Q #1) What is Automation Testing?
Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves use to a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn't require any manual intervention.

Q #2) What are the benefits of Automation Testing?
Benefits of Automation testing are:

1. Supports execution of repeated test cases
2. Aids in testing a large test matrix
3. Enables parallel execution
4. Encourages unattended execution
5. Improves accuracy thereby reducing human generated errors
6. Saves time and money

Q #3) Why should Selenium be selected as a test tool?
Selenium

1. is free and open source
2. have a large user base and helping communities
3. have cross Browser compatibility (Firefox, chrome, Internet Explorer, Safari etc.)
4. have great platform compatibility (Windows, Mac OS, Linux etc.)
5. supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. has fresh and regular repository developments
7. supports distributed testing

Q #4) What is Selenium? What are the different Selenium components?
Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.
Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.
The suite package constitutes of the following sets of tools:

- **Selenium Integrated Development Environment (IDE)** – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plugin.
- **Selenium Remote Control (RC)** – Selenium RC is a server that allows user to create test scripts in a desired programming language. It also allows executing test scripts within the large spectrum of browsers.
- **Selenium WebDriver** – WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.

- **Selenium Grid** – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

Q #5) What are the testing types that can be supported by Selenium?
Selenium supports the following types of testing:

1. Functional Testing
2. Regression Testing

Q #6) What are the limitations of Selenium?
Following are the limitations of Selenium:

- Selenium supports testing of only web based applications
- Mobile applications cannot be tested using Selenium
- Captcha and Bar code readers cannot be tested using Selenium
- Reports can only be generated using third party tools like TestNG or Junit.
- As Selenium is a free tool, thus there is no ready vendor support though the user can find numerous helping communities.
- User is expected to possess prior programming language knowledge.

Q #7)What is the difference between Selenium IDE, Selenium RC and WebDriver?

Feature	Selenium IDE	Selenium RC	WebDriver
Feature	Selenium IDE	Selenium RC	WebDriver
Browser Compatibility	Selenium IDE comes as a Firefox plugin, thus it supports only Firefox	Selenium RC supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera	WebDriver supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera. Also supports HtmlUnitDriver which is a GUI less or headless browser.
Record and Playback	Selenium IDE supports record and playback feature	Selenium RC doesn't supports record and playback feature	WebDriverdoesn't support record and playback feature
Server Requirement	Selenium IDE doesn't require any server to be started	Selenium RC requires server to be started before	WebDriver doesn't require any server to be started before

Feature	Selenium IDE	Selenium RC	WebDriver
Feature	Selenium IDE	Selenium RC	WebDriver
	before executing the test scripts	executing the test scripts	executing the test scripts
Architecture	Selenium IDE is a Javascript based framework	Selenium RC is a JavaScript based Framework	WebDriver uses the browser's native compatibility to automation
Object Oriented	Selenium IDE is not an object oriented tool	Selenium RC is semi object oriented tool	WebDriver is a purely object oriented tool
Dynamic Finders (for locating web elements on a webpage)	Selenium IDE doesn't support dynamic finders	Selenium RC doesn't support dynamic finders	WebDriver supports dynamic finders
Handling Alerts, Navigations, Dropdowns	Selenium IDE doesn't explicitly provides aids to handle alerts, navigations, dropdowns	Selenium RC doesn't explicitly provides aids to handle alerts, navigations, dropdowns	WebDriver offers a wide range of utilities and classes that helps in handling alerts, navigations, and dropdowns efficiently and effectively.
WAP (iPhone/Android) Testing	Selenium IDE doesn't support testing of iPhone/Andriod applications	Selenium RC doesn't support testing of iPhone/Andriod applications	WebDriver is designed in a way to efficiently support testing of iPhone/Android applications. The tool comes with a large range of drivers for WAP based testing. For example, AndroidDriver, iPhoneDriver
Listener Support	Selenium IDE	Selenium RC	WebDriver

Feature	Selenium IDE	Selenium RC	WebDriver
Feature	Selenium IDE	Selenium RC	WebDriver
	doesn't support listeners	doesn't support listeners	supports the implementation of Listeners
Speed	Selenium IDE is fast as it is plugged in with the web-browser that launches the test. Thus, the IDE and browser communicates directly	Selenium RC is slower than WebDriver as it doesn't communicates directly with the browser; rather it sends selenese commands over to Selenium Core which in turn communicates with the browser.	WebDriver communicates directly with the web browsers. Thus making it much faster.

Q #8) When should I use Selenium IDE?
Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

Q #9) What is Selenese?
Selenese is the language which is used to write test scripts in Selenium IDE.

Q #10)What are the different types of locators in Selenium?
Locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have different types of locators in Selenium:

- ID
- ClassName
- Name
- TagName
- LinkText
- PartialLinkText
- Xpath
- CSS Selector
- DOM

Q #11)What is difference between assert and verify commands?
Assert: Assert command checks whether the given condition is true or false. Let's say we assert whether the given element is present on the web page or not. If the condition is true then the program control will execute the next test step but if the condition is false, the execution would stop and no further test would be executed.
Verify: Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn't halts i.e. any failure during verification would not stop the execution and all the test steps would be executed.

Q #12) What is an Xpath?
Xpath is used to locate a web element based on its XML path. XML stands for Extensible Markup Language and is used to store, organize and transport arbitrary data. It stores data in a key-value pair which is very much similar to HTML tags. Both being markup languages and since they fall under the same umbrella, Xpath can be used to locate HTML elements.
The fundamental behind locating elements using Xpath is the traversing between various elements across the entire page and thus enabling a user to find an element with the reference of another element.

Q #13) What is the difference between "/" and "//" in Xpath?

Single Slash "/" - Single slash is used to create Xpath with absolute path i.e. the xpath would be created to start selection from the document node/start node.
Double Slash "//" - Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document.

Q #14) What is Same origin policy and how it can be handled?
The problem of same origin policy disallows to access the DOM of a document from an origin that is different from the origin we are trying to access the document.
Origin is a sequential combination of scheme, host and port of the URL. For example, for a URL http:// http://www.softwaretestinghelp.com/resources/, the origin is a combination of http, softwaretestinghelp.com, 80 correspondingly.
Thus the Selenium Core (JavaScript Program) cannot access the elements from an origin that is different from where it was launched. For Example, if I have launched the JavaScript Program from "http://www.softwaretestinghelp.com", then I would be able to access the pages within the same domain such as "http://www.softwaretestinghelp.com/resources" or "http://www.softwaretestinghelp.com/istqb-free-updates/". The other domains like google.com, seleniumhq.org would no more be accessible.
So, In order to handle same origin policy, Selenium Remote Control was introduced.

Q #15)When should I use Selenium Grid?

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

Q #16) What do we mean by Selenium 1 and Selenium 2?
Selenium RC and WebDriver, in a combination are popularly known as Selenium 2. Selenium RC alone is also referred as Selenium 1.

Q #17) How do I launch the browser using WebDriver?
The following syntax can be used to launch Browser:
*WebDriver driver = **new** FirefoxDriver();*
*WebDriver driver = **new** ChromeDriver();*
*WebDriver driver = **new** InternetExplorerDriver();*

Q #18) What are the different types of Drivers available in WebDriver?
The different drivers available in WebDriver are:

- FirefoxDriver
- InternetExplorerDriver
- ChromeDriver
- SafariDriver
- OperaDriver
- AndroidDriver
- IPhoneDriver
- HtmlUnitDriver

Q #19)How to type in a textbox using Selenium?

User can use sendKeys("String to be entered") to enter the string in the textbox.
Syntax:
WebElement username = drv.findElement(By.id("Email"));
// entering username
username.sendKeys("sth");

Q #20)How can you find if an element in displayed on the screen?

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

1. isDisplayed()
2. isSelected()
3. isEnabled()

Syntax:
isDisplayed():
boolean*buttonPresence = driver.findElement(By.id("gbqfba")).isDisplayed();*
isSelected():
boolean*buttonSelected = driver.findElement(By.id("gbqfba")).isDisplayed();*

isEnabled():
boolean*searchIconEnabled = driver.findElement(By.id("gbqfb")).isEnabled();*

Q #21)How can we get a text of a web element?

Get command is used to retrieve the inner text of the specified web element. The command doesn't require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.
Syntax:
String Text = driver.findElement(By.id("Text")).getText();

Q #22) How to select value in a dropdown?
Value in the drop down can be selected using WebDriver's Select class.
Syntax:
selectByValue:
*Select selectByValue = **new** Select(driver.findElement(By.id("SelectID_One")));*
selectByValue.selectByValue("greenvalue");
selectByVisibleText:
*Select selectByVisibleText = **new** Select (driver.findElement(By.id("SelectID_Two")));*
selectByVisibleText.selectByVisibleText("Lime");
selectByIndex:
*Select selectByIndex = **new** Select(driver.findElement(By.id("SelectID_Three")));*
selectByIndex.selectByIndex(2);

Q #23) What are the different types of navigation commands?
Following are the navigation commands:
navigate().back() – The above command requires no parameters and takes back the user to the previous webpage in the web browser's history.
Sample code:
driver.navigate().back();
navigate().forward() – This command lets the user to navigate to the next web page with reference to the browser's history.
Sample code:
driver.navigate().forward();
navigate().refresh() – This command lets the user to refresh the current web page there by reloading all the web elements.
Sample code:
driver.navigate().refresh();
navigate().to() – This command lets the user to launch a new web browser window and navigate to the specified URL.
Sample code:
driver.navigate().to("https://google.com");

Q #24) How to click on a hyper link using linkText?

driver.findElement(By.linkText("Google")).click();
The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.
The above mentioned link can also be accessed by using the following command.
driver.findElement(By.partialLinkText("Goo")).click();
The above command find the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element with the specified substring and then clicks on it.

Q #25)How to handle frame in WebDriver?
An inline frame acronym as iframe is used to insert another document with in the current HTML document or simply a web page into a web page by enabling nesting.
Select iframe by id
driver.switchTo().frame("ID of the frame");
Locating iframe using tagName
driver.switchTo().frame(driver.findElements(By.tagName("iframe").get(0));
Locating iframe using index
frame(index)
driver.switchTo().frame(0);

frame(Name of Frame)
driver.switchTo().frame("name of the frame");
frame(WebElement element)
Select Parent Window
driver.switchTo().defaultContent();
Q #26) When do we use findElement() and findElements()?
findElement(): findElement() is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.
Syntax:
WebElement element =
driver.findElements(By.xpath("//div[@id='example']//ul//li"));
findElements(): findElements() is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of WebElements.
Syntax:
List <WebElement>elementList =
driver.findElements(By.xpath("//div[@id='example']//ul//li"));

Q #27)How to find more than one web element in the list?
At times, we may come across elements of same type like multiple hyperlinks, images etc arranged in an ordered or unordered list. Thus, it makes absolute sense to deal with such elements by a single piece of code and this can be done using WebElement List.
Sample Code

```
1 // Storing the list
2 List <WebElement>elementList =
  driver.findElements(By.xpath("//div[@id='example']//ul//li"));
3 // Fetching the size of the list
4 intlistSize = elementList.size();
5 for(inti=0; i<listSize; i++)
6 {
7 // Clicking on each service provider link
8 serviceProviderLinks.get(i).click();
9  // Navigating back to the previous page that stores link to service providers
10 driver.navigate().back();
11 }
```

Q #28) What is the difference between driver.close() and driver.quit command?
close(): WebDriver's close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does is return any value.
quit(): Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does is return any value.

Q #29) Can Selenium handle windows based pop up?
Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

Q #30) How can we handle web based pop up?
WebDriver offers the users with a very efficient way to handle these pop ups using Alert interface. There are the four methods that we would be using along with the Alert interface.

- void dismiss() – The accept() method clicks on the "Cancel" button as soon as the pop up window appears.
- void accept() – The accept() method clicks on the "Ok" button as soon as the pop up window appears.
- String getText() – The getText() method returns the text displayed on the alert box.
- voidsendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

Syntax:

```
// accepting javascript alert
        Alertalert = driver.switchTo().alert();
alert.accept();
```

Q #31) How can we handle windows based pop up?
Selenium is an automation testing tool which supports only web application testing, that means, it doesn't support testing of windows based applications. However Selenium alone can't help the situation but along with some third party intervention, this problem can be overcome. There are several third party tools available for handling window based pop ups along with the selenium like AutoIT, Robot class etc.

Q #32) How to assert title of the web page?
//verify the title of the web page
assertTrue("The title of the window is incorrect.",driver.getTitle().equals("Title of the page"));

Q #33) How to mouse hover on a web element using WebDriver?
WebDriver offers a wide range of interaction utilities that the user can exploit to automate mouse and keyboard events. Action Interface is one such utility which simulates the single user interactions.
Thus, In the following scenario, we have used Action Interface to mouse hover on a drop down which then opens a list of options.
Sample Code:

```
1 // Instantiating Action Interface
2 Actions actions=newActions(driver);
3 // howering on the dropdown
4 actions.moveToElement(driver.findElement(By.id("id of the dropdown"))).perform();
5 // Clicking on one of the items in the list options
6 WebElementsubLinkOption=driver.findElement(By.id("id of the sub link"));
7 subLinkOption.click();
```

Q #34) How to capture screenshot in WebDriver?

```
importorg.junit.After;
importorg.junit.Before;
importorg.junit.Test;
importjava.io.File;
 importjava.io.IOException;
 importorg.apache.commons.io.FileUtils;
 importorg.openqa.selenium.OutputType;
 importorg.openqa.selenium.TakesScreenshot;
 importorg.openqa.selenium.WebDriver;
                        importorg.openqa.selenium.firefox.FirefoxDriver;

 publicclassCaptureScreenshot {
```

```
  WebDriver driver;
  @Before
  publicvoidsetUp() throwsException {
      driver = newFirefoxDriver();
      driver.get("https://google.com");
  }
  @After
  publicvoidtearDown() throwsException {
      driver.quit();
  }

  @Test
  publicvoidtest() throwsIOException {
      // Code to capture the screenshot
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
      // Code to copy the screenshot in the desired location
FileUtils.copyFile(scrFile, newFile("C:\\CaptureScreenshot\\google.jpg"));
  }
}
```

Q #35) What is Junit?
Junit is a unit testing framework introduced by Apache. Junit is based on Java.

Q #36) What are Junit annotations?
Following are the Junit Annotations:

- **@Test:** Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
- **@Before:** Method annotated as @Before lets the system know that this method shall be executed every time before each of the test method.
- **@After:** Method annotated as @After lets the system know that this method shall be executed every time after each of the test method.
- **@BeforeClass:** Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test method.
- **@AfterClass:** Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test method.
- **@Ignore:** Method annotated as @Ignore lets the system know that this method shall not be executed.

Q #37)What is TestNG and how is it better than Junit?

TestNG is an advance framework designed in a way to leverage the benefits by both the developers and testers. With the commencement of the frameworks, JUnit gained an enormous popularity across the Java applications, Java developers and Java testers with remarkably increasing the code quality. Despite being easy to use and straightforward, JUnit has its own limitations which give rise to the need of bringing TestNG into the picture. TestNG is an open source framework which is distributed under the Apache software License and is readily available for download.
TestNG with WebDriver provides an efficient and effective test result format that can in turn be shared with the stake holders to have a glimpse on the product's/application's health thereby eliminating the drawback of WebDriver's incapability to generate test reports. TestNG has an inbuilt exception handling mechanism which lets the program to run without terminating unexpectedly.
There are various advantages that make TestNG superior to JUnit. Some of them are:

- Added advance and easy annotations
- Execution patterns can set
- Concurrent execution of test scripts
- Test case dependencies can be set

Q #38)How to set test case priority in TestNG?
Setting Priority in TestNG Code Snippet

```
packageTestNG;
importorg.testng.annotations.*;
publicclassSettingPriority {
    @Test(priority=0)
    publicvoidmethod1() {
    }
    @Test(priority=1)
    publicvoidmethod2() {
    }
    @Test(priority=2
)
    publicvoidmethod3() {
    }
}
```

Test Execution Sequence:

1. Method1
2. Method2
3. Method3

Q #39) What is a framework?

Framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing.

Q #40)What are the advantages of Automation framework?
Advantage of Test Automation framework

- Reusability of code
- Maximum coverage
- Recovery scenario
- Low cost maintenance
- Minimal manual intervention
- Easy Reporting

Q #41) What are the different types of frameworks?
Below are the different types of frameworks:

1. **Module Based Testing Framework:** The framework divides the entire "Application Under Test" into number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts taken together builds a larger test script representing more than one module.
2. **Library Architecture Testing Framework:** The basic fundamental behind the framework is to determine the common steps and group them into functions under a library and call those functions in the test scripts whenever required.
3. Data Driven Testing Framework: Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The data is conventionally stored in "Key-Value" pairs. Thus, the key can be used to access and populate the data within the test scripts.
4. **Keyword Driven Testing Framework:** The Keyword driven testing framework is an extension to Data driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.
5. **Hybrid Testing Framework:** Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.
6. **Behavior Driven Development Framework:** Behavior Driven Development framework allows automation of functional validations in easily readable and understandable format to Business Analysts, Developers, Testers, etc.

Q #42) How can I read test data from excels?
Test data can efficiently be read from excel using JXL or POI API.

Q #43) What is the difference between POI and jxl jar?

#	JXL jar	POI jar

#	JXL jar	POI jar
1	JXL supports ".xls" format i.e. binary based format. JXL doesn't support Excel 2007 and ".xlsx" format i.e. XML based format	POI jar supports all of these formats
2	JXL API was last updated in the year 2009	POI is regularly updated and released
3	The JXL documentation is not as comprehensive as that of POI	POI has a well prepared and highly comprehensive documentation
4	JXL API doesn't support rich text formatting	POI API supports rich text formatting
5	JXL API is faster than POI API	POI API is slower than JXL API

Q #44)What is the difference between Selenium and QTP?

Feature	Selenium	Quick Test Professional (QTP)
Browser Compatibility	Selenium supports almost all the popular browsers like Firefox, Chrome, Safari, Internet Explorer, Opera etc	QTP supports Internet Explorer, Firefox and Chrome. QTP only supports Windows Operating System
Distribution	Selenium is distributed as an open source tool and is freely available	QTP is distributed as a licensed tool and is commercialized
Application under Test	Selenium supports testing of only web based applications	QTP supports testing of both the web based application and windows based application
Object Repository	Object Repository needs to be created as a separate entity	QTP automatically creates and maintains Object Repository
Language Support	Selenium supports multiple programming languages like Java, C#, Ruby, Python, Perl etc	QTP supports only VB Script
Vendor Support	As Selenium is a free tool, user would not get the vendor's support in troubleshooting issues	Users can easily get the vendor's support in case of any issue

Q #45) Can WebDriver test Mobile applications?
WebDriver cannot test Mobile applications. WebDriver is a web based testing tool, therefore applications on the mobile browsers can be tested.

Q #46) Can captcha be automated?
No, captcha and bar code reader cannot be automated.

Q #47) What is Object Repository? How can we create Object Repository in Selenium?

Object Repository is a term used to refer to the collection of web elements belonging to Application Under Test (AUT) along with their locator values. Thus, whenever the element is required within the script, the locator value can be populated from the Object Repository. Object Repository is used to store locators in a centralized location instead of hard coding them within the scripts.
In Selenium, objects can be stored in an excel sheet which can be populated inside the script whenever required.
That's all for now.
Hope in this article you will find answers to most frequently asked Selenium and WebDriver Interview questions. The answers provided here are also helpful for understanding the Selenium basics and advanced WebDriver topics.

Q #48) Are you familiar with Selenium? If yes, what is it?

Selenium is a suite of software testing and automation tools built for web applications. Using Selenium, software tests can be written in languages like Java, Perl, Python, and more. The framework itself can be deployed on the three major operating systems: Windows, Mac, and Linux.

Q #49) What kinds of tests can be run with the Selenium framework?

Selenium can be used for load testing, regression testing, and functional testing of web applications.

Q #50) Explain some disadvantages to manual software testing.

Manual software testing takes huge amounts of time and resources, both human and machine. It's a potentially exhausting process that can end up costing more time and money for the company than if the process was simply automated, owing to employee fatigue and its consequences: inaccuracy, missed issues, lack of clarity.

Q #51) Is automation testing a complete replacement for manual software testing?

No. Proper automation requires as little intervention from humans as possible, since the tools used are built to run tests once they're setup. As convenient as this might be, it should not be a complete replacement for manual testing – only for repetitive tasks like load testing, where thousands of virtual users are required. Engineers should not automate things like test scripts, if those scripts can only be expected to run occasionally, nor should they automate code reviews, or bug testing for new builds of software that might require human interaction to detect specific issues. Large-scale, repetitive tasks are better fit for automation.

Q #52) Does automation testing have any disadvantages?

Designing the tools and tests to run software through takes a lot of manual, human effort, though there are frameworks and tests ready made for engineers to use. Even with automated testing, human error is still a factor – tools can be buggy, inefficient, costly, and sometimes even technologically limited in what kinds of tests they can run on their own.

Q #53) What are the differences between open source tools, vendor tools, and in-house tools?

Open source tools are free to use frameworks and applications. Engineers build the tool, and have the source code available for free on the internet for other engineers to use. Vendor tools are developed by companies that come with licenses to use, and often cost money. Because they are developed by an outside source, technical support is often available for use. Example vendor tools include WinRunner, SilkTest, Rational Robot, QA Director, QTP, LR, QC, RFT, and RPT. An in-house tool is a tool that a company builds for their own use, rather than purchasing vendor tools or using open source tools.

Q #54) How do you choose which automation tool is best for your specific scenario?

In order to choose the proper automation testing tool, you must consider:

- the scope of the project
- the limitation of the tool
- how much the tool costs
- the tool's usability and convenience
- the testing environment
- compatibility

Q #55) What are the different types of scripting techniques for automation testing?

Test automation scripting techniques include key and data driven techniques, shared, structured, and linear.

Q #56) What is the Selenium WebDriver?

The Selenium WebDriver is used for automating tests for websites.

Q #57) What is the Selenium IDE and what is it used for?

The Selenium IDE is an add-on for Firefox that includes numerous features for quality assurance and engineers to record and play back browser-based actions, such as typing and mouse clicks. Some of these particular features are: debugging functions, record/playback ability, user add-on capability. Users can speed up and slow down executions with the use of a built-in slider. They can also use the Selenium IDE as a side-bar, or as a separate pop-up window.

Q #58) Does the Selenium IDE have any drawbacks?

The Selenium IDE lacks conditional statements, logging and reporting functionality, loops, database testing, and it can not handle exceptions or automatically re-run tests that have failed. It also can't take screenshots. Another downside is that it's Firefox only. If the Selenium IDE is used in the Firefox browser's side-bar, then the quality engineer can't use it to record any actions undertaken by a user in a separate window.

Q #59) How can we get the font size, font colour, font text used for the particular text on the web page use in the selenium?

By using getCCSValue("font-size");
It's like that driver.findElement(By.id()).getCCSValue("font-size");
It's like that driver.findElement(By.id()).getCCSValue("font-colour");
It's like that driver.findElement(By.id()).getCCSValue("font-type");
It's like that driver.findElement(By.id()).getCCSValue("background-colour");

Q #60) What is the difference between driver.get and driver.navigate().to("URL")?

Both does the same thing but navigate also have some another methods like back(), refresh(), forward(),

Q #61) How to clear cache using selemium?

There is multiple ways to delete the cookies
1: driver.manage().deleteAllCookies();

2: If above command does not work then use this
DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();
capabilities.setCapability(InternetExplorerDriver.IE_ENSURE_CLEAN_SESSION, true);

Q #62) How to install add on the Firefox?

Using the firefox profile we can add the extension

```
packagecom.helloselenium.selenium.test;

importjava.io.File;

importorg.openqa.selenium.WebDriver;
importorg.openqa.selenium.firefox.FirefoxDriver;
importorg.openqa.selenium.firefox.FirefoxProfile;
importorg.openqa.selenium.remote.DesiredCapabilities;

publicclassRunFirefoxWithAddons{

 publicstaticvoidmain(String[] args) {

  WebDriver driver = null;

  FirefoxProfile firefoxProfile = newFirefoxProfile();
  File addonpath = newFile("path of addon/extension (.xpi file)");
  firefoxProfile.addExtension(addonpath);

  DesiredCapabilities capabilities = DesiredCapabilities.firefox();
  capabilities.setCapability(FirefoxDriver.PROFILE, profile);

  driver = newFirefoxDriver(capabilities);

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

  driver.quit();
 }

}
```

Q #63) HOW TO RUN SELENIUM WEBDRIVER SCRIPT IN FIREFOX BROWSER USING DESIREDCAPABILITIES?

First get the desiredCapabilities and then set

```
packagecom.helloselenium.selenium.test;

importorg.openqa.selenium.WebDriver;
importorg.openqa.selenium.firefox.FirefoxDriver;
importorg.openqa.selenium.remote.DesiredCapabilities;

publicclassOpenHelloSeleniumBlogInFirefox{

 publicstaticvoidmain(String[] args) {
```

```
  WebDriver driver = null;

  DesiredCapabilities capabilities = DesiredCapabilities.firefox();
  capabilities.setCapability(capability arg1, capability arg2);
  driver = newFirefoxDriver(capabilities);

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

  driver.quit();
 }

}
```

Q #64) What does Thread.sleep() method does?

It is used to pause execution of the program for the define time.

Q #65) How to read data from the .properties files?

```
package framework.vtiger.UtilMethods;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Properties;

public class ConfigFile {


public static void main(String[] args) throws IOException {
                fn_ConfigFileRead("ExcelSheet\\config.properties");
}


public static void fn_ConfigFileRead(String propertyFile) throws IOException{

        Properties propObj = new Properties();
        FileInputStream fisObj = new FileInputStream(propertyFile);
        propObj.load(fisObj);
   String uname = propObj.getProperty("username");
System.out.println(uname);
   String uPassword = propObj.getProperty("password");
System.out.println(uPassword);
```

```
}
}
```

Q #66) How to Kill all the browser at the same time?
We are using the .bat file in this we used "taskkill" command which work on the cmd

```
taskkill /F /IM IEDriverServer.exe /T
taskkill /F /IM chromedriver.exe /T
taskkill /F /IM iexplore.exe /T
taskkill /F /IM firefox.exe /T
exit
```

Q #67) What is the difference betweet explicitly wait and implicitly wait?
There are primarily two types of waits available in Selenium WebDriver.

- Implicit Wait
- Explicit Wait

Implicit Wait

Implicit waits are used to provide the latency within each and every test step of the test script. Thus, the program control would wait for the specified time before moving the execution control to the next step. Thus the implicit waits are applied to all the test step of the testscript. In other words we can say that the system would wait for the specified period of time in order to load the desired object into the DOM.

```
//Create WebDriver instance variable
WebDriver driver;
//Launch browser
driver=new FirefoxDriver();
//Apply implicit wait
driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
```

Explicit Wait

Explicit waits are smarter waits than implicit waits. Explicit waits can be applied at certain instances instead of applying on every web element within the testscript. Suppose if we are creating a login script for Gmail. We know that we are most likely to wait for a few seconds to let the home page load successfully. Thus, in such cases explicit waits can be used. Another important benefit that we get from explicit wait is that it waits maximum for the specified period of time or a condition to be met. Thus, if our condition (Let's say an element should be visible before we click on it) is

met before the specified time has elapsed, then the system control would move forward rather than waiting for the complete time to elapse in order to save our execution time.

Code Sample

```
public void explicitWait(WebDriver driver)
{
WebDriverWait wait = new WebDriverWait(driver, 20);
wait.until(ExpectedConditions.elementToBeClickable(driver.findElement(By.id
("element id"))));
}
```

In the above method WebDriver would wait for the expected condition (elementToBeClickable) to be met or for the timeout (20 seconds) to occur. As soon as the condition is met, the WebDriver would execute the next test step.

Q #68) In XPath, I wants to do partial match on attribute value from beginning. Tell me two functions using which I can do It.

We can use bellow given two functions with XPath to find element using attribute value from beginning.

contains()
starts-with()
Where to use these:

- When we do not have complete knowledge about the web elements HTML properties
- When the values of the attributes are dynamic i.e. changing
- When we would like to create a list of web elements containing same partial attribute value

BY Attributes:

```
driver.findElement(By.xpath("//input[contains(@name,'user_name')]")).sendKeys("ad
min");
```

By Text():

```
driver.findElement(By.xpath("//a[contains(text(), 'Marketing')]")).click();
```

```
<a href="index.php?module=Campaigns&action=index&parenttab=Marketing">Market
ing</a>
```

Starts-with()

starts-with() method is used when we know about the initial partial attribute value or initial partial text associated with the web element. User can also use this method to locate web elements those are consist of both the static(initial) and dynamic(trailing) values.

By Attribute

- //a[starts-with(@id,'link-si')]

- //a[starts-with(@id,'link-sign')]

Q #69) How to know the selected box is already selected using selenium?
By using isSelected() to check that check box is already selected or not.

```
if(!(driver.findElement(By.id(Objects.getProperty(object))).isSelected())){
driver.findElement(By.id(Objects.getProperty(object))).click();
 System.out.println("Checkbox "+object+ " has been selected" );
```

Q #70) How to verify the text inside the text box is present or not?

```
String Textboxvalue = "";
Textboxvalue=driver.findElement(By.xpath(Objects.getProperty(object))).getAttribut
e("value");
if(Textboxvalue.contains(expectedData)){
        message="Pass";
```

Q #71) How to accept exceptions in testNG?
using the @Test(expectedExceptions = ArithmeticException.class, NullPointerException.class)

Q #72) I have used findElements In my test case. It Is returning NoSuchElementException when not element found. Correct me If I am wrong.

It Is Incorrect. findElements will never return NoSuchElementException. It will return just an empty list.

Q #73) My Firefox browser Is not Installed at usual place. How can I tell FirefoxDriver to use It?

If Firefox browsers Is Installed at some different place than the usual place then you needs to provide the actual path of Firefox.exe file as bellow.

```
System.setProperty("webdriver.firefox.bin","C:\\Program Files\\Mozilla Firefox\\Firefox.exe");
driver =new FirefoxDriver();
```

Q #74) How to handle Untrusted SSL certificate error in IE browser

```
System.setProperty("webdriver.ie.driver", "path of IEDriverServer.exe ");
DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();
// this line of code is to resolve protected mode issue
capabilities.setCapability(InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_IGNORING_SECURITY_DOMAINS, true);
capabilities.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);
driver = new InternetExplorerDriver(capabilities);
```

Q #75) Can we run testNG class code without using any TestNg annotation?

No, you have to add one “@Test” annotation compulsory.

Q #76) What Is TestNG?

TestNG Is Open Source(Freeware) framework which Is Inspired from NUnit and JUnit with Introducing few new features and functionality compared to NUnit and JUnit to make It easy to use and more powerful.

We can use TestNg with selenium webdriver to configure and run test cases very easily, easy to understand, read and manage test cases, and to generate HTML or XSLT test reports.

Q #77) Can you describe major features of TestNG?

TestNG has many major features like support of @DataProvider annotation to perform data driven testing, can set test case execution dependency, test case grouping, generate HTML and XSLT test execution report etc..

Q #78) Describe the similarities and difference between JUnit and TestNG unit testing frameworks.

You can find all the similarities and difference between JUnit and TestNG framework on

Similarity:

JUnit	TestNG
We can create test suite	Same
We can skip any test method	
It is possible to create expected exception test. Few annotation are similar, @Test, @BeforeClass, @AfterClass	
And @Before@After Junit annotation are similar as @BeforeMethod@AfterMethod in the TestNG	

Difference:

JUnit	**TestNG**
Parameter configuration is very tuff	Parameter configuration is very eassy.
JUnit does not support group test.	TestNG support group test @Test(group ={sanity})
These are not supportable.	TestNG support @BeforeTest, @AfterTest, @BeforeSuite, @AfterSuite, @BeforeGroups, @AfterGroups which are not supported in JUnit.
Test prioritizing and parallel testing is not possible	In testNG test priority is possible using @Test(priority = 1) Parallel test case is also possible by

calling the multiple test in a <suite>
<test>
…..
<\test>

Q #79) How to Install TestNG In Eclipse? How do you verify that TestNg Is Installed properly In Eclipse?

To Install TestNG In Eclipse. We can directly install the TestNG in the Eclipse using help->eclipseMarket. Or we can download the .jar .

Q #80) What are different annotations supported by TestNG ?
TestNG supports many different annotations to configure Selenium WebDriver test.
@Test
@Test annotation describes method as a test method or part of your test.

@Test (Priority = 1)
@ to the priority of the function

@Test(group = {"smoke"})
@ set the method name with group, when testNG will run "smoke" group only those methods will run which are only belongs to the "smoke" group
@Parameters
When you wants to pass parameters in your test methods, you need to use @Parameters annotation.
In the .xml file
(<parameter name="browser" value="FFX" />

In the class, use just above the methods
@Test
@Parameters ({"browser"})

@BeforeMethod
Any method which is marked with @BeforeMethod annotation will be executed before each and every @test annotated method.

@AfterMethod

Same as @BeforeMethod, If any method is annotated with @AfterMethod annotation then it will be executed after execution of each and every @test annotated method.

@BeforeClass
Method annotated using @BeforeClass will be executed before first @Test method execution. @BeforeClass annotated method will be executed once only per class so don't be confused.

@AfterClass
Same as @BeforeClass, Method annotated with @AfterClass annotation will be executed once only per class after execution of all @Test annotated methods of that class.

@BeforeTest
@BeforeTest annotated method will be executed before the any @Test annotated method of those classes which are inside <test> tag in testng.xml file.

@AfterTest
@AfterTest annotated method will be executed when all @Test annotated methods completes its execution of those classes which are inside <test> tag in testng.xml file.

@BeforeSuite
Method marked with @BeforeSuite annotation will run before the all suites from test.

@AfterSuite
@AfterSuite annotated method will start running when execution of all tests executed from current test suite.

@DataProvider
When you use @DataProvider annotation for any method that means you are using that method as a data supplier. Configuration of @DataProvider annotated method must be like it always return Object[][] which we can use in @Test annotated method.
If you want to provide the test data, the DataProvider way, then we need to declare a method that returns the data set in the form of two **dimensional object array Object[][]. The first array represents a data set whereas the second array contains the parameter values**.
There are two way to provide data using @DataProvideer annotation

Dynamic way: passing data through another method:

We have provided the DataProvider method getData within the test class itself. Note that it is annotated with @DataProvider. Since it doesn't have the name attribute, its name by default will be getData. It returns two sets of data, each set of which contains two values, an integer and a string value.

```
Public class InstanceDataProviderExample {

@Test(dataProvider="getData")
        public void instanceDbProvider(int p1, String p2) {
        System.out.println("Instance DataProvider Example: Data(" + p1 + ", " + p2 +
")");
        }

@DataProvider
        public Object[][] getData() {
                return new Object[][]{{5, "five"}, {6, "six"}};
        }
}
```

OUTPUT:

```
[TestNG] Running:
  C:\javacodegeeks_ws\testNgDataProvider\test\com\javacodegeeks\testng\testng.xml

Instance DataProvider Example: Data(5, five)
Instance DataProvider Example: Data(6, six)
```

Static Data Provider: DataProvider method can also be defined in a separate class as a static method, in which case, the test method using it has to specify both the DataProvider name and its class in the @Test attributes dataProvider and dataProviderClass.

```
public class StaticDataProviderExample {

@Test(dataProvider="client1", dataProviderClass=DataProviderSource.class)
        public void client1Test(Integer p) {
                System.out.println("Client1 testing: Data(" + p + ")");
        }
```

```
@Test(dataProvider="client2", dataProviderClass=DataProviderSource.class)
        public void client2Test(Integer p) {
                System.out.println("Client2 testing: Data(" + p + ")");
        }
}
```

OUTPUT:
C:\javacodegeeks_ws\testNgDataProvider\test\com\javacodegeeks\testng\staticDataProviderTestng.xml

Client1 testing: Data(1)
Client2 testing: Data(2)

@BeforeGroups
@BeforeGroups annotated method will run before the first test run of that specific group.

@AfterGroups
@AfterGroups annotated method will run after all test methods of that group completes its execution.

@Factory
When you wants to execute specific group of test cases with different values, you need to use @Factory annotation. An array of class objects is returned by @Factory annotated method and those TestNG will those objects as test classes.

@Listeners
@Listeners are used to with test class. It is helpful for logging purpose.

Q #81) What Is the usage of testng.xml file?
In selenium WebDriver, We are using testng.xml file to configure our whole test suite In single file. Few of the tasks which we can specify In testng.xml file are as below.
We can define test suite using set of test cases to run them from single place.
Can Include or exclude test methods from test execution.
Can specify a group to Include or exclude.
Can pass parameter to use In test case.
Can specify group dependencies.
Can configure parallel test execution.
Can define listeners.

Q #82) How to pass parameter with testng.xml file to use It In test case?
We can define parameter In testng.xml file using syntax like bellow.

<parameter name="browser" value="FFX" />

Here, name attribute defines parameter name and value defines value of that parameter. Then we can use that parameter In selenium webdriver test case using bellow given syntax.

@Parameters ({"browser"})

Q #83) I have a test case with two @Test methods. I want to exclude one @Test method from execution. Can I do It? How?

Yes you need to specify @Test method exclusion In testng.xml file as bellow.

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Test Exclusion Suite">
<test name="Exclusion Test" >
<classes>
<class name="Your Test Class Name">
<methods>
<exclude name="Your Test Method Name To Exclude"/>
</methods>
</class>
</classes>
</test>
</suite>
```

You need to provide @Test method name In exclude tag to exclude It from execution.

Q #84) Tell me syntax to skip @Test method from execution.
You can use bellow given syntax Inside @Test method to skip It from test execution.
Please see the code in the "**package** framework.vTigerDay1.TestNGClass and TestNG.xml file;"
Using (@Test (enabled= false)): we can skip any method even it include in the TestNG.xml file.

throw new SkipException("Test Check_Checkbox Is Skipped");

Q #85) Arrange bellow give testng.xml tags from parent to child.

<test>
<suite>
<class>
</methods>
</classes>

Parent to child arrangement for above testng tags Is as bellow.

<suite>
<test>
</classes>
<class>
</methods>

Q #86) How to set priority of @Test method? What Is Its usage?
In your test case, you can set priority for TestNG @Test annotated methods as bellow.
@Test(priority=0)

Using priority, We can control @Test method execution manner as per our requirement. That
means @Test method with priority = 0 will be executed 1st and @Test method with priority = 1 will be executed 2nd and so on.

Q #87) Tell me any 5 assertions of TestNG which we can use In selenium webdriver.
There are many different assertions available In TestNG but generally I am using bellow given assertions In my test cases.
assertEquals: This assertion is useful to compare expected and actual values in selenium webdriver. If both values match then its fine and will continue execution. But if fails then immediately it will mark that specific test method as fail and exit from that test method.
Assert.assertEquals(actual, expected);
Assert.assertEquals(Actualtext, "Tuesday, 28 January 2014");

assertNotEquals :It's function is opposite to assertEquals assertion. Means if both sides values will not match then this assertion will pass else it will fail
Assert.assertNotEquals(actual, expected, message)
Assert.assertNotEquals(Actualtext, "Tuesday, 28 January 2014", "Expected and actual match in assertion_method_1");
assertTrue : assertTrue assertion is generally used for boolean condition true. It will pass if condition returns "true". If it will return false then it will fail and skip test execution from that specific method.
assertTrue(condition)
Assert.assertTrue(driver.findelement(By.id(“ ”)).isSelected());

assertFalse :It will check boolean value returned by condition and will pass if returned value is "False". If returned value is pass then this assertion will fail and skip execution from current test method.
Assert.assertFalse(condition)
Assert.assertFalse(driver.findelement(By.id(“ ”)).isSelected());

assertNull : assertNull(object) assertion will check and verify that object is null. It will pass if object found null. If object found not null then it will return error message like "java.lang.AssertionError: expected [null] but found [true]". Whenever your assertion fails, it will mark that specific test method as fail in TestNG result.
assertNull(object)
txt1 = driver.findElement(By.xpath("//input[@id='text1']"));

Assert.assertNull(txt1.getAttribute("disabled"));

assertNotNull: assertNotNull assertion is designed to check and verify that values returned by object is not null. Means it will be pass if returned value is not null.
assertNotNull(object)
txt1 = driver.findElement(By.xpath("//input[@id='text1']"));

Assert.assertNotNull(txt1.getAttribute("disabled"));

Q #88) Can you tell me usage of TestNG Soft Assertion?

Using TestNG soft assertion, We can continue our test execution even if assertion fails. That means on failure of soft assertion, remaining part of @Test method will be executed and assertion failure will be reportedat the end of @Test method.
To use testng soft assertion, you have to use testng SoftAssert class. This class will helps to not throw an exception on assertion failure and recording failure. If you will use soft assertion then your test execution will remain continue even If any assertion fails. Another most Important thing Is your assertion failure will be reported In report so that you can view It at end of test. You can use soft assertion when you are using multiple assertions In same test method and you wants to execute all of them even If any one In between fails.

```
//Created object of testng SoftAssert class to use It's Properties.
  SoftAssert s_assert = new SoftAssert();

//Text on expected side Is written Incorrect intentionally to get fail this assertion.
Assert.assertEquals(Actualtext, "Tuesday, 01 January 2014", "1st assert failed.");
System.out.println("Hard Assertion -> 1st pagetext assertion executed."

//Text on expected side Is written Incorrect intentionally to get fail this assertion.
s_assert.assertEquals(Actualtext, "Tuesday, 01 January 2014", "1st assert failed.");
System.out.println("Soft Assertion -> 1st pagetext assertion executed.");
```


Q #89) How to write regular expression In testng.xml file to search @Test methods containing "product" keyword.

Regular expression to find @Test methods containing keyword "product" Is as bellow.

```
<methods>
<include name=".*product.*"/>
</methods>
```

Q #90) Which time unit we provide In time test? minutes? seconds? milliseconds? or hours? Give Example.

Time unit we provide on @Test method level or test suite level Is In milliseconds.

Q #91) What Is the syntax to get value from text box and store It In variable.

Most of the time, String In text box will be stored as value. So we need to access value attribute(getAttribute) of that text box as shown In bellow example.

String Result =
driver.findElement(By.xpath("//input[@id='Resultbox']")).getAttribute("value");

Q #92) What Is Parallelism In TestNG?

In general software term, Parallelism means executing two part of program simultaneously or executing program simultaneously or we can say multithreaded or parallel mode. TestNG has same feature using which we can start multiple threads simultaneously In parallel mode and test methods will be executed In them.

Example:

```
<suite name="Parallel Class Suite" parallel="classes" thread-count="2">
<test name="Parallel Class Test" >
<classes>
<class name="Testing_Pack.Test_One"/>
<class name="Testing_Pack.Test_Two"/>
</classes>
</test>
</suite>
```

Q #93) What Is dependency test In TestNG?

Dependency Is very good feature of testng using which we can set test method as dependent test method of any other single or multiple or group of test methods. That means depends-on method will be executed first and then dependent test method will be executed. If depends-on test method will fail then execution of dependent test method will be skipped automatically. TestNG dependency feature will works only If depends-on test method Is part of same class or part of Inherited base class.

Exmaple:

```
@Test(dependsOnMethods={"Login","checkMail"})
public void LogOut() {
System.out.println("LogOut Test code.");
   }
```

Q #94) How to use grid and while using grid how to run webDriver on ode machines?

To use grid we need two or multiple machines.
One would be host and other would be node.
HOST Machine:

Open command prompt and fire the below command or save that command in the .bat file. So that we can run by using file it self.

D:

cd D:\seleium tool\Guru 99\WebDriver Test cases\SeleniumClassMy\jars

java -jar selenium-server-standalone-2.45.0.jar -role hub

Node Machine:

Open command prompt and file the below command or save that command in the .bat file. So that we can run by using file itself.

And for using node machine we have to set the limitations and capabilities for the node machine so that save all the configurations, limitations and capabilities in the .json file. As all the files are in that path "D:\seleium tool\Guru 99\WebDriver Test cases\SeleniumClassMy\grid"

D:

cd D:\seleium tool\Guru 99\WebDriver Test cases\SeleniumClassMy

java -jar jars\selenium-server-standalone-2.45.0.jar -role node -hub

http://localhost:4444/grid/register nodeconfig grid\nodeConfig.json

In the code, how to set the webDriver for grid?

```
publicstatic LoginPage openWebPageGrid(String browserName, String url, String OS)
throws MalformedURLException {
DesiredCapabilities dc = null;

if (browserName.equalsIgnoreCase("FF") &&OS.contentEquals("windows")){
        dc = DesiredCapabilities.firefox();
        dc.setPlatform(Platform.WINDOWS);
}
elseif (browserName.equalsIgnoreCase("CH") &&OS.equalsIgnoreCase("windows")){
System.setProperty("webdriver.chrome.driver", "Drivers\\chromedriver.exe");
dc = DesiredCapabilities.chrome();
dc.setPlatform(Platform.WINDOWS);
}
elseif (browserName.equalsIgnoreCase("IE") &&OS.equalsIgnoreCase("Windows")) {
System.setProperty("webdriver.ie.driver", "Drivers\\IEDriverServer.exe");
dc = DesiredCapabilities.internetExplorer();
dc.setPlatform(Platform.WINDOWS);
} // same we can create more conditions

// WD is the predefine key word, wd- webdriver
```

```
URL urlObj = newURL("http://14.98.122.21:4444/wd/hub");

driver = new RemoteWebDriver(urlObj, dc);
driver.get(url);
driver.manage().timeouts().implicitlyWait(50, TimeUnit.SECONDS);
driver.manage().window().maximize();

return PageFactory.initElements(driver, LoginPage.class);
}
```

***Note**:-Default port no is 4444*

Q #95) Can we use implicitly wait() and explicitly wait() together in the test case?

No. we should not.

Before using the explicitly wait(), reset the implicitly wait() by using the below code.

Create a method for the explicitly wait():

```
Public void explicitlyWaitForWebElement(WebDriver driver, WebElement we){
driver.manage.timeouts().implicitywait(0, TimeUnit.SECONDS); // nullify the time
WebDriverWait wait = new WebDriverWait(driver, 10);
Wait.until(ExpectedConditions.presenceOfElementLocated(we));
driver.manage().timeouts().implicitywait(DEFAULT_WAIT_4_PAGE,
TimeUnit.SECONDS); // reset the time
}
```

Always call this function to set the explicitly wait().

Q #96) What is the syntax of URL?

Uniform Resource Locator

resource_type://hostname.domain:port/*path*/filename

- **resource_type** - defines the **type** of Internet service (most common is **http, ftp,https**)
- **HTTP**: hyper text transport protocol
- **HTTPS**: Secure Hyper Text transport protocol
- **hostname** - defines the **domain host** (default host for http is **www**)
- **domain** - defines the Internet **domain name** (w3schools.com)
- **port** - defines the **port number** at the host (default for http is **80**)
- **path** - defines a **path** at the server (If omitted: the root directory of the site)
- **filename** - defines the name of a document or resource

Q #97) What is cookies?

Cookies are text files retained on computers by browsers containing various information in regards to a specific website visit.
Cookies are used to identify users, store shopping cart information, and prepare customized web pages that utilize user information. The cookie may be used to remember a username, help you resume where you left off, for example, so that the name will auto-fill on the user's next visit. Cookies may be disabled, or cookie options customized, due to privacy concerns and the risk of some cookies being used as spyware. It should be noted that because cookies are not executable files, they cannot be considered viruses as they do not have the ability to replicate.
Session cookies last only for as long as a user is on a website; they expire after the browser window is closed or the session times out**. Persistent cookies** (also known as tracking cookies) remain active for a period of time on a user's machine and are used whenever the website is accessed**. Secure cookies** are used when accessing a website via HTTPS and are encrypted for greater safety.

Cookies cannot carry viruses, and cannot install malware on the host computer

Q #98) What type of framework use in the automation testing?
Three type of framework majorly use in the automation testing.

1. Data Driven framework
2. Keyword driven framework
3. Hybrid driven framework

Data Driven framework: It is use where testing relays on the huge number of input test data.
Example: Online Candidate forms fill up. Suppose it have lots of input data to be supplied by user(tester) with the different combination and permutation.
These different types of data cannot be hard code in the java test file. Hence data need to be pull from the external source (framework).

Keyword Driven framework: All information is written in code.
Example: here each operations are represented as a keyword like SendMail(), LoginIntoPage(), EnterDetails() etc.
Through this way new test case can reuse the existing test case.

But it required more programming knowledge and code becomes lengthier.

Hybrid framework: It is the combination of data driven and keyword driven both. Hybrid framework pulls the data from the external source and also it uses the Keywords to perform operation like SendMails(), LoginIntoPage().

Q #99) What types of Models choose while designing the framework?

Most commonly use models are:

1. Behavioral Driven Development
2. Page Object Model

1. Behavioral Driven Development:
2. Page Object Model: In this type of Model, Each UI has different type of object (elements) to interact. These object are identified and written in the code along with their identity by using the One page have one Class in java.

   Exp: @FindBy (name = "user_name")
   Public WebElement UserName;
   @FindBy (xpath = "//span[@text()='Marketing']")
   Public WebElement Link;

   Advantages: In page Object Model, one page objects save in the one class. This reduces the amount of duplicate code and If any UI changes occurs then need to change only in one place.

Q #100) What is the frame? Why we need to switch to the frame?

frames are used to divide your browser window into multiple sections where each section can load a separate HTML document.To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines how to divide the window into frames.

Frames are not the part of the same window handles because An IFrame (Inline Frame) is an HTML document embedded inside another HTML document on a website.

IFrame:*An inline frame is used to embed another document within the current HTML document. It means iframe is actually a webpage within the webpage which have its own DOM (Document object model) for every iframe on the page.* Web designers use IFrames to embed interactive applications in Web pages, including those that employ Ajax (Asynchronous JavaScript and XML), like Google Maps or ecommerce applications.

Q #101) What are the selenium exception you faced?

There are multiple exceptions in selenium:

NoSuchElementException: Thrown when element could not be found.

To handle this, use wait either implicit or explicit and first use WebElement.isDisplayed(), or WebElement.isEnable().

NoAlertPresentException: Thrown when switching to no presented alert.

NoSuchWindowException: Thrown when window target to be switched doesn't exist.

NoSuchFrameException: Thrown when frame target to be switched doesn't exist
ElementNotVisibleException: Thrown when an element is present on the DOM, but it is not visible, and so is not able to be interacted with.
To handle this exception, use explicit wait or try to find out the element by javaScript.
StaleElementReferenceException: This tells that element is no longer appearing on the DOM page.
To handle this exception use an explicit wait on the element to ensure the update is complete, then grab a fresh reference to the element again.
ElementNotSelectableException: Thrown when trying to select an unselectable element.

Q #102) How to use @DataProvider annotation in TestNG?

@DataProvider

When you use @DataProvider annotation for any method that means you are using that method as a data supplier. Configuration of @DataProvider annotated method must be like it always return Object[][] which we can use in @Test annotated method.
If you want to provide the test data, the DataProvider way, then we need to declare a method that returns the data set in the form of two **dimensional object array Object[][]. The first array represents a data set whereas the second array contains the parameter values**.
There are two way to provide data using @DataProvideer annotation

Dynamic way: passing data through another method:

We have provided the DataProvider method getData within the test class itself. Note that it is annotated with @DataProvider. Since it doesn't have the name attribute, its name by default will be getData. It returns two sets of data, each set of which contains two values, an integer and a string value.

Public class InstanceDataProviderExample {

```
@Test(dataProvider="getData")
	public void instanceDbProvider(int p1, String p2) {
	System.out.println("Instance DataProvider Example: Data(" + p1 + ", " + p2 +
")");
	}

@DataProvider
	public Object[][] getData() {
		return new Object[][]{{5, "five"}, {6, "six"}};
	}
}
```

OUTPUT:

```
[TestNG] Running:
  C:\javacodegeeks_ws\testNgDataProvider\test\com\javacodegeeks\testng\testng.xml

Instance DataProvider Example: Data(5, five)
Instance DataProvider Example: Data(6, six)
```

Static Data Provider: DataProvider method can also be defined in a separate class as a static method, in which case, the test method using it has to specify both the DataProvider name and its class in the @Test attributes dataProvider and dataProviderClass.

```
public class StaticDataProviderExample {

@Test(dataProvider="client1", dataProviderClass=DataProviderSource.class)
	public void client1Test(Integer p) {
		System.out.println("Client1 testing: Data(" + p + ")");
	}

@Test(dataProvider="client2", dataProviderClass=DataProviderSource.class)
	public void client2Test(Integer p) {
		System.out.println("Client2 testing: Data(" + p + ")");
	}
}
```

OUTPUT:

```
C:\javacodegeeks_ws\testNgDataProvider\test\com\javacodegeeks\testng\staticDataPr
oviderTestng.xml
```

Client1 testing: Data(1)
Client2 testing: Data(2)

Q #103) How to find out the dynamic element by webdriver?

Can find dynamic element by multiple ways:

- **Absolute xpath:**Xpath Position or Absolute Xpath are most frequently used to resolve the dynamic element issues.

web_element_name=html/body/div[30]/div[2]/div[2]/div/div/div/div[1]/table/tbody/tr/td[2]/table/tbody/tr/td[1]/table/tbody/tr/td[1]/table/tbody/tr[2]/td[2]/em/button//p[6]/label[2]/div/ins

- **Identity element by starts-with and contains text():**

If the dynamic elements have a definite pattern to them, then we can also use JavaScript functions like “starts-with” or “contains” in our element locators to separate the dynamic part of locator from static part.

XPath: //button[starts-with(@id, 'Submit-')]

XPath: //input[contains(@class, 'suggest')].

Q #104) What are the most common pre define functions of xpath?

Below are the most common pre define functions are;

- Last()
- Position()
- Contains()
- Start-with ()

How to use them?

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book>
  <title lang="en">Harry Potter</title>
  <price>29.99</price>
</book>
<book>
  <title lang="en">Learning XML</title>
  <price>39.95</price>
</book>
</bookstore>
```

/ = Selects from the root node

// =Selects nodes in the document from the current node that match the selection no matter where they are

. = Selects the current node

.. = Selects the parent of the current node

@ = Selects attributes

/bookstore = selects the root element bookstore

Note: If the path start with a slash(/), it always represents an absolute path to an elements.

//book = select all book elements no matter where they are in the document
bookstore/book = Selects all book elements that are children of bookstore
bookstore//book = Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang = Selects all attributes that are named lang

//bookstore/book[1] =Selects the first book element that is the child of the bookstore element.

Note: In IE 5,6,7,8,9 first node is[0], but according to W3C, it is [1]. To solve this problem in IE, set the SelectionLanguage to XPath:

In JavaScript: xml.setProperty("SelectionLanguage","XPath");
//bookstore/book[**last()**]=Selects the last book element that is the child of the bookstore element.
//bookstore/book[position()<3] = Selects the first two book elements that are children of the bookstore element.
*= matches any elements node
@* = matches any attribute node
Node() = matches any node of any kind

/bookstore/* = select all the child element of the bookstore element.
//* = select all elements in the document.
//tittle[@*] = select all title elements which have at least one attribute of any kind.

Q #105) How to set the browser for the GRID?
For the grid we also need to add OS information and for the remote machine we use RemoteWebDriver()

```
publicstatic LoginPage openWebPageGrid(String browserName, String url, String OS)
throws MalformedURLException {
        DesiredCapabilities dc = null;

        if (browserName.equalsIgnoreCase("FF") &&OS.contentEquals("windows")){
                dc = DesiredCapabilities.firefox();
                dc.setPlatform(Platform.WINDOWS);
        }
        elseif (browserName.equalsIgnoreCase("CH")
&&OS.equalsIgnoreCase("windows")){
                System.setProperty("webdriver.chrome.driver",
"Drivers\\chromedriver.exe");
                dc = DesiredCapabilities.chrome();
                dc.setPlatform(Platform.WINDOWS);
        }
        elseif (browserName.equalsIgnoreCase("IE")
&&OS.equalsIgnoreCase("Windows")) {
        System.setProperty("webdriver.ie.driver", "Drivers\\IEDriverServer.exe");
dc = DesiredCapabilities.internetExplorer();
dc.setPlatform(Platform.WINDOWS);
        } // same we can create more conditions
        // WD is the predefine key word, wd- webdriver
        URL urlObj = newURL("http://14.98.122.21:4444/wd/hub");

        driver = new RemoteWebDriver(urlObj, dc);
        driver.get(url);
        driver.manage().timeouts().implicitlyWait(50, TimeUnit.SECONDS);
        driver.manage().window().maximize();
        return PageFactory.initElements(driver, LoginPage.class);

}
```

Q #106) How to get the snapshot using selenium?

```
        publicstaticvoid fn_takeSanpShot(String path) throws IOException{
                        if (fileSnapShotObj == null) {
                        fileSnapShotObj = new File(snapShotFolderPath);
                        if (!fileSnapShotObj.exists()){
                        // this is using to create the directory for snapShot if not exist.
                                fileSnapShotObj.mkdir();
                        }
                }

                TakesScreenshot tss = (TakesScreenshot)driver;
```

```
            File scrObj = tss.getScreenshotAs(OutputType.FILE);
            File destObj = newFile(path);
            FileUtils.copyFile(scrObj, destObj);
        }
```

Q #107) How to shoot the snapshot using selenium?

```
publicstaticvoid shootSnapShot(WebElement we, String destPath)  {

        if (fileSnapShotObj == null) {
                fileSnapShotObj = new File(snapShotFolderPath);
                if (!fileSnapShotObj.exists()){
// this is using to create the directory for snapShot if not exist.
                fileSnapShotObj.mkdir();
                }
        }
        TakesScreenshot tss = (TakesScreenshot) driver;
        File srcObj = tss.getScreenshotAs(OutputType.FILE);
        Point p = we.getLocation();
        intheight = we.getSize().getHeight();
        intwidth = we.getSize().getWidth();
System.out.println("p.getx()= "+p.getX()+"p.gety()="+p.getY()+"height=
"+height+"width"+width);
try {
        BufferedImage biObj = ImageIO.read(srcObj);
//      Parameters:
//x - the X coordinate of the upper-left corner of the specified rectangular region
//y - the Y coordinate of the upper-left corner of the specified rectangular region
//w - the width of the specified rectangular region
//h - the height of the specified rectangular region
        BufferedImage dest = biObj.getSubimage(p.getX(), p.getY(), width, height);

        File destObj = newFile(destPath);
        ImageIO.write(dest, "jpeg", srcObj);
        FileUtils.copyFile(srcObj, destObj);
        Reporter.log("<a href=\"" + destPath  + "\">  SnapShot  </a>"  );
        } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        }
   }
}
```

Q #108) How to know that element is visible?

We can find the dimension of the element if it is visible.

```
privatestaticboolean fn_checkVisible(WebElement we){
        Dimension dimObj = we.getSize();
        if (dimObj.getHeight() > 0 &&dimObj.getWidth()> 0)
                returntrue;
        else
                returnfalse; }
```

Q #109) how to handle Alert using selenium?

By switching on the alert :

```
        publicstaticvoid fn_alert(String message) {

                Alert alert = driver.switchTo().alert();
                if (message.equalsIgnoreCase("yes"))
                        alert.accept();
                else
                        alert.dismiss();

        }
```

Q #110) How to switch on the given URL in multiple windows handle?

First find all the windows handle and then save it in the Iterator and then check the url of every windows and then switch.

```
        publicstaticvoid fn_switchWindowUsingURL(String fieldName){

                String fieldValue = fn_getValue(fieldName);
                Set<String>setObj = driver.getWindowHandles();
                Iterator<String>iteObj = setObj.iterator();

                while(iteObj.hasNext()) {

                        driver.switchTo().window(iteObj.next());
                        String currentURL = driver.getCurrentUrl();
                        if(currentURL.contains(fieldValue)){
                                System.out.println("Contains the url");
                                break;

                        }


                }
        }
```

Q #111) Explain the different between HTTP and HTTPS?

The differences between HTTP and HTTPS are following:

- Hypertext Transfer Protocol is a protocol for information to be passed back and forth between web servers and clients. Https is refers to the combination of a normal HTTP interaction over an encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS) transport mechanism
- HTTP use port number 80 whereas HTTPS use port number 443
- HTTP can support the client asking for a particular file to be sent only if it has been updated after a certain date and time whereas Hypertext Transfer Protocol over Secure Socket Layer is built into its browser that encrypts and decrypts user page requests as well as the pages that are returned by the Web server

Q #112) Why use pageObjectModel?

- It's a model though which we can create page specific code using all page functionality, collect all the locators at the time of page initialization.
- If there are some UI changes for the page, we just need to change the code with in the page class.
- Collect all the locators at the time of page initialization, we don't face the stateElementExceptions or ElementNotFoundException.
- By using the page object model, we can also make private to WebElement, WebDriver so that only base class can access it.
- Using the page object, when we are navigate to another page, return the other page (navigation page) object that will initialize the webElements of the class.

Q #113) How do you read data from excel ?

```
FileInputStream fis = new FileInputStream("path of excel file");
 Workbook wb = WorkbookFactory.create(fis);
 Sheet s = wb.getSheet("sheetName");
 String value = s.getRow(rowNum).getCell(cellNum).getStringCellValue();
```

Q #114) What are different types of locators ?

There are 8 types of locators and all are the static methods of the By class.
By.id(), By.name(), By.tagName(), By.className(), By.linkText(), By.partialLinkText(), By.xpath, By.cssSelector().

Q #115) What is the difference between Assert and Verify?

Assert- it is used to verify the result. If the test case fail then it will stop the execution of the test case there itself and move the control to other test case.
Verify- it is also used to verify the result. If the test case fail then it will not stop the execution of that test case.

Q #116) What is the alternate way to click on login button?

use submit() method but it can be used only when attribute type=submit.

Q #117) How do you verify if the checkbox/radio is checked or not ?

We can use isSelected() method.
Syntax –
driver.findElement(By.xpath("xpath of the checkbox/radio button")).isSelected();
If the return value of this method is true then it is checked else it is not.

Q #118) How do you handle alert pop-up ?

To handle alert pop-ups, we need to 1st switch control to alert pop-ups then click on ok or cancle then move control back to main page.
Syntax-
String mainPage = driver.getWindowHandle();
Alert alt = driver.switchTo().alert(); → to move control to alert popup
alt.accept(); ---> to click on ok.
alt.dismiss(); ---> to click on cancle.
Then move the control back to main web page-
driver.switchTo().window(mainPage); → to switch back to main page.

Q #119) How do you launch IE/chrome browser?

Before launching IE or Chrome browser we need to set the System property.
To open IE browser → System.setProperty("webdriver.ie.driver","path of the iedriver.exe file ");
WebDriver driver = new InternetExplorerDriver();
To open Chrome browser → System.setProperty("webdriver.chrome.driver","path of

the chromeDriver.exe file ”);
WebDriver driver = new ChromeDriver();

Q #120) How to perform right click using WebDriver?

use Actions class.
Actions act = new Actions(driver); // where driver is WebDriver type
act.moveToElement(webElement).perform();
act.contextClick().perform();

Q #121) How do perform drag and drop using WebDriver?

use Action class.
Actions act = new Actions(driver);
WebElement source = driver.findElement(By.xpath(“ -----”)); //source ele which you want to drag
WebElement target = driver.findElement(By.xpath(“ -----”)); //target where you want to drop
act.dragAndDrop(source,target).perform();

Q #122) Give the example for method overload in WebDriver.

frame(string), frame(int), frame(WebElement).

Q #123) How do you upload a file?

To upload a file we can use sendKeys() method.
Syntax - driver.findElement(By.xpath(“input field”)).sendKeys(“path of the file which u want to upload”);

Q #124) How do you click on a menu item in a drop down menu?

if that menu has been created by using select tag then we can use the methods selectByValue() or selectByIndex() or selectByVisibleText(). These are the methods of the Select class.
If the menu has not been created by using the select tag then we can simply find the xpath of that element and click on that to select.

Q #125) What is the difference between findElement and findElements?

Both methods are abstract method of WebDriver interface and used to find the WebElement in a web page.
findElement() - it used to find the one web element. It return only one WebElement type.
findElements()- it used to find more than one web element. It return List of WebElements.

Q #126) Write the code for Reading and Writing to Excel through Selenium ?

```
FileInputStream fis = new FileInputStream("path of excel file");
Workbook wb = WorkbookFactory.create(fis);
Sheet s = wb.getSheet("SheetName");
String value = s.getRow(rowNum).getCell(cellNum).getStringCellValue(); // read data
s.getRow(rowNum).getCell(cellNum).setCellValue(); // write data
FileOutputStream fos = new FileOutputStream("path of file");
wb.write(fos);
//save file
```

Q #127) How do you clear the contents of a textbox in selenium ?

use clear() method.
syntax- driver.findElement(By.xpath("xpath of box")).clear();

Q #128) How to get the number of frames on a page ?

```
List <WebElement> framesList = driver.findElements(By.xpath("//iframe"));
int numOfFrames = frameList.size();
```

Q #129) How do you simulate scroll down action ?

use java script to scroll down-

```
JavascriptExecutor jsx = (JavascriptExecutor)driver;
jsx.executeScript("window.scrollBy(0,4500)", ""); //scroll down, value 4500 you can change as per your req
```

```
        jsx.executeScript("window.scrollBy(450,0)", ""); //scroll up
ex-
public class ScrollDown {
   public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new FirefoxDriver();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.get("http://www.flipkart.com/womens-
clothing/pr?sid=2oq,c1r&otracker=hp_nmenu_sub_women_1_View%20all");
        driver.manage().window().maximize();
        JavascriptExecutor jsx = (JavascriptExecutor)driver;
        jsx.executeScript("window.scrollBy(0,4500)", ""); //scroll down
        Thread.sleep(3000);
        jsx.executeScript("window.scrollBy(450,0)", ""); //scroll up
   }
}
```

Q #130) What is the command line we have to write inside a .bat file to execute a selenium project when we are using testng ?

```
java -cp bin;jars/* org.testng.TestNG testng.xml
```

Q #131) How to check if an element is visible on the web page ?

use isDisplayed() method. The return type of the method is boolean. So if it return true then element is visible else not visible.

Syntax – driver.findElement(By.xpath("xpath of elemnt")).isDisplayed();

Q #132) How to check if a button is enabled on the page ?

use isEnabled() method. The return type of the method is boolean. So if it return true then button is enabled else not enabled.

Syntax – driver.findElement(By.xpath("xpath of button")).isEnabled();

Q #133) How to check if a text is highlighted on the page ?

To identify weather color for a field is different or not-

```
String color =
driver.findElement(By.xpath("//a[text()='Shop']")).getCssValue("color");
```

String backcolor = driver.findElement(By.xpath("//a[text()='Shop']")).getCssValue("background-color");
System.out.println(color);
System.out.println(backcolor);
Here if both color and backcolor different then that means that element is in different color.

Q #134) How to check the checkbox or radio button is selected ?

use isSelected() method to identify. The return type of the method is boolean. So if it return true then button is selected else not enabled.
Syntax – driver.findElement(By.xpath("xpath of button")).isSelected();

Q #135) How do u get the width of the textbox ?

driver.findElement(By.xpath("xpath of textbox ")).getSize().getWidth();
driver.findElement(By.xpath("xpath of textbox ")).getSize().getHeight();

Q #136) How do u get the attribute of the web element ?

driver.getElement(By.tagName("img")).getAttribute("src") will give you the src attribute of this tag. Similarly, you can get the values of attributes such as title, alt etc.
Similarly you can get CSS properties of any tag by using getCssValue("some propety name").

Q #137) How to check whether a text is underlined or not ?

Identify by getCssValue("border-bottom") or sometime getCssValue("text-decoration") method if the
cssValue is 'underline' for that WebElement or not.
ex- This is for when moving cursor over element that is going to be underlined or not-

```
public class UnderLine {
public static void main(String[] args) {
WebDriver driver = new FirefoxDriver();
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.get("https://www.google.co.in/?gfe_rd=ctrl&ei=bXAwU8jYN4W6iAf8zIDgDA&gws_rd=cr");
```

```
        String cssValue=
driver.findElement(By.xpath("//a[text()='Hindi']")).getCssValue("text-decoration");
        System.out.println("value"+cssValue);
        Actions act = new Actions(driver);
        act.moveToElement(driver.findElement(By.xpath("//a[text()='Hindi']"))).perform
();
        String cssValue1=
driver.findElement(By.xpath("//a[text()='Hindi']")).getCssValue("text-decoration");
        System.out.println("value over"+cssValue1);
        driver.close();
    }
}
```

Q #138) How to hover the mouse on an element ?

```
    Actions act = new Actions(driver);
        act.moveToElement(webelement); //webelement on which you want to move
cursor
```

Q #139) What is the use of getOptions() method ?

getOptions() is used to get the selected option from the dropdown list.

Q #140) What is the use of deSelectAll() method ?

It is used to deselect all the options which have been selected from the dropdown list.

Q #141) Which is the super interface of webdriver ?

SearchContext.

Q #142) How to enter text without using sendkeys() ?

Yes we can enter text without using sendKeys() method. We have to use combination of javascript and wrapper classes with WebDriver extension class, check the below code-

```
public static void setAttribute(WebElement element, String
```

```
attributeName, String value)
{
WrapsDriver wrappedElement = (WrapsDriver) element;
JavascriptExecutor driver = (JavascriptExecutor)
wrappedElement.getWrappedDriver();
driver.executeScript("arguments[0].setAttribute(arguments[1],
arguments[2])", element, attributeName, value);
}
```

call the above method in the test script and pass the text field attribute and pass the text you want to enter.

Q #143) There is a scenario whenever "Assert.assertEquals()" function fails automatically it has to take screenshot. How can you achieve this ?
By using EventFiringWebDriver.

Syntax-

```
EventFiringWebDriver eDriver=new EventFiringWebDriver(driver);
File srcFile = eDriver.getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(srcFile, new File(imgPath));
```

Q #144) How do you handle https website in selenium
By changing the setting of FirefoxProfile.
Syntax-

```
public class HTTPSSecuredConnection {
   public static void main(String[] args){
      FirefoxProfile profile = new FirefoxProfile();
      profile.setAcceptUntrustedCertificates(false);
      WebDriver driver = new FirefoxDriver(profile);
      driver.get("url");
   }
}
```

Q #145) How do you send ENTER/TAB keys in webdriver?

use click() or submit() [submit() can be used only when type='submit']) method for ENTER.
Or act.sendKeys(Keys.ENTER);
For Tab-

```
   act.sendKeys(Keys.TAB);
```

where act is Actions class type. (Actions act = new Actions(act);)

Q #146) What is Datadriven framework & Keyword Driven?

Datadriven framework- In this Framework , while Test case logic resides in Test Scripts, the Test Data is separated and kept outside the Test Scripts.Test Data is read from the external files (Excel File) and are loaded into the variables inside the Test Script. Variables are used both for Input values and for Verification values.

Keyword Driven framework- The Keyword-Driven or Table-Driven framework requires the development of data tables and keywords, independent of the test automation tool used to execute them . Tests can be designed with or without the Application. In a keyword-driven test, the functionality of the application-under-test is documented in a table as well as in step-by-step instructions for each test.

Q #147) While explaining the framework, what are points which should be covered ?

1. What is the frame work.
2. Which frame work you are using.
3. Why This Frame work.
4. Architecture.
5. Explanation of every component of frame work.
6. Process followed in frame work.
7. How & when u execute the frame work.
8. Code (u must write code and explain).
9. Result and reporting .
10. You should be able to explain it for 20 Minutes.

Q #148) How to switch back from a frame ?

use method defaultContent().
Syntax – driver.switchTo().defaultContent();

Q #149) How to type text in a new line inside a text area ?

Use \n for new line.
ex- webelement.sendKeys("Sanjay_Line1.\n Sanjay_Line2.");
it will type in text box as

Sanjay_Line1.
Sanjay_Line2.

Q #150) What is the use of AutoIt tool ?

Some times while doing testing with selenium, we get stuck by some interruptions like a window based pop up. But selenium fails to handle this as it has support for only web based application. To overcome this problem we need to use AutoIT along with selenium script. AutoIT is a third party tool to handle window based applications. The scripting language used is in VBScript.

Q #151) How to perform double click using WebDriver ?

use doubleClick() method.
Syntax- Actions act = new Actions(driver);
act.doubleClick(webelement);

Q #152) How to press Shift+Tab ?

```
String press = Keys.chord(Keys.SHIFT,Keys.TAB);
webelement.sendKeys(press);
```


Q #153) What is the use of contextClick() ?

It is used to right click.

Q #154) What is the difference b/w getWindowHandles() and getWindowHandle() ?

getWindowHandles()- is used to get the address of all the open browser and its return type is Iterator<String>.
getWindowHandle()- is used to get the address of the current browser where the conrol is and return type is String.

Q #155) How do you accommodate project specific methods in your framework ?

1st go through all the manual test cases and identify the steps which are repeating. Note down such steps and make them as methods and write into ProjectSpecificLibrary.

Q #156) What are different components of your framework ?

Library- Assertion, ConfigLibrary, GenericLibrary, ProjectSpecificLibrary, Modules.
Drivers folder, Jars folder, excel file.

Q #157) How to check all checkboxes in a page ?

```
List<webElement>  chkBox =
driver.findElements(By.xpath("//htmltag[@attbute='checkbox']"));
      for(int i=0; i<=chkBox.size(); i++){
   chkBox.get(i).click();
   }
```

Q #158) Count the number of links in a page.

use the locator By.tagName and find the elements for the tag //a then use loop to count the number of elements found.

```
Syntax-  int  count = 0;
 List<webElement>  link = driver.findElements(By.tagName("a"));
System.out.println(link.size());  // this will print the number of links in a page.
```

Q #159) How do you identify the Xpath of element on your browser ?

And- to find the xpath , we use Firebug addons on firefox browser and to identify the xpath written we use Firepath addons.

Syntax- //htmltag[@attname='attvalue'] or //html[text()='textvalue'] or //htmltag[contains(text(),'textvalue')] or //htmltag[contains(@attname,'attvalue')]

Q #160) What is Selenium Webdriver ?

WebDriver is the name of the key interface against which tests should be written in Java. All the methods of WebDriver have been implementated by RemoteWebDriver.

Q #161) What are the different assertions or check points used in your script?

The common types of validations are:

a) Is the page title as expected
b) Validations against an element on the page
c) Does text exist on the page

d) Does a javascript call return an expected value
method used for validation – Assert.assertEquals();

Q #162) What is the difference between before method and before class ?

@BeforeMethod- this will execute before every @Test method.
@BeforeClass- this will execute before every class.

Q #163) What are the different attributes for @Test annotation?

alwaysRun, dataProvider, dependsOnMethods, enabled, expectedExceptions, timeOut etc.
ex- @Test(expectedExceptions = ArithmeticException.class),
@Test(timeOut = 2000).

Q #164) What is object repository ?

An object repository is a very essential entity in any UI automation tool. A repository allows a tester to store all the objects that will be used in the scripts in one or more centralized locations rather than letting them be scattered all over the test scripts. The concept of an object repository is not tied to WET alone. It can be used for any UI test automation. In fact, the original reason why the concept of object repositories were introduced was for a framework required by QTP.

Q #165) What is testing strategy ?

A Test Strategy document is a high level document and normally developed by project manager. This document defines "Software Testing Approach" to achieve testing objectives. The Test Strategy is normally derived from the Business Requirement Specification document.

Q #166) Difference between Web driver listener and TestNG Listener.

TestNG and Web driver Listener have different interfaces to implement and call them. They both modify respective behaviour. You can use Listeners in Annotation.

Q #167) Which is the best way to locate an element?

Finding elements by ID is usually going to be the fastest option, because at its root, it eventually calls down to document.getElementById(), which is optimized by many browsers.

Finding elements by XPath is useful for finding elements using very complex selectors, and is the most flexible selection strategy, but it has the potential to be very slow,

particularly in IE. In IE 6, 7, or 8, finding by XPath can be an order of magnitude slower than doing the same in Firefox. IE provides no native XPath-over-HTML solution, so the project must use a JavaScript XPath implementation, and the JavaScript engine in legacy versions of IE really is that much slower.

If you have a need to find an element using a complex selector, I usually recommend using CSS Selectors, if possible. It's not quite as flexible as XPath, but will cover many of the same cases, without exhibiting the extreme performance penalty on IE that XPath can.

Q #168) Why we refer Firefox driver to the web driver inheritance.

web Driver driver = new FireFoxDriver();

WebDriver is an interface which contain several abstract methods such as get(...), findElamentBy(...) etc.

We simply create reference of web Driver and we can assign objects (Firefox driver, CromeDriver, IEDriver, Andriod driver etc) to it.

Ex :

WebDriver driver = new FireFoxDriver();----------(1)

If we are using (1) we can do the same thing by using

FireFoxDriver driver = new FireFoxDriver();---------(2)

We can use (1) and (2) for same purpose but if we want to switch to another browser in same program then again we have to create the object of other class as for example

CromeDriver driver = new CromeDriver();.

creating object of several class is not good. So we create the reference of WebDriver and we assign the objects of another class as for example

WebDriver driver; // it is created only one time in the program

driver = new FireFoxDriver();// any where in the program

driver = new CromeDriver(); // any where in the program

Q #169) What is the difference between thread.Sleep() and selenium. Set Speed ("2000")?

If the application is taking time to load the page then we use selenium.waitforpageload(" "). This command is doesn't wait upto the given time whenever the page load is completed.

If the application is taking time to refresh the page, then we use Thread. Sleep ( ).it is a standard wait it simply wait to the given time.

selenium.setSpeed

1. Takes a single argument in string format

Ex: selenium.setSpeed("2000") - will wait for 2 seconds

2. Runs each command in after setSpeed delay by the number of milliseconds mentioned in set Speed.

thread.sleep

1. Takes a single argument in integer format

ex: thread. Sleep(2000) - will wait for 2 seconds

2. Waits for only once at the command given at sleep.

Q #170) In what situation selenium finding element get fails?

- · Element loading issue
- · Dynamic id of web element

Q #171) How we can retrieve the dynamically changing Ids?

It can be handled through customized xpath

- ✓ preceding-sibling
- ✓ following-sibling
- ✓ contains method
- ✓ starts-with() method

Q #172) What is the basic use of Firefox profiles and how can we use them using selenium?

A profile in Firefox is a collection of bookmarks, browser settings, extensions, passwords, and history; in short, all of your personal settings. We use them to change user agent, changing default download directory, changing versions etc.

Q #173) Customize the name of file going to be downloaded?

You have to download AUTO IT.exe file and has to be install and later you have create .au3 file (in this file you have to specify the commands in
VB script like your file name, where have to save, it will be easy may be 3 or 4 steps)
using AUTOIT...then right click the .au3 file you have to compile ....after that you will
get the .exe file with the name of .au3 file ..In eclipse you will give the code like this

<----ProcessBuildderps = new ProcessBuilder("path of the .exe file of au3")
.start();--->

Q #174) How to handle internationalisation throughweb driver?

FirefoxProfile profile = new FirefoxProfile();
profile.set Preference("intl.accept_languages","jp");

Web driver driver = new FirefoxDriver(profile); driver.get(google.com) will open google in
Japanese Lang

Q #175) How to overcome same origin policy through web driver?

- · Proxy server.

DesiredCapabilities capability=new DesiredCapabilities.firefox();

capability.setCapability(CapabilityType.PROXY,"your desire proxy")

WebDriver driver=new FirefoxDriver(capability);

Q #176) Difference between flex and flash application.

In flash there is no code just based on creativity(design) we will complete the work(time consuming process) whereas flex contain some small functions
which is integrated with mxml,PHP..(no tool is there to develop

flex we want to use the properties of css and style sheet)

Q #177) What is Error Collector in TestNG? What is its use?

This class allows the collection of errors during the process of retrieving the test data for the test method parameters

Q #178) How to run tests in multiple browser parallel? Is there any other option other than selenium grid?

You create a class with a method something like this:

```
public class LaunchBrowser {

WebDriver driver=null;


// Pass parameter browser from test.xml
@Parameters("browser")
public void initiateBrowser(String browser){

// compare browser to fire fox and then open firefox driver
if(browser.equals("Firefox"))
{

driver = new FirefoxDriver();
}
else
{
\ set path to the IE driver correctly here
System.setProperty("webdriver.ie.driver", "\iexploredriver.exe");
driver =new InternetExplorerDriver();
}
}
```

Now create YourClassName class and call extend the above class something like this

```
@Test
public class YourClassName extends LaunchBrowser{

public void gotoGoogle(){


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

Q #179) How to prepare Customized html Report using TestNG in hybrid framework.

Below are the 3 ways:

• Junit: with the help of ANT.

• TestNG: using inbuilt default.html to get the HTML report. Also XST reports from ANT,

Selenium, TestNG combination.

• Using our own customized reports using XSL jar for converting XML content to HTML.

Q #180) How the TestNG interacts with Selenium Core? Explain me steps and internal architecture?"What is TestNG?

So far we had been doing Selenium tests without generating a proper format for the test results. From this point on, we shall tackle how to make these reports using a test framework called TestNG.

TestNG is a testing framework that overcomes the limitations of another popular testing framework called JUnit. The "NG" means "Next Generation". Most Selenium users use this more than JUnit because of its advantages. There are so many features of TestNG, but we will only focus on the most important ones that we can use in Selenium.Advantages of TestNG over JUnit

There are three major advantages of TestNG over JUnit:

- Annotations are easier to understand
- Test cases can be grouped more easily
- Parallel testing is possible

Q #181) Is it possible test web services using selenium?

Using Jmeter we can test how one website is talking to each other means time taken to
send data, feeds, messages from one website to other website. Jmeter does a nice job
of doubling for performance and api tests.

Q #182) How to refresh a page without using context click?

1.UsingsendKeys.Keys method

2.Usingnavigate.refresh() method

3.Usingnavigate.refresh() method

4.Using get() method

5.UsingsendKeys() method

1.UsingsendKeys.Keys method

driver.get("https://accounts.google.com/SignUp");

driver.findElement(By.id("firstname-placeholder")).sendKeys(Keys.F5);

2.Usingnavigate.refresh() method

driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/100-selenium-interview-questions.html");

driver.navigate().refresh();

3.Using navigate.to() method

driver.get("http://ruchi-myseleniumblog.blogspot.in/2014/01/selenium-hybrid-framework-using.html");

driver.navigate().to(driver.getCurrentUrl());

4.Using get() method

driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/basic-core-java-interview-questions.html");

driver.get(driver.getCurrentUrl());

5.UsingsendKeys() method

driver.get("https://accounts.google.com/SignUp");

driver.findElement(By.id("firstname-placeholder")).sendKeys("\uE035");

Q #183) Can you send a code for printing in selenium?

There are two cases:

Case1. Any hyperlink/button on a web page, n clicking that link/button a print dialog box
opens. (Performing an action on web page)

Case2.or do u want to open print dialog box within ur own script, not by performing any
action on web page.

So If Case 1: just a call for WebElement.click() event will work to open it.

If Case 2: Call a Printer Job object (Use Awt API).

For code: Google it.

Q #184) How to find broken images in a page using Selenium Web driver.

1. Get xpath and then using tag name; get all the links in the page

2. Click on each and every link in the page

3. In the target page title, look for 404/500 error.

How to find broken images in a page using Selenium

```
package programs;

import java.util.List;
import org.apache.http.HttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class findbrokenimages {
static intinvalidimg;
static WebDriver driver ;
public static void main(String[] args) {
try {
driver = new FirefoxDriver();
driver.get("http://ruchi-myseleniumblog.blogspot.in");
invalidimg = 0;
List allImages = driver.findElements(By.tagName("img"));
System.out.println("Total images are " + allImages.size());
for (int i = 0; i <allImages.size(); i++) {
WebElementimg = (WebElement) allImages.get(i);
if (img != null) {
verifyimgActive(img);
```

```
}
}

System.out.println("Total invalid images are " + invalidimg);
driver.quit();
} catch (Exception e) {
e.printStackTrace();
System.out.println(e.getMessage());
}
}



public static void verifyimgActive(WebElementimg) {
try {
HttpResponse response = new DefaultHttpClient().execute(new
HttpGet(img.getAttribute("src")));
if (response.getStatusLine().getStatusCode() != 200)
invalidimg++;
}
catch (Exception e) {
e.printStackTrace();
}
}
}
```


Q #185) How to handle Ajax popup window?

By using getWindowHandles() and obj.switchTo.window(windowid) we can handle popups using
explicit wait and driver.swtchT0.window("name") commands for your requirements.

Q #186) How to handle auto complete box in web driver?

How to handle autocomplete box in web driver

How to handle autocomplete box in web driver?

driver.findElement(By.id("your searchBox")).sendKeys("your partial keyword");

Thread.sleep(3000);

List <WebElement>listItems = driver.findElements(By.xpath("your list item locator"));

listItems.get(0).click();

```
driver.findElement(By.id("your searchButton")).click();
```

Q #187) How to get the name of browser using Web Driver?

```
public class JsExecute

{

WebDriver driver;

JavascriptExecutorjs;

@Before

public void setUp() throws Exception

{

driver=new FirefoxDriver();

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

}

@Test

public void test()

{

JavascriptExecutorjs = (JavascriptExecutor) driver;

System.out.println(js.executeScript("return navigator.appCodeName"));

}}
```

OR

```
String s = (String) ((JavascriptExecutor) driver).executeScript("return
navigator.userAgent;");

System.out.println("Browser name : " + s);
```

Q #188) How to handle colors in web driver?

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

Example

String col = driver.findElement(By.id(locator)).getCssValue("color");

Q #189) How to pass parameters from testng.xml into test case.

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.htmlunit.HtmlUnitDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class Parallelexecution {

private WebDriver driver = null;

@BeforeTest
@Parameters({ "BROWSER" })
public void setup(String BROWSER) {
System.out.println("Browser: " + BROWSER);

if (BROWSER.equals("FF")) {
System.out.println("Firefox Browser is selected");
driver = new FirefoxDriver();
} else if (BROWSER.equals("IE")) {
System.out.println("Internet Explorer Browser is selected");
driver = new InternetExplorerDriver();
} else if (BROWSER.equals("HU")) {
System.out.println("Html Unit Browser is selected");
driver = new HtmlUnitDriver();
} else if (BROWSER.equals("CH")) {
System.out.println("Google chrome Browser is selected");
driver = new ChromeDriver();
}
}
```

```
@Test
public void testParallel() throws Exception {
driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/100-selenium-
interview-questions.html");

}
}
```

Q #190) How to get text from captcha image??

```
driver.findElement(By.xpath(".//*[@id='SkipCaptcha']")).click();

String attr = ie.findElement(By.xpath(".//*[@id='SkipCaptcha']")).getAttribute("value");

System.out.println("The value of the attribute 'Name' is " + attr);
```

Q #191) Is there a way to click hidden LINK in web driver?

```
String Block1 = driver.findElement(By.id("element ID"));

JavascriptExecutor js1=(JavascriptExecutor)driver;

js1.executeScript("$("+Block1+").css({'display':'block'});");
```

Q #192) What Class Extends Web Driver?

AndroidDriver, ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver, InternetExplorerDriver, IPhoneDriver, PhantomJSDriver, RemoteWebDriver, SafariDriver

Q #193) What are the APIs that support Web Driver?

API are nothing but collection of all selenium commands for Locating UI Elements (WebElements),Fetching a Page,User Input etc…

Q #194) How to disable cookies in browser.

Using deleteAllVisibleCookies() in selenium

Q #195) We have heard about frameworks well it can be broadly classified into these TDD, BDD and ATDD frameworks .What's the Difference?

TDD- Test Driven Development, Behaviour Driven Development & Acceptance TestDriven Development

Well, you could see the above Acronyms buzzing over all Automation folks. I was not sure on what it means and How it differs each other. How each methodology will benefit? and where exactly it will help in the Development Life cycle.

Finally, after some analysis I had found out the differences and posting it here. Readers are always welcomed to correct me if I am wrong.

First lets list out what exactly each methodology does means

TDD – Test Driven Development

Its also called test-driven design, is a method of software development in which unit testing is repeatedly done on source code. Write your tests watch it fails and then refactor it. The concept is we write these tests to check if the code we wrote works fine. After each test, refactoring is done and then the same or a similar test is performed again. The process is iterated as many times as necessary until each unit is functionally working as expected. TDD was introduced first by XP. I believe I have explained enough in simple terms.

BDD – Behaviour Driven Development

Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design

DDD-Domain Driven Testing
BDD is similar in many ways to TDD except that the word "test" is replaced with the word "Behaviour". It's purpose is to help the the folks devising the system (i.e., the developer) identify appropriate tests to write–that is, tests that reflect the behavior desired by the stakeholders. BDD is usually done in very English-like language helps the Domain experts to understand the implementation rather than exposing the code level tests. Its defined in a GWT format, GIVEN WHEN & THEN.

Q #196) How to change user agent in Firefox by selenium web driver.

FirefoxProfile profile = new FirefoxProfile();

profile.setPreference("general.useragent.override", "some UA string");

Web Driver driver = new FirefoxDriver(profile);

Q #197) What is the MOST challenging test problem in my career in Automation?

In my career

- Changing XPATHS' between testing server and production server-by keeping generic xpath
- Keep separate property files for production and UAT
- automating flash apps
- Mobile Automation

Q #198) Suppose developer changed the existing image to new image with same xpath. Is test case pass or fail?

Pass

Q #199) How to handle network latency using selenium?

Using driver.manage.pageLoadingtime for network latency

Q #200) How does u handle dynamic elements without usingxpath (with example?)

By using classname or css.

Q #201) How to work with radio button in web driver?

We can select the value from the drop down by using 3 methods.

selectByVisibleText - select by the text displayed in drop down

selectByIndex - select by index of option in drop down

selectByValue - select by value of option in drop down

<select id="44"><option value="1">xyz</option>

<option value="2">abc</option>

<option value="3">pqr</option>

</select>

WebElement e = driver.findElement(By.id("44"));

Select selectElement=new Select(e);

// both of the below statements will select first option in the weblist

```
selectElement.selectByVisibleText("xyz");

selectElement.selectByValue("1");
```

Q #202) How to work with dynamic web table?

You can get the total number of **<tr>** tags within a **<td> or <th>** tag by giving the xpath of the **<td> or <th>** element by using this function -

```
List<WebElement>ele = driver.findElements(By.xpath("Xpath of the table"));
```

Now you can use a for each loop to loop through each of the **<tr>** tags in the above list and then read each value by using getText() method.

Q #203) Detail about TestNG Test Output folder.

It is the directory where reports are generated. Every time tests run in a suite, TestNG creates index.html and other files in the output directory.

Q #204) In frame if no frame Id as well as no frame name then which attribute I should consider throughout our script.

You can go like this.....driver.findElements(By.xpath("//iframe"))...

Then it will return List of frames then switch to each and every frame and search for the locator which you want then break the loop.

Q #205) What is object repository?

It is collection of object names their properties, attributes and their values .It maye be excel, XML, property file or text file.

Q #206) What are the different Parameters for @Test annotation?

Parameters are keywords that modify the annotation's function.

Q #207) Can we run group of test cases using TestNG?

Test cases in group in Selenium using TestNG will be executed with the below options.

If you want to execute the test cases based on one of the group like regression test or smoke test

```
@Test(groups = {"regressiontest", "smoketest"})
```

Q #208) In what all case we have to go for "JavaScript executor".

Consider FB main page after you login. When u scrolls down, the updates get loaded. To handle this activity, there is no selenium command. So you can go for javascript to set the scroll down value like driver.executeScript("window.scrollBy(0,200)", "");

Q #209) What are the different assertions in SIDE?

Assertions are like Accessors, but they verify that the state of the application conforms to what is expected. Examples include "make sure the page title is X" and "verify that this checkbox is checked".

All Selenium Assertions can be used in 3 modes: "assert", "verify", and "waitFor".

For example, you can "assertText", "verifyText" and "waitForText". When an "assert" fails, the test is aborted. When a "verify" fails, the test will continue execution, logging the failure. This allows a single "assert" to ensure that the application is on the correct page, followed by a bunch of "verify" assertions to test form field values, labels, etc.

"waitFor" commands wait for some condition to become true (which can be useful for testing Ajax applications). They will succeed immediately if the condition is already true. However, they will fail and halt the test if the condition does not become true within the current timeout setting.

Q #210) How to handle alerts and confirmation boxes.

Confirmation boxes and Alerts are handled in same way in selenium.

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

alert.dismiss(); //Click Cancel or Close window operation

alert.accept(); //Click OK

Handle Confirmation boxes via JavaScript,

driver.executeScript("window.confirm = function(message){return true;};");

Q #211) How to mouse hover on an element?

Actions action = new Actions(webdriver);

```
WebElement we = webdriver.findElement(By.xpath("html/body/div[13]/ul/li[4]/a"));

action.moveToElement(we).moveToElement(webdriver.findElement(By.xpath("/expre
ssion-here"))).click().build().perform();
```

Q #212) How to switch between the windows?

```
private void handlingMultipleWindows(String windowTitle) {

Set<String> windows = driver.getWindowHandles();

for (String window : windows) {

driver.switchTo().window(window);

if (driver.getTitle().contains(windowTitle))

{

return;

} } }
```

Q #213) How to switch between frames?

WebDriver's driver.switchTo().frame() method takes one of the three possible arguments:

•A number.

Select a frame by its (zero-based) index. That is, if a page has three frames, the first frame would be at index "0", the second at index "1" and the third at index "2". Once the frame has been selected, all subsequent calls on the WebDriver interface are made to that frame.

•A name or ID.

Select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID.

•A previously found WebElement.

Select a frame using its previously located WebElement.

Get the frame by its id/name or locate it by driver.findElement() and you'll be good.

Q #214) What is actions class in web driver?

Actions class with web Driver help is Sliding element, Resizing an Element, Drag & Drop, hovering a mouse, especially in a case when dealing with mouse over menus.

Dragging & Dropping an Element:

```
importorg.openqa.selenium.By;

importorg.openqa.selenium.WebDriver;

importorg.openqa.selenium.WebElement;

importorg.openqa.selenium.firefox.FirefoxDriver;

importorg.openqa.selenium.interactions.Actions;

public class testDragandDrop {

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

WebDriver driver = new FirefoxDriver();

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

WebElementdraggable = driver.findElement(By.xpath("//*[@id='draggable']"));

WebElement droppable = driver.findElement(By.xpath("//*[@id='droppable']"));

Actions action = new Actions(driver);

action.dragAndDrop(draggable, droppable).perform();

}

}
```

Sliding an Element:

```
importorg.openqa.selenium.By;

importorg.openqa.selenium.WebDriver;

importorg.openqa.selenium.WebElement;
```

```
importorg.openqa.selenium.firefox.FirefoxDriver;
importorg.openqa.selenium.interactions.Actions;
public class testSlider {
/**
* @paramargs
* @throws InterruptedException
*/
public static void main(String[] args) throws InterruptedException {
WebDriver driver = new FirefoxDriver();
driver.get("http://jqueryui.com/resources/demos/slider/default.html");
WebElement slider = driver.findElement(By.xpath("//*[@id='slider']/a"));
Actions action = new Actions(driver);
Thread.sleep(3000);
action.dragAndDropBy(slider, 90, 0).perform();
}
}
```

Re-sizing an Element:

```
importorg.openqa.selenium.By;
importorg.openqa.selenium.WebDriver;
importorg.openqa.selenium.WebElement;
importorg.openqa.selenium.firefox.FirefoxDriver;
importorg.openqa.selenium.interactions.Actions;
public class testResizable {
```

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

WebDriver driver = new FirefoxDriver();

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

WebElement resize = driver.findElement(By.xpath("//*[@id='resizable']/div[3]"));

Actions action = new Actions(driver);

action.dragAndDropBy(resize, 400, 200).perform();

}

}
```

Q #215) Write down scenarios which we can't automate?

Barcode Reader, Captcha etc.

Q #216) Differences between jxl and ApachePOI.

- jxl does not support XLSX files
- jxl exerts less load on memory as compared to ApachePOI
- jxl doesn't support rich text formatting while ApachePOI does.
- jxl has not been maintained properly while ApachePOI is more up to date.
- Sample code on Apache POI is easily available as compare to jxl.

Q #217) How to ZIP files in Selenium with an Example?

```
// Sample Function to make zip of reports
public static void zip(String filepath){
try
{
File inputFolder=new File('Mention file path her");
File outputFolder=new File("Reports.zip");
ZipOutputStream out = new ZipOutputStream(new BufferedOutputStream(new
FileOutputStream(outputFolder)));
BufferedInputStream in = null;
byte[] data = new byte[1000];
String files[] = inputFolder.list();
for (int j=0; j<files.length; i++)
{
in = new BufferedInputStream(new FileInputStream
```

```
(inputFolder.getPath() + "/" + files[j]), 1000);
out.putNextEntry(new ZipEntry(files[j]));
inttotalcount;
while((totalcount= in.read(data,0,1000)) != -1)
{
out.write(data, 0, totalcount);
}
out.closeEntry();
}
out.flush();
out.close();
}
catch(Exception e)
{
e.printStackTrace();
return "Fail - " + e.getMessage();
}
}
```

Q #218) How to do Applet testing using selenium?

```
// selenium setup
selenium = new DefaultJavaSelenium("localhost",4444, browserString , url);
selenium.start();
selenium.open(url);

// get the appletfixure to control fest JAppletFixture
AppletFixture dialog = selenium.applet(LIST_APPLET_ID)

// fest similar API for autmation testing
dialog.comboBox("domain").select("Users");
dialog.textBox("username").enterText("alex.ruiz");
dialog.button("ok").click();
```

Q #219) If Default port no is busy how to change port no?

We can use any port number which is valid.. First create an object to remote control configuration.
Use 'setPort' method and provide valid port number(4545,5555,5655, etc).. There after attach this
remote control configuration object to selenium server..i.e

```
RemoteControlConfiguration r= new RemoteControlConfiguration();
```

r.setPort(4567);

SeleniumServer s= new SeleniumServer(r);

Q #220) Does Selenium support https protocols?

Yes

Q #221) Majorly asked test scenario with framework in Interviews?

Majorly asked are:

· Login for Gmail scenario

· Goggle search and finding no of results

· Downloading a file and save it

· Checking mails and deleting them

· Do shopping in flipkart.com

Q #222) I want to find the location of ""b"" in the below code, how can I find out without using xpath, name,id, csslocator, index.

<Button>a</button>

<Button>b</button>

<Button>c</button>

Ans

driver.findElement(By.xpath("//*[contains(text(),'b')]")).click();

or

//div/button[contains(text(),'b']