

Selenium Interview questions and answers

This post is mainly for people preparing for interviews.

Before posting Selenium Interview questions and answers let me clear that it totally depends on number of factors.

- 1- For which position you are applying.*
- 2- Who is taking your interview?*
- 3- For which company you are applying.*
- 4- Does Interviewer's themselves knows about automation or not?*

You must be wondering why I am covering all these factors right. I have taken couple of interviews and based on my experience I will tell you what questions an Interviewer might ask.

If you are applying for Sr Automation Engg, Lead or Test Manager then be ready with some high-level topics, which anyways I will be covering in this post.

However, if you are applying as a fresher, then you should be ready with some Java program and some Selenium related code.

Before getting started keep in mind that there is not hard and fast rule that, the interviewer will ask questions from a specific domain. It might cover multiple domain/tools.

- 1- What is Selenium Webdriver, Selenium RC, Selenium IDE and Selenium Grid and which version you have used?

2-

Ans : Selenium Webdriver : WebDriver is a web automation tool that allows you to **execute your tests against different browsers**, not just Firefox (unlike Selenium IDE). It is faster than the Selenium RC. Because it directly communicate with browser no need of external server like Selenium RC.

Selenium IDE : **Selenium IDE** (Integrated Development Environment) is the simplest tool in the **Selenium Suite**. It is a Firefox add-on that creates tests very quickly through its record-and-playback functionality.

Selenium RC : **Selenium Remote Control (RC)** is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser.

Selenium Grid: **Selenium Grid** is a part of the **Selenium Suite** that specializes on running multiple tests across different browsers, operating systems, and machines in parallel.

*****_*****

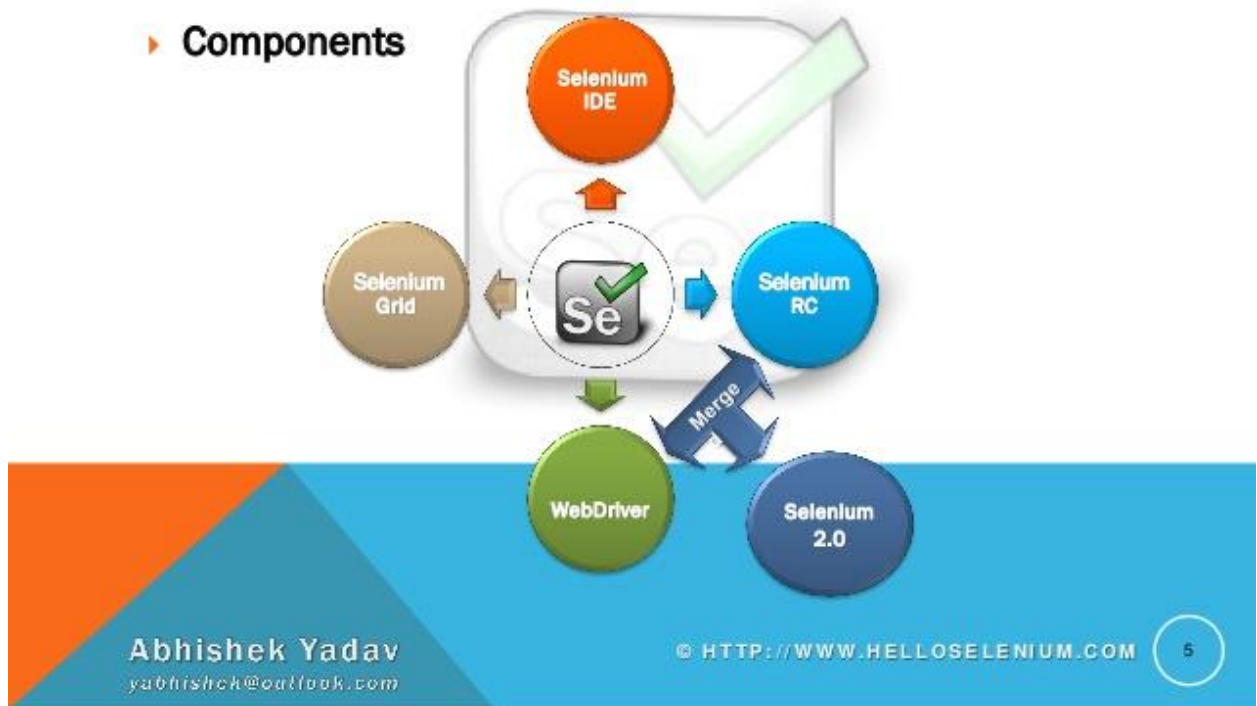
3- Can you please explain Selenium Architecture?

4-

SELENIUM OVERVIEW

HelloSe
+91 8512 888 161

► Components



5- Have you worked on different browsers? Have you ever performed Cross browser testing?

Ans: Yes, I have worked on different browsers like Mozilla (the most common), Chrome, and IE, performed cross browser testing demo. But while performing cross browser testing in my automation framework. It was getting slower and data would not be going on to the proper browser.

Ans- Every month a new browser is coming into market and it became very important to test our web application on different browser. Selenium supports Cross browser testing. Please check the detailed article here.

<http://learn-automation.com/cross-browser-testing-using-selenium-webdriver/>

6- What type of test you have automated?

Ans- Automation mainly focuses on regression testing, smoke testing, Sanity Testing and some time you can for End to End test cases.

5- How many test cases you have automated per day?

Ans- It totally depends on your manual test cases. Sometimes we can automate 3-4 test cases per day which contain limited operation and some validation. Some test cases may take 1 day or more than one day as well. It totally depends on test case complexity.

6- How to work with Chrome, IE or any other third party driver? Why we need separate driver for them?

Ans- Selenium by default supports only Firefox browser so in order to work with Chrome and IE browser we need to download separate drivers for them and specify the path.

<http://learn-automation.com/launch-chrome-browser-using-selenium-webdriver/>

<http://learn-automation.com/execute-selenium-script-in-ie-browser/>

7- Have you created any framework? Which framework you have used?

Ans. Yes. I have created Data-Driven Framework and added some functionalities of Hybrid framework and Page object model.

8- Can you please explain framework architecture?



9- What is POM (Page Object Model) and what is the need of it?

Ans- Page Object Model Framework has now a days become very popular test automation framework in the industry and many companies are using it because of its easy test maintenance and reduces the duplication of code. The main advantage of Page Object Model is that if the UI changes for any page, it doesn't require us to change any tests, we just need to change only the code within the page objects (Only at one place). Many other tools which are using selenium, are following the page object model.

The Page Object model provides the following advantages.

1. There is clean separation between test code and page specific code such as locators (or their use if you're using a UI map) and layout. 2. There is single repository for the services or operations offered by the page rather than having these services scattered throughout the tests.

10- What are the challenges you have faced while automating your application?

Ans- Challenges faced that are as follows:

- Frequently changing UI. It always need to make changes in code most of the time.
- Stuck somewhere while running automation scripts in chrome browser getting error that element is not visible, element not found.
- New kind of element like ck-editor, bootstrap calendar and dynamic web tables. But get the solution always.
- Reused of test scripts.
- To be continued.....

11- What are the different type of exception available in Selenium? Have you faced any exception while automation?

Ans- Yes. I have faced lots of exception. List are as follows:

1. **ElementNotSelectableException**
2. **ElementNotVisibleException**
3. **NoSuchAttributeException**
4. **NoSuchElementException**
5. **NoSuchFrameException**
6. **TimeoutException**
7. **Element not visible at this point**

12- What is Alert window/ JavaScript Alert and How to handle alert in Selenium Webdriver?

Ans: There are two types of alerts that we would be focusing on majorly:

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

As we know that handling windows based pop ups is beyond WebDriver's capabilities, thus we would exercise some third party utilities to handle window pop ups. Handling pop up is one of the most challenging piece of work to automate while testing web applications. Owing to the diversity in types of pop ups complexes the situation even more.

Generally JavaScript popups are generated by web application and hence they can be easily controlled by the browser.

Webdriver offers the ability to cope with javascript alerts using Alerts API [Click here to view Alert API Details](#)

// Get a handle to the open alert, prompt or confirmation

Alert alert = driver.switchTo().alert();

Alert is an interface. There below are the methods that are used

//Will Click on OK button.

alert.accept();

// Will click on Cancel button.

alert.dismiss()

//will get the text which is present on th Alert.

alert.getText();

//Will pass the text to the prompt popup

alert.sendkeys();

//Is used to Authenticate by passing the credentials

alert.authenticateUsing(Credentials credentials)

Example to handle alert in selenium Webdriver:

The below is the sample code for alerts, please copy and make an html file and pass it to the webdriver.

```
<html>
<head>
<title>Selenium Easy Alerts Sample </title>
```

```

</head><body>
<h2>Alert Box Example</h2>
<fieldset>
<legend>Alert Box</legend><p>Click the button to display an alert box.</p>
<button onclick="alertFunction()">Click on me</button>
<script>
function alertFunction()
{
alert("I am an example for alert box!");
}
</script>
</fieldset>
</body>
</html>

```

The below program will demonstrate you working on Alerts popup using above html file.

```

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Test;

public class PopupsHandling {
    WebDriver driver=new FirefoxDriver();
    @Test
    public void ExampleForAlert() throws InterruptedException
    {
        driver.manage().window().maximize();
        driver.get("file:///C:/path/alerts.html");
        Thread.sleep(2000);
        driver.findElement(By.xpath("//button[@onclick='alertFunction()']")).click();
        Alert alert=driver.switchTo().alert();
        System.out.println(alert.getText());
        alert.accept();
    }
}

```

```
}
```

```
*****-----*****
```

13- How to handle multiple windows in Selenium?

Ans:

When we have multiple windows in test automation, all we need to do is switching the focus from one window to another. Let us understand the same in the following way:

Window A has a link "Link1" and we need to click on the link (click event).

Window B displays and we perform some actions.

The entire process can be fundamentally segregated into following steps:

Step 1 : Clicking on Link1 on Window A

A new Window B is opened.

Step 2 : Save reference for Window A

Step 3 : Create reference for Window B

Step 3 : Move Focus from Window A to Window B

Window B is active now

Step 3 : Perform Actions on Window B

Complete the entire set of Actions

Step 4 : Move Focus from Window B to Window A

Window A is active now

Let us understand the same with a small coding example.

```
import java.util.List;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
```

```
public class MultipleWindowsHandle {
```

```

WebDriver driver;
@Before
public void setup() throws Exception {
driver=new FirefoxDriver();
String URL="http://www.seleniummaster.com";
driver.get(URL);
driver.manage().window().maximize();
}
@Test
public void test() throws Exception {
// Opening site
driver.findElement(By.xpath("//img[ @alt='SeleniumMasterLogo']")).click();
// Storing parent window reference into a String Variable
String Parent_Window = driver.getWindowHandle();
// Switching from parent window to child window
for (String Child_Window : driver.getWindowHandles())
{
driver.switchTo().window(Child_Window);
// Performing actions on child window
driver.findElement(By.id("dropdown_txt")).click();
List dropdownitems=driver.findElements(By.xpath("//div[ @id='DropDownitems']/div"));
int dropdownitems_Size=dropdownitems.size();
System.out.println("Dropdown item size is:"+dropdownitems_Size);
((WebElement) dropdownitems.get(1)).click();
driver.findElement(By.xpath("//*[ @id='anotherItemDiv']")).click();
}
//Switching back to Parent Window
driver.switchTo().window(Parent_Window);
//Performing some actions on Parent Window
driver.findElement(By.className("btn_style")).click();
}
@After
public void close() {
driver.quit();
}
}

```

*****-----*****

14- Have you ever worked on frames? In addition, how to handle frames in Selenium?

Yes. In Selenium to work with iFrames, we have different ways to handle frame depending on the need.

Please look at the below ways of handling frames.

driver.switchTo().frame(int arg0);

Select a frame by its (zero-based) index. That is, if a page has multiple frames (more than 1), the first frame would be at index "0", the second at index "1" and so on.

Once the frame is selected or navigated , all subsequent calls on the WebDriver interface are made to that

frame. i.e the driver focus will be now on the frame. Whatever operations we try to perform on pages will not work and throws element not found as we navigated / switched to Frame.

Example: if iframe id=webclipper-publisher-widget-container-frame, it can be written as
driver.switchTo().frame("webclipper-publisher-widget-container-frame"); Below is the code snippet to work with switchToFrame using frame id.

Code:

```
public void switchToFrame(int frame) {
    try {
        driver.switchTo().frame(frame);
        System.out.println("Navigated to frame with id " + frame);
    } catch (NoSuchFrameException e) {
        System.out.println("Unable to locate frame with id " + frame +
            e.getStackTrace());
    } catch (Exception e) { System.out.println("Unable to navigate to
        frame with id " + frame + e.getStackTrace()); } }
```

driver.switchTo().frame(WebElement frameElement);

Some times when there are multiple Frames (Frame in side a frame), we need to first switch to the parent frame and then we need to switch to the child frame. below is the code snippet to work with multiple frames.

```
public void switchToFrame(String ParentFrame, String ChildFrame) { try {
    driver.switchTo().frame(ParentFrame).switchTo().frame(ChildFrame); System.out.println("Navigated to
    innerframe with id " + ChildFrame + "which is present on frame with id" + ParentFrame); } catch
    (NoSuchFrameException e) { System.out.println("Unable to locate frame with id " + ParentFrame + " or "
    + ChildFrame + e.getStackTrace()); } catch (Exception e) { System.out.println("Unable to navigate to
    innerframe with id " + ChildFrame + "which is present on frame with id" + ParentFrame +
    e.getStackTrace()); } }
```

After working with the frames, main important is to come back to the web page. if we don't switch back to the default page, driver will throw an exception. Below is the code snippet to switch back to the default content.

```
public void switchtoDefaultFrame() { try { driver.switchTo().defaultContent();
    System.out.println("Navigated back to webpage from frame"); } catch (Exception e) { System.out
    .println("unable to navigate back to main webpage from frame" + e.getStackTrace()); } }
```

*****_____*****

15- What are different locators available in Selenium?

Ans- There are 8 types of locators are available in selenium that are as follows:

id, xpath, name, byClassName, css selector, byTagName, linkText, partialLinkText.

16- Can you please explain XPATH and CSS technique? How to handle dynamic changing elements?

CSS Selector:

CSS mainly used to provide style rules for the web pages and we can use for identifying one or more elements in the web page using css.

If you start using css selectors to identify elements, you will love the speed when compared with XPath.

Check this for more details on [Css selectors examples](#)

We can use Css Selectors to make sure scripts run with the same speed in IE browser. CSS selector is always the best possible way to locate complex elements in the page.

XPath Selector:

XPath is designed to allow the navigation of XML documents, with the purpose of selecting individual elements, attributes, or some other part of an XML document for specific processing

There are two types of xpath

1. Native Xpath, it is like directing the xpath to go in direct way. like

Example:

html/head/body/table/tr/td

Here the advantage of specifying native path is, finding an element is very easy as we are mention the direct path. But if there is any change in the path (if some thing has been added/removed) then that xpath will break.

2. Relative Xpath.

In relative xpath we will provide the relative path, it is like we will tell the xpath to find an element by telling the path in between.

Advantage here is, if at all there is any change in the html that works fine, until unless that particular path has changed. Finding address will be quite difficult as it need to check each and every node to find that path.

Example:

//table/tr/td

Dynamic CSS Selector in Selenium using multiple ways

Difference between CSS selector and xpath

CSS Selector	Xpath
It is style sheet.	Xpath is xml path.
Performance of css selectors faster than xpath.	Slower than css selector.
CSS selectors behaves perfectly in IE and other browser	Xpath sometimes get changed in case of IE browser.

Symbol used while writing css selector

	attribute	Symbol used
1	Using id	use # symbol
2	Using class name	use . symbol
3	Using attribute	tagname[attribute='value']
4	Using multiple attribute	tagname[attribute1='value1'] [attribute2='value2']
5	Contains	* symbol
6	Starts with	^ symbol
7	Ends with	\$ symbol

Examples :

- Find css selector using single attribute. Syntax- tagname[attribute='value'] , Example- input[id='user_login']
- Css selector using multiple attribute: Syntax- tagname[attribute1='value1'][attribute2='value2']
- Find css using id and class name. Syntax for id: tagname#id, syntax for class: tagname.className
- Find css using contains. Syntax : tagname[attribute*='value']
- Find css using start-with. Syntax: tagname[attribute^='value']
- Find css using end-with. Syntax: tagname[attribute\$='value']

Xpath Examples :

- Using multiple attribute. Syntax: //tagname[@attribute1='value1'][attribute2='value2'], Example: //a[@id='id1'][@name='namevalue1'] , //img[@src=''][@href='']
- Using single attribute. Syntax: // tagname[@attribute-name='value1'] , Example: // a [@href='http://www.google.com'] , //input[@id='name'] , //input[@name='username']
- Using contains method. Syntax: //tagname[contains(@attribute,'value1')] , Example: //input[contains(@id,'')] , //input[contains(@name,'')] , //a[contains(@href,'')] , //img[contains(@src,'')] , //div[contains(@id,'')]
- Using starts with. Syntax: //tagname[starts-with(@attribute-name,'')] , Example: //id[starts-with(@id,'')]

- Using following node. Syntax: `//input[@id='']/following::input[1]`
- Using preceding node. Syntax: `//input[@id='']/preceding::input[1]`

17- How to verify checkbox (any element) is enable/disable/ checked/Unchecked/ displayed/ not displayed?

Ans- We have different Boolean methods for enable / disable, checked / unchecked and displayed / not displayed that are as follows:

1. There's a method **"isEnabled()", that checks whether a WebElement is enabled or not**. You can use the below code to check for that;

```
boolean enabled = driver.findElement(By.xpath("//xpath of the checkbox")).isEnabled();
```

2. **to check whether the checkbox is checked/selected or not, you can use "isSelected()" method**, which you can use like this;

```
boolean checked = driver.findElement(By.xpath("//xpath of the checkbox")).isSelected();
```

18- How to work with dropdown?

Ans- WebDriver's support classes called "Select", which provides useful methods for interacting with select options. User can perform operations on a select dropdown and also de-select operation using the below methods.

We can select or deselect option in dropdown by using following methods.

Syntax:

```
Select Se=new Select(element);
```

```
Se.selectByIndex(index);
```

```
Se.selectByvalue(value);
```

```
Se.selectByVisibleText(text);
```

We can also deselect the item using same thing that is just above method like.

19- Have you worked with Web table (Calendar)? If yes, then what was your approach.

Ans- Yes. First need to analysis its web page html code for this element. To find which type of calendar is, then you can decide you can solve this calendar by using selenium Webdriver or using JavaScript executer. It all depends on the scenario the code. Now a day, there would be no. of new type of calendar are using by dev teams. We can't handle these by using selenium but using JavaScript executer we get solution.

20- Can you tell me some navigation commands?

Ans- To access the navigation's method, just type ***driver.navigate()***.. The intellisense feature of eclipse will automatically display all the public methods of ***Navigate Interface***.

Command - *driver.navigate().to(appUrl);*

It does exactly the same thing as the ***driver.get(appUrl)*** method. Where ***appUrl*** is the website address to load. It is best to use a fully qualified URL.

forward() : void – This method does the same operation as clicking on the ***Forward Button*** of any browser. It neither accepts nor returns anything.

Command - *driver.navigate().forward();*

Takes you forward by one page on the browser's history.

back() : void – This method does the same operation as clicking on the ***Back Button*** of any browser. It neither accepts nor returns anything.

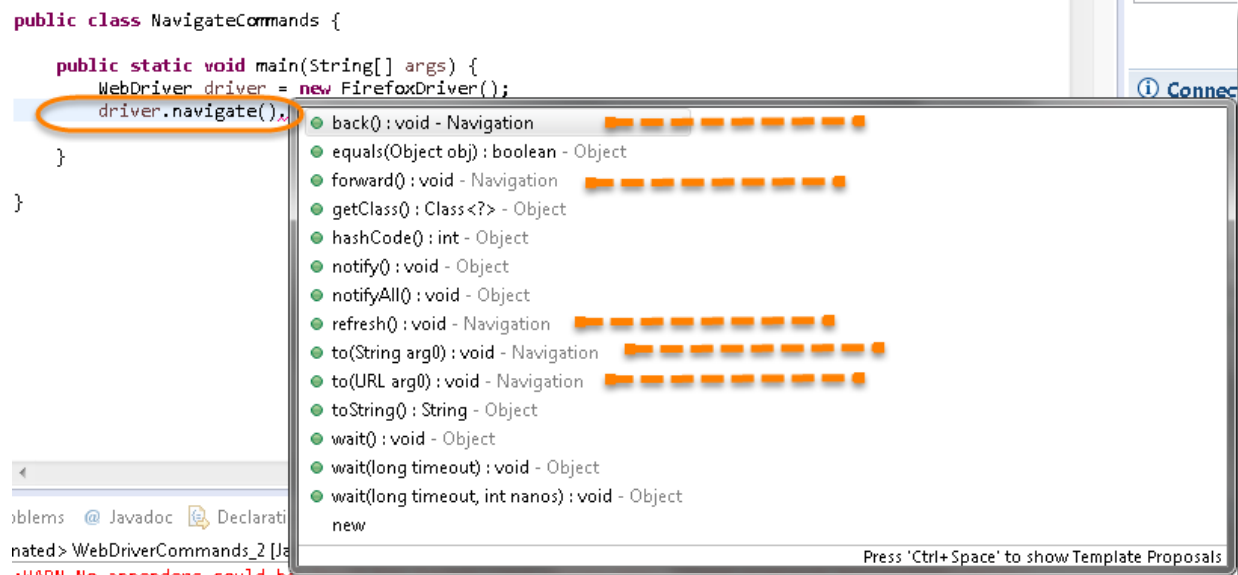
Command - *driver.navigate().back();*

Takes you back by one page on the browser's history.

refresh() : void – This method ***Refresh*** the current page. It neither accepts nor returns anything.

Command - *driver.navigate().refresh();*

Perform the same function as pressing F5 in the browser.



*****_*****

21- Difference between QUIT and Close?

Ans- driver.close and driver.quit are two different methods for closing the browser session in Selenium WebDriver. Understanding both of them and knowing when to use which method is important in your test execution. Therefore, in this article, we have tried to throw light on both these methods.

- driver.close – It closes the browser window on which the focus is set.
- driver.quit – It basically calls driver.dispose method which in turn closes all the browser windows and ends the WebDriver session gracefully.

You should use driver.quit whenever you want to end the program. It will close all opened browser window and terminates the WebDriver session. If you do not use driver.quit at the end of program, WebDriver session will not close properly and files would not be cleared off memory. This may result in memory leak errors.

*****-----*****

22- Can you find the number of links available on Webpage?

Ans- **Step to follow...**

- 1) **Navigate** to the interested webpage for e.g. www.toolsqa.com.
- 2) **Create** a list of type **WebElement** to store all the **Link elements** in to it.
- 3) **Collect** all the links from the webpage. All the links are associated with the **Tag 'a'**.
- 4) Now **iterate** through every link and print the **Link Text** on the console screen.

```
1 package practiceTestCases;
2
3 import org.openqa.selenium.By;
4
5 import org.openqa.selenium.WebDriver;
6
7 import org.openqa.selenium.WebElement;
8
9 import org.openqa.selenium.firefox.FirefoxDriver;
10
11 public class FindAllLinks {
12
13     public static void main(String[] args) {
14
15         WebDriver driver = new FirefoxDriver();
16
17         driver.get("http://toolsqa.com/");
18
19         java.util.List<WebElement> links = driver.findElements(By.tagName("a"));
20
```

```

21         System.out.println(links.size());
22
23         for (int i = 1; i<=links.size(); i=i+1)
24         {
25
26             System.out.println(links.get(i).getText());
27
28         }
29     }
30
31 }
32
33 }

```

The same way you can easily be able to find any type of WebElements on a WebPage:

Find total number of Checkboxes on a Webpage :

```

java.util.List<WebElement> checkboxes =
driver.findElements(By.xpath("//input[@type='checkbox']")); System.out.println(checkboxes.size());

```

Find total number of Menus on a Webpage :

```

2 java.util.List<WebElement> dropdown = driver.findElements(By.tagName("select"));
3 System.out.println(dropdown.size());

```

Find total number of TextBoxes on a Webpage :

```

java.util.List<WebElement> textboxes =
driver.findElements(By.xpath("//input[@type='text'[@class='inputtext']]"));
System.out.println(textboxes.size());

```

23- What is Page Load Timeout?

Ans-When automation script run on the browser. Sometimes scripts are faster than the web application that time scripts looking for an element but it can't be found because the web page not loaded completely and throws an element not found or element not visible exception. To eliminate these kind of exception and ensuring script run smoothly for this we mention or set page load time out.

Ex: driver.manage().timeouts().pageLoadTimeout(10, TimeUnit.SECONDS);

24- Can you explain implicit wait, explicit wait and fluent wait and what is the difference between all of them? Which one you have used frequently?

Ans- Implicit Wait

Selenium WebDriver has borrowed the idea of **implicit waits** from **Watir**. This means that we can tell Selenium that we would like it to wait for a certain amount of time before throwing an **exception** that it

cannot find the element on the page. We should note that implicit waits will be in place for the entire time the browser is open. This means that any search for elements on the page could take the time the implicit wait is set for.

```
Ex: WebDriver driver = new FirefoxDriver();
```

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

```
driver.get("http://url_that_delays_loading");
```

```
WebElement myDynamicElement = driver.findElement(By.id("myDynamicElement"));
```

Fluent Wait

Each **FluentWait** instance defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition. Furthermore, the user may configure the wait to ignore specific types of exceptions whilst waiting, such as **NoSuchElementException** when searching for an element on the page.

Ex: // Waiting 30 seconds for an element to be present on the page, checking

```
// for its presence once every 5 seconds.
```

```
Wait wait = new FluentWait(driver)
```

```
.withTimeout(30, SECONDS)
```

```
.pollingEvery(5, SECONDS)
```

```
.ignoring(NoSuchElementException.class);
```

```
WebElement foo = wait.until(new Function() {
```

```
    public WebElement apply(WebDriver driver) {
```

```
        return driver.findElement(By.id("foo"));
```

```
    }
```

```
});
```

Explicit Wait

It is more extendible in the means that you can set it up to wait for any condition you might like. Usually, you can use some of the prebuilt **ExpectedConditions** to wait for elements to become clickable, visible, invisible, etc.

Ex:

```
WebDriverWait wait = new WebDriverWait(driver, 10);  
2  
3 WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("someid")));
```

Difference Between Implicit, Explicit and Fluent Wait

Implicit Wait: During Implicit wait if the Web Driver cannot find it immediately because of its availability, the WebDriver will wait for mentioned time and it will not try to find the element again during the specified time period. Once the specified time is over, it will try to search the element once again the last time before throwing exception. The default setting is zero. Once we set a time, the Web Driver waits for the period of the WebDriver object instance.

Explicit Wait: There can be instance when a particular element takes more than a minute to load. In that case you definitely not like to set a huge time to Implicit wait, as if you do this your browser will be going to wait for the same time for every element.

To avoid that situation, you can simply put a separate time on the required element only. By following this your browser implicit wait time would be short for every element and it would be large for specific element.

Fluent Wait: Let's say you have an element which sometime appears in just 1 second and some time it takes minutes to appear. In that case it is better to use fluent wait, as this will try to find element again and again until it finds it or until the final timer runs out.

25- What is JavaScript Executor and where you have used JavaScript executor?

Ans- **JavascriptExecutor it is an interface.** It Indicates that a driver can execute JavaScript, providing access to the mechanism to do so.

There were lots of scenarios' their we need java-script should be executing for some element that are as follows:

1. when element is not clickable using locators then we can have used JavaScript.
2. While working with frames we used JavaScript.
3. The most recently while working with ck-editor I used JavaScript executors.
4. For the bootstrap calendar when the conditions for using selenium command it can't be possible that time I used JavaScript executor etc.

Syntax: JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript(Script,Arguments);

Examples: 1. How to generate *Alert Pop window* in selenium?

```
JavascriptExecutor js = (JavascriptExecutor)driver;  
Js.executeScript("alert('hello world');");
```

26- How to capture Screenshot in Selenium? Can we capture screenshot only when test fails?

Ans- For taking screenshots Selenium has provided TakesScreenShot interface in this interface you can use getScreenshotAs method which will capture the entire screenshot in form of file then using FileUtils we can copy screenshots from one location to another location.

Ex: // Take screenshot and store as a file format

```
File src= ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
try {
    // now copy the screenshot to desired location using copyFile //method
    FileUtils.copyFile(src, new File("C:/selenium/error.png"));
}

catch (IOException e)
{
    System.out.println(e.getMessage());
}
```

And we can capture screen-shot while test fails we can do this by following way.

Ex: create one method for capturing screen-shot. And then call this method in test methods catch block.

```
public static void captureScreenShot(WebDriver ldriver){

    // Take screenshot and store as a file format
    File src= ((TakesScreenshot)ldriver).getScreenshotAs(OutputType.FILE);
    try {
        // now copy the screenshot to desired location using copyFile method

        FileUtils.copyFile(src, new File("C:/selenium/"+System.currentTimeMillis()+".png"));
    }

    catch (IOException e)

    {

        System.out.println(e.getMessage());

    }

}
```

Now call this method in test methods catch block.

*****-----*****

27- What is Web Driver Listener and Usage of the same?

Ans- In general, terms, Listeners are whom that listen to you. If you talk about Webdriver Listener so you should make a note of some classes and interfaces.

1- WebDriverEventListener – This is an interface, which have some predefined methods so we will implement all of these methods. 2-EventFiringWebDriver- This is a class that actually fire Webdriver event.

Why we are using Webdriver Listeners: If you talk about Webdriver we are doing some activity like type, click, navigate etc this is all your events which you are performing on your script so we should have activity which actually will keep track of it. Take an example if you perform click then what should happen before click and after click. To capture these events, we will add listener that will perform this task for us.

Ex: package listerDemo;

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.events.WebDriverEventListener;

public class ActivityCapture implements WebDriverEventListener {

    @Override
    public void afterChangeValueOf(WebElement arg0, WebDriver arg1) {

    }

    @Override
    public void afterClickOn(WebElement arg0, WebDriver arg1) {

        System.out.println("After click "+arg0.toString());

    }

    @Override
    public void afterFindBy(By arg0, WebElement arg1, WebDriver arg2) {

        System.out.println("After FindBy "+arg0.toString());

    }

    @Override
    public void afterNavigateBack(WebDriver arg0) {
```

```
System.out.println("After navigating back "+arg0.toString());
```

```
}
```

```
@Override
```

```
public void afterNavigateForward(WebDriver arg0) {
```

```
System.out.println("After navigating forward "+arg0.toString());
```

```
}
```

```
@Override
```

```
public void afterNavigateTo(String arg0, WebDriver arg1) {
```

```
System.out.println("After navigating "+arg0.toString());
```

```
System.out.println("After navigating "+arg1.toString());
```

```
}
```

```
@Override
```

```
public void afterScript(String arg0, WebDriver arg1) {
```

```
}
```

```
@Override
```

```
public void beforeChangeValueOf(WebElement arg0, WebDriver arg1) {
```

```
}
```

```
@Override
```

```
public void beforeClickOn(WebElement arg0, WebDriver arg1) {
```

```
System.out.println("before click "+arg0.toString());
```

```
}
```

```
@Override
```

```
public void beforeFindBy(By arg0, WebElement arg1, WebDriver arg2) {
```

```
System.out.println("before FindBY "+arg0.toString());
```

```
}
```

```
@Override
```

```
public void beforeNavigateBack(WebDriver arg0) {
```

```
System.out.println("Before navigating back "+arg0.toString());
```

```
}
```

```
@Override
public void beforeNavigateForward(WebDriver arg0) {
System.out.println("Before navigating Forward "+arg0.toString());

}
```

```
@Override
public void beforeNavigateTo(String arg0, WebDriver arg1) {

System.out.println("Before navigating "+arg0.toString());
System.out.println("Before navigating "+arg1.toString());

}
```

```
@Override
public void beforeScript(String arg0, WebDriver arg1) {

}
```

```
@Override
public void onException(Throwable arg0, WebDriver arg1) {

System.out.println("Testcase done"+arg0.toString());
System.out.println("Testcase done"+arg1.toString());
}

}
```

Let's Discuss one of these methods

```
@Override

public void afterClickOn(WebElement arg0, WebDriver arg1) {

System.out.println("After click "+arg0.toString());

}
```

In above method we are simply printing on console and this method will automatically called once click events done. In same way you have to implement on methods.

Note- We generally use Listener to generate log events

Step 2- Now create your simple script, create EventFiringWebDriver object, and pass your driver object.

```
EventFiringWebDriver event1=new EventFiringWebDriver(driver);
```

Step 3- Create an object of the class who has implemented all the method of WebDriverEventListener so in our case ActivityCapture is a class who has implemented the same.

```
ActivityCapture handle=new ActivityCapture();
```

Step 4- Now register that event using register method and pass the object of ActivityCapture class
event1.register(handle);

```
*****-----*****
```

28-How to scroll in Selenium Webdriver?

Ans- Selenium support auto scroll to find an element but sometimes we need to scroll based on requirement like scroll up and down. We can perform this using Java Script. In this video, we will discuss How to Scroll page up and down in Selenium Webdriver.

```
ex: WebDriver driver = new FirefoxDriver();
JavascriptExecutor jse = (JavascriptExecutor)driver;
jse.executeScript("window.scrollTo(0,250)", "");
```

```
*****-----*****
```

29- Can you write login script for Gmail or any other application?

Ans-Yes.

```
*****-----*****
```

30- How to upload files in Selenium? Have you ever used AutoIT?

Ans-We can upload file in web application by using directly sending path in sendKeys. But at sometimes selenium path could not accept or upload the things, for this we used AutoIT tool.

1-AutoIt is freeware automation tool that can work with desktop application too. 2-It uses a combination of keystrokes, mouse movement and window/control manipulation in order to automate tasks in a way not possible or reliable with other languages (e.g. VBScript and SendKeys).

```
*****-----*****
```

31- How to handle untrusted Certificate in Selenium?

Ans- Untrusted SSL certificates: Whenever We try to access HTTPS website so many time so will face untrusted SSL certificate issue. This issue comes in all browser like IE, Chrome,Safari, Firefox etc.

This certificate some in multiple conditions and we should know all of them so that we can rectify them easily.

1- Each secure site has Certificate so it certificate is not valid *up-to date*.

2- Certificate has been expired on *date*

3- Certificate is only valid for (*site name*)

4- The certificate is not trusted because the issuer certificate is unknown due to many reason.

Handle untrusted certificate in selenium.

Step 1-We have to create FirefoxProfile in Selenium.

Step 2- We have some predefined method in Selenium called setAcceptUntrustedCertificates() which accept Boolean values(true/false)- so we will make it true.

Step 3-Open Firefox browser with above created profile.

Handle untrusted certificate in Firefox

```
public class SSLCertificate {  
  
    public static void main(String[] args) {  
  
        //It create firefox profile  
        FirefoxProfile profile=new FirefoxProfile();  
  
        // This will set the true value  
        profile.setAcceptUntrustedCertificates(true);  
  
        // This will open firefox browser using above created profile  
        WebDriver driver=new FirefoxDriver();  
  
        driver.get("pass the url as per your requirement");  
  
    }  
  
}
```

Handle untrusted certificate in Chrome

```
// Create object of DesiredCapabilities class  
DesiredCapabilities cap=DesiredCapabilities.chrome();  
  
// Set ACCEPT_SSL_CERTS variable to true
```

```
cap.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);

// Set the driver path
System.setProperty("webdriver.chrome.driver", "Chrome driver path");

// Open browser with capability
WebDriver driver=new ChromeDriver(cap);
```

Handle untrusted certificate in IE
// Create object of DesiredCapabilities class

```
DesiredCapabilities cap=DesiredCapabilities.internetExplorer();
```

```
// Set ACCEPT_SSL_CERTS variable to true
cap.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
```

```
// Set the driver path
System.setProperty("webdriver.ie.driver", "IE driver path");
```

```
// Open browser with capability
WebDriver driver=newInternetExplorerDriver(cap);
```

Handle untrusted certificate in Safari
// Create object of DesiredCapabilities class

```
DesiredCapabilities cap=DesiredCapabilities.safari();
```

```
// Set ACCEPT_SSL_CERTS variable to true
cap.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
```

```
// Set the driver path
System.setProperty("webdriver.safari.driver", "Safari driver path");
```

```
// Open browser with capability
WebDriver driver=new SafariDriver();
```

*****_*****

32- How to handle window authentication window in Selenium Webdriver?

Ans- Each company having some proxy setting for specific application so while running script using Selenium you will get this authentication window which ask for Username and password so until we don't handle this you cannot navigate to parent window.

There are two ways to handle this issue.

1. Handle windows authentication using selenium Webdriver. You can provide credentials in URL itself it means we will add username and password in URL so while running script it will bypass the same.

2. We can use AutoIT to handle these kind of windows authentication.

We can use AutoIT again to handle this authentication window for this Please check whether you have AutoIT installed or not. 1) Open the Url ENGPROD on which the authentication is required and open the AutoIt Window Info tool to get the name of the class and the text of the authentication window.

*****_*****

33- What is Firefox Profile?

Ans: **Firefox Profile**

Firefox saves your personal information such as bookmarks, passwords, and user preferences in a set of files called your **profile**, which is stored in a separate location from the Firefox program files. You can have multiple Firefox profiles, each containing a separate set of user information. The **Profile Manager** allows you to **create**, **remove**, **rename**, and **switch** profiles.

Need of New Firefox Profile

The default Firefox profile is not very automation friendly. When you want to run automation reliably on a Firefox browser it is advisable to make a separate profile. Automation profile should be light to load and have special proxy and other settings to run good test.

You should be consistent with the profile you use on all development and test execution machines. If you used different profiles everywhere, the SSL certificates you accepted or the plug-ins you installed would be different and that would make the tests behave differently on the machines.

- - *There are several times when you need something special in your profile to make test execution reliable. The most common example is a SSL certificate settings or browser plug-ins that handles self-signed certs. It makes a lot of sense to create a profile that handles these special test needs and packaging and deploying it along with the test execution code.*
- - *You should use a very lightweight profile with just the settings and plug-ins you need for the execution. Each time Selenium starts a new session driving a Firefox instance, it copies the entire profile in some temporary directory and if the profile is big, it makes it, not only slow but unreliable as well.*

Steps to create new profile in Firefox:

1. Close Firefox browser if open.
2. Run -> firefox.exe -p
3. It opens Firefox user profile window.

4. Create new profile by clicking create profile and follow the steps.

You can use this in your scripts by using following code:

Code:

```
ProfilesIni profile = new ProfilesIni();
FirefoxProfile myprofile = profile. getProfile("profileToolsQA");
WebDriver driver = new FirefoxDriver(myprofile);
*****_*****
```

34- What are the issues or Challenges you faced while working with IE Browser?

Ans- **Challenges with IE browser in selenium Webdriver.**

- ✓ Openqa.selenium.NoSuchWindowException. (This is a common issue with Selenium and you can avoid this by doing some IE setting, which we are going to discuss now.)
- ✓ sendKeys works very slow it takes 1 to 2 seconds to type each character. (This is a known issue with Selenium and it only happens once you work with IE 64 bit driver.) **Solution:** You can download IE Driver 32 bit and start using it, even you are working with 64 bit OS this 32 bit IE driver works every time.
- ✓ Unexpected error while launching Internet Explorer.Protected mode must be set to the same value.(When I started working with IE this was the first exception, which I used to get, and I was sure that this related to some browser setting.) We can solve this issue using do some setting in internet explorer. To enabling one by one all security zones in internet options of IE.
- ✓ Unexpected error launching Internet Explorer.Browser zoom level was set to 0%. (By the name itself, you can see that we have to set the zoom level to 100 % to make it work.)
- ✓ Handle Untrusted SSL certificate error in IE browser in different ways
Solution: IE is the product of Microsoft and IE is much worried about security so when you start working with some https application you will get a untrusted certificate. Selenium has so many ways to handle this, but we will see 2 ways which work all the time for me.

1. Open the application for which SSL certificate is coming so use below code after passing the URL.

```
driver.get("ur app URL");
driver.navigate().to("javascript:document.getElementById('overridelink').click()");
```

// you can use your code now.

2. You can handle this certificate using Desired Capabilities as well. To accepting untrusted ssl certificates.

```
*****_*****
```

35- Have you ever faced any proxy issue if yes then how you handled?

Ans:

Handle proxy in Selenium Webdriver

When you try to access some secure applications you will get proxy issues so many times. Until we do not set proxy, we cannot access the application itself.

Handle Proxy in Selenium Webdriver

1- Change the proxy setting manually and open default browser.

2- Change the proxy setting using Webdriver code.

In this post, we will see second approach

To handle proxy setting in Selenium we have separate class called Proxy that is available inside org.openqa.selenium package

Sample Demo Code:

```
import org.openqa.selenium.Proxy;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.remote.DesiredCapabilities;
```

```
public class ProxyDemo
{

    public static void main(String[] args) {

        // Create proxy class object

        Proxy p=new Proxy();

        // Set HTTP Port to 7777

        p.setHttpProxy("localhost:7777");

        // Create desired Capability object

        DesiredCapabilities cap=new DesiredCapabilities();

        // Pass proxy object p

        cap.setCapability(CapabilityType.PROXY, p);

        // Open  firefox browser
```

```

WebDriver driver=new FirefoxDriver(cap);

// from here onwards code will be same as normal script

}

}

*****-----*****

```

36- What is Actions class in Selenium (How to perform Mouse Hover, Keyboard events, DragAndDrop etc?)

Ans-In Webdriver, handling keyboard events and mouse events (including actions such as Drag and Drop or clicking multiple elements With Control key) are done using the advanced user interactions API . It contains Actions and Action classes which are needed when performing these events. For all advance activity in Selenium Webdriver we can perform easily using Actions class like Drag and Drop, mouse hover, right click, clickandhold, releasemouse many more. We have predefined method called dragAndDrop(source, destination) which is method of Actions class.

To use mouse actions, we need to use current location of the element and then perform the action. The following are the regularly used mouse and keyboard events:

Method :clickAndHold()

Purpose: Clicks without releasing the current mouse location

Method : contextClick()

Purpose: Performs a context-click at the current mouse location.

[How to work with context menu by taking a simple example](#)

Method: doubleClick()

Purpose: Performs a double click at the current mouse location

Method: dragAndDrop(source,target)

Parameters: Source and Target

Purpose: Performs click and hold at the location of the source element and moves to the location of the target element then releases the mouse.

Method : dragAndDropBy(source,x-offset,y-offset)

Parameters: Source, xOffset - horizontal move, y-Offset - vertical move Offset

Purpose: Performs click and hold at the location of the source element moves by a given off set, then releases the mouse.

Method: **keyDown(modifier_key)**

Parameters: Modifier_key (keys.ALT or Keys.SHIFT or Keys.CONROL)

Purpose: Performs a modifier key press, doesn't release the modifier key. Subsequent interactions may assume it's kept pressed

Method: **keyUp(modifier_key)**

Parameters: Modifier_key (keys.ALT or Keys.SHIFT or Keys.CONROL)

Purpose: Performs a key release.

Mouse Hover:

```
public class mouseHover{

    public static WebDriver driver;

    public static void main(String[] args) {

        driver = new FirefoxDriver();

        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        driver.get("http://www.onlinestore.toolsqa.com");

        WebElement element = driver.findElement(By.linkText("Product Category"));

        Actions action = new Actions(driver);

        action.moveToElement(element).build().perform();

        driver.findElement(By.linkText("iPads")).click();

    }

}
```

Drag and Drop Example

```
public class DemoDragDrop {

    public static void main(String[] args) throws Exception {

        // Initiate browser
        WebDriver driver=new FirefoxDriver();

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

        // Open webpage
        driver.get("http://jqueryui.com/resources/demos/droppable/default.html");

    }

}
```

```
// Add 5 seconds wait
Thread.sleep(5000);

// Create object of actions class
Actions act=new Actions(driver);

// find element which we need to drag
WebElement drag=driver.findElement(By.xpath("//*[ @id='draggable']"));

// find element which we need to drop
WebElement drop=driver.findElement(By.xpath("//*[ @id='droppable']"));

// this will drag element to destination
act.dragAndDrop(drag, drop).build().perform();

}

}
```

*****_*****

37- Do we have object repository concept in Selenium if yes then please explain how we can achieve it?

Ans-Yes We have object Repository concept in selenium. Whenever you talk about repository by the name itself you can think that it is kind of storage. Object repository is the collection of object and object here is locator. Here locator means web element id, name, CSS, XPath, class name etc. Basically we can use object repository in automation framework because we need to handle huge amount of element locator separate so that it easy to handle and maintain.

*****_*****

38- What is Headless testing? Can we capture Screenshot in Headless mode?

Ans-If you have ever worked on [Firefox](#), [Chrome](#), IE browser in Selenium you might have faced so many issues like xpath is changing every time when new feature is getting added. Using these browser, some common issue which I faced during my daily automation like Sync issues and speed as well. Some browser takes time to load in my experience I faced performance issues with IE browser. To increase speed of your test script i.e. performance of your script we can try some testcases using Headless Browser Testing using Selenium.

Headless Testing / Headless Browser:

A browser, which does not have any GUI it means which runs in background. If you run your programs in Firefox, Chrome, IE and different browser then you can see how the browser is behaving but in headless browsers, you cannot.

Advantage and Disadvantage of headless browsers

One of the most Important advantage is Performance.

1-When we run your test case using headless browsers then you will get result just in seconds, you will see the result in below program.

2-When we have to run test case to create some sample test data or just you have to verify some messages and functionality then you can should try headless browsers.

3- When we have to run test case in remote machine or server, which does not have any browser, but still you have to execute test case then you can try with headless browsers again.

I hope you get the clear picture of this so let us start with some program and output as well. There are so many headless browser available in market, which do the same like Nodejs etc. When you build your test case using [Jenkins](#) then also it run in Headless mode

Program for headless Testing:

```
public class HtmlDemoProgram1 {

    public static void main(String[] args) throws InterruptedException {

        // Declaring and initialize HtmlUnitWebDriver
        WebDriver driver = new HtmlUnitDriver();

        // open facebook webpage
        driver.get("http://www.facebook.com");

        // Print the title
        System.out.println("Title of the page "+ driver.getTitle());

        // find the username field
        WebElement username = driver.findElement(By.id("email"));

        // enter username
        username.sendKeys("mukeshotwani.50@gmail.com");

        // find the password field
        WebElement password = driver.findElement(By.id("pass"));

        // Click the loginbutton
        password.sendKeys("pjs@903998");

        // find the Sign up button
```

```

WebElement Signup_button = driver.findElement(By.id("loginbutton"));

// Click the loginbutton
Signup_button.click();

// wait for 5 second to login
Thread.sleep(5000);

// You will get new title after login
System.out.println("After login title is = " + driver.getTitle());

}}
*****-----*****

```

39-Difference between findElement and FindElements?

Ans-findElement () will return only single WebElement and if that element is not located or we use some wrong selector then it will throw NoSuchElementException exception.

findElements() will return List of WebElements – for this we need to give locator in such a way that it can find multiple elements and will return you list of webelements then using List we can iterate and perform our operation.

*****-----*****

40- Can we highlight Element in Selenium?

Ans-Yes. In Selenium, we can use [JavascriptExecutor](#) (interface) to execute Javascript code into webdriver. In this post we, will execute Javascript which will highlight the element.

Code:

```

public class Highlight {

    public static void main(String []args){

        WebDriver driver=new FirefoxDriver();

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

        driver.get("http://www.facebook.com");

        // Create the JavascriptExecutor object
        JavascriptExecutor js=(JavascriptExecutor)driver;

        // find element using id attribute
        WebElement username= driver.findElement(By.id("email"));

        // call the executeScript method
    }
}

```



```

js.executeScript("arguments[0].setAttribute('style','border: solid 2px red')", username);
}
}

```

*****-----*****

41- What is log4j and How to generate log files in Selenium?

Ans- **Log4j**: Log4j is free open source tool given by Apache foundation for creating log files It help us to generate log file in various output target.

Generate Log Files: In my automation framework I use these following steps.

Step1: Download log4j.jar and add into your project.

Step2. Then create log4j.xml. That are some available log4j files on internet.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/" debug="false">
<appender name="fileAppender" class="org.apache.log4j.FileAppender">
<param name="Threshold" value="INFO" />
<param name="File" value="src/config/Application.log"/>
<layout class="org.apache.log4j.PatternLayout">
<param name="ConversionPattern" value="%d %-5p [%c{1}] %m %n" />
</layout></appender>
<root>
<level value="INFO"/>
<appender-ref ref="fileAppender"/>
</root>
</log4j:configuration>

```

Step3: We need to add Log.Java and then we can used logs in our automation framework that can we see while running script in console and the above log4j.xml files create application.log file.

```

public class Log {
// Initialize Log4j logs
private static Logger Log = Logger.getLogger(Log.class.getName());

public static void info(String message) {
Log.info(message);
} //We can create a lots of method like warn, error, fatal, etc as per our need.
}

```

*****-----*****

42-How to capture Page title, tooltip and error message and how to verify them?

Ans- **Verify Page title:**

```
public class verifyTitle {
WebDriver driver = new FirefoxDriver();
String title;

@Test
public void getTitlePage()
{
driver.get("https://www.salesforce.com/login");
driver.manage().window().maximize();

title = driver.getTitle();

System.out.println("Page title is: "+title);

Assert.assertTrue(title.contains("salesforce.com"));

driver.close();
}
}
```

Verify Tooltip:

```
public class Tooltip {

    public static void main(String[] args) {

        // This will open browser
        WebDriver driver=new FirefoxDriver();

        // This will maximize your browser
        driver.manage().window().maximize();

        // Open Gmail account creation page
        driver.get("https://accounts.google.com/SignUp?service=mail&continue=https%3A%2F%2Fmail.google.com%2Fmail%2F&ltmpl=default");

        // Click on username textbox
        driver.findElement(By.xpath(".*[@id='GmailAddress']")).click();

        // Create action class object
        Actions builder=new Actions(driver);

        // find the tooltip/help text message xpath
        WebElement username_tooltip=driver.findElement(By.xpath("html/body/div[2]/div[1]"));

        // Mouse hover to that text
        builder.moveToElement(username_tooltip).perform();

        // Extract the text from tooltip using getText
        String tooltip_msg=username_tooltip.getText();

        // Print the tooltip message
        System.out.println("Tooltip message is "+tooltip_msg);
    }
}
```

```
// This is expected message should come and store in variable
String expected_tooltip="You can use letters, numbers, and periods.";

// It will compare if actual matches with expected then TC will fail else it will fail

Assert.assertEquals(tooltip_msg, expected_tooltip);

System.out.println("Message verified");

    }

}
```

Verify Error Message:

```
public class TestNaukri {

    @Test
    public void TestError()
    {

        // Open browser
        FirefoxDriver driver=new FirefoxDriver();

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

        // Open URL
        driver.get("http://www.naukri.com/");

        // Click on login button
        driver.findElement(By.id("p0submit")).click();

        // This will capture error message
        String actual_msg=driver.findElement(By.id("emailId_err")).getText();

        // Store message in variable
        String expect="plz enter valid email";

        // Here Assert is a class and assertEquals is a method which will compare two values if// both matches it will run fine but in case
        //if does not match then it will throw an
        //exception and fail testcases

        // Verify error message
        Assert.assertEquals(actual_msg, expect);

    }

}
```

*****_____*****

43-How to download files in Selenium?

Ans-

```
public class FileDownloadExample
{
    public static String downloadPath = "D:\\seleniumdownloads";
    @Test
    public void testDownload() throws Exception
    {
        WebDriver driver = new FirefoxDriver(FirefoxDriverProfile());
        driver.manage().window().maximize();
        driver.get("http://spreadsheetpage.com/index.php/file/C35/P10/");
        driver.findElement(By.linkText("smilechart.xls")).click();
    }
    public static FirefoxProfile FirefoxDriverProfile() throws Exception
    {
        FirefoxProfile profile = new FirefoxProfile();
        profile.setPreference("browser.download.folderList", 2);
        profile.setPreference("browser.download.manager.showWhenStarting", false);
        profile.setPreference("browser.download.dir", downloadPath);
        profile.setPreference("browser.helperApps.neverAsk.openFile",
            "text/csv,application/x-msexcel,application/excel,application/x-excel,application/vnd.ms-excel,image/png,image/jpeg,text/html,text/plain,application/msword,application/xml");
        profile.setPreference("browser.helperApps.neverAsk.saveToDisk",
            "text/csv,application/x-msexcel,application/excel,application/x-excel,application/vnd.ms-excel,image/png,image/jpeg,text/html,text/plain,application/msword,application/xml");
        profile.setPreference("browser.helperApps.alwaysAsk.force", false);
        profile.setPreference("browser.download.manager.alertOnEXEOpen", false);
        profile.setPreference("browser.download.manager.focusWhenStarting", false);
        profile.setPreference("browser.download.manager.useWindow", false);
        profile.setPreference("browser.download.manager.showAlertOnComplete", false);
        profile.setPreference("browser.download.manager.closeWhenDone", false);
        return profile;
    }
}
```

We will explain you the preferences that we have set to Firefox browser.

setPreference("browser.download.folderList", 2);

Default Value: 1

The value of browser.download.folderList can be set to either 0, 1, or 2. When set to 0, Firefox will save all files downloaded via the browser on the user's desktop. When set to 1, these downloads are stored in the Downloads folder. When set to 2, the location specified for the most recent download is utilized again.

setPreference("browser.download.manager.showWhenStarting", false);

Default Value: true

The browser.download.manager.showWhenStarting Preference in Firefox's about:config interface allows the user to specify whether or not the Download Manager window is displayed when a file download is initiated.

browser.helperApps.alwaysAsk.force

True: Always ask what to do with an unknown MIME type, and disable option to remember what to open it with False (default): Opposite of above

browser.helperApps.neverAsk.saveToDisk

A comma-separated list of MIME types to save to disk without asking what to use to open the file. Default value is an empty string.

browser.helperApps.neverAsk.openFile

A comma-separated list of MIME types to open directly without asking for confirmation. Default value is an empty string.

browser.download.dir

The last directory used for saving a file from the "What should (browser) do with this file?" dialog.

browser.download.manager.alertOnEXEOpen

True (default): warn the user attempting to open an executable from the Download Manager

False: display no warning and allow executable to be run

Note: In Firefox, this can be changed by checking the "Don't ask me this again" box when you encounter the alert.

browser.download.manager.closeWhenDone

True: Close the Download Manager when all downloads are complete

False (default): Opposite of the above

browser.download.manager.focusWhenStarting

True: Set the Download Manager window as active when starting a download

False (default): Leave the window in the background when starting a download

44- Have you ever performed Database testing in Selenium?

No.

45- Have you integrated Selenium with other tools like Sikuli, Ant, Maven, and Jenkins? If yes, then how you have used them?

Ans- Yes I am very much interested. Currently I am using Ant in my automation framework. And I am planning to use maven in next project. But individually I have performed some demo on Sikuli, maven, Jenkins as well.

46- Does Selenium support Mobile Automation if yes then How?

Ans-Yes. But it needs some external tools to communicate with mobile app and perform automation.

There is various tool available for mobile automation. Robotium, Monkey-runner, Ranorex, Appium and UI Automator.

47- What is Limitation of Selenium?

Ans-Some limitations of Selenium Automation tool are as follows:

1. It does not support and non-web-based applications, it only supports web based applications.
2. You need to know at least one of the supported language very well in order to automate your application successfully.
3. No inbuilt reporting capability so you need plugins like JUnit and TestNG for test reports.
4. Lot of challenges with IE browser.

48- Have you ever heard of POC? What is POC?

Ans-Proof of Concept in Automation Testing

How we create POC for current project/ application? Why POC is important for Automation testing?

What is POC in Automation?

A POC stands for Proof of Concept. It is just a document or some time a demo piece of code, which

describe points, and some questions/answer.

In Other words- It simply words that we have to proof that what are doing and what will be outcome of the same.

Example- When we are running automation script for some testcase what will be total saving or effort you have saved via automation and so on.

How we create POC for current project/ application?

While creating POC documents/dem you should be ready with some Points.

1. What type of your application/project we are Automating?

Answer can be a web-based application or a desktop application or mobile application or some other type of application?

2. In your project, what type of testing tasks will be performed?

Answer can be smoke testing, regression testing and sometime E2E (End to End)test cases also etc.

3. Does your company have budget for same?

Does it be willing to go in for high-end commercial tools like QTP or would it only free tools (Selenium)?

4. Identify the task which has to be done before automation?

It can be setting up the setting up your application, test data, setting the project / setup of tools etc. After discussion of all above question, let us conclude below points

1. Select one or two suitable tool.

Important point-If it is a commercial tool then you could probably get a trial version of the same or ask demo for paid product and if it matches with your requirement then go for it. If it is free-ware then check the limitation of the tool and compare both tools.

2. For each tool, create a simple automation script for desired testing tasks and compare the result, pros and cons.

3. When you have the automation-working fine then you can present it to your manager, lead, or client for next action. If they agree then you can adapt the same

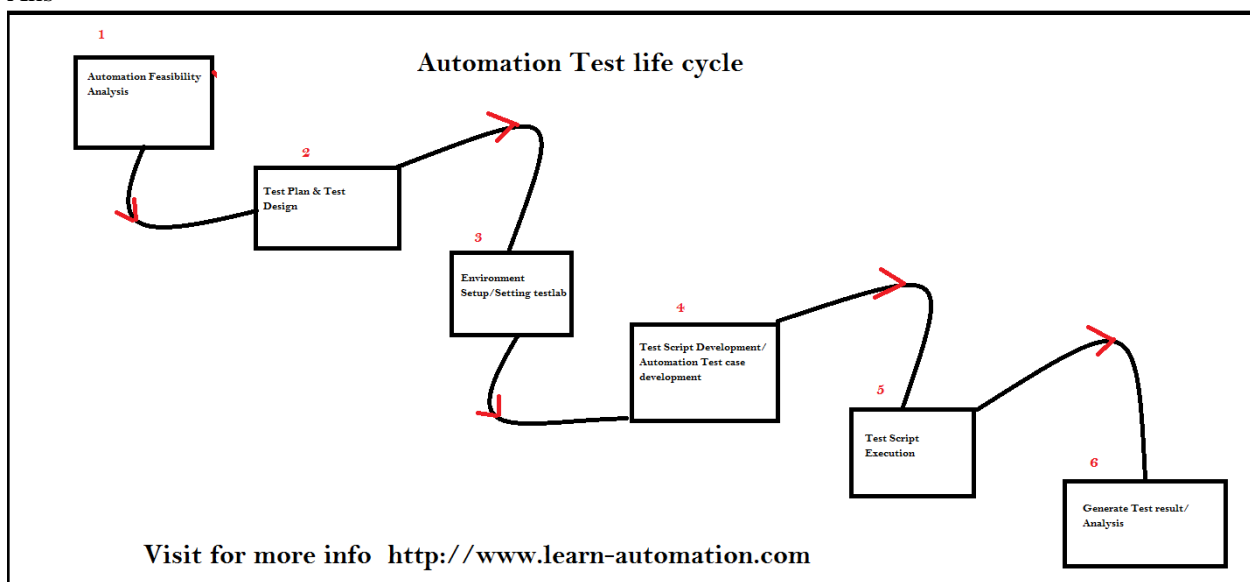
Proof of Concept in Automation Testing

POC document is very important for companies before moving to automation like why they should move to manual to automation. What will be saving- Effort saving/ Time saving/Money saving etc. Because automation tools require kind of programming knowledge. POC document will give result that whether we should move from Manual to Automation for the fowling project or not.

*****-----*****

49- What is ATLC (Automation Test life Cycle)?

Ans-



Let's start each of the phase of Automation test life cycle.

1- Automation feasibility analysis

In this section you have to think from different perspective. The main objective of this phase will be to check feasibility of automation.

- 1- Which test case can be automated and how we can automate them.
- 2- Which module of your application can be tested and which cannot be automated.
- 3- Which tools we can use for our application (like Selenium,QTP,Sahi,OATS, Telrik etc) and which tools will be best of our application.
- 4- Take following factors into consideration like Team size,Effort and cost involved for tools which we will use.

2- Test Plan/Test Design

This phase plays very important role in Automation test life cycle. In this phase you have to create a Test plan by considering below point into considerations.

- 1- Fetch all the manual test case from test management tool that which TC has to automate.
- 2- Which framework to use and what will be advantage and disadvantage of the framework which we will use.
- 3- Create a test suite for Automation test case in Test Management tool.
- 4- In test plan you can mention background, limitation, risk and dependency between application and tools.

7- Approval from client/ Stake holders.

3- Environment Setup/Test lab setup

By name itself you can understand that we need to setup machine or remote machine where our test case will execute.

- 1- In this section you can mention how many machine you want.
- 2- What should be the configuration in terms of hardware and software.

4-Test Script development/ Automation test case development

In this phase you have to start develop automation script and make sure all test script is running fine and should be stable enough.

- 1- Start creating test script based on your requirement
- 2- Create some common method or function that you can reuse throughout your script.
- 3- Make your script easy, reusable, well-structured and well documented so if third person check your script then he/she can understand your scripts easily.
- 4- Use better reporting so in case of failing you can trace your code
- 5- Finally review your script and your script should be ready before consumption.

5-Test script execution

Now its time for execution of test scripts, in this phase you have to execute all your test script.

Some points to remember while execution.

- 1- Your script should cover all the functional requirement as per testcase.
- 2- Your script should be stable so it should run in multiple environment and multiple browsers (depends on your requirement)
- 3- You can do batch execution also if possible so it will save time and effort.
- 4- In case of failure your script should take screen shots.

5- If test case is failing due to functionality, you have to raise a bug/defect.

6- Generate test result / Analyses of result

This is the last phase of Automation test life cycle in which we will gather test result and will share with team/client/stack holders.

1- Analyze the output and calculate how much time it takes to complete the testcase.

2- You should have good report generation like XSLT report, TestNG report, ReporterNG etc.

*****-----*****

50- What is Automation Test Plan?

Ans-

Automation test plan sections

1. Scope
2. Test strategy
3. Resources/roles and responsibilities
4. Tools
5. Schedules
6. Environment
7. Deliverables
8. Risks
9. Test data
10. Reports/results

Section #1: Scope

- Choose the test cases/scenarios that are to be regressed over and over across multiple cycles.
- Sometimes the simplest of test cases need lots of complicated solutions to be automated. If these are just for a one time use, it obviously does not make sense. Reusability should be your focus.
- Automation Testing does not/cannot perform exploratory testing.

Section #2: Test strategy

- This section is referred to as Framework in the automation world. Some frameworks are extremely challenging to create and also are effective – but time, effort and competency wise they are demanding. Always look for a middle ground and do the best you can without jeopardizing over utilization of resources.
- Decide on coding best practices to be used, naming conventions, locations for test assets to be stored, format of test results, etc. to maintain uniformity and increase productivity.

Section #3: Resources/roles and responsibilities

- The first step in this direction is to understand the team's capabilities and anticipate ahead the scope of automation coming into picture. This will help choose a team that suits both the automation and manual testing needs. Also, pick people who have the right attitude – those do not think that manual testing is beneath their stature.
- Choose a team well versed with AUT, test management, defect management and other SDLC activities
- Section #1: Scope

Section #4: Tools

Pick automation tools based on the following rules:

- Does the company already have licenses for a certain tool, try and see if you can use it?
- Look for open source (but reliable) tools
- Do the team members know the tool already or do we need to bring in someone new? Or train the existing ones?

Section #5: Schedules

- Include time for code-walkthroughs and inspection of the automation scripts
- Maintain the scripts on a timely basis. If you create a piece of code that you are not going to use for the next 6 months or so, make sure to periodically maintain it to lessen its chances of failure.

Section #6: Environment

- The target environment that your AUT is going to run and the automation tool that you want to use should be compatible. This is one of the factors to be considered pre-licensing for the tool.
- Also, analyze if the rest of the management tools in place and the automation tool you are trying to bring in are inter-connectible for additional benefit.

Section #7: Deliverables

- Your test scripts are your deliverables. However, not everyone is automation/programming language savvy. So, plan on creating a “How-to” document that will help the current users and future team members to be able to understand this script even when you are not around.
- Include comments in your script too.

Section #8: Risks

If you are going to propose an automation solution, be sure to choose cost effective tools and solutions to make sure that the automation endeavor does not burden the project. It is important to set the expectation that ROI for an automation project cannot be positive immediately but can be clearly seen over long periods of time.

Therefore, if you propose automating a system, pick the one that is

- Stable and not too much maintenance
- Has a scope for huge regression suites
- Does not have too much of manual intervention or does not depend of a human's intuition

Section #9: Test data

- Take into consideration the security aspects of the data
- Do not hard code any test data into the scripts. This just leads to too much script maintenance and might induce errors during modification.
- Be very specific. For a manual test step – ‘enter first name’, you can say enter any 5 character name. While testing, a tester can type “Swati” or “seela” or anything else. But for a tool it can’t make such suppositions. Therefore, provide exact values.

Section #10: Reports/results

- Script execution results are also technical and might not be easily understood by the rest of the teams. Plan on writing detailed results to notepad or excel sheets as an additional measure.
- Detailed framework documents, review results, defect reports, execution status reports are also expected.

Selenium Webdriver/RC/IDE Interview questions and Answers

All these question that we discussed now that is the combination of all level (Beginner, Advance). They will definitely ask so many questions from Framework itself and they will try to drag you in this topics because most of the people will stuck and interviewer will get to know that person has actually worked on Selenium or not.

Please make sure you are giving proper answer

52- Have you designed framework in your team or you are using existing framework, which already implemented by other members?

Ans-Yes. I have implemented and created framework in my team.

53- Can you create design of your framework?

Ans- Yes.

54- Which framework you have used and why?

Ans- I have created Data-Driven Framework with some capabilities of Page object model framework. After analyzing condition of project I recommend myself to make data driven framework. And after sometime added some functionalities. As per what I need. And the most important thing is that it easier maintain and create.

55- Can you create one sample script using your framework?

Ans- No. Because it needs to take time to make framework. And at same time I did not know all the things but I can.

*****_*****

There is again no limitation or specific question so be ready with any type of question but make sure whenever you are giving answer it should have valid point or you can directly say that I am not sure about it.

If you answer 6-7 out of 10 then it is enough.

We have more question for you, which you must know because we all have worked on TestNG with Selenium and it is one of my favorite Tool, which I have used it with Selenium and I really enjoyed a lot. Since TestNG is not a very big tool so do not worry about it and it has very good documentation on their official site so if you want to check then check out below link –

<http://testng.org/doc/documentation-main.html>

Here is a list of questions for TestNG

56- What is TestNG?

Ans: TestNG is a testing framework inspired from JUnit and NUnit, but introducing some new functionalities that make it more powerful and easier to use.

TestNG is an open source automated testing framework; where NG means Next Generation. TestNG is similar to JUnit (especially JUnit 4),

but it is not a JUnit extension. It is inspired by JUnit. It is designed to be better than JUnit, especially when testing integrated classes.

TestNG Features Supports annotations.

1. TestNG uses more Java and OO features.
2. Supports testing integrated classes (e.g., by default, no need to create a new test class instance for every test method).
3. Separates compile-time test code from run-time configuration/data info.
4. Flexible runtime configuration.

5. Introduces 'test groups'. Once you have compiled your tests, you can just ask TestNG to run all the "front-end" tests, or "fast", "slow", "database" tests, etc.

6. Supports Dependent test methods, parallel testing, load testing, and partial failure.

7. Flexible plug-in API.

8. Support for multi threaded testing.

*****-----*****

57- Why you have used TestNG in your framework? Can you compare JUNIT with TestNG framework?

Ans: Difference between testng and junit.

1. WebDriver has no native mechanism for generating reports.

2. TestNG can generate the report in a readable format

3. TestNG simplifies the way the tests are coded

4. There is no more need for a static main method in our tests.

The sequence of actions is regulated by easy-to-understand annotations that do not require methods to be static.

5. Uncaught exceptions are automatically handled by TestNG without terminating the test prematurely. These exceptions are reported as failed steps in the report. TestNG fares better than JUnit on following parameters.

1) Support for Annotations

Both the frameworks have support for annotations. In JUnit 4, the @BeforeClass and @AfterClass methods have to be declared as static.

TestNG does not have this constraint. TestNG has provided three additional setup/teardown pairs

for the suite, test and groups, i.e. @BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest,

@BeforeGroup and @AfterGroup.

2) Dependent Tests

We cannot create dependent tests in JUnit. It's possible only in TestNG.

TestNG uses "dependOnMethods" to implement the dependency testing as following.

@Test

```

public void test1()

{

System.out.println("This is test 1");

}

@Test(dependsOnMethods={"test1"})

public void test2()

{

System.out.println("This is test2");

}

```

The “test2()” will execute only if “test1()” is run successfully, else “test2()” will skip the test.

3) Parallel Execution of the tests

JUnit does not have inherent support for parallel execution. However, we can use maven-surefire-plugin and Gradle test task attribute maxParallelForks to execute the tests in parallel. TestNG has inherent support for parallelization by means of “parallel” and “thread-count” attributes in . @Test annotation has threadPoolSize attribute. JUnit is often shipped with mainstream IDEs by default, which contributes to its wider popularity. However, TestNG’s goal is much wider, which includes not only unit testing, but also support of integration and acceptance testing, etc. Which one is better or more suitable depends on use contexts and requirements.

4) In TestNG, Parameterized test configuration is very easy while It is very hard to configure

Parameterized test in JUnit.

5) TestNG support group test but it is not supported in JUnit.

6) TestNG support @BeforeTest, @AfterTest, @BeforeSuite, @AfterSuite, @BeforeGroups, @AfterGroups which are not supported in JUnit.

7) Test prioritization, Parallel testing is possible in TestNG. It is not supported by JUnit.

58- What are different annotation present in TestNG?

Here is the list of annotations that TestNG supports:

Annotation	Description
@BeforeSuite	The annotated method will be run only once before all tests in this suite have run.
@AfterSuite	The annotated method will be run only once after all tests in this suite have run.
@BeforeClass	The annotated method will be run only once before the first test method in the current class is invoked.
@AfterClass	The annotated method will be run only once after all the test methods in the current class have run.
@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.
@BeforeGroups	The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.
@AfterGroups	The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.
@BeforeMethod	The annotated method will be run before each test method.
@AfterMethod	The annotated method will be run after each test method.
@DataProvider	Marks a method as supplying data for a test method. The annotated method must return an Object[][], where each Object[] can be assigned the parameter list of the test method. The @Test method that wants to receive data from this DataProvider needs to use a dataProvider name equals to the name of this annotation.
@Factory	Marks a method as a factory that returns objects that will be used by TestNG as Test classes. The method must return Object[].
@Listeners	Defines listeners on a test class.
@Parameters	Describes how to pass parameters to a @Test method.
@Test	Marks a class or a method as a part of the test.

*****-----*****

59- What is priority feature in TestNG? In addition, how we can use this?

Ans:

1. In TestNG "Priority" is used to schedule the test cases. When there are multiple test cases, we want to execute test cases in order.
2. In order to achieve, we need to add annotation as `@Test(priority=?)`. The default value will be zero for priority.
3. If we define priority as "priority=", these test cases will get executed only when all the test cases which don't have any priority as the default priority will be set to "priority=0".

Eg. The below examples shows using the priority for test cases.

```
public class testNGPriorityExample {

    @Test
    public void registerAccount(){

        System.out.println("First register your account");
    }
    @Test(priority=2)
    public void sendEmail(){
        System.out.println("Send email after login");
    }
    @Test(priority=1)
    public void login(){
        System.out.println("Login to the account after registration");} }

    *****-----*****
```

60- What is dependsOnMethods and depends on group feature in TestNG?

Ans: DependsOnMethods: Sometimes it may be required for a test method to depend upon multiple other methods. This feature is very well supported by TestNG as part of the dependency support.

Code:

```
public class DependentTestExamples {

    @Test(dependsOnMethods = { "testTwo", "testThree" })
    public void testOne() {
        System.out.println("Test method one");
    }
    @Test
    public void testTwo() {
        System.out.println("Test method two");
    }

    @Test
    public void testThree() {
        System.out.println("Test method three");
    }
}
```

The preceding test class contains three test methods which print a message name onto the console when executed. Here test method testOne depends on test methods testTwo and testThree. This is configured by using the attribute dependsOnMethods while using the Test annotation.

Depends on group: Similar to dependent methods TestNG also allows test methods to depend on groups. This makes sure that a group of test methods get executed before the dependent test method.

Code:

```
public class DependentTestExamples{

    @Test(dependsOnGroups = { "test-group" })
    public void groupTestOne() {
        System.out.println("Group Test method one");
    }
    @Test(groups = { "test-group" })
    public void groupTestTwo() {
        System.out.println("Group test method two");
    }
    @Test(groups = { "test-group" })
    public void groupTestThree() {
        System.out.println("Group Test method three");}}}
```

The preceding test class contains two test methods which print a message name onto the console when executed. Here, test method testOne depends on test method testTwo. This is configured by using the attribute dependsOnMethods while using the Test annotation.

*****_*****

61- What is testng.xml file in TestNG?

Ans: In testng.xml file we can specify multiple name (s) which needs to be executed. In a project there may be many classes, but we want to execute only the selected classes. We can pass class names of multiple packages also. If say suppose, we want to execute two classes in one package and other class from some other package. The below is the example testng.xml which will execute the class names that are specified.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="example suite 1" verbose="1" >
<test name="Regression suite 1" >
<classes>
<class name="com.first.example.demoOne"/>
<class name="com.first.example.demoTwo"/>
<class name="com.second.example.demoThree"/>
</classes> </test>
</suite>
```

We need to specify the class names along with packages in between the classes tags. In the above xml, we have specified class name as “com.first.example.demoOne” and “com.first.example.demoOne” which are in “com.first.example” package. And class name demoThree is in package “com.second.example.” All the classes specified in the xml will get executes which have TestNG annotations. Below are the two example classes under “com.first.example” package which is executed.

```
public class demoOne {
    @Test
    public void firstTestCase(){
        System.out.println("im in first test case from demoOne Class");
    }
    @Test
    public void secondTestCase(){
        System.out.println("im in second test case from demoOne Class");} }
```

We need to run the testng.xml file. (Right click on testng.xml and select Run as ‘TestNG Suite’)

*****-----*****

62- How to group test cases in TestNG?

Ans: TestNG allows us to perform sophisticated groupings of test methods. Using TestNG can we can execute only set of groups while excluding another set. This gives us the maximum flexibility in divide tests and doesn't require us to recompile anything if you want to run two different sets of tests back to back. Groups are specified in testng.xml file and can be used either under the or tag. Groups specified in the tag apply to all the tags underneath.

Code:

```
public class groupExamples {

    @Test(groups="Regression")
    public void testCaseOne()
    {
        System.out.println("Im in testCaseOne - And in Regression Group");
    }
    @Test(groups="Regression")
    public void testCaseTwo(){
        System.out.println("Im in testCaseTwo - And in Regression Group");
    }
    @Test(groups="Smoke Test")
    public void testCaseThree(){
        System.out.println("Im in testCaseThree - And in Smoke Test Group");
    }
}
```

```

    }
    @Test(groups="Regression")
    public void testCaseFour(){
        System.out.println("Im in testCaseFour - And in Regression Group");
    }
}

```

The below is the XML file to execute, the test methods with group. We will execute the group “Regression” which will execute the test methods which are defined with group as “Regression”

```

<?xml version="1.0" encoding="UTF-8"?>
<suite name="Sample Suite">
<test name="testing">
<groups>
<run>
<include name="Regression"/>
</run>
</groups>
<classes>
<class name="com.example.group.groupExamples" />
</classes></test></suite>

```

*****-----*****

63- How to execute multiple test cases in Selenium?

Ans: TestNG provides an option to execute multiple tests in a single configuration file (testng.xml). It allows to divide tests into different parts and group them in a single tests. We can group all the tests related to database into one group, Regression tests in one group. And all the test cases related to Unit test cases into one group and so on. In testng.xml file we can specify multiple name (s) which needs to be executed. In a project there may be many classes, but we want to execute only the selected classes. We can pass class names of multiple packages also. If say suppose, we want to execute two classes in one package and other class from some other package.

Code:

```

<?xml version="1.0" encoding="UTF-8"?>
<suite name="Sample Suite" verbose="1" >
<test name="Unit Level Test" >
<classes>

```

```

<class name="com.easy.entry.AddTestCase" />
<class name="com.easy.entry.EditTestCase" />
</classes></test>
<test name="Regression Test">
<classes>
<class name="com.easy.records.AddUserTestCase" />
<class name="com.easy.records.DeleteUserTestCase" />
</classes></test></suite>
*****-----*****

```

64- How to execute parallel test cases in Selenium?

Ans: TestNG provides an ability to run test methods, test classes and tests in parallel. By using parallel execution, we can reduce the 'execution time' as tests are started and executed simultaneously in different threads. In testNG we can achieve parallel execution by two ways. One with testng.xml file and we can Configure an independent test method to run in multiple threads. First let us look at basic example for Parallel Execution of Test Methods using testng.xml. We will create a class with Two test methods and try to execute in different threads.

```

public class TestParallelOne {
@Test
public void testCaseOne() {
//Printing Id of the thread on using which test method got executed
System.out.println("Test Case One with Thread Id:- " + Thread.currentThread().getId());
}
@Test
public void testCaseTwo() {
//Printing Id of the thread on using which test method got executed
System.out.println("Test Case two with Thread Id:- " + Thread.currentThread().getId());
}}

```

The below is the simple testng.xml file, if you observe, we are defining two attributes 'parallel' and 'thread-count' at suite level. As we want test methods to be executed in parallel, we have provided 'methods'. And 'thread-count' attribute is to use to pass the number of maximum threads to be created.

```

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Parallel test suite" parallel="methods" thread-count="2">
<test name="Regression 1">

```

```

<classes>
<class name="com.parallel.TestParallelOne"/>
</classes>
</test>
</suite>
*****

```

65- What is TestNG listener?

Ans: A TestNG listener always extends the marker interface org.testng.ITestNGListener. Using listeners, one can extend TestNG in their dealings with notifications, reports and test behavior. Below are the listeners that TestNG provides:

IExecutionListener

IAnnotationTransformer

ISuiteListener

ITestListener

IConfigurationListener

IMethodInterceptor

IInvokedMethodListener

IHookable

IRReporter

```

*****

```

66- What is Data provider in TestNG?

Ans-An important features provided by TestNG is the **DataProvider feature**. It helps you to write data-driven tests, which essentially means that **same test method can be run multiple times with different data-sets**. Please note that DataProvider is the second way of passing parameters to test methods (first way we already discussed in [@Parameters example](#)). It helps in providing complex parameters to the test methods as it is not possible to do this from XML.

To use the DataProvider feature in your tests you have to declare a method annotated by `@DataProvider` and then use the said method in the test method using the '***dataProvider***' attribute in the Test annotation.

1) Using @DataProvider and Test in Same Class

The below test class contains a test method which takes one argument as input and prints it to console when executed. A DataProvider method is also available in the same class by using the `@DataProvider` annotation of TestNG. The name of the said DataProvider method is mentioned using the ***name*** attribute of the `@DataProvider` annotation. The DataProvider returns a double Object class array with two sets of data i.e. "data one" and "data two".

```
public class SameClassDataProvider
{
    @DataProvider(name = "data-provider")
    public Object[][] dataProviderMethod() {
        return new Object[][] { { "data one" }, { "data two" } };
    }

    @Test(dataProvider = "data-provider")
    public void testMethod(String data) {
        System.out.println("Data is: " + data);
    }
}
```

Now run above test. Output of above test run is given below:

```
Data is: data one
Data is: data two
PASSED: testMethod("data one")
PASSED: testMethod("data two")
```

2) Using @DataProvider and Test in Different Class

DataProvider.java

```
public class DataProviderClass
```

```

{
    @DataProvider(name = "data-provider")
    public static Object[][] dataProviderMethod()
    {
        return new Object[][] { { "data one" }, { "data two" } };
    }
}

```

TestClass.java

```

import org.testng.annotations.Test;

public class TestClass
{
    @Test(dataProvider = "data-provider", dataProviderClass =
DataProviderClass.class)
    public void testMethod(String data)
    {
        System.out.println("Data is: " + data);
    }
}

```

Now run above test. Output of above test run is given below:

```

Data is: data one
Data is: data two
PASSED: testMethod("data one")
PASSED: testMethod("data two")

```

*****_*****

67- How to disable particular test case?

Ans-Just add an attribute enabled=false in test declaration annotation.

Ex: @Test(enabled=false)

*****_*****

68- How to generate reports in TestNG?

Ans- We just need to run an annotated TestNG annotation scripts and refresh the project you can see the test-output folder is generated in project explorer. Just click on to it, and then click on to the emailable-report.html you can view the testNG report in HTML format

*****_*****

69- How to generate log in TestNG?

Ans-Reporter is a separate class in TestNG that is available under org.testng package. In Selenium you can specify steps, which we are performing so that we can check our output, and in case any issue we can debug at which point our test cases failed. We can see the logs in report.

Code:

```
public class ReporterDemo {

    @Test
    public void testReport(){

        WebDriver driver=new FirefoxDriver();

        Reporter.log("Browser Opened");

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

        Reporter.log("Browser Maximized");

        driver.get("http://www.google.com");

        Reporter.log("Application started");

        driver.quit();

        Reporter.log("Application closed");

    }

}

*****_____*****
```

70- How to execute only failed test cases in Selenium?

Ans-We can achieve this thing by using IRetryAnalyzer Interface.

Code:

```
// implement IRetryAnalyzer interface

public class Retry implements IRetryAnalyzer{

    // set counter to 0

    int minretryCount=0;

    // set maxcounter value this will execute our test 3 times
```

```

int maxretryCount=2;

// override retry Method

public boolean retry(ITestResult result) {

    // this will run until max count completes if test pass within this frame it will come out of for loop

    if(minretryCount<=maxretryCount)
    {

        // print the test name for log purpose

        System.out.println("Following test is failing===="+result.getName());

        // print the counter value

        System.out.println("Retrying the test Count is=== "+ (minretryCount+1));

        // increment counter each time by 1

        minretryCount++;

        return true;

    }

    return false;

}

}

```

Now we are done almost only we need to specify this in the test case.

```
@Test(retryAnalyzer=Retry.class)
```

In above statement, we are giving instruction to our test case that if the test case fails then it will call Retry class that we created above.

*****_*****

Now if an interviewer having good knowledge on Selenium and have worked on different tools then be ready with other question too that will be related to Selenium only.

We have so many tools in market that we can integrate with Selenium like Maven, Sikuli, Jenkins, AutoIT, Ant and so on.

I will try to summarize question-based on the tools, which I have used.

Maven

71- Can you please explain what is apache maven and Apache ant?

Ans- **Apache Maven:** Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Apache Ant: Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. Ant can also be used effectively to build non Java applications, for instance C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks.

*****_*****

72- Do you have used Maven project in your organization? If yes, then have you created build to execute your test?

Ans- No I don't create maven project for company. But I have created demo for it.

*****_*****

75- Can you please explain Maven life cycle?

Ans-

Maven Lifecycle and Phases

- The build lifecycle is the process of building and distributing an artifact.
- A phase is a step in the build lifecycle.
- Most important default phases:
 - Validate
 - Compile
 - Test
 - Package
 - Install
 - Deploy
- Some common phases not default:
 - Clean
 - Site



*****_*****

Sikuli

76- Have you heard of Sikuli? If yes, can you please explain what exactly Sikuli does?

Ans: Sikuli automates anything you see on the screen. It uses image recognition to identify and control GUI components. It is useful when there is no easy access to a GUI's internal or source code.

*****_*****

78- Advantage of Sikuli, Limitation of Sikuli.

Ans- 1. Sikuli is open source.

2. Sikuli IDE gives enough API's to interact with desktop based applications. Adding on to that we can use most of the API's of python scripting as well. Which totally depends upon how we code basically.

3. Image/Picture recognition in case of Sikuli is pretty much accurate. If we want more accuracy, we have to specify the regions of the images on the screen properly (This solves the problem of identification of small images on the screen).

4. Pretty much useful if the application demands so much of interaction from the user.

5. Behavior part of an application can be effectively analyzed.

6. We can easily identify the application crashes and bugs if we write script efficiently.

7. We can easily perform Boundary values testing of an application, which again depends on the scripting.

8. One of the biggest advantage of Sikuli is that, it can easily automate Flash objects.

9. makes easy to automate windows application.

Cons:

1. Running of batch wise sikuli scrips is little tricky and will not work some times in case we have 10 sikuli scripts and if we want to run them one by one automatically.

2. sikuli IDE hangs often and some time doesnt even gets opened untill we clear the registry.

3. sikuli is resolution dependent. (Which means the script written in 1366 * 768 screen resolution might not work in other resolutions).

4. sikuli scripting is platform dependent.

5. Error handling during run time has to be taken care properly in scripting if we run the scripts for long time.

*****-----*****

79- How to integrate Sikuli with Selenium script?

Ans-Step #1: Create a new Java Project in eclipse by clicking New -> Java project.

Step #2: Right click on the Project Go to Build Path -> Configure Build Path. Switch to Libraries Tab. Click on "Add External Jars" and Add Selenium library jars as well as Sikuli-scrip.jar

Step #3: Create a package inside src/ folder and create a class under that package.

Eg.

```
import org.junit.Test;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.firefox.FirefoxDriver;
```

```

import org.sikuli.script.App;
import org.sikuli.script.FindFailed;
import org.sikuli.script.Pattern;
import org.sikuli.script.Screen;

public class sikuliFirstTest {
    @Test
    public void functionName() throws FindFailed {

// Create a new instance of the Firefox driver
WebDriver driver = new FirefoxDriver();

// And now use this to visit Google
driver.get("http://www.google.com");
//Create and initialize an instance of Screen object
Screen screen = new Screen();
//Add image path
Pattern image = new Pattern("C:\\searchButton.png");
//Wait 10ms for image
screen.wait(image, 10);
//Click on the image
screen.click(image); }}

```

*****-----*****

80- Can you tell us the scenario where you have used Sikuli with Selenium?

Ans- 1) Here the scenario is to identify the “I’m Feeling Lucky” button and to click it using Sikuli.

Open Google home page from your browser and Capture “I’m Feeling Lucky” button using any screen capturing tool and save it to your local machine.

Note: Please don’t change the image by highlighting it or by editing. If you do so, then Sikuli may throw an error like “Can’t find image on the screen”.

*****-----*****

Jenkins

This is very vast topic and very interested but as an automation tester you will use it based on requirement

81- What is CI (Continuous integration) and what are different tools available in market.

Ans- **Continuous Integration** (CI) is a development practice that requires developers to **integrate** code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

Tools available for continues integration are as follow:

1. Jenkins
2. Buildbot
3. Travis CI
4. Strider
5. Go
6. Integrity

*****_*****

82- How to integrate Jenkins with Selenium?

Ans- 1. Open your web browser and then Navigate to Below URL <http://jenkins-ci.org> this is the official website of Jenkins.

2. Now download Jenkins.war file and save it.

3. Go to location where Jenkins.war is available.

4. Step 2- Open Command prompt knows as CMD and navigate till project home directory and Start Jenkins server Start- cmd> Project_home_Directory> java -jar jenkins.war

5. Open any browser and type the url <http://localhost:8080>

6. Click on > Manage Jenkins

7. Click on Configure System, Navigate to JDK section and Click on Add JDK button, Uncheck Install automatically check box so Jenkins will only take java which we have mention above.

8. Give the name as JAVA_HOME and Specify the JDK path

9. Part 3- Execute Selenium build using Jenkins

10.Part 4-Schedule your build in Jenkins for periodic execution

*****_*****

83- How to schedule test cases for nightly execution?

Ans- Open Task Scheduler by clicking the Start button Picture of the Start button, clicking Control Panel, clicking System and Security, clicking Administrative Tools, and then double-clicking Task Scheduler. Administrator permission required If you're prompted for an administrator password or confirmation, type the password or provide confirmation.

- Click the Action menu, and then click Create Basic Task.
- Type a name for the task and an optional description, and then click Next.
- Do one of the following:
 - To select a schedule based on the calendar, click Daily, Weekly, Monthly, or One time, click Next; specify the schedule you want to use, and then click Next.
 - To select a schedule based on common recurring events, click When the computer starts or When I log on, and then click Next.
 - To select a schedule based on specific events, click When a specific event is logged, click Next; specify the event log and other information using the drop-down lists, and then click Next.
 - To schedule a program to start automatically, click Start a program, and then click Next.
- Click Browse to find the program you want to start, and then click Next.
- Click Finish.

*****-----*****

84- Can you send email through Jenkins?

Ans- Yes.

- install Email-ext plugin
- Install Email-ext plugin at plug-in install page of Jenkins Configure System “Jenkins Location” section
- Enter valid email address to “System Admin e-mail address” “Extended E-mail Notification” section
- Enter your email address to “Default Recipients” “E-mail Notification” section
- Enter your SMTP server name to “SMTP server”
- Click “Advanced”
- Click “Use SMTP Authentication”
- Enter required information’s
- Check “Test configuration by sending test e-mail”
- Click “Test configuration” to send test email

- Click “Save” in the bottom of the page
- Configure a project to send email at every build
- Click “Add post-build action”
- Click “Editable Email Notification”
- Click “Advanced Settings...”
- Click “Add Trigger”
- Click “Always”
- Save
- Test-run
- Click “Build Now”
- Check Console output and received email

*****-----*****

85-Jenkins master slave concept and so on?

Ans- A "master" operating by itself is the basic installation of Jenkins and in this configuration the master handles all tasks for your build system. In most cases installing a slave doesn't change the behavior of the master. It will serve all HTTP requests, and it can still build projects on its own. Once you install a few slaves you might find yourself removing the executors on the master in order to free up master resources (allowing it to concentrate resources on managing your build environment) but this is not a necessary step. If you start to use Jenkins a lot with just a master you will most likely find that you will run out of resources (memory, CPU, etc.). At this point you can either upgrade your master or you can setup slaves to pick up the load. As mentioned above you might also need several different environments to test your builds. In this case using a slave to represent each of your required environments is almost a must.

A slave is a computer that is set up to offload build projects from the master and once setup this distribution of tasks is fairly automatic. The exact delegation behavior depends on the configuration of each project; some projects may choose to "stick" to a particular machine for a build, while others may choose to roam freely between slaves. For people accessing your Jenkins system via the integrated website (<http://yourjenkinsmaster:8080>), things work mostly transparently. You can still browse javadoc, see test results, download build results from a master, without ever noticing that builds were done by slaves. In other words, the master becomes a sort of "portal" to the entire build farm. Since each slave runs a separate program called a "slave agent" there is no need to install the full Jenkins (package or compiled binaries) on a slave. There are various ways to start slave agents, but in the end the slave agent

and Jenkins master needs to establish a bi-directional communication link (for example a TCP/IP socket.) in order to operate.

*****-----*****

Random Questions

86- What is robot class and where have we used this in Selenium?

Ans- robot class is a class of java programming. It uses to facilitate automated testing of java platform implementations. It provides no. of methods to fire windows keyboard and mouse events. Syntax and example are as follows:

Some commonly and popular used methods of Robot API during web automation:

-need to create object of robot class : Robot robot=new Robot();

- **keyPress(): Example:** robot.keyPress(KeyEvent.VK_DOWN) : This method with press down arrow key of Keyboard
- **mousePress() : Example :** robot.mousePress(InputEvent.BUTTON3_DOWN_MASK) : This method will press the right click of your mouse.
- **mouseMove() : Example:** robot.mouseMove(point.getX(), point.getY()) : This will move mouse pointer to the specified X and Y coordinates.
- **keyRelease() : Example:** robot.keyRelease(KeyEvent.VK_DOWN) : This method with release down arrow key of Keyboard
- **mouseRelease() : Example:** robot.mouseRelease(InputEvent.BUTTON3_DOWN_MASK) : This method will release the right click of your mouse

In certain Selenium Automation Tests, there is a need to control keyboard or mouse to interact with OS windows like Download pop-up, Alerts, Print Pop-ups, etc. or native Operation System applications like Notepad, Skype, Calculator, etc. Selenium Webdriver cannot handle these OS pop-ups/applications.

*****-----*****

87- Be ready with some basic Java programs which every automation Engg should know

Like string reverse, count the number of characters in a given string and so on.

If you will apply for Amazon, Flipkart then make sure you know Data structure very well because question level will be very high.

*****_*****

88- List down all locator with performance.

Ans-As per performance of locator from top to bottom.

id - *->Select element with the specified @id attribute.*

Name - *-> Select first element with the specified @name attribute.*

Linktext - *-> Select link (anchor tag) element which contains text matching the specified link text*

Partial Linktext - *-> Select link (anchor tag) element which contains text matching the specified partial link text*

Tag Name - *-> Locate Element using a Tag Name .*

Class name - *->Locate Element using a Tag Name ..*

Css - *->Select the element using css selectors. You can check here for [Css examples](#) and You can also refer [W3C CSS Locatros](#)*

Xpath - *->Locate an element using an XPath expression.*

*****_*****

89- Find broken links on Web page.

Ans: **Scenario for find broken links using selenium**

Before jumping to the code let's take one simple example to get the actual concept.

Example1- Suppose we have one application which contains 400 links and we need to verify the link is broken or not.

Program for finding broken links

```
public class VerifyLinks {  
  
    public static void main(String[] args)  
  
    {  
  
        WebDriver driver=new FirefoxDriver();  
  
        driver.manage().window().maximize();  
  
        driver.get("http://www.google.co.in/");  
  
        List<WebElement> links=driver.findElements(By.tagName("a"));  
  
        System.out.println("Total links are "+links.size());  
    }  
}
```

```

        for(int i=0;i<links.size();i++)
        {
            WebElement ele= links.get(i);

            String url=ele.getAttribute("href");

            verifyLinkActive(url);

        }
    }

    public static void verifyLinkActive(String linkUrl){

    try
    {

        URL url = new URL(linkUrl);

        HttpURLConnection httpURLConnection=(HttpURLConnection)url.openConnection();

        httpURLConnection.setConnectTimeout(3000);

        httpURLConnection.connect();

        if(httpURLConnection.getResponseCode()==200)
        {

            System.out.println(linkUrl+" - "+httpURLConnection.getResponseMessage());

        }

        if(httpURLConnection.getResponseCode()==HttpURLConnection.HTTP_NOT_FOUND)

        {

            System.out.println(linkUrl+" - "+httpURLConnection.getResponseMessage() + " - "+
            HttpURLConnection.HTTP_NOT_FOUND);

        }catch(Exception e){}}

    }
}

*****_*****

```

90- How to read write excel files using Apache POI.

Ans: Basics of APACHE POI

There are two main prefixes which you will encounter when working with Apache POI:

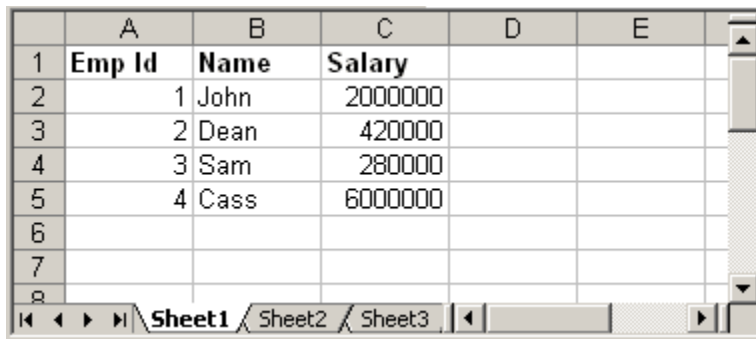
- **HSSF**: denotes the API is for working with Excel 2003 and earlier.
- **XSSF**: denotes the API is for working with Excel 2007 and later.

And to get started the Apache POI API, you just need to understand and use the following 4 interfaces:

- **Workbook**: high level representation of an Excel workbook. Concrete implementations are: **HSSFWorkbook** and **XSSFWorkbook**.
- **Sheet**: high level representation of an Excel worksheet. Typical implementing classes are **HSSFSheet** and **XSSFSheet**.
- **Row**: high level representation of a row in a spreadsheet. **HSSFRow** and **XSSFRow** are two concrete classes.
- **Cell**: high level representation of a cell in a row. **HSSFCell** and **XSSFCell** are the typical implementing classes.

Now, let's walk through some real-life examples.

Reading from Excel File Examples



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E
1	Emp Id	Name	Salary		
2	1	John	2000000		
3	2	Dean	420000		
4	3	Sam	280000		
5	4	Cass	6000000		
6					
7					
8					

The spreadsheet has three sheets: Sheet1, Sheet2, and Sheet3. Sheet1 is the active sheet.

```
try {
```

```
    FileInputStream file = new FileInputStream(new File("C:\\test.xls"));
```

```
    //Get the workbook instance for XLS file
```

```
    HSSFWorkbook workbook = new HSSFWorkbook(file);
```

```
    //Get first sheet from the workbook
```

```
    HSSFSheet sheet = workbook.getSheetAt(0);
```

```
    //Iterate through each rows from first sheet
```

```
    Iterator<Row> rowIterator = sheet.iterator();
```

```
    while(rowIterator.hasNext()) {
```

```

Row row = rowIterator.next();

//For each row, iterate through each columns
Iterator<Cell> cellIterator = row.cellIterator();
while(cellIterator.hasNext()) {

    Cell cell = cellIterator.next();

    switch(cell.getCellType()) {
        case Cell.CELL_TYPE_BOOLEAN:
            System.out.print(cell.getBooleanCellValue() + "\t\t");
            break;
        case Cell.CELL_TYPE_NUMERIC:
            System.out.print(cell.getNumericCellValue() + "\t\t");
            break;
        case Cell.CELL_TYPE_STRING:
            System.out.print(cell.getStringCellValue() + "\t\t");
            break;
    }
}
System.out.println("");
}
file.close();
FileOutputStream out =
    new FileOutputStream(new File("C:\\test.xls"));
workbook.write(out);
out.close();

} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

```

Create new Excel File

```

import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
//..
HSSFWorkbook workbook = new HSSFWorkbook();
HSSFSheet sheet = workbook.createSheet("Sample sheet");

```

```
//Create a new row in current sheet
Row row = sheet.createRow(0);
//Create a new cell in current row
Cell cell = row.createCell(0);
//Set value to new value
cell.setCellValue("Blahblah");
```

Write data into the Excel File

```
HSSFWorkbook workbook = new HSSFWorkbook();
HSSFSheet sheet = workbook.createSheet("Sample sheet");

Map<String, Object[]> data = new HashMap<String, Object[]>();
data.put("1", new Object[] { "Emp No.", "Name", "Salary" });
data.put("2", new Object[] { 1d, "John", 1500000d });
data.put("3", new Object[] { 2d, "Sam", 800000d });
data.put("4", new Object[] { 3d, "Dean", 700000d });

Set<String> keyset = data.keySet();
int rownum = 0;
for (String key : keyset) {
    Row row = sheet.createRow(rownum++);
    Object [] objArr = data.get(key);
    int cellnum = 0;
    for (Object obj : objArr) {
        Cell cell = row.createCell(cellnum++);
        if(obj instanceof Date)
            cell.setCellValue((Date)obj);
        else if(obj instanceof Boolean)
            cell.setCellValue((Boolean)obj);
        else if(obj instanceof String)
            cell.setCellValue((String)obj);
        else if(obj instanceof Double)
            cell.setCellValue((Double)obj);
    }
}

try {
    FileOutputStream out =
        new FileOutputStream(new File("C:\\new.xls"));
    workbook.write(out);
    out.close();
    System.out.println("Excel written successfully..");
}
```

```

} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

```

*****-----*****

91- Have you ever done connection with Database using JDBC?

Ans-Yes. Used MySQL as database.

*****-----*****

92- Read CSV files.

Ans-Program to read csv file.

```

package blog;
import java.io.FileReader;
import java.util.Iterator;
import java.util.List;

import au.com.bytecode.opencsv.CSVReader;

public class ReadCsvFiles {

    public static void main(String[] args) throws Exception {

        // This will load csv file
        CSVReader reader = new CSVReader(new
        FileReader("C:\\Users\\mukesh_otwani\\Desktop\\demo.csv"));

        // this will load content into list
        List<String[]> li=reader.readAll();
        System.out.println("Total rows which we have is "+li.size());

        // create Iterator reference
        Iterator<String[]>i1= li.iterator();

        // Iterate all values
        while(i1.hasNext()){

            String[] str=i1.next();

            System.out.print(" Values are ");

            for(int i=0;i<str.length;i++)
            {

                System.out.print(" "+str[i]);

```



```

}
System.out.println(" ");

}

}

}
*****-----*****

```

93- How to read properties files in Selenium?

Ans- Program for reading properties file.

```

public class ReadFileData {
public static void main(String[] args) {
File file = new File("D:/Dev/ReadData/src/datafile.properties");
FileInputStream fileInput = null; try
{
    fileInput = new FileInputStream(file);
} catch (FileNotFoundException e) { e.printStackTrace(); }
Properties prop = new Properties(); //load properties file try
{
    prop.load(fileInput);
} catch (IOException e) { e.printStackTrace(); }
WebDriver driver = new FirefoxDriver();
driver.get(prop.getProperty("URL"));
driver.findElement(By.id("Email")).sendKeys(prop.getProperty("username"));
driver.findElement(By.id("Passwd")).sendKeys(prop.getProperty("password"));
driver.findElement(By.id("SignIn")).click();
System.out.println("URL ::" + prop.getProperty("URL")); System.out.println("User name::"
+prop.getProperty("username"));
System.out.println("Password::" +prop.getProperty("password"));
} }

```

The below is the Output after executing the program: We are passing the properties values to the webdriver and printing the values at end

URL ::http://gmail.com

User name::testuser

Password::password123

*****_*****

94- Does Selenium support mobile automation?

Ans-using Appium or other automation tool (need to confirm more on it.)

*****_*****

95- What is Selendroid?

Ans-Selendroid can be used to test already built apps. Those Android apps (apk file) must exist on the machine, where the *selendroid-standalone* server will be started. The reason for this is that a customized *selendroid-server* for the app under test (AUT) will be created. Both apps (selendroid-server and AUT) must be signed with the same certificate in order to install the apks on the device.

*****_*****

96- What is Appium?

Ans- Appium aims to automate any mobile app from any language and any test framework, with full access to back-end APIs and DBs from test code. Write tests with your favorite dev tools using all the above programming languages, and probably more (with the Selenium WebDriver API and language-specific client libraries).

*****_*****

97- What is same origin policy in Selenium?

Ans-First of all “**Same Origin Policy**” is introduced for security reason, and it ensures that content of your site will never be accessible by a script from another site. As per the policy, any code loaded within the browser can only operate within that website’s domain.

Same Origin policy prohibits JavaScript code from accessing elements from a domain that is different from where it was launched. Example, the HTML code in www.google.com uses a JavaScript program "**testScript.js**". The **same origin policy** will only allow **testScript.js** to access pages withingoogle.com such as **google.com/mail**, **google.com/login**, or **google.com/signup**. However, it cannot access pages from different sites such as **yahoo.com/search** or **fbk.com** because they belong to different domains.

To avoid “**Same Origin Policy**” proxy injection method is used, in proxy injection mode the **Selenium Server** acts as a client configured HTTP proxy , which sits between the browser and application under test and then masks the **AUT** under a fictional URL

Selenium uses java script to drives tests on a browser; Selenium injects its own js to the response which is returned from aut. But there is a java script security restriction (**same origin policy**) which lets you modify html of page using js only if js also originates from the same domain as html. This security

restriction is of utmost important but spoils the working of Selenium. This is where Selenium server comes to play an important role.

98- What is Selenium grid, hub, node and commands that used in Selenium Grid?

Ans- **Selenium Grid:** Selenium Grid is a tool that distributes the tests across multiple physical or virtual machines so that we can execute scripts in parallel (simultaneously). It dramatically accelerates the testing process across browsers and across platforms by giving us quick and accurate feedback.

Hub: The hub can also be understood as a server which acts as the central point where the tests would be triggered. A Selenium Grid has only one Hub and it is launched on a single machine once.

Node: Nodes are the Selenium instances that are attached to the Hub which execute the tests. There can be one or more nodes in a grid which can be of any OS and can contain any of the Selenium supported browsers.

99-What is the difference between / and // in XPATH?

Ans- “/” It’s starts search selection from root element in document. (absolute path)

“//” It start selection from anywhere in XML document. (relative path)

100- Can we automate Flash application in Selenium?

Ans-The straightforward answer to it is that Selenium has no interface to interact with your Flash content. Flash files are programmed in ActionScript and are very similar to JavaScript. Flash files also contains programming Elements just like other languages. For e.g. they too have buttons, text box etc. Interacting with these elements cause the Flash file to call some internal methods to do the task.

It’s a fairly simple process. All you have to do is use the ExternalInterface class. We will start it with a small example. This is a small flash application which contains 3 buttons. Add, Subtract and Multiply. These three buttons when clicked sends a text to a Text control and write down what the name of the button is. So if you click Add you will find text “Add” being added to the Text control.
