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	WebDriver doesn't support record and playback feature
Server Requirement	Selenium IDE doesn't require any server to be started before executing the test scripts	Selenium RC requires server to be started before executing the test scripts	WebDriver doesn't require any server to be started before 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	Selenium RC is semi object	WebDriver is a purely

Feature	Selenium IDE	Selenium RC	WebDriver
	object oriented tool	oriented tool	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/Andriod) 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 doesn't support listeners	Selenium RC doesn't support listeners	WebDriver 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) Which is the latest Selenium tool?

WebDriver

Q #18) 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 #19) 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 #20) What are the different types of waits available in WebDriver?

There are two types of waits available in WebDriver:

1. Implicit Wait

2. Explicit Wait

Implicit Wait: Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

Explicit Wait: Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

Q #21) 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 #22) 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 #23) 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 #24) 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 #25) 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 #26) 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 #27) 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 #28) 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 #29) 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   int listSize = elementList.size();
5   for (int i=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 #30) 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 #31) 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 #32) 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.
- void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

Syntax:
// accepting javascript alert
Alert alert = driver.switchTo().alert();
alert.accept();

Q #33) 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 #34) 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 #35) 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=new Actions(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  WebElement subLinkOption=driver.findElement(By.id("id of the sub link"));
7  subLinkOption.click();
```

Q #36) How to retrieve css properties of an element?

The values of the css properties can be retrieved using a get() method:

Syntax:
driver.findElement(By.id("id")).getCssValue("name of css attribute");
driver.findElement(By.id("id")).getCssValue("font-size");

Q #37) How to capture screenshot in WebDriver?

 File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

Q #38) What is Junit?

Junit is a unit testing framework introduced by Apache. Junit is based on Java.

Q #39) 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 #40) 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 #41) How to set test case priority in TestNG?

Setting Priority in TestNG

Code Snippet

```
1  package TestNG;
2  import org.testng.annotations.*;
3  public class SettingPriority {
4   @Test(priority=0)
5   public void method1() {
6   }
7  @Test(priority=1)
8   public void method2() {
9   }
10   @Test(priority=2)
11   public void method3() {
12   }
13   }
```
Test Execution Sequence:

1. Method1
2. Method2
3. Method3

Q #42) 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 #43) 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 #44) 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 #45) How can I read test data from excels?

Test data can efficiently be read from excel using JXL or POI API. See detailed tutorial here.

Q #46) What is the difference between POI and jxl 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 #47) 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 #48) 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 #49) Can captcha be automated?

No, captcha and bar code reader cannot be automated.

Q #50) 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.

Q #51) What is the difference between setSpeed() and sleep() methods?

Both will delay the speed of execution.

Thread.sleep () : It will stop the current (java) thread for the specified period of time. Its done only once

- It takes a single argument in integer format

Ex: thread.sleep(2000)- It will wait for 2 seconds

- It waits only once at the command given at sleep

SetSpeed () : For specific amount of time it will stop the execution for every selenium command.

- It takes a single argument in integer format

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

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

Q #52) Explain what is the difference between Borland Silk and Selenium?

Silk Test Tool	Selenium Test Tool
• Borland Silk test is not a free testing tool	• Selenium is completely free test automation tool
• Silk test supports only Internet Explorer and Firefox	• Selenium supports many browsers like Internet Explorer, Firefox, Safari, Opera and so on
• Silk test uses test scripting language	• Selenium suite has the flexibility to use many languages like Java, Ruby,Perl and so on
• Silk test can be used for client server applications	• Selenium can be used for only web application

Q #53) Mention 5 different exceptions you had in Selenium web driver?

The 5 different exceptions you had in Selenium web drivers are

- WebDriverException
- NoAlertPresentException

- NoSuchWindowException
- NoSuchElementException
- TimeoutException

Q #54) How will you use Selenium to upload a file ?

You can use "type"command to type in a file input box of upload file. Then, you have to use "Robot" class in JAVA to make file upload work.

Q #55) What is the difference between getWindowhandles() and getwindowhandle() ?

getwindowhandles(): It is used to get the address of all the open browser and its return type is iterator<String>

getwindowhandle(): It is used to get the address of the current browser where the control is and return type is string

Q #56) Explain how you can use recovery scenario with Selenium?

Recovery scenarios depends upon the programming language you use. If you are using Java then you can use exception handling to overcome same. By using "Try Catch Block" within your Selenium WebDriver Java tests

Q #57) How will you handle working with multiple windows in Selenium ?

We can use the command **selectWindow** to switch between windows. This command uses the title of Windows to identify which window to switch to

Q #58) Mention what is desired capability? How is it useful in terms of Selenium?

The desired capability is a series of key/value pairs that stores the browser properties like browser name, browser version, the path of the browser driver in the system, etc. to determine the behavior of the browser at run time.

For Selenium,

- It can be used to configure the driver instance of Selenium WebDriver.
- When you want to run the test cases on a different browser with different operating systems and versions.

Q #59) Mention when to use AutoIT?

Selenium is designed to automate web-based applications on different browsers. But to handle window GUI and non-HTML popups in the application you need AutoIT.

Q #60) How many types of Webdriver APIs are available in Selenium?

Below is the list of driver classes that you can use for the browser automation.

- AndroidDriver,
- ChromeDriver,
- EventFiringWebDriver,

- FirefoxDriver,
- HtmlUnitDriver,
- InternetExplorerDriver,
- IPhoneDriver,
- IPhoneSimulatorDriver,
- RemoteWebDriver.

Q #61) How do you read a JavaScript variable in Selenium WebDriver?

It's easy to access any JavaScript variable from the Selenium Webdriver test scripts.

```
// Initialize the JS object.
JavascriptExecutor JS = (JavascriptExecutor) webdriver;
// Get the current site title.
String sitetitle = (String)JS.executeScript("return document.title");
System.out.println("My Site Title: " + sitetitle);
```

Q #62) How can you check the state of a checkbox/radio button?

We can call the isSelected() method to test the status of these elements.

```
boolean test = driver.findElement(By.xpath("checkbox/radio button Xpath")).isSelected();
```

Q #63) How would you simulate the right click operation in WebDriver?

You can make use of the Actions class method.

```
Actions test = new Actions(driver); // Here, driver is the object of WebDriver class.
 test.moveToElement(element).perform();
 test.contextClick().perform();
```

Q #64) What is the name of the super interface of the Webdriver?

SearchContext

Q #65) Can we test APIs or web services using selenium webdriver?

No selenium webdriver uses browser's native method to automate the web applications. Since web services are headless, so we cannot automate web services using selenium webdriver.

Q #66) How can we locate an element by only partially matching its attributes value in Xpath?

Using contains() method we can locate an element by partially matching its attribute's value. This is particularly helpful in the scenarios where the attributes have dynamic values with certain constant part.

xPath expression = //*[contains(@name,'user')]

The above statement will match the all the values of name attribute containing the word 'user' in

them.

Q #67) How can we move to parent of an element using XPath?

Using '..' expression in XPath we can move to parent of an element.

Q #68) What is the syntax of finding elements by id using CSS Selector?

By #idValue we can select all the element belonging to a particluar class e.g. '#userId' will select the element having id - userId.

Q #69) How can we submit a form in selenium?

Ans. Using submit() method we can submit a form in selenium.

driver.findElement(By.id("form1")).submit();

Q #70) what are various ways to refresh browser using selenium?

There a multiple ways to refresh a page in selenium-

- Using *driver.navigate().refresh()* command
- Using *sendKeys(Keys.F5)* on any textbox on the webpage
- Using driver.get("URL") on the current URL or using *driver.getCurrentUrl()*
- Using driver.navigate().to("URL") on the current URL or
  driver.navigate().to(driver.getCurrentUrl());

Q #71) How to delete cookies in selenium?

 Using deleteAllCookies() method-

driver.manage().deleteAllCookies();

Q #72) What are some expected conditions that can be used in Explicit waits?

Ans. Some of the commonly used expected conditions of an element that can be used with expicit waits are-

- elementToBeClickable(WebElement element or By locator)
- stalenessOf(WebElement element)
- visibilityOf(WebElement element)
- visibilityOfElementLocated(By locator)
- invisibilityOfElementLocated(By locator)
- attributeContains(WebElement element, String attribute, String value)
- alertIsPresent()
- titleContains(String title)
- titleIs(String title)
- textToBePresentInElementLocated(By, String)

Q #73) What are the different keyboard operations that can be performed in selenium?

The different keyboard operations that can be performed in selenium are-

1. .sendKeys("sequence of characters") - Used for passing charcter sequesnce to an input or textbox element.
2. .pressKey("non-text keys") - Used for keys like control, function keys etc that ae non text.
3. .releaseKey("non-text keys") - Used in conjuntion with keypress event to simulate releasing a key from keyboard event.

Q #74) Write the code to right click an element in selenium?

Code to right click an element in selenium-

Actions action = **new** Actions(driver);

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

action.contextClick(element).perform();

Q #75) How can we check if an element is enabled for interaction on a web page?

Ans. Using isEnabled method we can check if an element is enabled or not.

driver.findElement(By locator).isEnabled();

Q #76) How to handle HTTPS website in selenium? or How to accept the SSL untrusted connection?

Ans. Using profiles in firefox we can handle accept the SSL untrusted connection certificate. Profiles are basically set of user preferences stored in a file.

FirefoxProfile profile = **new** FirefoxProfile();

profile.setAcceptUntrustedCertificates(**true**);

profile.setAssumeUntrustedCertificateIssuer(**false**);

WebDriver driver = **new** FirefoxDriver(profile);

Q #77) What is Page Object Model or POM?

Ans. Page Object Model(POM) is a design pattern in selenium. A design pattern is a solution or a set of standards that are used for solving commonly occuring software problems.
Now coming to POM - POM helps to create a framework for maintaining selenium scripts. In POM for each page of the application a class is created having the web elements belonging to the page and methods handling the events in that page. The test scripts are maintained in seperate files and the methods of the page object files are called from the test scripts file.

Q #78) What are the advantages of POM?

Ans. The advantages are POM are-

1. Using POM we can create an Object Repository, a set of web elements in seperate files along with their associated functions. Thereby keeping code clean.
2. For any chnage in UI(or web elements) only page object files are required to be updated leaving test files unchanged.
3. It makes code reusable and maintable.

Q #79) What is Page Factory?

Ans. Page factory is an implementation of Page Object Model in selenium. It provides @FindBy annotation to find web elements and PageFactory.initElements() method to initialize all web elements defined with @FindBy annotation.

Q #80) What are some advantages of selenium grid?

The advantages of selenium grid are-

1. It allows running test cases in parallel thereby saving test execution time.
2. Multi browser testing is possible using selenium grid by running the test on machines having different browsers.
3. It is allows multi-platform testing by configuring nodes having different operating systems.

Q #81) What is a hub in selenium grid?

A hub is server or a central point in selenium grid that controls the test executions on the different machines.

Q #82) What is a node in selenium grid?

Nodes are the machines which are attached to the selenium grid hub and have selenium instances running the test scripts. Unlike hub there can be multiple nodes in selenium grid.

Q #83) What are some advantages of testNG?

Following are the advantages of testNG-

1. TestNG provides different assertions that helps in checking the expected and actual results.
2. It provides parallel execution of test methods.
3. We can define dependency of one test method over other in TestNG.
4. We can assign priority to test methods in selenium.
5. It allows grouping of test methods into test groups.
6. It allows data driven testing using @DataProvider annotation.
7. It has inherent support for reporting.
8. It has support for parameterizing test cases using @Parameters annotation.

Q #84) What is the use of testng.xml file?

testng.xml file is used for configuring the whole test suite. In testng.xml file we can create test suite, create test groups, mark tests for parallel execution, add listeners and pass parameters to test scripts. Later this testng.xml file can be used for triggering the test suite.

Q #85) How can we create data driven framework using testNG?

Using @DataProvider we can create a data driven framework in which data is passed to the associated test method and multiple iteration of the test runs for the different test data values passed from the @DataProvider method. The method annotated with @DataProvider annotation return a 2D array of object.

```
@DataProvider(name = "dataProvider1")
  public Object[][] dataProviderMethod1() {
    return new Object[][] {{"admin","admin123"}, {"admin123","admin"},
{"admin123","admin123"}};
  }


  //This method is bound to the above data provider returning 2D array of 3*2 matrix
  //The test case will run 3 times with different set of values
  @Test(dataProvider = "dataProvider1")
  public void sampleTest(String s1, String s2) {
    System.out.println(s1 + " " + s2);
  }
```

Q #86) What is difference between @Factory and @DataProvider annotation?

Ans. @Factory method creates instances of test class and run all the test methods in that class with different set of data.
Whereas, @DataProvider is bound to individual test methods and run the specific methods multiple times.

Q #87) How can we make one test method dependent on other using TestNG?

Using dependsOnMethods parameter inside @Test annotation in testNG we can make one test method run only after successful execution of dependent test method.

```
@Test(dependsOnMethods = { "preTests" })
```

Q #88) What are some common assertions provided by testNG?

Some of the common assertions provided by testNG are-

1. assertEquals(String actual, String expected, String message)
2. assertNotEquals(double data1, double data2, String message)
3. assertFalse(boolean condition, String message)
4. assertTrue(boolean condition, String message)
5. assertNotNull(Object object)
6. fail(boolean condition, String message)
7. true(String message)

Q #89) How can we run test cases in parallel using TestNG?

Ans. In order to run the tests in parallel just add these two key value pairs in suite-

- parallel="{methods/tests/classes}"
- thread-count="{number of thread you want to run simultaneously}".

`<suite name="ArtOfTestingTestSuite" parallel="methods" thread-count="5">`

Q #90) Name an API used for logging in Java.

Ans. Log4j is an open source API widely used for logging in Java. It supports multiple levels of logging like - ALL, DEBUG, INFO, WARN, ERROR, TRACE and FATAL.

Q #91) How to login into any site if its showing any authetication popup for user name and pass ?

pass the username and password with url.

Syntax- http://username:password@url
ex- http://creyate:seleniumwebdriver@alpha.beta.com

Q #92) How to count total number of rows of a table using Selenium ?

```
List {WebElement} rows = driver.findElements(By.className("//table[@id='tableID']/tr"));
int totalRow = rows.size();
```

Q #93) How to call AutoIT script from Selenium Webdriver?

```
Runtime.getRuntime().exec("complete AutoIt .exe filepath");
```

Q #94) Which technique should you consider using throughout the script "if there is neither frame id nor frame name"?

If neither frame name nor frame id is available, then we can use **frame by index**.

Let's say, that there are 3 frames in a web page and if none of them have frame name and frame id, then we can still select those frames by using frame (zero-based) index attribute. Each frame will have an index number. 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 will be made to that frame.

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

Q #95) How to skip a method or a code block in TestNG?

If you want to skip a particular test method, then you can set the 'enabled' parameter in test annotation to false.

@Test(enabled = false)

By default, the value of 'enabled' parameter will be true. Hence it is not necessary to define the annotation as true while defining it.

Q #96) How to handle internationalisation through web 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 #97) 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 #98) What are the technical challenges with Selenium?
According to contributors on Guru99.com, the technical challenges with Selenium include.

- Selenium supports only web based applications
- It does not support the Bitmap comparison
- For any reporting-related capabilities, the user must depend on third-party tools
- There is no vendor support, unlike the backup available for commercial tools such as HP UFT
- Maintainability of objects can be difficult, as Selenium has no object repository concept.

Q #99) What is Object Repository ?

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

Q #100) For Database Testing in Selenium Webdriver what API is required?

For Database Testing in Selenium Webdriver, you need JDBC (Java Database Connectivity) API. It allows you to execute SQL statements.