

Listeners can be implemented at suite level and class level.

The important listeners we generally use in the daily work are

1. ITestListener

Its having list of implemented methods which perform based on the test execution.

Examples:

onStart(), onFinish(), onTestStart(),onTestSkipped(), onTestSuccess(), onTestFailure()

We can call the implemented class in the testscript class as

@Listeners(TestNgListeners.ListenersBlog.class)

1. IAnnotationTransformer

It contains the retry logic to run the failure test cases based on the retry count we specify.

1. ISuiteListener

It is implemented at a suite level

**onStart:** This method is invoked prior the test suite execution.

**onFinish:** This method is invoked post the test suite execution.

# How to handle failure e test cases in your framework

We can handle the failure testcases multiple ways.

1. Using IRetryAnalyzer and IAnnotationTransformer, We can specify the number of times the failure testcases to run.
2. By going through TestNG-Failed.xml and run through it. So only the failure test cases will be triggered.

# What are the different Maven commands

1. mvn Clean
2. mvn Build
3. mvn Install
4. mvn Test
5. mvn clean install
6. mvn compile

# How to run the test using Maven

Mvn test **or** mvn clean test **or** mvn install or mvn clean install

# Importance of Maven in building project

Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. It simplifies the build process like ANT. ... maven make the day-to-day work easier and generally help with the comprehension of any Java-based project.

# What are the different plugins you used in Maven and usage of it?

1. We generally uses Maven Compiler and Maven surefire plugins

# What are the OOPS concepts?

1. Inheritence
2. Abstraction
3. Polymorphism
4. Encapsulation

# What are the different collections you used?

Collections are used to store and manipulate the group of objects.

Mainly we used

1. ArrayList
2. LinkedList
3. HashSet

# What is Map? What are the different classes in Map Interface?

A map contains values on the basis of key, i.e. key and value pair. Each key and value pair is known as an entry. A Map contains unique keys

A Map is useful if you have to search, update or delete elements on the basis of a key

1. HashMap
2. LinkedHashMap
3. TreeMap

A Map doesn't allow duplicate keys, but you can have duplicate values. HashMap and LinkedHashMap allow null keys and values, but TreeMap doesn't allow any null key or value.

# Difference between Abstraction and Interface?

1. Abstract class can **have abstract and non-abstract** methods and Interface can have **only abstract** methods.
2. Abstract class **doesn't support multiple inheritance** and interface **supports multiple inheritance**
3. Abstract class **can have final, non-final, static and non-static variables**. And Interface has **only static and final variables**.
4. The **abstract keyword** is used to declare abstract class. The **interface keyword** is used to declare interface.
5. An **abstract class** can be extended using keyword "extends". An **interface** can be implemented using keyword "implements".

Example: Abstract Class

public abstract class Shape{

int a=10;

abstract string str=”Mahesh”;  
public abstract void draw();

public void display();  
}

Child Class

Public rectangle extends Shape{

Public void draw(){

SYSO(“Iam in rectangle class”);

}

}

**Example: Interface**

public interface Drawable{

public abstract void draw1();

public abstract void draw2();

}

public class implmentingClass extends Shape Implements Drawable{

public abstract void draw();

public abstract void draw1();

public abstract void draw2();

}

# What is Inheritence? How do you handle it? Where it is used?

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.

The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).

**extends** is the keyword used to inherit the properties of a class

**Usage**:

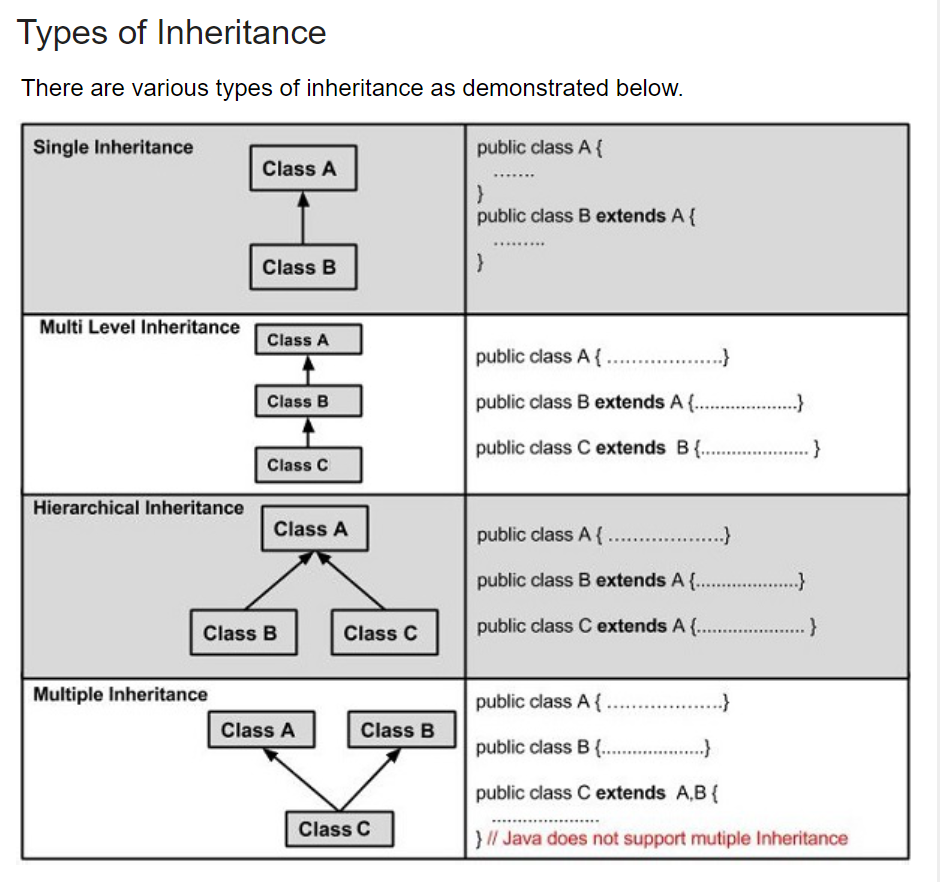
We used the parent child relationship for the driver passing between our test methods/classes. The driver is initialized in the base class and stored in the configuration class as a static variable. Using the extends keyword the driver moves from one class to another. Even in the page object mechanism, the driver is initialized in the pagebase class and the same driver is sent to the child pages which inherits the properties of page base class.

# Example of Interface you used in Selenium?

For example WebDriver is an interface. We can define the WebDriver for chrome, firefox, internetexplorer, safari, phantomjs browsers.

1. In the Listeners concept, we are utilizing the interface concept to define all the methods implementing by the listener. Example: ITestListener contains list of abstract methds. We define the body of the methods in the inherited class.

# Different types of inheritance in Java



# Importance of static keyword in Java, where do you use it ?

The static can be:

1. Variable (also known as a class variable)
2. Method (also known as a class method)
3. Block
4. Nested class

When a member is declared static, then the member is shared by all the instances of a class without making copies of per instance.

it can be accessed without using an object.

* The static data members of the class are accessible to the static method. The static method can even change the values of the static data member.
* A static method cannot refer to non-static data members or variables and cannot call non-static methods too.

# What are the different methods available in HashMap?

We can define the HashMap in multiple ways

 HashMap<Integer, String> hm1 = new HashMap<>();

 HashMap<Integer, String> hm1 = new HashMap<>(10); // 10 is the capacity

HashMap<Integer, String> hm1  = new HashMap<>(5, 0.75f);

Adding to the Map:

 hm1.put(1, "Geeks");

        hm1.put(2, "For");

        hm1.put(3, "Geeks");

Changing the value using the Key

hm.put(2, "For");

Removing the value using key

 hm.remove(2);

Traversing in the Map

for (Map.Entry<String, Integer> e : map.entrySet())

            System.out.println("Key: " + e.getKey()

                               + " Value: " + e.getValue());

    }

containsKey(Object key) Returns true if this map contains a mapping for the specified key.

entrySet() Returns a Set view of the mappings contained in this map.

clear() Removes all of the mappings from this map.

# What is the difference between FileInputStream and BufferedReader

# How do you read data from Excel files? Define the syntax

Using the Apache POI jars, we are getting the input data stores in xls or xlsx files.

The syntax:

File file= new File(“provide the location of your xlsx file”);

FileInputStream FIS = new FileInputStream(file);

//POI methods

WorkBook WB= new XSSFWorkBook(FIS);

XSSFSheet sh= WB.getSheet(“sheetName” or Index);

XSSFRow row=sh.getRow(index);

XSSFColumn column= row.getColoumn(columnIndex);

String CellValue= row.column.getStringCellValue();

# How to extract records from SQL DB? Explain the logic?

# What are the different String methods you used? Explain few of them?

# What is the difference between Super and This keyword

* Super is used to **differentiate the members** of superclass from the members of subclass, if they have same names.
* It is used to **invoke the superclass** constructor from subclass

# Why should Selenium be selected as a testing tool for web applications or systems?

Selenium provides the following advantages, which make it an excellent automated testing framework:

* It is free and open-source software with a large user base and supports providing community.
* It has cross-browser compatibility and supports multiple browsers like Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
* It supports multiple operating systems such as Windows, Linux, macOS, etc.
* It facilitates the usage of multiple programming languages including Scala, Ruby, Python, PHP, Perl, Java, Groovy, C#, etc.
* It provides support for distributed testing as well.

# What are the disadvantages of using Selenium as a testing tool?

The following are the disadvantages of using Selenium as a testing tool:

* **Tests web applications only:** Selenium supports the testing of only web-based applications. Mobile applications, Captcha, and Barcode readers cannot be tested using Selenium unless integrated with third-party tools like Appium and TestNG.
* **No built-in reporting and test management facility:** Selenium can generate reports only using third-party tools like TestNG or JUnit.
* **Unavailability of reliable tech support:** Since Selenium is an open-source tool, no dedicated support for user issues is available.
* **May require the knowledge of programming languages:** Some prior programming knowledge is required to use Selenium.

# What is meant by Selenium Suite and what are its different components?

Selenium is a package of several testing tools and is therefore often referred to as a Selenium Suite with each of these tools designed to cater to a different testing requirement.

Following are the different components of Selenium Suite:

* **Selenium Integrated Development Environment (IDE):** It is a Firefox/Chrome plug-in that is developed to speed up the creation of automation scripts by recording the user actions on the web browser and exporting them as a reusable script.
* **Selenium Remote Control (RC):** It is a server that enables users to generate test scripts in their preferred programming language. It accepts commands from the test scripts and sends them to the browser as Selenium core JavaScript commands, for the browser to behave accordingly.
* **Selenium WebDriver:** It is a programming interface that helps create and run test cases by directly communicating with the web browser and using its native compatibility to automate. Unlike RC, it doesn’t require an additional server to create and run test cases.
* **Selenium Grid:** It allows parallel execution of tests on different browsers and operating systems by distributing commands to different machines simultaneously.

# What is meant by Selenese? Explain different types of Selenium commands

The language used for writing test scripts in Selenium IDE is called Selenese. It is a set of commands used to test your web application or system. Selenium commands could be divided into 3 major categories:

1. **Actions:** These are the commands interacting directly with web applications.
2. **Accessors:** These are the commands which allow users to store values to a user-defined variable.
3. **Assertions:** They enable a comparison of the current state of the application with its expected state

# State the major difference between “assert” and “verify” commands in Selenium.

Both “assert” and “verify” commands check whether the given condition is true or false and the only difference between them is that:

* **Assert:** assert condition stops the execution of the testing if the given condition is false else would continue with the further tests.
* **Verify:** verify the condition doesn’t stop the flow of execution irrespective of the condition being true or false.

# What is meant by an exception test in Selenium?

An exception test is a test that expects an exception to be thrown inside a test class. It expects a @Test annotation followed by the expected exception name in the brackets.

**Eg:** @Test(expectedException = NoSuchElementException.class) is an exception test for missing elements in Selenium.

# What is meant by XPath in Selenium. Explain XPath Absolute and XPath Relative.

XPath, also defined as XML-Path (Extensible Markup Language Path), is a language used to query XML documents and provide functionalities like locating elements in Selenium by iterating through each element in a webpage. In XPath, data is stored in a key-value pair format similar to an HTML tag. It uses a single slash, i.e. ‘ / ’ for creating an absolute path, and a double slash, i.e. ‘ // ’ for creating a relative path for an element to be located on a webpage.

# Can selenium be used to launch web browsers?

Yes, Selenium provides good support to launch web browsers like Google Chrome, Mozilla Firefox, Internet Explorer, etc.

The following commands can be used to launch web browsers using Selenium:

* WebDriver driver = new FirefoxDriver();
* WebDriver driver = new ChromeDriver();
* WebDriver driver = new InternetExplorerDriver();

# With the help of code snippets, explain how we can create right-click and mouse hover actions in Selenium.

The following code can replicate right-click action:

actions action = newActions(driver);

WebElement element = driver.findElement(By.id("elementId"));

action.contextClick(element).perform();

The following code can replicate mouse hover action:

actions action = newActions(driver);

WebElement element = driver.findElement(By.id("elementId"));

action.moveToElement(element).perform();

# Can we handle a windows-based pop-up in Selenium, and if not, then what are the alternatives?

No, Selenium doesn’t support windows based pop-ups as it’s an automated testing tool built for web application based testing. However, with the support of third-party tools like AutoIT, Robot class, etc., windows-based pop-ups can be handled in selenium.

# What is Object Repository ?

An object repository is an essential entity in any UI automations which allows a tester to store all object that will be used in the scripts in one or more centralized locations rather than scattered all over the test scripts.

# Explain how you can find broken images in a page using Selenium Web driver ?

To find the broken images in a page using Selenium web driver is

* Get XPath and get all the links in the page using tag name
* In the page click on each and every link
* Look for 404/500 in the target page title

# Explain how you can handle colors in web driver?

To handle colors in web driver you can use

Use getCssValue(arg0) function to get the colors by sending ‘color’ string as an argument

#### Using web driver how you can store a value which is text box?

You can use following command to store a value which is text box using web driver

driver.findElement(By.id(“your Textbox”)).sendKeys(“your keyword”);

# Explain how you can switch between frames?

To switch between frames webdrivers **[ driver.switchTo().frame() ]** method takes one of the three possible arguments

* A number:  It selects the number by its (zero-based) index
* A name or ID: Select a frame by its name or ID
* Previously found WebElement: Using its previously located WebElement select a frame

# How can you find if an element is displayed on the screen?

WebDriver allows user to check the visibility of the web elements. These web elements can be buttons, radio buttons, drop, checkboxes, boxes, labels etc. which are used with the following methods.

* isDisplayed()
* isSelected()
* isEnabled()

**Syntax:**

1. isDisplayed():
2. **boolean** buttonPresence = driver.findElement(By.id("gbqfba")).isDisplayed();
3. isSelected():
4. **boolean** buttonSelected = driver.findElement(By.id("gbqfba")).isSelected();
5. isEnabled():
6. **boolean** searchIconEnabled = driver.findElement(By.id("gbqfb")).isEnabled();

# **What type of test cases not to be automated?**

Types of Test Cases Not To Be Automated are

* Subjective Validation
* New Functionalities
* Strategic Development
* User Experience
* Complex Functionality
* Quality Control
* Low return on investment
* Installation and setup testing

# **What are the types of WebDriver APIs available in Selenium?**

* Firefox Driver
* Gecko Driver
* InternetExplorer Driver
* Chrome Driver
* HTMLUnit Driver
* Opera Driver
* Safari Driver
* Android Driver
* iPhone Driver
* EventFiringWebDriver

# **Which WebDriver implementation claims to be the fastest?**

The fastest implementation of WebDriver is the HTMLUnitDriver. It is because the HTMLUnitDriver does not execute tests in the browser. Starting a browser and running test cases took more time compared to running the scripts without a browser. HTMLUnitDriver took a simple HTTP request-response mechanism for test case execution.

# **What is the difference between “/” and “//”**

**Single Slash “/” –**Single slash is used to create XPath with absolute path i.e. the XPath would be created to start selection from the document node/start node.

**Double Slash “//” –** Double slash is used to create XPath with relative path i.e. the XPath would be created to start selection from anywhere within the document.

# **What is the difference between setSpeed () and sleep () methods?**

Both sleep() and setSpeed() are used to delay the execution speed.

**setSpeed():** It set up speed that will apply a delay time before every Selenium operation.

***Example:*** setSpeed(“5000”) – It waits for 5 seconds

**sleep():** It set up wait only for once when called in our Selenium script.

***Example:*** sleep(5000) – It waits for 5 seconds

# **We do create a reference variable ‘driver’ of type WebDriver as shown below. What is the purpose of doing this way?**

WebDriver driver = new FirefoxDriver();

instead of creating

FirefoxDriver driver = new FirefoxDriver();

If we create a reference variable driver of type WebDriver then we could use the same driver variable to work with any browser of our choice such as IEDriver, SafariDriver etc.,

# **How to handle hidden elements in Selenium WebDriver?**

We can handle hidden elements by using javaScript executor

(JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");

# **How do you read test data from excels?**

Test data can efficiently be read from excel using JXL or POI API. POI API has many advantages than JXL.

# **Is it possible to automate the captcha using Selenium?**

No, It’s not possible to automate captcha and bar code reader.

# **How can you use the Recovery Scenario in Selenium WebDriver?**

By using “Try Catch Block” within Selenium WebDriver Java tests.

|  |  |
| --- | --- |
| 1  2  3  4  5 | try {       driver.get("www.softwaretestingmaterial.com");  }catch(Exception e){       System.out.println(e.getMessage());  } |

# **How to connect a Database in selenium?**

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

We use JDBC Driver to connect the Database in Selenium (While using Java Programming Language).

1. Establish a Database connection
2. Send SQL Queries to the Database
3. Process the results

**Loading the required JDBC (Java DataBase Connectivity) Driver class:**

Class.forName("net.sourceforge.jtds.jdbc.Driver");

**Establishing a connection to the DataBase**

Connection con = DriverManager.getConnection("DataBaseURL", "userName", "password");

**Executing SQL Queries:**

String sqlQuery = "SELECT \* FROM table\_name WHERE condition";

ResultSet resSet = sqlStatement.executeQuery(sqlQuery);

**Fetching data from result set:**

|  |  |
| --- | --- |
| 1  2  3 | while (resSet.next()) {      System.out.println(resSet.getString(required\_column\_name));  } |

**Disconnecting the Database connection:**

**Con.close()**

# **What is desired capabilities?**

In Selenium we use desired capabilities to handle SSL certificates in chrome browser

We need to create an instance of DesiredCapabilities

DesiredCapabilities desiredCapability = DesiredCapabilities.chrome();