**1) What is the difference between Manual and Automation Testing?**

* Manual Testing shows lower accuracy due to the higher possibilities of human errors. Automation Testing depicts a higher accuracy due to computer-based testing eliminating the chances of errors.
* Manual Testing needs time when testing is needed at a large scale. Automation Testing easily performs testing at a large scale with the utmost efficiency.
* Manual Testing takes more time to complete a cycle of testing, and thus the turnaround time is higher. Automation Testing completes a cycle of testing within record time and thus the turnaround time is much lower.
* Manual Testing should be used to perform Exploratory Testing, Usability Testing and Ad-hoc Testing to exhibit the best results. Automation Testing should be used to perform Regression Testing, Load Testing, Performance Testing and Repeated Execution for best results.
* Exploratory testing is possible in Manual Testing. Automation does not allow random testing.
* The initial investment and ROI in the Manual testing is lower compared to Automation testing in the long run. The initial investment in the automated testing is higher and the ROI is better in the long run.

**2) What are the benefits of Automation Testing?**

* **ROI:** Even though the initial investment is high, return on investment over time is much better.
* **Running tests:** Tests can be run automatically or can be scheduled at a particular time.
* **Fewer human resources:** Manual testers are not required.
* **Reusability:** Scripts are reusable.
* **Bugs Automation** helps you find bugs in the early stages of software development.
* **Reliability:** Automated testing is more reliable and way quicker.
* **Parallel testing:** Automated tests can be run parallel on multiple devices.

**3) Which Test cases needs to be automated?**

* Tests that need to be run against every build/release of the application, such as smoke test, sanity test and regression test.
* Tests that need to run against multiple configurations — different OS & Browser combinations.
* Tests that involve inputting large volumes of data, such as filling up very long forms.
* Tests that take a long time to perform and may need to be run during breaks or overnight.

**4) What are the popular test automation tools for functional testing?**

* Selenium
* Unified Functional Testing
* Test Complete
* Ranorex
* Tosca

**5) What is the main purpose of Automation Testing?**

Automation testing is the best way to increase the effectiveness, efficiency and coverage of your software testing.

**6) What is the goal of Automation Testing?**

* To reduce Testing Cost and Time.
* To reduce Redundancy.
* To speed up the Testing Process.
* To help improve Quality.
* To improve Test coverage.
* To reduce Manual Intervention.

**7) Why Selenium should be selected as a Test tool?**

* Language & Framework support: Selenium supports all major languages like Java, Python, JavaScript, C#, Ruby, and Perl programming languages for software test automation. Also, every Selenium supported language has dedicated frameworks which help in writing test script for Selenium test automation.
* Open source: Being an open source tool, Selenium is a publicly accessible automation framework and is free, with no upfront costs.
* Multi browser support: Selenium script is compatible with all browsers like Chrome, Safari, IE, etc …
* Support across various OS: Selenium is yet a highly portable tool that supports and can work across different operating systems like Windows, Linux, Mac OS, UNIX, etc.
* Ease of implementation: Selenium automation framework is very easy-to-use tool. Selenium provides a user-friendly interface that helps create and execute test scripts easily and effectively.
* Reusability and Integrations: Selenium needs third-party frameworks and add-ons to broaden the scope of testing. For example, it can integrate with TestNG and JUnit for managing test cases and generating reports.
* Parallel Test Execution: With the help of Selenium Grid, we can execute multiple tests in parallel, hence reducing the test execution time.

**8) What are the testing types that can be supported by Selenium?**

* Functional Testing
* Regression Testing
* Sanity Testing
* Smoke Testing
* Responsive Testing
* Cross Browser Testing
* UI testing (black box)
* Integration Testing

**9) What are the limitations of Selenium?**

* Selenium does not support automation testing for desktop applications.
* Since Selenium is open source software, you have to rely on community forums to get your technical issues resolved.
* It does not have built-in Object Repository like UTF/QTP to maintain objects/elements in centralized location.
* Selenium does not have any inbuilt reporting capability and you have to rely on plug-ins like JUnit and TestNG for test reports.

**10) What is the difference between Selenium IDE, Selenium RC and Selenium WebDriver?**

**Selenium IDE:**  
– It only works in Mozilla browser.  
– It supports Record and playback  
– Doesn’t required to start server before executing the test script.  
– It is a GUI Plug-in  
– Core engine is Javascript based  
– Very simple to use as it is record & playback.  
– It is not object oriented  
– It does not supports listeners  
– It does not support to test iphone/Android applications

**Selenium RC:**  
– It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc.  
– It doesn’t supports Record and playback  
– Required to start server before executing the test script.  
– It is standalone java program which allow you to run Html test suites.  
– Core engine is Javascript based  
– It is easy and small API  
– API’s are less Object oriented  
– It does not supports listeners  
– It does not support to test iphone/Android applications

**Selenium Webdriver:**  
– It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc.  
– It doesn’t supports Record and playback  
– Doesn’t required to start server before executing the test script.  
– It is a core API which has binding in a range of languages.  
– Interacts natively with browser application  
– As compared to RC, it is bit complex and large API.  
– API’s are entirely Object oriented  
– It supports the implementation of listeners  
– It support to test iphone/Android applications

**11) When should I use Selenium IDE?**

Because of its simplicity, Selenium IDE should only be used as a prototyping tool, not an overall solution for developing and maintaining complex test suites.

**12) What is Selenese?**

Selenese is the set of selenium commands which are used to test your web application. Tester can test the broken links, existence of some object on the UI, Ajax functionality, Alerts, window, list options and lot more using selenese.

**13) What is the difference between Assert and Verify commands?**

In case of the “Assert” command, as soon as the validation fails the execution of that particular test method is stopped and the test method is marked as failed. Whereas, in case of “Verify”, the test method continues execution even after the failure of an assertion statement.

**14) What is Same Origin Policy and how it can be handled? How to overcome same origin policy through web driver?**

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

**15) How do you use findElement() and findElements()?**

WebElement login= driver.findElement(By.linkText("Login"));

List<WebElement> listOfElements = driver.findElements(By.xpath("//div"));

**16) Can Selenium handle window based pop up?**

No. Selenium cannot handle window based pop-up on its own but can use third party tools.

**17) How can we handle window based pop up using Selenium?**

We can handle window based popups using some third party tools such as AutoIT, Robot class.

**18) How can we get** **Window Handle in Selenium?**

**Step 1:** After opening the website, we need to get the main window handle by using driver.getWindowHandle();  
**Step 2:** We now need to get all the window handles by using driver.getWindowHandles();  
**Step 3:** We will compare all the window handles with the main Window handles and perform the operation the window which we need.

**19) How to assert title of the web page?**

String actualTitle = driver.getTitle();

String expectedTitle = "Title of Page";

assertEquals(expectedTitle,actualTitle);

**20) How to mouse hover on a web element using WebDriver?**

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

builder.moveToElement(hoverElement).perform();

**21) How to retrieve CSS Properties of an element?**

driver.findElement(By.id("by-id")).getCssValue("font-size");

**22) What is JUnit?**

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks.

**23) What are JUnit annotations?**

Annotation is a special form of syntactic meta-data that can be added to Java source code for better code readability and structure.  
Some of them are:  
– @Before  
– @After  
– @BeforeClass  
– @AfterClass  
– @Ignore  
– @RunWith  
– @Test

**24) What is TestNG and what is its use?**

TestNG is an automation testing framework in which NG stands for “Next Generation”. TestNG is inspired from JUnit which uses the annotations (@). Using TestNG you can generate a proper report, and you can easily come to know how many test cases are passed, failed and skipped.

**25) How is TestNG better than JUnit?**

* In TestNG Annotations are easy to understand over JUnit.
* TestNG enable you to grouping of test cases easily which is not possible in JUnit.
* TestNG allows us to define the dependent test cases each test case is independent to other test case.
* Parallel execution of Selenium test cases is possible in TestNG.

**26) Define Selenium?**

SELENIUM is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms.  
You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts.

**27) What are the top Selenium alternatives available for free?**

– Robot Framework  
– Ranorex  
– TestComplete  
– Tricentis Tosca  
– Soap UI

**28) What are different versions of Selenium available you have used and what are the additional features you have seen from the previous versions?**

Mostly worked with version 3 but now version 4 is released. Following are additional features which are expected in new version:  
– Selenium 4 WebDriver is completely W3C Standardized  
– The Selenium IDE support for Chrome is available now  
– Improved Selenium Grid  
– Better Debugging  
– Better Documentation

**29) What is the principle difference between a Data-driven framework and a Keyword Driven Framework?**

Data driven framework includes different test data sources like flat files, databases or XML but Keyword Driven Framework involves business keywords which represent a feature or user actions.

**30) What are the two most common practices for automation testing?**

– Identify which test cases can be automated and which cannot be automated.  
– Do not rely completely on UI Automation.

**31) What is Test Driven Development (TDD) Framework?**

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, then the code is improved so that the tests pass.  
The following is an sequence of steps which are followed:  
– Add a test  
– Run all tests and see if it fails  
– Write the code  
– Run Tests  
– Refactor code  
– Repeat

**32) What is Behavior Driven Development (BDD) Framework?**

BDD is a software development process for teams to create simple scenarios on how an application should behave from the end user’s perspective. The goal of implementing BDD testing is to improve collaboration between stakeholders, such as developers, testers, product managers, and business analysts.

**33) What are the main traits of a good Software Test Automation framework?**

Maintainability, Reliability, Flexibility, Efficiency, Portability, Robustness, and Usability are main attributes of a good automation framework.

**34) What are the challenges have you faced with Selenium and how did you overcome them?**

– Handling dynamic content: We can use Implicit Wait or Explicit Wait to overcome this challenge  
– Handling popup up windows: We can use getWindowHandle and getWindowHandles methods to handle popups.  
– Handling alerts: We can use methods provided by Alert interface to handle an alert.  
– Handling false positives: We can use different Assertions to look out for false positives.  
– Synchronization issues: We can use different wait methods provided by selenium.  
– Window based dialogs: We can use AutoIt tool to automate window based dialogs

**35) What are the different components of Test Automation Framework?**

– Object Repository  
– Driver Script  
– Test Scripts  
– Function Library  
– Test data resources

**36) What are the benefits does WebDriver have over Selenium RC?**

– Webdriver architecture is simpler than RC  
– Webdriver is also faster than RC  
– WebDriver interacts with page elements in a more realistic way  
– WebDriver’s API is simpler than Selenium RC’s  
– WebDriver can support the headless HtmlUnit browser but RC cannot

**37) Which of the WebDriver APIs is the fastest and why?**

HTML Unit Driver is the fastest because it works on a simple HTTP request-response mechanism and doesn’t interact with the browser UI for execution.

**38) What is the command to bind a node to Selenium Grid?**

driver = new RemoteWebDriver(new URL(nodeURL),capability);

**39) Which of Java. C-Sharp or Ruby can we use with Selenium Grid?**

All the languages can be used with Selenium Grid.

**40) What are Selenium Grid Extras and the additional features does it add to Selenium Grid?**

Selenium Grid Extras is a project that helps you set up and manage your local Selenium Grid. Below are the additional features:  
– Killing any browser instance by name  
– Stopping any Process by PID  
– Moving mouse to specific location  
– Get Memory usage and disk statistics  
– Automatically upgrade WebDriver binaries  
– Restart node after a set number of test executions  
– Central storage of configurations for all nodes on the HUB server  
– Screenshots at the OS level

**41) Explain the concept of Object Repository?**

An Object Repository is a map between UI element and its locator. It can also be written as an Object Map between UI element and the way to find it.

**42) What is the difference between findElement() and findElements(), its return type and few examples of where you have used in Selenium Projects?**

findElement returns only first matching element but findElements returns a list of all matching elements.  
Return type for findElement is WebElement while for findElements is List<WebElements>  
findElements can be used in a scenario where we want to find all broken links in a webpage.

**43) Which method can be used to get the text of an element?**

We can use getText() method to get text of any element.

**44) How to check which check-box from multiple check-box options is selected previously using Selenium?**

isSelected() method is used to know whether the Checkbox is toggled on or off.

**45) What is the return type of isSelected() method in Selenium?**

Return type is boolean.

**46) What are the different methods which can be used to verify the existence of an element on a web page?**

– driver.findElements(By.xpath(“value”)).size() != 0  
– driver.findElement(By.id(id)).isDisplayed()  
– driver.findElement(By.id(id)).isEnabled()

**47) What is XPath Axes and what are the different Axes available?**

XPath axes search different nodes in XML document from current context node. XPath Axes are the methods used to find dynamic elements.  
– following: Selects all elements in the document of the current node  
– ancestor: The ancestor axis selects all ancestors element (grandparent, parent, etc.) of the current node  
– child: Selects all children elements of the current node  
– preceding: Select all nodes that come before the current node  
– following-sibling: Select the following siblings of the context node.  
– parent: Selects the parent of the current node  
– descendant: Selects the descendants of the current node

**48) How to fetch an element when its attributes are changing frequently?**

We can use different XPath methods like contains(), using or/and, starts-with, text(),ends-with

**49) What are the different ways to click on a button using Selenium?**

– using click() method

– using return key: sendKeys(Keys.Return)

– **using JavaScriptExecutor**  
JavascriptExecutor js = (JavascriptExecutor) driver;  
js.executeScript(“document.getElementsByName(‘login’)[0].click()”);

– **using Actions class**  
Actions actions = new Actions(driver);  
actions.moveToElement(button).click().perform();

**50) What are the different types of Exceptions in Selenium?**

– NoSuchElementException  
– NoSuchWindowException  
– NoSuchFrameException  
– NoAlertPresentException  
– InvalidSelectorException  
– ElementNotVisibleException  
– ElementNotSelectableException  
– TimeoutException  
– NoSuchSessionException  
– StaleElementReferenceException

**51) How to handle Selenium WebDriver Exceptions?**

We can handle selenium exceptions by using try catch block methods of Java.

**try**{

driver.findElement(by.id("button")).click();

}

**catch**(NoSuchElementException e){

System.out.println("Element not present");

}

**52) There are four browser windows opened and you don’t have any idea where the required element is present. What will be your approach to find that element?**

– use getWindowHandles() method to get Window handles of all browser windows  
– use switchTo() method to switch to each browser window using the handle id  
– Find the element in each browser window and close the window if not present

**53) How do you handle an alert pop-up in Selenium?**

We can use the following methods to handle an alert in Selenium:

- dismiss()

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

- accept()

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

**54) How do you retrieve the text displayed on an Alert?**

**String** text = driver.switchTo().alert().getText();

**55) How do you type text into the text box on an Alert?**

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

**56) Is Alert in Selenium an Interface or Class?**

Alert is an interface in Selenium.

**57) How do you handle frames in Selenium?**

We can switch to frames by following methods:  
– **By Index**  
driver.switchTo().frame(0);  
– **By Name or Id**  
driver.switchTo().frame(“id of the element”);  
– **By Web Element**  
driver.switchTo().frame(WebElement);

**58) Give an example for method overloading concept that you have used in Selenium?**

Implicit Wait in Selenium use method overloading as we can provide different Timestamp or TimeUnit like SECONDS, MINUTES, etc.

**59) How do you select a value from a drop-down field and what are the different methods available?**

We can select value from drop-down using methods of Select class. Following are the methods:  
– selectByVisibleText  
– selectByValue  
– selectByIndex

Select elements = new Select(driver.findElement(By.id("button"));

elements.selectByVisibleText("Selenium");

elements.selectByIndex(1);

**60) When your XPath is matching more than one element, how do you handle it to locate the required element?**

We can use index of the element to locate it or we can use different Xpath axes methods to locate the element like Following, Ancestor, Child, Preceding or Following-sibling

**61) How do you capture screen-shots in Selenium and what is the best place to have the screen-shot code?**

//Convert web driver object to TakeScreenshot

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

//Call getScreenshotAs method to create image file

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

//Move image file to new destination

File DestFile=new File(fileWithPath);

//Copy file at destination

FileUtils.copyFile(SrcFile, DestFile);

**62) Write the code for connecting to Excel files and other operations?**

XSSFWorkbook srcBook = new XSSFWorkbook("Demo.xlsx");

XSSFSheet sourceSheet = srcBook.getSheetAt(0);

**int** rownum=rowcounter;

XSSFRow sourceRow = sourceSheet.getRow(rownum);

XSSFCell cell1=sourceRow.getCell(0);

**63) How do you read and write into a PDF file?**

BufferedInputStream file = new BufferedInputStream("Path of PDF file");

PDFParser pdf = new PDFParser(file);

pdf.parse();

**String** text = new PDFTestStripper().getText(pdf.getPDDocument());

**64) What are the disadvantages of Selenium?**

– It supports only web applications and cannot automate desktop applications  
– No default reporting mechanism  
– No default object repository  
– Cannot automate captcha

**65) How do you debug your automation code when it is not working as expected?**

– Add breakpoints on the lines of code where it is not working

– Run code in debugging mode

– Use different actions like F5(Step into), F6(Step Over), F9(Step Out) to debug the problem

**66) What are the end methods you use for verifying whether the end result is achieved by our Selenium automation scripts?**

We can use different assertion methods available in different test frameworks like TestNG or Junit.

**67) How do you clear the cookies of a browser using Selenium, before starting the execution?**

driver.manage().deleteAllCookies();

**68) How do you implement collections in your framework?**

Collections can be used in framework in situations where you have to store large number of objects. For example, findElements() method returns a list of all matching elements.

**69) Give a scenario where inheritance is used in your framework?**

We create a Base Class in the Framework to initialize WebDriver interface, WebDriver waits, Property files, Excels, etc., in the Base Class. We extend the Base Class in other classes such as Tests and Utility Class.

**70) Give a scenario where interface is used in your framework?**

WebDriver is an interface and when we create an instance of the driver object to use its different methods.

**71) Write a code using JavascriptExecutor to scroll the web page?**

//This will scroll the web page till end.

js.executeScript("window.scrollTo(0, document.body.scrollHeight)");

**72) What is the use of property file in Selenium?**

Property file can be used to store the different web elements of an application or to store all the different application, framework configurations.

**73) How do you handle multiple browsers selection in Selenium?**

We can select different browsers in Selenium using TestNG framework.

**74) What do you use for reporting in your Selenium Project?**

We can use the default TestNG or Cucumber report. We can also use different reporting libraries like Extent reports.

**75) How Cross Browser testing is handled in Selenium?**

@BeforeTest

@Parameters("browser")

**public** **void** setup(**String** browser) **throws** Exception{

//Check if parameter passed from TestNG is 'firefox'

**if**(browser.equalsIgnoreCase("firefox")){

//create firefox instance

System.setProperty("webdriver.gecko.driver", ".\\geckodriver.exe");

driver = new FirefoxDriver();

}

//Check if parameter passed as 'chrome'

**else** **if**(browser.equalsIgnoreCase("chrome")){

//set path to chromedriver.exe

System.setProperty("webdriver.chrome.driver",".\\chromedriver.exe");

//create chrome instance

driver = new ChromeDriver();

}

testng.xml:

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

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

<**suite** name="TestSuite" thread-count="2" parallel="tests" >

<**test** name="ChromeTest">

<**parameter** name="browser" value="Chrome" />

<**classes**>

<**class** name="com.qascript.crossbrowsertests">

</**class**>

</**classes**>

</**test**>

<**test** name="FirefoxTest">

<**parameter** name="browser" value="Firefox" />

<**classes**>

<**class** name="com.qascript.crossbrowsertests">

</**class**>

</**classes**>

</**test**>

</**suite**>

**1) Can Selenium automate Client Server applications?**

No Selenium cannot automate Client Server applications but there are other tools like Winium and AutoIt which can be used  
to automate desktop applications.

**2) What are the limitations of Selenium WebDriver?**

– It supports Web based applications only  
– Limited reporting capabilities  
– Handling captcha

**3) Tell me about the Selenium WebDriver architecture?**

WebDriver is made of following components:  
– Language Bindings  
– JSON Wire Protocol  
– Browser Drivers  
– Real Browsers

When a test script is executed with the help of WebDriver, the following tasks are performed in the background:

– An HTTP request is generated and it is delivered to the browser driver for every Selenium Command.  
– The HTTP request is received by the driver through an HTTP server.  
– All the steps/instructions to be executed on the browser is decided by an HTTP server.  
– The HTTP server then receives the execution status and in turn sends it back to the automation scripts.

**4) How to identify the web elements?**

In order to identify WebElements accurately and precisely, Selenium makes use of following locators:  
-ID  
-Name  
-CSS  
-LinkText  
-XPath

**5) When do you go for an XPath?**

Although XPath can be used as a locator for any webelement, it is particularly useful when elements are dynamically changing or  
don’t have any unique properties.

**6) How to execute the tests on Firefox Browser in Selenium?**

System.setProperty("webdriver.gecko.driver",Path\_of\_Firefox\_Driver");

WebDriver driver = new FirefoxDriver(); //Creating an object of FirefoxDriver

driver.get("https://qascript.com)

**7) What is the difference between id and name?**

id is used to identify the HTML element through the DOM and is expected to unique within the page  
name correspond to the form element and identified what is posted back to server

**8) How to handle dynamic web elements in Selenium?**

Dynamic web elements can be handled in the following ways:  
– By starting text  
– containing text  
– By index  
– By following-sibling  
– By preceding text

**9) What is the default timeout of Selenium WebDriver?**

Default timeout is 30 seconds

**10) When do we use implicit and explicit waits in Selenium?**

– The implicit wait will tell to the web driver to wait for certain amount of time before it throws a “No Such Element Exception”.  
Once we set the time, web driver will wait for that time before throwing an exception.  
– The explicit wait is used to tell the Web Driver to wait for certain conditions (Expected Conditions) or the maximum time exceeded  
before throwing an “ElementNotVisibleException” exception.

**11) How to select a date in a Calendar on a web page using Selenium?**

**public** **class** DatePicker

{

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

{

**String** dot="9/October/2018";

**String** date,month,year;

**String** caldt,calmonth,calyear;

/\*

\* Split the String into String Array

\*/

**String** dateArray[]= dot.split("/");

date=dateArray[0];

month=dateArray[1];

year=dateArray[2];

ChromeDriver driver=new ChromeDriver();

driver.get("http://cleartrip.com");

driver.findElement(By.id("DepartDate")).click();

WebElement cal;

cal=driver.findElement(By.className("calendar"));

calyear=driver.findElement(By.className("ui-datepicker-year")).getText();

/\*\*

\* Select the year

\*/

**while** (!calyear.equals(year))

{

driver.findElement(By.className("nextMonth")).click();

calyear=driver.findElement(By.className("ui-datepicker-year")).getText();

System.out.println("Displayed Year::" + calyear);

}

calmonth=driver.findElement(By.className("ui-datepicker-month")).getText();

/\*\*

\* Select the Month

\*/

**while** (!calmonth.equalsIgnoreCase(month))

{

driver.findElement(By.className("nextMonth ")).click();

calmonth=driver.findElement(By.className("ui-datepicker-month")).getText();

}

cal=driver.findElement(By.className("calendar"));

/\*\*

\* Select the Date

\*/

List<WebElement> rows,cols;

rows=cal.findElements(By.tagName("tr"));

**for** (**int** i = 1; i < rows.size(); i++)

{

cols=rows.get(i).findElements(By.tagName("td"));

**for** (**int** j = 0; j < cols.size(); j++)

{

caldt=cols.get(j).getText();

**if** (caldt.equals(date))

{

cols.get(j).click();

**break**;

}

}

}

}

}

**12) Can I navigate back and forth in a browser using Selenium WebDriver?**

Yes. We can use Navigate method to move back and forth in a browser.

driver.navigate().forward();

driver.navigate().back();

**13) How to execute the Selenium scripts on different browsers?**

We can use a framework like TestNg or Junit and configure them to run Selenium Scripts on multiple browsers.

**14) What is the purpose of isDisplayed() function in Selenium WebDriver?**

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

**15) What is the difference between isDisplayed() and isEnabled() functions in Selenium WebDriver?**

isDisplayed() is capable to check for the presence of all kinds of web elements available.  
isEnabled() is the method used to verify if the web element is enabled or disabled within the webpage.

**16) Can you test flash images in Selenium?**

You can also automate the flash using Selenium web driver through the Flashwebdriver object and  
then call a method to operate flash object. You need to download flashwebdriver jar files

**17) What is a Framework?**

A framework defines a set of rules or best practices which we can follow in a systematic way to achieve the desired results.

**18) How to select a third value from a drop-down field?**

Select select = new Select(listFrameworks);

select.selectByIndex(2);

**19) How to get columns from a table?**

WebElement table = driver.findElement(By.xpath("WebTableXPath"));

List<WebElement> totalRows = table.findElements(By.tagName("tr"));

**for**(**int** i=0;i<totalRows.size-1;i++){

List<WebElement> totalColumns = totalRows[i].findElements(By.tagName("td"));

}

**20) How many scripts are you writing and executing per a day?**

It is all dependent on the automation framework and the application under test.

**21) Which driver implementation will allow headless mode?**

HtmlUnit driver can be used to run tests in headless mode.

**22) Which reporting mechanism you have used in your Selenium projects?**

We used the Maven Cucumber Reporting plugin to generate detailed html reports.

**23) Why did you choose Selenium in your project, when there are so many tools?**

We chose Selenium because of the following reasons:  
– It is open source and free  
– It is easy to learn and setup  
– It is the most widely used and popular automation tool  
– All the web applications in our project are compatible with Selenium  
– Multi-browser and parallel testing is possible with Selenium

**24) How do you make use of JSON files in Selenium Grid?**

We can configure our hub and nodes using Json file in Selenium Grid.

**25) How to pause a test execution for 5 seconds at a specific point ?**

We can put a breakpoint on the line where we want to pause the test execution.

**1) How to get the html source code of a particular web element using Selenium WebDriver?**

We can get the html source code of an element using getAttribute method.

driver.getAttribute("innerhtml");

**2) What are the different driver classes available in Selenium WebDriver API?**

Selenium WebDriver API consists of different types of Browser driver classes like ChromeDriver,IEDriver, FirefoxDriver, etc…

**3) What automation tools could be used for post-release validation with continuous intergration?**

We can use continuous monitoring tools such as Nagios and Splunk to perform post release validation.

**4) Does the latest version of Selenium WebDriver support Mobile Testing?**

Selenium directly doesn’t support Mobile testing but it is possible by the help of other tools like Appium.

**5) What is the major differences between XPath Expressions and CSS Selectors?**

Using XPath we can traverse both forward and backward whereas CSS selector only moves forward in HTML DOM.

**6) How to select a check box in Selenium?**

We can select a checkbox by clicking on it.

driver.findElement(By.id("chkbox")).click();

**7) How to verify whether the checkbox option or radio button is selected or not?**

By using isSelected() method.

driver.findElement(By.id("chkbox")).isSelected();

**8) What is the alternative way to click on login button?**

We can use JavaScriptExecutor to click on login button.

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].click();", button);

**9) How can you find the value of different attributes like name, class, value of an element?**

By using the getAttribute() method.

**String** name = driver.findElement(By.id("login")).getAttribute("name");

**10) How to verify whether a button is enabled on the page?**

We can verify by using isEnabled() method.

driver.findElement(By.id("btn")).isEnabled();

**11) What kind of mouse actions can be performed using Selenium?**

Following mouse actions can be performed:  
– **doubleClick():** Performs double click on the element  
– **clickAndHold():** Performs long click on the mouse without releasing it  
– **dragAndDrop():** Drags the element from one point and drops to another  
– **moveToElement():** Shifts the mouse pointer to the center of the element  
– **contextClick():** Performs right-click on the mouse

**12) What kind of keyboard operations can be performed in Selenium?**

Following keyboard actions can be performed:  
– **sendKeys():** Sends a series of keys to the element  
– **keyUp():** Performs key release  
– **keyDown():** Performs keypress without release

**13) Can Bar Code Reader be automated using Selenium?**

It is not possible to automate bar code reader in Selenium.

**14) How to locate a link using its text in Selenium?**

We can use By.linkText() to locate a link using text in Selenium.

**15) Write the program to locate/fetch all the links on a specific web page?**

List<WebElements> allLinks = driver.findElements(By.tagName("a"));

**16) How can we run test cases in parallel using TestNG?**

By using the parallel attribute in testng.xml. The parallel attribute of suite tag can accept four values:

– tests: All the test cases inside <test> tag of Testing xml file will run parallel.  
– classes: All the test cases inside a Java class will run parallel  
– methods: All the methods with @Test annotation will execute parallel.  
– instances: Test cases in same instance will execute parallel

**17) How do you get the height and width of a text box field using Selenium?**

We can get the height and width of a text box using getSize() method.

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

System.out.println(element.size());

**18) Which package can be imported while working with WebDriver?**

Following package can be imported: org.openqa.selenium.WebDriver

**19) What is the purpose of deselectAll() method?**

Clears all selected entries. This is only valid when the SELECT supports multiple selections.

**20) What is the purpose of getOptions() method?**

It Returns all the option elements displayed in the select tag for dropdown list.

**21) How to handle alerts in Selenium WebDriver?**

We can use switchTo().Alert() method to switch to the alert dialog and perform the below actions:

driver.switchTo.Alert().accept(); //To click on Ok button of alert

driver.switchTo().Alert().dismiss(); //To click on Cancel button of alert

**22) What is hybrid framework?**

Hybrid Framework is a combination two or more frameworks like Data-Driven, Modular or Keyword-Driven. It uses the best features of each framework to build a highly reusable and maintainable framework.

**23) Can you explain the line of code WebDriver driver = new FirefoxDriver();?**

It starts a new Firefox browser driver instance.

**24) What could be the cause for Selenium WebDriver test to fail?**

There could be many reasons for test failure. Some of them are listed below:  
– Driver is null or not found  
– Element is not found on the page  
– Element is present but not intractable  
– Page Synchronization issues

**25) What is the difference between @Factory and @DataProvider annotation?**

@DataProvider – A test method that uses @DataProvider will be executed multiple number of times based on the configuration provided in it.  
The test method will be executed using the same instance of the test class to which the test method belongs.  
@Factory – A factory will execute all the test methods present inside a test class using separate instances of the class.

**26) How do you achieve synchronization in WebDriver?**

Synchronization can be achieved by using the following wait commands in WebDriver:  
– ImplicitWait  
– ExplicitWait  
– FluentWait