2. How to handle pop-up handling?

We’re going to Alerts handling the pop-ups Using a switchTo.accept if it’s a positive flow. And if it’s negative flow well use dismiss.

3. What are the different types of EXCEPTIONS in Selenium WebDriver?

Google types of exceptions\*

4. what are the different types of method in Selenium?

Mostly everything’s a method, .sendKeys , .pageloadTimeout , .click , .getText etc.

5. Can we enter text without using .sendKeys() ?

We can use Java script executor to enter without sendkeys.

6. Write program for string reverse.

We need to write a loop to reverse it. .Reverse or use a string Builder.

9. How to find more than one Element in the list?

We can use find element it will give you the list of value then if you give the .Size will give the size of the list.

10. What is the difference between driver.close & driver.quit?

driver.close will close only one page that is currently open. And driver.quit will close the entire driver.

12. How to count the number of links in a page?

Using find elements we will be doing it.

13. Does java script support multiple inheritance?

No Java doesn’t support multiple inheritance, but with interfaces we can achieve the multiple inheritance function. MI inheritance means 2 child accessing parents at the same time.

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

Both are used for jUnit or framework. Assert means if there is an issue it will stop at that. And using Verify will not stop it’ll continue.

15. How to take a screenshot using selenium webdriver?

Using Take screenshot method, then create image file and then copy to desired location.

16. How will you execute your login script using chrome browser from your editor using Selenium?

Webdriver driver = new chrome driver.

17. Can we use multiple catch in TRY, if so how?

Yes, we can, So TRY is like if-else statement, for a single IF you can have a lot Else-If statement

18. Difference between WebDriver/Firefox Driver?

Web driver means it can be used for all the browsers, Firefox, chrome, and IE. And if only using for Firefox it can only be used for Firefox.

19. What is the difference between “GET” and “NAVIGATE” to open a webpage in selenium web driver? Both will be performing the same. Get will wait till all the pages are loaded and navigate wont but with navigate you can go forward backward and refresh the pages.

20. How can we get the font size, color and type used for a particular text on webpage using selenium webdriver? Webdriver driver.manage then It’ll give you options for choosing font.

21. How do you manage the code version in your project?

You can manage with Maven.

22. How do we handle web elements without using Xpath?

We can use with Find Elements by using other options such as Tag name, text etc.

23.How to handle alerts and confirmation boxes.

Alert is nothing but pop-ups, we can use switchTo to handle it either switchTo. Accept or dismiss.

24. How to handle colors in Webdriver ?

Get CSS value is used for handling colors font etc. this method is used for everything.

25. How to press shift +tab?

Using robot class or using Keys.Chord we can press combination of keys

26.How to take a screenshot in selenium.

We can use a take screenshot method.

27.Is there a way to click hidden link in Webdriver.

Yes, using Java Script Executor

28. Login for Gmail scenario ?

29. What are the technical challenges that you faced with selenium?

There are lots of challenges. Application having dynamic objects, lot of hidden elements. My app will have random pop-ups so how we’ve handled is whenever the popup is throwing, we would if-else conditions try catch conditions. Having lots of dynamic elements defining Xpaths will be different finding difficult Xpaths would enable us to use parent child ancestor type, starts with ,ends with type of methods. And synchronization issues my app will be someday extremely fast my webdriver will freeze and won’t be performing as expected so depending on the issues I would have to invoke Implicit explicit wait times and I have created lots of reusable functions for Implicit explicit. Another example would be page loading time, if there is a page navigation from one page to the other sometimes it would take a lot of time for example if it’s a Hotel or flight booking app once if an end user clicks submit to booking after the getting the confirmation ID it will take sometimes up to 5 mins so what we would do is basically check the DB to see if the ID were created or not.

31. What classes extends webdriver ?

Remote Webdriver is a class and webdriver is an interface (Google the WEB driver Architecture\*\*).

32. What is action class in webdriver?

Whenever the keyboard actions or the most actions. Those action classes are used for mouse hovering drag and drop and right click, so whatever is abnormal we use the action class to perform it.

35. What is the hierarchy of TestNG Annotation?

* @BeforeSuite – The @BeforeSuite method in TestNG runs before the execution of all other test methods.
* @AfterSuite – The @AfterSuite method in TestNG runs after the execution of all other test methods.
* @BeforeTest – The@BeforeTest method in TestNG runs before the execution of all the test methods that are inside that folder.
* @AfterTest – The @AfterTest method in TestNG executes after the execution of all the test methods that are inside that folder.
* @BeforeClass – The @BeforeClass method in TestNG will run before the first method invokes of the current class.
* @AfterClass – The @AfterClass method in TestNG will execute after all the test methods of the current class execute.
* @BeforeMethod – The @BeforeMethod method in TestNG will execute before each test method.
* @AfterMethod – The @AfterMethod method in TestNG will run after each test method is executed.
* @BeforeGroups – The @BeforeGroups method in TestNG run before the test cases of that group execute. It executes just once.
* @AfterGroups – The @AfterGroups method in TestNG run after the test cases of that group execute. It executes only once.

36. What is Selenium and what are the different components?

Its an automation tool. Selenium IDE- record and playback , Webdriver-, RC- Remote control, selenium grid which used for parallel execution with different machines.

37. What is Xpath?

It’s nothing but XML path, we have two types of Xpath relative- to the particular node and absolute- means starts from the beginning.

38. How to refresh a page without using context click?

Driver.navigate.to.refresh

39. How to handle autocomplete box in web driver?

Nothing but a list. Iterate it using a for-loop and when the value is matching then use the if-else condition to perform it.

41. Difference between the getWindowHandle and getWindowHandles?

**getWindowHandle**() returns the window handle **of** currently focused window / tab. **getWindowHandles**() returns all windows/tabs handles launched/opened by same driver instance including all parent and child window.

Return type **of getWindowHandle**() is String while return type **of getWindowHandles**() is Set<String>.

42. Write a code for number of characters in string ?

1st we would need to break the string and then write the If-Else condition character. Todigit. Or to num.

43. How to Read particular cell from html table ?

By using Xpath

44. How to Connect Java Application with Oracle database

How to connect from Excel, property file and DB. These 3 are the commonly used one.

45. How to switch between frames in Selenium WebDriver?

Driver.switchTo.frame\

46. What are the different types of Locators in Selenium?

8 **different types of locators** namely id, name, className, tagName, linkText, partialLinkText, CSS selector and xpath.

47. How to handle AJAX pop-window

Using similar methods to Alerts and if Popup is not from the website then use switchTo

48. Difference between Interface and Abstract class?

Abstract class can inherit another class using extends keyword and implement an interface. Interface can inherit only an inteface. Abstract class can be inherited using extends keyword. Interface can only be implemented using implements keyword

49. How to connect to Excel code for Excel (POI) API to read Excel.

1. Excel sheet

2. Property file

3. Database

50. Retry executing only failed tests using TestNG

We need to write logic for Retry executing only failed tests using TestNG

\* This method decides how many times a test needs to be rerun.

\* TestNg will call this method every time a test fails. So we

\* can put some code in here to decide when to rerun the test.

\* Note: This method will return true if a tests needs to be retried

\* and false it not.

\* \*/

@Override

public boolean retry(ITestResult result) {

if(counter < retryLimit)

{

counter++;

return true;

}

return false;

}

}

51. Relative Xpath Method

which ever starts with double slashes , it means find from the middle coming from the 1st node

52. Cucumber options and its uses

Used in a runner class its used for giving you wherever the test cases are located, feature files are located, types of reports you want

53. Purpose of Scenario outline

WE have something Scenario – we cant give you multiple data we can run the test case only once and scenario outline – you can run the same test case multiple time.

54. Testing in multiple browsers using Selenium and Testing

Launching with Chrome driver – ‘Webdriver driver = new chrome driver;’ same goes for IE, firefox and etc

55. Find out broken links on website using selenium webdriver and HTTP Client

Whichever link is not giving results, 2 ways we can find result for the broken link we can find the face of the link the other way is we can get the response of it and find the link

try {

huc = (HttpURLConnection)(new URL(www.gmail.com).openConnection());

huc.setRequestMethod("HEAD");

huc.connect();

respCode = huc.getResponseCode();

if(respCode >= 400){

System.out.println(url+" is a broken link");

}

Else {

System.out.println(url+" is a valid link");

}

56. Find broken / invalid images on a page

For checking the broken links, you will need to do the following steps.

1. Collect all the links in the web page based on <a> tag.
2. Send HTTP request for the link and read HTTP response code.
3. Find out whether the link is valid or broken based on HTTP response code.
4. Repeat this for all the links captured.

Explaining the code Example

Step 1: Import Packages

Step 2: Collect all links in web page

Step 3: Identifying and Validating URL

Step 4: Send http request

Step 5: Validating Links

57. How to handle Internationalization through webdriver

**Internationalization in Selenium WebDriver**, or Chrome and Firefox **in** this case, is done **by** setting intl. accept\_languages preference to the necessary BCP 47 tag **in** the profile. The profile should be set **in** the DriverOptions so that the updated locale is reflected **in** the profile

58. Page Object Model (POM)

is a design pattern, popularly used in test automation that creates Object Repository for web UI elements? The advantage of the model is that it reduces code duplication and improves test maintenance.

59. Taking Screenshot only for failed tests

// Import all classes and interface

// Create Webdriver reference

// Initiate Firefox browser

// Pass application url

// Here we are forcefully passing wrong id so that it will fail our testcase

// It will execute after every test execution

// Here will compare if test is failing then only it will enter if condition

// Create reference of TakesScreenshot

* TakesScreenshot ts=(TakesScreenshot)driver;

// Call method to capture screenshot

* File source=ts.getScreenshotAs(OutputType.FILE);

// Copy files to specific location here it will save all screenshot in our project home directory and

// result.getName() will return name of test case so that screenshot name will be same

* FileUtils.copyFile(source, new File("./Screenshots/"+result.getName()+".png"));

// close application

* driver.quit();

60. WAIT commands in Webdriver

There are **4** types of WAITS available

**PageLoadTimeOut** - Sets the amount of time to wait for a page-load to complete before throwing an error. If the timeout is negative, page loads can be indefinite.

**ExplicitWait -** Explicit waits are confined to a particular web element. Explicit Wait is code you define to wait for a certain condition to occur before proceeding further in the code.

**ImplicitWait** - we can tell Selenium that we would like it to wait for a certain amount of time before throwing an **exception** that it cannot find the element on the page

**FluentWait** - Each instance defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition

61. WHAT IS TESTNG LISTENER ?

***TestNG listeners are the piece of code that listens to the events occurring in the TestNG***. If the event matches the event for which we want the listener to listen, it executes the code, which ultimately results in modifying the default behavior of TestNG. For example, we want to print the exception error onto the reports only if the test fails. Here, we can apply a TestNG listener that will listen to the event of “*failing of test case*” and when it does, it will log the error.

62. HOW TO SELECT an ELEMENT FROM DROPDOWN ?

WebDriver provides three ways to select an option from the drop-down menu.

**1. selectByIndex** - It is used to select an option based on its index, beginning with 0.

dropdown.selectByIndex(5);

**2. selectByValue** - It is used to select an option based on its 'value' attribute.

dropdown.selectByValue("Database");

**3. selectByVisibleText** - It is used to select an option based on the text over the option.

dropdown.selectByVisibleText("Database Testing");

63. WHAT IS WEBDRIVER EVENT LISTENER ?

Is an interface holding predefined methods. They are helpful for viewing events triggered by the WebDriver. The WebDriverEventListener interface can implement classes and methods

64. HOW DOES THE SELENIUM WEBDRIVER WORK? 

65. HOW DO YOU CREATE HTML TEST REPORT FROM YOUR TEST SCRIPT

### index.html

1. Right click on the index.html from the project directory.
2. Select option open with web browser option. It will display the result in the following order.

66. HOW DO YOU HANDLE HTTPS WEBSITE IN SELENIUM

SSL (Secure Socket Layer) Certificate ensures secure transformation of data across the server and client application using strong encryption standard or digital signature. One has to install an SSL certificate or a code signing certificate.

67. HOW THE TESTNG INTERACT WITH SELENIUM CORE

68. HOW TO CHANGE USER AGENT IN FIREFOX BY SELENIUM WEBDRIVER

1. **Open a new tab**
2. **Browse to "about:config"**
3. **Dismiss the warning**
4. **Create a new preference item**
5. **Close the Preferences window**
6. **Resetting your user agent in Firefox**

69. HOW TO READ PARTICULAR CELL FROM HTML TABLE

This is the way to get the value of each cell of a web table using selenium

**1. Table initialize**

In your case, the table has the id tbr, the way to initialize it :

WebElement tbl = driver.findElement(By.id("tbr"));

**2. Row initialize**

The name tag for the web table row in general is tr, the way to initialize it :

List<WebElement> rows = tbl.findElements(By.tagName("tr"));

**3. Column initialize**

The name tag for the web table column in general are th or td, the way to initialize it :

th tag

List<WebElement> cols = rows.get(rowIndex).findElements(By.tagName("th"));

td tag

List<WebElement> cols = rows.get(rowIndex).findElements(By.tagName("td"));

**So you can get particular cell values by :**

String cell = cols.get(indexCol).getText();

The bellow code is function to get all value cell table (without getting header th value)

WebElement tbl = driver.findElement(By.id("tbr"));

//check all row, identification with 'tr' tag

List<WebElement> rows = tbl.findElements(By.tagName("tr"));

//row iteration

for(int i=0; i<rows.size(); i++) {

//check column each in row, identification with 'td' tag

List<WebElement> cols = rows.get(i).findElements(By.tagName("td"));

//column iteration

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

System.out.println(cols.get(j).getText());

}

//This is to get the cell value you want

////get col no 2

//System.out.println(cols.get(1).getText());

////get col no 8

//System.out.println(cols.get(7).getText());

}

70. HOW TO CHECK ALL CHECK BOXES IN A PAGE

// Find all checked checkboxes with xpath

List<WebElement> checkBoxList=driver.findElements(By.xpath("//input[@type='checkbox' and @checked='checked']"));

for(WebElement checkBox:checkBoxList)

{

checkBox.click();

}

71. HOW TO DISABLE COOKIES IN A BROWSER IN SELENIUM?

**selenium**. chrome. The method DeleteAllCookies(); will **delete** all the **cookies** for the current domain [Handling **Cookies** in **Webdriver**]. If you want to **delete** all **cookies** for all domain you should use solution from Chrome **settings** "Clear browsing data".

driver.manage().getCookies();

72. HOW TO GET THE NAME OF BROWSER USING WEBDRIVER?

driver.get(URL);

// Or can be written as

String URL = "URL";

driver.get(URL);

73. HOW TO HANDLE AJAX POPUP WINDOW IN SELENIUM?

Thread.Sleep()

Implicit Wait()

Explicit Wait()

WebdriverWait

Fluent Wait

74. HOW TO HANDLE AUTO COMPLETE BOX IN WEBDRIVER IN SELENIUM?

By using Try catch Method

75. HOW TO HANDLE NETWORK LATENCY USING SELENIUM ?

Again you need describe about wait time

Thread.sleep

Implicit wait

Explicit wait

Fluent wait

76. HOW TO KNOW ALL THE METHODS SUPPORTED IN WEBDRIVER AND ITS SYNTAX IN SELENIUM? ?

**There's** a total of 5 **Selenium Method** Categories. The categories are Browser **Methods**, WebElement **Methods**, Navigation **Methods**, Wait **Methods**, and Switch **Methods**. Each category has a group of **methods** that perform actions via **Selenium**: Browser **Methods** perform actions on a browser.

77. HOW TO LOGIN INTO ANY SITE IF ITS SHOWING ANT AUTHENTICATION POP UP FOR USERNAME AND PASSWORD DURING LAUNCH OF THE URL IN SELENIUM?

WebDriverWait wait = new WebDriverWait(driver, 10);

Alert = wait.until(ExpectedConditions.alertIsPresent());

alert.authenticateUsing(new UserAndPassword(\*\*username\*\*, \*\*password\*\*));

78. HOW TO MOUSE HOVER ON AN ELEMENT IN SELENIUM?

1. Instantiate Action Class

2.Retrieve WebElement 'Music' to perform mouse hover

3.Mouse hover menu Option 'Music'

4. Now Select 'Rock' from sub menu which has got displayed on mouse hover of 'Music'

5. Mouse hover menu Option 'Rock'

6. Now , finally, it displays the desired menu list from which required option needs to be selected

7. Now Select 'Alternative' from sub menu which has got displayed on mouse hover of 'Rock'

79. HOW TO PASS PARAMETERS FROM TESTNG.XML INTO TEST CASES IN SELENIUM? we can define the parameters in the testng.xml file and then reference those parameters in the source files. Create a java test class, say, ***ParameterizedTest.java*** and add a test method say **parameterizedTest()** to the test class. This method takes a string as input parameter. Add the annotation **@Parameters(“browser”)**to this method.

80. HOW TO PERFORM DOUBLE CLICK USING WEBDRIVER IN SELENIUM?

Actions = new Actions(driver);

WebElement elementLocator = driver.findElement(By.id("ID"));

actions.doubleClick(elementLocator).perform();

81. HOW TO PREPARE CUSTOMIZED HTML REPORT USING TESTNG IN HYBRID FRAMEWORK IN SELENIUM?

To customize selenium TestNG report, you can use testng listeners.

1. [ITestListener](http://testng.org/javadocs/org/testng/ITestListener.html): Log Result/Screenshot on test pass/fail/skip.
2. [IReporter](http://testng.org/javadocs/org/testng/IReporter.html): To generate html report from xml suite results and log.

There can be two ways we can customize TestNG report.

public void generateReport(List<XmlSuite> xmlSuites, List<ISuite> suites,

String outputDirectory) {

}

...

//paste the content of TestHTMLReporter.java here

82. HOW TO PUT TEXT IN ANY SEARCH BOX USING SELENIUM WEBDRIVER ?

1. Find the Text field using the id locator.
2. Enter text into the searchbox using the Selenium sendkeys method.
3. Enter any words in the field using the sendKeys() method.

83. HOW TO REFRESH A PAGE WITHOUT USING CONTEXT CLICK IN SELENIUM?

**Using** navigate.**refresh**() method. driver.navigate().**refresh**();

84. HOW TO RUN TESTS IN MULTIPLE BROWSER PARALLEL (SELENIUM GRID)

public void setup(String browserName) throws Exception{

if (browserName.equalsIgnoreCase("Firefox")) {

driver = new FirefoxDriver();

}

else if (browserName.equalsIgnoreCase("Chrome")) {

System.setProperty("webdriver.chrome.driver",

"C:/Users/MSTEMP/Downloads/Softwares/chromedriver\_win32/chromedriver.exe");

driver = new ChromeDriver();

}

else if (browserName.equalsIgnoreCase("ie")) {

System.setProperty("webdriver.ie.driver",

"C:/Users/MSTEMP/Downloads/Softwares/IEDriverServer/IEDriverServer.exe");

driver = new InternetExplorerDriver();

85. HOW TO STORE A VALUE WHICH IS TEXT BOXES USING WEBDRIVER ?

To obtain the **value** attribute of an element in the html document, we have to **use** the getAttribute() method. Then the **value** is passed as a parameter to the method.

86. HOW TO SWITCH BETWEEN THE WINDOWS in Selenium?

driver.switchTo().window(parentWindow);

87. HOW TO TYPE NEXT IN A NEW LINE INSIDE A TEXT AREA in Selenium?

**WebElement = driver.findElement(By.id("cke\_1\_contents"));**

**webelement.sendKeys(“This is line one.\n This is line two.”);**

88. HOW TO WORK WITH DYNAMIC WEBTABLE in Selenium?

Go to Website

**Step 2)**Right click on web element whose x-path is to be fetched.

**Step 3)** Right Click on highlighted web element > Select Copy -> Copy x-path option.

**Step 4)** Use the copied X-path "//\*[@id="leftcontainer"]/table/thead/tr/th [1]" in Selenium WebDriver to locate the element.

89. HOW TO WORK WITH RADIO BUTTON IN WEBDRIVER in Selenium?

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

90. HOW WE CAN RETRIEVE THE DYNAMICALLY CHANGING ADS in Selenium?

The elements for the dynamic ads cannot be found because it isn't fully loaded yet on load page. I recommend adding an explicit wait time for the specific ad is being looked for.

1.)Implement WebDriverWait: WebDriverWait wait = new WebDriverWait(driver, 20);

2.) Change your Dynamic\_ads data to wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("map"))); then do Dynamic\_ads.click(); afterwards.

91. IF A SELENIUM FUNCTION REQUIRES A SCRIPT ARGUMENT ,WHAT WOULD THAT ARGUMENT LOOK LIKE IN A GENERAL TERMS ?

That should be java script argument.

92. IF TESTNG I HAVE SOME TEST’S TEST-1 TEST-2 TEST-3 TEST-4 TEST-5 I WANT TO RUN MY EXECUTION ORDER IS TEST-5 TEST-4 TEST-3 TEST-2 TEST-1 .HOW DO YOU SET THE EXECUTION ORDER CAN YOU EXPLAIN FOR THAT ?

modifying the previous piece of code with **Priority Parameter** so that each test should run against to the priority assigned to them.

Example:

@Test (priority=1) Webdriver driver = new ChromeDriver

@Test (priority=2) Google

@Test (priority=3) FaceBook

93. LATEST VERSION OF FIREFOX AND SELENIUM IN MARKET AND HIM VERSION ON WHICH YOU ARE TESTING

Selenium is 3.0 and chrome version 88

94. LIST THE BROWSERS, OS SUPPORTED BY THE SELENIUM

Internet Explorer, Mozilla Firefox, Google Chrome, Safari and Opera.

95. PLEASE TELL ME THE DIFFERENCE B/W IMPLICIT WAIT AND EXPLICIT WAIT

**Implicit Wait** : If **wait** is set, it will **wait** for specified amount of time for each findElement/findElements call. ... **Explicit Wait** : If **wait** is set, it will **wait** and move on to next step when the provided condition becomes true else it will throw an exception after **waiting** for specified time.

96. PROVIDE DETAIL ABOUT TESTNG TEST OUTPUT FOLDER.

1. Right-click the MyFirstTestNGProject **folder** and select Refresh. A **test**-**output folder** will be created:
2. Open the **test**-**output folder**.
3. Right-click the index. html file.
4. Select Open with > Web Browser. The **report** will look like this:
5. Click (show). ...
6. Click the Chronological view.

97. SUPPOSE DEVELOPER CHANGED THE EXISTING IMAGE TO NEW IMAGE WITH SAME X PATH . WILL THE TEST CASE PASS OR FAIL ?

Then our testcase will be passing.

98. THERE IS A SCENARIO WHENEVER “ASSERT.ASSERTEQUALS()” FUNCTION FAILS AUTOMATICALLY IT HAS TO TAKE SCREENSHOT .HOW CAN YOU ACHIEVE THIS

EventFiringWebDriver eDriver=new EventFiringWebDriver(driver);

File srcFile = eDriver.getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(srcFile, new File(imgPath));

99. WHAT ARE OOPS CONCEPT ?

Object-Oriented Programming is a paradigm that provides many concepts, such as **inheritance**, **data binding**, **polymorphism**, etc.

100. WHAT ARE BENEFITS OF USING TESTNG ?

* It gives the ability to produce HTML Reports of execution.
* Annotations made testers life easy.
* Test cases can be Grouped & Prioritized more easily.
* Parallel testing is possible.
* Generates Logs.
* Data Parameterization is possible.

101. WHAT ARE BROWSER SUPPORTED BY SELENIUM IDE ?

Firefox and Chrome

102. WHAT ARE DIFFERENT ACCESS SPECIFIER IN JAVA ?

Java provides four types of **access modifiers** or visibility **specifiers** i.e. default, public, private, and protected. The default modifier does not have any keyword associated with it. When a class or method or variable does not have an **access specifier** associated with it, we assume it is having default **access**.

103. WHAT ARE THE DIFFERENT ASSERTIONS OR CHECK POINTS USED IN YOUR SCRIPT ?

* Is the page title as expected?
* Validations against an element on the page?
* Does text exist on the page?
* Does a javascript call return an expected value?

***Assert.assertEquals()*** is the method **used for validation**.

104. WHAT ARE THE DIFFERENT PARAMETERS FOR @ TEST ANNOTATION?

* String
* int/Integer
* boolean/Boolean
* byte/Byte
* char/Character
* double/Double
* float/Float
* long/Long
* short/Short

105. WHAT ARE DIFFERENT TYPES OF DRIVER IMPLEMENTATION ?

* FirefoxDriver
* InternetExplorerDriver
* ChromeDriver
* SafariDriver
* OperaDriver
* AndroidDriver
* IPhoneDriver
* HTMLUnitDriver

106. WHAT ARE THE FEATURES OF TESTNG?

* Supports annotations.
* TestNG uses more Java and OO features.
* Supports testing integrated classes (e.g., by default, no need to create a new test class instance for every test method).
* Separates compile-time test code from run-time configuration/data info.
* Flexible runtime configuration.
* Introduces ‘test groups’. Once you have compiled your tests, you can just ask TestNG to run all the "front-end" tests, or "fast", "slow", "database" tests, etc.
* Supports Dependent test methods, parallel testing, load testing, and partial failure.
* Flexible plug-in API.
* Support for multi threaded testing.

107. WHAT ARE THE OPERATING SYSTEM SUPPORTED BY SELENIUM ?

**Selenium supports OS** X, all versions of MS Windows, Ubuntu and other builds with ease.

108. WHAT ARE THE POSSIBLE SCENARIOS WHERE SELENIUM FINDING ELEMENT GET FAILS ?

* Selectors move on screen. ...
* Selector functionality changes. ...
* Incorrect comparisons. ...
* Over-precise comparisons. ...
* Underlying data changes. ...
* System in an unknown state. ...
* Errors may only become apparent later in the test. ...
* Timing errors.

109. WHAT IS THE BASIC USE OF FIREFOX PROFILES AND HOW CAN WE USE THEM USING SELENIUM ?

**Firefox profile** is the collection of settings, customization, add-ons and other personalization settings that **can** be done on the **Firefox** Browser. You **can** customize **Firefox profile to** suit your **Selenium** automation requirement. Also, **Firefox** or any other browser handles the SSL certificates settings

**Step 1)** First of all close the Firefox if open.

**Step 2)** Open Run (windows key + R) and type firefox.exe –p and click OK

**Step 3)** A dialogue box will open named Firefox – choose user profile

**Step 4)** Select option "Create Profile" from the window, and a wizard will open. Click on next

**Step 5)** Give your profile name which you want to create and click on finish button

110. WHAT IS THE DEFAULT TIME FOR SELENIUM IDE AND WEBDRIVER

By **default**, a **Selenium** test will have a maximum run **time** of 1800 seconds (30 minutes). If you need more **time** you can change that by passing the max\_duration capability along with a value.

111. WHAT IS THE DIFFERENCE B/W GET WINDOW HANDLE AND GET WINDOW HANDLES

getWindowHandles() – It stores the set of **handles** for all the pages opened simultaneously. driver. getWindowHandle() – It fetches the **handle** of the web page which is in focus. It gets the address of the active browser and it has a return type of String

113. WHAT IS DIFFERENCE BETWEEN @ BEFOR METHOD AND @ BEFORE CLASS

@BeforeTest will execute only one time **before** any test **methods**. **Methods** will run **before** executing any @Test annotated test **method** that is part **of** the <test> tag in testNG. xml file. @BeforeMethod will execute **before** every **method** annotated with @Test

114. WHAT IS DIFFERENCE BETWEEN ABSOLUTE XPATH AND RELATIVE XPATH

**Absolute Xpath**: It uses Complete path from the Root Element to the desire element. **Relative Xpath**: You can simply start by referencing the element you want and go from there. **Relative** Xpaths are always preferred as they are not the complete paths from the root element. ... So Always use **Relative** Xpaths in your Automation

116. WHAT IS DIFFERENCE BETWEEN SELENIUM RC AND WEBDRIVER

They both allow you to **use a programming language** in designing your test scripts.

They both allow you to **run your tests against different browsers.**

* WebDriver is **faster than Selenium RC** because of its simpler architecture.
* WebDriver **directly talks to the browser** while Selenium RC needs the help of the RC Server in order to do so.
* WebDriver's API is more**concise** than Selenium RC's.
* WebDriver **can support HtmlUnit** while Selenium RC cannot.
* The only drawbacks of WebDriver are:
  + It **cannot readily support new browsers**, but Selenium RC can.
  + It **does not have a built-in command** for automatic generation of test results.

117. WHAT IS DIFFERENCE BETWEEN SINGLE AND DOUBLE SLASH

**single slash** '/' anywhere in XPath signifies to look **for** the element immediately inside its parent element. A **double slash** '//' signifies to look **for** any child or any grand-child (descendant) element inside the parent element.

118. WHAT IS DIFFERENCE BETWEEN THREAD.SLEEP() AND SELENIUMSETSPEED(“2000”)

**Thread**. **sleep** : It causes the current **thread** to suspend execution **for** a specified period. **Selenium**. **setSpeed()** will stop the execution **for** the specified amount **of** time **for** every **selenium** command

119. WHAT IS DIFFERENCE BETWEEN THREAD.SLEEP() AND SELENIUM.SETSPEED()

The main **difference between** them is that: **setSpeed** sets a speed that will apply a delay time before every **Selenium** operation. **thread**. **sleep()** will set up wait only **for** once when called

120. WHAT IS MOST CHALLENGING TEST PROBLEM IN MY CAREER AUTOMATION

1. Pop-Up Windows

2. Identifying Dynamic Elements

3. Timeout or Sync Issue

4. Cross Browser Testing

5. Testing of Mobile Apps

6. Page Loading

7. Scalability

8. Multi-Tab Testing

9. Limited Reporting

10. Limited Captcha Handling

11. Supports Only Web-Based Applications

**Conclusion**

Selenium is one of the best tools to automate your testing process, but every good thing comes with some drawbacks. In this article, we have covered the most common challenges that are often faced by developers and testing teams during automation testing with Selenium.

 121. WHAT MOBILE DEVICES IT SUPPORT

Selenium willnot support, we need Appium to support mobile testing.

122. WHICH ARE THE DIFFERENT METHODS TO LOCATE AN ELEMENT

Name all 8 locators

123. WHICH ONE IS BETTER ONE XPATH OR CSS

**XPath** and **CSS** Selectors is that, with the **XPath** we can traverse both forward and backward whereas **CSS** selector only moves forward. Although **CSS** selectors perform far **better** than **Xpath** and it is well documented in Selenium community

124. WHICH REPOSITORY YOU HAVE USED TO STORE THE TEST SCRIPTS

QAs **store** all the element locators in a separate file, also referred to as a property file (. properties) in **Selenium**. The property file **stores** information in a key-value pair format. This file serves as an object **repository** in **Selenium WebDriver**

125. WHY WE REFER FIREFOX DRIVER TO THE WEBDRIVER INHERITANCE

**FirefoxDriver** is a class that has been written specifically for the **Firefox** browser. It has methods that are implemented and it can be instantiated. It can perform all functions (or methods) on the **Firefox** browser as defined in the interface **WebDriver**

126. WRITE A CODE TO MAKE USE OF ASSERT IF MY USERNAME IS INCORRECT

try{  
Assert.assertEquals(expUserName, actUserName);  
}catch(Exception e){  
Syste.out.println(“name is invalid”);  
}

127. WRITE A JAVA PROGRAM FOR SWAPPING OF TWO NUMBERS

**import** java.util.\*;

**class** Swap\_With {

**public** **static** **void** main(String[] args) {

**int** x, y, t;// x and y are to swap

       Scanner sc = **new** Scanner(System.in);

       System.out.println("Enter the value of X and Y");

       x = sc.nextInt();

       y = sc.nextInt();

       System.out.println("before swapping numbers: "+x +"  "+ y);

       /\*swapping \*/

       t = x;

       x = y;

       y = t;

       System.out.println("After swapping: "+x +"   " + y);

       System.out.println( );

128. WRITE DOWN SCENARIOS WHICH WE CAN’T AUTOMATE.

Bitmap comparison is not possible using Selenium WebDriver.

Automating Captcha is not possible using Selenium WebDriver.

We can not read bar code using Selenium WebDriver.

We can not automate OTP submission.

There are more things that cannot be automated using Selenium WebDriver.

129. DIFFERENCES BETWEEN SELENIUM WEBDRIVER RC , IDE AND RC ?

| Functionalities | Selenium IDE | Selenium RC | Selenium Webdriver |
| --- | --- | --- | --- |
| Record and playback | It has the record and playback feature. | It does not have a record and playback. | It does not have a record and playback. |
| Server | It requires no server to start execution of test cases. | It requires the server to start execution of test cases. | It requires no server to start execution of test cases. |
| Browser | It can be used for testing only in Firefox. | It can be used for testing in the majority of browsers. | It can be used for testing in the majority of browsers including in headless mode. |
| Object Oriented | It is based on Selenese which is a procedural language. | It can be partially used for object oriented programming. | It is majorly used for object oriented programming. |
| Dynamic Locators | Elements cannot be identified. | Elements cannot be identified. | Elements can be identified. |
| Alerts | Cannot handle alerts. | Cannot handle alerts. | Can handle alerts |
| Mouse Actions | Cannot handle mouse actions. | Cannot handle mouse actions. | Can handle mouse actions. |
| Dropdown | Cannot handle dropdown. | Cannot handle dropdown. | Can handle dropdown. |
| iPhone/Android | Cannot perform iPhone/Android testing. | Cannot perform iPhone/Android testing. | Can perform iPhone/Android testing with the help of Android Driver, iPhone Driver. |
| Listener | Does not have a Listener. | Does not have a Listener. | Have Listeners. |
| Performance | Fast [comes as a Firefox plugin]. It interacts with the browser directly. | It does not interact with the browser directly. Hence on a slower side compared to webdriver. | Fast as it interacts directly with the browser. |
| Architecture | Derived from Javascript. | Derived from Javascript. | Not derived from Javascript. |
| Usage | UI interface available to create scripts. | Standalone Jars available to execute test cases in the browser. | Contains API and supported by languages like Java, Python, and Ruby and so on. |
| Xpath | Only has absolute xpath. | Only has absolute xpath. | Has both absolute and relative xpath. |
| Navigation | Cannot handle navigation. | Cannot handle navigation. | Can handle navigation |

130. DO SHOPPING IN FLIPKART ?

131. WHAT DOES DESIRED CAPABILITES DO?

**are** a set of key-value pairs encoded as a JSON object. It helps QAs define basic test requirements such as operating systems, browser combinations, browser versions, etc. within **Selenium** test scripts

132. WHAT IS ACTIONS CLASS IN WEBDRIVER ?

 is an ability provided by **Selenium** for handling keyboard and mouse events. In **Selenium WebDriver**, handling these events includes operations such as drag and drop, clicking on multiple elements with the control key, among others. These operations are performed using the advanced user interactions API.

133. WHAT IS DIFFERENCE BETWEEN OVERLOAD AND OVERRIDE

**Overloading** happens when 2 methods **in the** same class have the same name but different parameters. **Overriding** means having 2 methods with the same name and same parameters, one being **in a** parent class and the other **in a** child class that inherits from the parent class

134. WHAT IS DIFFERENCE BETWEEN WEBDRIVER AND LISTENER AND TESTNG LISTENER

WebDriver Event **Listener** is to listen the events triggered by **webdriver** like beforeClickOn, afterClickOn, beforeFindBy, afterFindBy, etc and take actions. It is mainly used to write log file **for selenium** test execution. **TestNG listener** mainly used to generate the report **for** the test

135. Read cell from html table

**TestNG** allows you to perform ordered groupings of **test** methods. You **can** not only declare that methods belong to **groups**, but you **can** also specify **groups** that contain other **groups**. ... **Groups** are specified in your **testng**. xml file and **can** be found either under the <**test**> or <suite> tag

136.  Logic for looping the table, xpath to iterate rows & columns

This is the way to get the value of each cell of a web table using selenium

**1. Table initialize**

In your case, the table has the id tbr, the way to initialize it :

WebElement tbl = driver.findElement(By.id("tbr"));

**2. Row initialize**

The name tag for the web table row in general is tr, the way to initialize it :

List<WebElement> rows = tbl.findElements(By.tagName("tr"));

**3. Column initialize**

The name tag for the web table column in general are th or td, the way to initialize it :

th tag

List<WebElement> cols = rows.get(rowIndex).findElements(By.tagName("th"));

td tag

List<WebElement> cols = rows.get(rowIndex).findElements(By.tagName("td"));

**So you can get particular cell values by :**

String cell = cols.get(indexCol).getText();

The bellow code is function to get all value cell table (without getting header th value)

WebElement tbl = driver.findElement(By.id("tbr"));

//check all row, identification with 'tr' tag

List<WebElement> rows = tbl.findElements(By.tagName("tr"));

//row iteration

for(int i=0; i<rows.size(); i++) {

//check column each in row, identification with 'td' tag

List<WebElement> cols = rows.get(i).findElements(By.tagName("td"));

//column iteration

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

System.out.println(cols.get(j).getText());

}

137. Can we run Group of test cases using TestNG

**TestNG** allows you to perform ordered groupings of **test** methods. You **can** not only declare that methods belong to **groups**, but you **can** also specify **groups** that contain other **groups**. ... **Groups** are specified in your **testng**. xml file and **can** be found either under the <**test**> or <suite> tag

140. Code for opening firefox browser

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

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

141. Does selenium support https protocols ?

Yes it can support https protocols

142. Downloading a file and save it?

options = Options()

options.add\_experimental\_option("prefs", {

"download.default\_directory": r"C:\Users\xxx\downloads\Test",

"download.prompt\_for\_download": False,

"download.directory\_upgrade": True,

"safebrowsing.enabled": True

})

143. Explain any 3 testNG annotation

|  |  |
| --- | --- |
| 1 | **@BeforeSuite**  The annotated method will be run only once before all tests in this suite have run. |
| 2 | **@AfterSuite**  The annotated method will be run only once after all tests in this suite have run. |
| 3 | **@BeforeClass**  The annotated method will be run only once before the first test method in the current class is invoked. |

144. Google search and finding no of results

145. TestNG- Write sample code to select browser depending on parameter given in testing.xml

146. Parameter annotation, how to add system property for chrome/iedriver.exe for chrome & IE

147. How to check result with expected

148. Assert.assertEquals().

149. implicit wait and explicit wait- WebDriverWait or Thread.sleep()

150. Can we execute java code without main()no, only can compile

Yes, **we can execute** a **java program without** a **main** method by using a static block. Static block in **Java** is a group of statements that gets executed **only** once when the class is loaded into the memory by **Java** ClassLoader, It is also known as a static initialization block

151. OOPS concept and diff between overloading and overriding

What is **Overloading and Overriding**? When two or more methods in the same class have the same name but different parameters, it's called **Overloading**. When the **method** signature (name and parameters) are the same in the superclass and the child class, it's called **Overriding**.

152. Find the top from array list(10 number) using java code

import java.util.Arrays;

public class GFG {

    public static void main(String[] args){

        int arr[] = {10, 324, 45, 90, 9808};

        int max = Arrays.stream(arr).max().getAsInt();

        System.out.println("Largest in given array is " +max);

    }

}

153. check logic for Fibonacci series, even/odd number, top value from 5th number

154. what are the access modifier and difference.public, private, protected, default

A **private** member ( i ) is only accessible within the same class as it is declared. A member with no **access modifier** ( j ) is only accessible within classes in the same package. A **protected** member ( k ) is accessible within all classes in the same package and within subclasses in other packages

155. What is abstract class with example

* **Abstract class:** is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class).

abstract class Animal {

public abstract void animalSound();

public void sleep() {

System.out.println("Zzz");

}

156. Polymorphism

157. Inheritance

158. Collection-set,list,map

159. Exception-exception,error,assertion

160. Oops-Type casting of object

161. Wrapper classes

Wrapper classes allows us to convert the primitive types into an object type. Java is not 100% object-oriented programming language because of the 8 primitive types. Then wrapper classes are introduced to give the primitive types of an object form. So, the primitive types can also be stored as an object of its respective wrapper class the 8 primitive types and its wrapper classes are, byte. - Byte int - Integer short - Short long - Long float - Float double - Double char - Character Boolean - Boolean all this wrapper classes are available in java. Lang package Now if you want to store an integer as an object type you can write it as Integer i=new Integer(10);

This is known as Boxing, converting a primitive type into an object. Now to get that integer value back, int a=i.intValue(); This operation is known as Unboxing converting the value of a wrapper class object into a primitive type. After java 1.5/5 Boxing and Unboxing became automatic. You can directly assign the primitive as an object of wrapper class. Integer i=10; Before java 1.5/5, databases store only Object class objects so to store a primitive type into an Array List or Vector wrapper classes are used. Primitive types are converted into wrapper class object through boxing and upcasted to Object type to store in the database. The significance of wrapper class comes when you want to write a program which will work with any type of value. To write such a program declare the arguments as Object type since Object class is extended by all the other class in java Object class type can store any kind of objects. Let’s take an example, you want to write a program which will work if you pass any kind of values public static void printHello(Object a) { System.out.println("Hello"); } This program will work with all kind of values. Since Object class is the super most class, Object type variable can accept any objects. the wrapper class comes into picture when you pass a primitive type of value. Imagine you are calling the function with a value 10 printHello(10); Now the 10 will be boxed and becomes an object of Integer class(wrapper class) since Integer extends Object the method will work fine. There are many other uses which comes while overriding the build in methods like compare(), equals() etc because all these functions have Object type as parameters.

162. Constructor-abstract class

163. Fibonacii,String reverse, remove duplicate from array, sort array, find min-max value from array

164. JDBC

165. TestNG-

166. Annotations

167. Listeners

168. Parameterization/Data provider

169. Parallel run- threading(XML)

170. Factory

171. how to create testng.xml programmatically

172. Webdriver-

173. Eventfiringwebdriver

174. Capabilities

175. IE issues & resolution-6-7

176. Exceptions

177. Class interface e.g.

178. Synchronization-waits

179. Frame

180. xpath,css-functions-contains,sibilings,starts-ends with, relative-absolute xpath,nth-child,nth-of-typ