1. What is Automation Testing?

Automation testing is the process of testing the application using an automation tool to find the defects.

In this process, executing the test scripts and generating the results are performed automatically by automation tools.

Some most popular tools to do automation testing are HP QTP/UFT, Selenium WebDriver, etc.,

2. What are the benefits of Automation Testing?

* Saves time and money.
* Automation testing is faster in execution.
* Reusability of code. Create one time and execute multiple times with less or no maintenance.
* Easy reporting. It generates automatic reports after test execution.
* Easy for compatibility testing.
* It enables parallel execution in the combination of different OS and browser environments.
* Low-cost maintenance. It is cheaper compared to manual testing in a long run.
* Automated testing is more reliable.
* Automated testing is more powerful and versatile.
* It is mostly used for regression testing. Supports execution of repeated test cases.
* Minimal manual intervention. Test scripts can be run unattended.
* Maximum coverage. It helps to increase the test coverage.

3. What type of tests have you automated?

Our main focus is to automate test cases to do Regression testing, Smoke testing, and Sanity testing.

4. How many test cases you have automated per day?

It depends on Test case scenario complexity and length. I did automate 2-5 test scenarios per day when the complexity is limited. Sometimes just 1 or fewer test scenarios in a day when the complexity is high.

5. What is a Framework?

A set of guidelines like coding standards, test-data handling, object repository treatment etc... It’s beneficial to increase code re-usage , higher portability , reduced script maintenance cost etc. (Mind you these are just guidelines and not rules; they are not mandatory and you can still script without following the guidelines.)

There are different types of automation frameworks and the most common ones are:

* Data Driven Testing Framework - In this Framework, while[Test Case](https://www.guru99.com/test-case.html)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 Files, Text Files, CSV Files, ODBC Sources, DAO Objects, ADO Objects) and are loaded into the variables inside the Test Script. Variables are used both for Input values and for Verification values.
* Keyword Driven / table driven Testing Framework - The Keyword-Driven or Table-Driven framework requires the development of data tables and keywords, 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 is documented in a table as well as in step-by-step instructions for each test.
* Hybrid Testing Framework

6. Have you created any Framework?

If you are a beginner: No, I didn’t get a chance to create a framework. I have used the framework which is already available.

If you are an experienced tester: Yes, I have created a framework.  Or I have involved in the creation of the framework.

7. Can you explain the Framework which you have used in your Selenium Project?

Here we have clearly explained each component of Framework.

8. Why do you prefer Selenium Automation Tool?

* Free and open source
* Have large user base and helping communities
* Cross browser compatibility (Google Chrome, Firefox, Safari, IE)
* Platform compatibility (Windows, Mac OS, Linux)
* Multiple programming languages support (Java, Python, C#, Perl, PHP)

9. What is Selenium?

Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers.

Selenium is a set of different software tools. Each tool has a different approach in supporting web based automation testing.

It has four components namely,

1. Selenium IDE (Integrated Development Environment)
2. Selenium RC (Remote Control) – selenium 1
3. Selenium WebDriver – selenium 2 & 3
4. Selenium Grid

*(Commercial tool HP QTP (Quick Test Professional) AKA HP UFT (Unified Functional Testing)).*

Selenium Installation steps:

JDK setup Steps

JDK and JRE

TestNG SetUp Steps

Jenkins Setup Steps

10. What is Selenium IDE?

Selenium IDE (Integrated Development Environment) is a Firefox plugin. It allows us to record and playback the scripts. Even though we can create scripts using Selenium IDE, we need to use Selenium RC or Selenium WebDriver to write more advanced and robust test cases.

Selenium IDE is almost deprecated. It has already stopped working since Firefox 55

11. What is Selenese?

Selenese is the language which is used to write test scripts in Selenium IDE.

12. Which is the only browser that supports Selenium IDE to be used?

Firefox

13. What is Selenium RC?

Selenium RC AKA Selenium 1. Selenium RC was the main Selenium project for a long time before the WebDriver merge brought up Selenium 2. Selenium 1 is still actively supported (in maintenance mode). It relies on JavaScript for automation. It supports Java, Javascript, Ruby, PHP, Python, Perl and C#. It supports almost every browser out there.

14. What is Selenium WebDriver?

Selenium WebDriver AKA Selenium 2 is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.

15. What is Selenium Grid?

Selenium Grid is a tool used to run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems.

In simple words, it is used to distribute your test execution on multiple platforms and environments concurrently.

16. When do you 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

17. What are the advantages of Selenium Grid?

It allows running test cases in parallel thereby saving test execution time.

It allows multi-browser testing

It allows us to execute test cases on multi-platform

18. What is a hub in Selenium Grid?

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

19. What is a node in Selenium Grid?

Node is the machine which is attached to the hub. There can be multiple nodes in Selenium Grid.

20. What are the types of WebDriver APIs available in Selenium?

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

21. Which WebDriver implementation claims to be the fastest?

The fastest implementation of WebDriver is the HTMLUnitDriver. It is because the HTMLUnitDriver does not execute tests in the browser.

22. What are the Programming Languages supported by Selenium WebDiver?

* Java
* C#
* Python
* Ruby
* Perl
* PHP

23. What are the Operating Systems supported by Selenium WebDriver?

* Windows
* Linux
* Apple

24. What are the Open-source Frameworks supported by Selenium WebDriver?

* JUnit
* TestNG
* CUCUMBER
* JBHEAVE

25. What are the Locators available in Selenium?

Different types of locators are:

1. ID –
2. ClassName –
3. Name –
4. TagName –
5. LinkText –
6. PartialLinkText –
7. XPath –
8. CSS Selector –

26. What is a XPath?

XPath (xml Path) is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements such as textbox, button, checkbox, Image etc., in a web page.

Difference between XML & HTML?

27. What is the difference between “/” and “//”

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

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

28. What is the difference between Absolute Path and Relative Path?

Absolute XPath starts from the root node and ends with desired descendant element’s node. It starts with top HTML node and ends with input node. It starts with a single forward slash(/) as shown below.

/html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input

Relative XPath starts from any node in between the HTML page to the current element’s node(last node of the element). It starts with a double forward slash(//) as shown below.

//input[@id='email']

29. What is the difference between Assert and Verify in Selenium?

Assert: Assertion is used as a  verification point. It verifies what is expected.  The types of assertion are “assert” , “verify”

In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.

Verify: In simple words, there won’t be any halt in the test execution even though the verify condition is true or false.

In TestNG, we use only Assert Statements.

For detailed post check the below link.

30. What are Soft Assert and Hard Assert in Selenium?

Soft Assert: Soft Assert collects errors during *@Test* Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement

Hard Assert: Hard Assert throws an AssertException immediately when an assert statement fails and test suite continues with next *@Tes.* To achieve this, we need to handle the Assertion error which is thrown with a try-catch block. After the suit is completed, after completion of the execution, particular test has been marked as passed instead of FAIL

31. What are the verification points available in Selenium?

In Selenium IDE, we use Selenese Verify and Assert Commands as Verification points

In Selenium WebDriver, there is no built-in features for verification points. It totally depends on our coding style. some of the Verification points are

To check for page title

To check for certain text

To check for certain element (text box, button, drop down, etc.)

32. How to launch a browser using Selenium WebDriver?

WebDriver is an Interface. We create Object of a WebDriver Interface.

<2.53 – no geckodriver

3.x – geckodriver for FF

To launch Firefox Driver: WebDriver driver = new FirefoxDriver();

To launch Chrome Driver: WebDriver driver = new ChromeDriver();

To launch Internet Explorer Driver: WebDriver driver = new InternetExplorerDriver();

33. Is the FirefoxDriver a Class or an Interface?

FirefoxDriver is a Java class, and it implements the WebDriver interface.



34. What is the super interface of WebDriver?

SearchContext.

35. Explain the line of code Webdriver driver = new FirefoxDriver(); ?

‘WebDriver‘ is an interface and we are creating an object reference of type WebDriver instantiating an object of FirefoxDriver class.

36. We do create a reference variable ‘driver’ of type WebDriver

WebDriver driver = new FirefoxDriver();

instead of creating

FirefoxDriver driver = new FirefoxDriver();

|  |  |
| --- | --- |
| 1  2  3  4  5 | WebDriver driver = newFirefoxDriver();    instead of creating    FirefoxDriverdriver = newFirefoxDriver(); |

What is the purpose of doing this way?

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

//FirefoxDriver driver = new FirefoxDriver();

ChromeDriver driver = new ChromeDriver();

driver.get(“http://www.google.com”);

WebDriver driver = newFirefoxDriver();

37. What are the different exceptions you have faced in Selenium WebDriver?

* WebDriverException
* TimeoutException
* NoAlertPresentException
* NoSuchWindowException
* NoSuchElementException
* StaleElementReferenceException
* IllegalStateException

38. How To Login Into Any Site If It Is Showing Any Authentication Pop-Up For Username And Password?

To do this we pass username and password with the URL

http://username:password@url

e.g. <http://admin:admin123@xyz.com>

39. What are the types of waits available in Selenium WebDriver?

In Selenium we could see three types of waits such as Implicit Waits, Explicit Waits and Fluent Waits.

* Implicit Waits – Sets a timeout for all successive Web Element searches. For the specified amount of time it will try looking for element again and again before throwing a NoSuchElementException.  It waits for elements to show up.

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

* Explicit Waits – It is a one-timer, used for a particular search.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. Once we declare explicit wait we have to use "**ExpectedCondtions**".

WebDriverWaitwait = new WebDriverWait(WebDriverRefrence,TimeOut);

WebElement guru99seleniumlink = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(xxxxxxx)));

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

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)

.withTimeout(30, TimeUnit.SECONDS)

.pollingEvery(5, TimeUnit.SECONDS)

.ignoring(NoSuchElementException.class);

* PageLoadTimeOut
* Thread.sleep() – static wait

40. What is Implicit Wait In Selenium WebDriver?

Implicit waits tell to the WebDriver to wait for a certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception. The default setting is 0 (zero). We need to set some wait time to make WebDriver to wait for the required time.

41. What is WebDriver Wait In Selenium WebDriver?

WebDriverWait is applied on a certain element with defined expected condition and time. This wait is only applied to the specified element. This wait can also throw an exception when an element is not found.

42. What is Fluent Wait In Selenium WebDriver?

FluentWait can define the maximum amount of time to wait for a specific condition and frequency with which to check the condition before throwing an “ElementNotVisibleException” exception.

43. How to input text in the text box using Selenium WebDriver?

By using sendKeys() method

WebDriver driver = new FirefoxDriver();

driver.get("https://www.gmail.com");

driver.findElement(By.xpath("xpath")).sendKeys("test");

44. How to input text in the text box without calling the sendKeys()?

// To initialize js object

JavascriptExecutor JS = (JavascriptExecutor)driver;

// To enter username

JS.executeScript("document.getElementById(‘User').value=test.com'");

45. How to clear the text in the text box using Selenium WebDriver?

By using clear() method

WebDriver driver = new FirefoxDriver();

driver.get("https://www.gmail.com");

driver.findElement(By.xpath("xpath\_of\_element1")).sendKeys("Software Testing Material Website");

driver.findElement(By.xpath("xpath\_of\_element1")).clear();

46. How to get a text of a web element?

By using getText() method

47. How to get an attribute value using Selenium WebDriver?

By using getAttribute(value);

48. How to click on a hyperlink using Selenium WebDriver?

We use click() method in Selenium to click on the hyperlink

driver.findElement(By.linkText(“Software Testing Material Website”)).click();

|  |  |
| --- | --- |
| 1 | driver.findElement(By.linkText(“Software Testing Material Website”)).click(); |

49. How to submit a form using Selenium WebDriver?

We use “submit” method on element to submit a form

driver.findElement(By.id("form\_1")).submit();

|  |
| --- |
| 1 |

Alternatively, you can use click method on the element which does form submission

50. How to press ENTER key on text box In Selenium WebDriver?

To press ENTER key using Selenium WebDriver, We need to use Selenium Enum Keys with its constant ENTER.

driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER);

51. How to pause a test execution for 5 seconds at a specific point?

By using java.lang.Thread.sleep(long milliseconds) method we could pause the execution for a specific time. To pause 5 seconds, we need to pass parameter as 5000 (5 seconds)

Thread.sleep(5000)

52. Is Selenium Server needed to run Selenium WebDriver Scripts?

When we are distributing our Selenium WebDriver scripts to execute using Selenium Grid, we need to use Selenium Server.

53. What happens if I run this command. driver.get(“www.softwaretestingmaterial.com”) ;

An exception is thrown. We need to pass HTTP protocol within driver.get() method.

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

|  |  |
| --- | --- |
| 1 | driver.get("http://www.softwaretestingmaterial.com"); |

54. What is the alternative to driver.get() method to open an URL using Selenium WebDriver?

Alternative method to driver.get(“url”) method is driver.navigate.to(“url”)

55. What is the difference between driver.get() and driver.navigate.to(“url”)?

driver.get(): To open an URL and it will wait till the whole page gets loaded

driver.navigate.to(): To navigate to an URL and It will not wait till the whole page gets loaded

56. Can I navigate back and forth in a browser in Selenium WebDriver?

We use Navigate interface to do navigate back and forth in a browser. It has methods to move back, forward as well as to refresh a page.

driver.navigate().forward(); – to navigate to the next web page with reference to the browser’s history

driver.navigate().back(); – takes back to the previous webpage with reference to the browser’s history

driver.navigate().refresh(); – to refresh the current web page thereby reloading all the web elements

driver.navigate().to(“url”); – to launch a new web browser window and navigate to the specified URL

57. What are the different types of navigation commands?

Refer above question (Can I navigate back and forth in a browser)

58. How to fetch the current page URL in Selenium?

To fetch the current page URL, we use getCurrentURL()

driver.getCurrentUrl();

59. How can we maximize browser window in Selenium?

To maximize browser window in selenium we use maximize() method. This method maximizes the current window if it is not already maximized

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

60. How to delete cookies in Selenium?

To delete cookies we use deleteAllCookies() method

driver.manage().deleteAllCookies();

61. What are the ways to refresh a browser using Selenium WebDriver?

There are multiple ways to refresh a page in selenium

* Using driver.navigate().refresh() command as mentioned in the question 45
* 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());
* Using sendKeys(Keys.F5) on any textbox on the webpage

62. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in Selenium WebDriver?

driver.getWindowHandle() – It returns a handle of the current page (a unique identifier)

driver.getWindowHandles() – It returns a set of handles of the all the pages available.

63. What is the difference between driver.close() and driver.quit() methods?

Purpose of these two methods (driver.close and driver.quit) is almost same. Both allow us to close a browser but still, there is a difference.

driver.close(): To close current WebDriver instance

driver.quit(): To close all the opened WebDriver instances

64. What is the difference between driver.findElement() and driver.findElements() commands?

The difference between driver.findElement() and driver.findElements() commands is-

* findElement() returns a single WebElement (found first) based on the locator passed as parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
* Syntax of findElement()-
* WebElement textbox = driver.findElement(By.id(“textBoxLocator”));
* Syntax of findElements()-
* List <WebElement> elements = driver.findElements(By.id(“value”));
* Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.

List<WebElement> list = driver.findElements(By.tagName(“a”));

Sop(list.size()); ==40

65. How to find whether an element is displayed on the web page?

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

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 1 | 1. boolean elePresent = driver.findElement(By.xpath("xpath")).isDisplayed(); |  |  |

1. isSelected()
2. boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected();

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 1 | 1. boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected(); |  |  |

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

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 1 | 1. boolean eleEnabled= driver.findElement(By.xpath("xpath")).isEnabled(); |  |  |

66. How to select a value in a dropdown?

By using Select class

WebElement mySelectElement = driver.findElement(By.name("dropdown"));

Select dropdown = new Select(mySelectElement);

dropdown.selectByVisibleText(Text);

dropdown.selectByIndex(Index);

dropdown.selectByValue(Value);

[:](http://www.softwaretestingmaterial.com/handle-drop-down-and-multi-select-list-using-selenium/)

67. How to capture Screenshot in Selenium WebDriver?

By using TakesScreenshot Interface

In Selenium 3, we may face few issues while capturing Screenshots. To overcome we use aShot utility. Click on below links to see posts related to the normal way of capturing a screenshot and capturing a screenshot using aShot utility.

68. How to mouse hover on a web element using WebDriver?

By using Actions class

WebElement ele = driver.findElement(By.xpath("xpath"));

//Create object 'action' of an Actions class

Actions action = new Actions(driver);

//Mouseover on an element

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

[.](http://www.softwaretestingmaterial.com/mouse-hover-actions-using-selenium/)

69. How can we handle web based pop-up?

To handle alerts popups we need to do switch to the alert window and call Selenium WebDriver Alert API methods.

[.](http://www.softwaretestingmaterial.com/javascript-alerts-popups-selenium/)

70. How can we handle windows based pop up?

Selenium doesn’t support windows based applications. It is an automation testing tool which supports only web application testing. We could handle windows based popups in Selenium using some third party tools such as AutoIT, SIKULI, Robot class etc.

71. How to handle hidden elements in Selenium WebDriver?

It is one of the most important selenium interview questions.

We can handle hidden elements by using javaScript executor

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

72. How can you find Broken Links in a page using Selenium WebDriver?

[.](http://www.softwaretestingmaterial.com/broken-links-using-selenium/)

73. How to find more than one web element in the list?

// To store the list

List <WebElement>eleList = driver.findElements(By.xpath("xpath"));

// To fetch the size of the list

intlistSize = eleList.size();

//for loop

for (inti=0; i<listSize; i++)

{

// Clicking on each link

links.get(i).click();

// Navigating back to the previous page that stores the links

driver.navigate().back();

}

74. How to read a JavaScript variable in Selenium WebDriver?

By using JavascriptExecutor

// To initialize the JS object.

JavascriptExecutor JS = (JavascriptExecutor) webdriver;

// To get the site title.

String title = (String)JS.executeScript("return document.title");

System.out.println("Title of the webpage : " + title);

75. How do you read test data from excels?

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

76. Is it possible to automate the captcha using Selenium?

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

77. List some scenarios which we cannot automate using Selenium WebDriver?

1. Bitmap comparison Is not possible using Selenium WebDriver

2. Automating Captcha is not possible using Selenium WebDriver

3. We cannot read bar code using Selenium WebDriver

4. windows OS based pop ups

5. third party calendars/element

6. Image

7. Word/PDF

78. What is Object Repository in Selenium WebDriver?

Object Repository is used to store element locator values in a centralized location instead of hard coding them within the scripts. We do create a property file (.properties) to store all the element locators and these property files act as an object repository in Selenium WebDriver.

79. How can you use the Recovery Scenario in Selenium WebDriver?

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

try {

driver.get("www.xyz.com");

}catch(Exception e){

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

}

80. How to Upload a file in Selenium WebDriver?

There are two cases which are majorly used to upload a file in Selenium WebDriver such as using SendKeys Method and using AutoIT Script.

[.](http://www.softwaretestingmaterial.com/upload-file-using-autoit/)

Browser Button – type =“file”

SendKeys (c:\\test\\naveen.jpg);

81. How to Download a file in Selenium WebDriver?

By using AutoIT script, we could download a file in Selenium WebDriver.

82. How to run Selenium WebDriver Test from the command line?

Class A{

}

cd c

c: javac A.java

c: java A.java

javaorg.testng.TestNG C:\Users \Desktop\ \workspace\testing\testng.xml

83. How to switch between frames in Selenium?

By using the following code, we could switch between frames.

driver.switchTo().frame();

84. How to connect a Database in selenium?

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

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

85. How To Resize Browser Window Using Selenium WebDriver?

To resize the browser window to particular dimensions, we use ‘Dimension’ class to resize the browser window.

//Create object of Dimensions class

        Dimensiond=newDimension(480,620);

        //Resize the current window to the given dimension

        driver.manage().window().setSize(d);

86. How To Scroll Web Page Down Or UP Using Selenium WebDriver?

JavaScript scrollBy() method scrolls the document by the specified number of pixels.

87. How To Perform Right Click Action (Context Click) In Selenium WebDriver?

We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

action.contextClick(driver.findElement(By.xpath()).build().perform();

88. How To Perform Double Click Action In Selenium WebDriver?

We use Actions class to do Double click action in selenium.

89. How To Perform Drag And Drop Action in Selenium WebDriver?

We use Actions class to do Drag And Drop Action

90. How To Highlight Element Using Selenium WebDriver?

By using JavascriptExecutor interface, we could highlight the specified element

91. What is TestNG? Features?

TestNG is an automation testing framework in which NG stands for "Next Generation".

* Generate the report in a proper format including a number of test cases runs, the number of test cases passed, the number of test cases failed, and the number of test cases skipped.
* Multiple test cases can be grouped more easily by converting them into testng.xml file. In which you can make priorities which test case should be executed first.
* The same test case can be executed multiple times without loops just by using keyword called 'invocation count.'
* Using testng, you can execute multiple test cases on multiple browsers, i.e., cross browser testing.
* The testing framework can be easily integrated with tools like Maven, Jenkins, etc.
* Annotations used in the testing are very easy to understand ex: @BeforeMethod, @AfterMethod, @BeforeTest, @AfterTest (Annotations in TestNG are lines of code that can control how the method below them will be executed. They are always preceded by the @ symbol. )
* WebDriver has no native mechanism for generating reports. TestNG can generate the report in a readable format like the one shown below.
* TestNG simplifies the way the tests are coded. There is no more need for a static main method in our tests. The sequence of actions is regulated by easy-to-understand annotations that do not require methods to be static.
* Uncaught exceptions are automatically handled by TestNG without terminating the test prematurely. These exceptions are reported as failed steps in the report.

92. Which attribute you should consider throughout the script in frame for “if no frame Id as well as no frame name”?

You can use…..driver.findElements(By.xpath(“//iframe”))….

This will return list of frames.

You will need to switch to each and every frame and search for locator which we want.

Then break the loop.

93. What is BDD and TDD?

94. Difference between Java Class and Java Interface?

95. ‘StaleElementReferenceException’ Vs. ‘NoSuchElementFound’

* What is Static dropdown and how to read static drop down:

In Static drop down, values are pre-defined and do not change on any event.

When you inspect the element, mostly it’s tag name is <Select>.

We handle it using ‘Select’ Class.

* Why Maven?
* How to automation tooltip
* How to automate pie chart or graphs
* How to download file using selenium
* How to check data in Excel file using selenium

Real time questions:

* Explain current Project (Project descriptions/ modules/ my job responsibilities)
* Which Framework did you use (Page Objects/cucumber/ data driven)?
* What the advantages of the framework you used?
* Page Object Model: tell about PageClasses, Elements/Actions/PageFactory in PageClasses, Object Repository is separated from Test verification, Readable/Maintainable/Reusable,
* Why this framework you choose
* What are the challenges you faced during framework building
  + Timing issue (used explicit wait)
  + DOM structure/locators (tell some example and how it resolved) like duplicate elements i.e. Sort option is in header also in footer of the grid, Save button is in header and in footer also. Using difference axis relation like parent/ancestor/sibling I resolved it.
  + Dynamically changing elements i.e. New ClientName and ProjectName text box field while on new Client/Project page. Resolved it using starting part and ending part of the name of the fieldname as
  + @FindBy(xpath="//input[contains(@name,'][qty]') and contains(@name,'cart[')]")

WebElement QtyTextBox;

* + To generate Extent Report in reqiored format. Frankly speaking, I took help from google to create extent report, but it was missing something like attaching screenshot of failed test cases. I took me little time to make the changes
* Which framework you used to execute test cases (Testng or junit)

Why TestNG: easy to use/more annotations than jUnit like @BeforeSuite/@AfterSuite/@BeforeTest/@AfterTest, supports dependent methods, support parameters, flexible test configuration, it generates test reports, grouping of test cases, supports parallel execution

* Why do we declare some variables/methods static and non-static? Also, public and private?

Static variables: Static variables can be created at Class Level only. Static variables are Global variables and available to all methods/instance of the class. Static Variables can be accessed by Class Name. It can be accessed by static/non-static methods.

Non-Static variables: Two types - Local variables and Instance variables. Local variables can be created inside the block /inside the method. Scope of these variables are within the block in which it gets declared. Instance variables are declared in a class; these are created when an object of a class is created and destroyed when the object gets destroyed. Instance variable can be accessed only by creating objects.

Public variables: Public variables, are variables that are visible to all classes.

Private variables: Private variables, are variables that are visible only to the class to which they belong.

Protected variables:, are variables that are visible only to the class to which they belong, and any subclasses.

* How did you avoid Hard coding the things in your project? (used config.properties file)
* Which Source Code Management tool did you use?

GitHub => how to check-in/check-out/branching/commit/conflict/conflict resolution/versioning/pull/push/merging code

* How many members in your team

Tell total members including dev/managers/testers and also tell the follow agile/scrum methodology and we perform manual + automation

* How much time it take to complete test execution for your framework for all test cases
* How did you reduce the time of execution or what we can do to reduce execution time

=> ran test cases/classes parallel which are not dependent. Understand how parallel execution work for test cases/classes in testng.xml

* How to make sure that failure of test cases are not intermittent
* Used the concept of re-try in testng. Understand how it works (running the failed test cases only using ReTryListener class).
* How do you debug failed test cases or failures ?
* Look at logs and screenshots and then check the feature manually and then code if any mistake
* Understand the concept of loggings and taking screenshot on failures
* How did you perform logging and taking screenshots on failures (using ITestListener)
* Tell me any component in your framework that can be useful in future requirements also?
* How did you handle synchronization problem in your framework?

Q – Any common exceptions in selenium?

<https://www.softwaretestinghelp.com/exception-handling-framework-selenium-tutorial-19/>

**TimeoutException**: This exception is thrown when a command performing an operation does not complete in the stipulated time

When You have provided a certain time span to wait. So driver will wait for that time and then if driver is not able to find that element on the page, it will throw timeout exception.

Just remove specified wait timespan to solve this problem

**NoSuchElementException**: This exception is thrown when an element with given attributes is not found on the web page

**ElementNotVisibleException**: This exception is thrown when the element is present in DOM (Document Object Model), but not visible on the web page

**StaleElementException**: This exception is thrown when the element is either deleted or no longer attached to the DOM

Q – What is Error and Exception?

Error: When a scenario is fatal/deadly and program cannot recover or handle them then JVM throws an Error. Errors cannot be handled by Try/Catch block. Example – Assertion Error, OutOfMemory Error

Exception: When a scenario or a problematic condition can be handled by program while compilation time or execution time. Exception gives more details like Type of Exception, Line number. It can be handled by Try/Catch block to get more details. Example – FileNotFoundException, IOException, ArrayIndexOutOfBoundException

EventFiringWebDriver – for logging purpose

TestListenerAdapter – to take screenshots

Q - Challenges faced using selenium automation testing, and how to solve them:

1. Image or text overlapping issue

2. No facility to deal with Captcha, Bar Code

3. Doesn’t support any non web based (Like Win 32, Java Applet, Java Swing, .Net Client Server etc) applications

4. When you compare selenium with QTP, Silk Test, Test Partner and RFT, there are many challenges in terms of maintainability of the test cases

5. Since Selenium is a freeware tool, there is no direct support if one is in trouble with the support of applications

6. Bitmap comparison is not supported by Selenium

7. Any reporting related capabilities, you need to depend on third party tools

8. You need to learn any one of the native language like (.Net, Java, Perl, Python, PHP, Ruby) to work efficiently

9. Difficult to identify dynamic objects

10. Working with frames

11. Selenium test playback is slow (IDE)

12. JavaScript sandbox, Flash, Applets, Silverlight, and HTML 5’s Canvas all present problems in Selenium

13. Dealing with pop-up windows: Selenium can sometimes fail to record common popups in web apps. The Alert interface brings with it the following commands: void dismiss(), void accept (), getText(), void sendKeys(String stringToSend). The first two basically click on the “cancel” and “OK” buttons respectively on a popup window.

14. Timeout resulting from synchronization problems: One should ideally use selenium.IsElementPresent(locator) to verify that the object is in a loop with Thread.Sleep

15. Testing Flash apps: To automate flash apps with Selenium, one can use Flex Monkium. The application source code must be compiled with the swc files generated by Flex Monkium. Then the app and the Selenium IDE are connected, and the tests can be recorded with IDE.

16. Unexpected error launching Internet Explorer. Browser zoom level should be set to 100% by default for the IE browser to overcome this error

17. Protected Mode must be set to the same valueerror occurs when trying to run Selenium WebDriver on a fresh Windows machine. This issue can be fixed by using capabilities when launching IE

18. Cross Browser Testing Issues

19. Ajax Components