1. What is Automation Testing?

Automation testing is the process of testing the software 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. Sometimes based on the project and the test time estimation, we do focus on End to End 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. Why do you prefer Selenium Automation Tool?

· Free and open source

· Have large user base and helping communities

· Cross browser compatibility

· Platform compatibility

· Multiple programming languages support

6. What is Selenium?

Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers. It has gained a lot of popularity in terms of web-based automated testing and giving a great competition to the famous commercial tool HP QTP (Quick Test Professional) AKA HP UFT (Unified Functional Testing).

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,

i Selenium IDE (Integrated Development Environment)

ii Selenium RC (Remote Control) – selenium 1

iii Selenium WebDriver – selenium 2 & 3

iv Selenium Grid

7. What is Selenium IDE?

Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. 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 supports only firefox browser. We cannot record the action using selenium ide on any other browser.
* It doesn’t allow usage of the conditional or looping statements
* We cannot parameterize our test scenarios in Selenium IDE
* It doesn’t support scripting of test in a programming language, it only supports html
* We cannot generate reports, or effective logs with IDE

8 What is Selenese?

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

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

Firefox

10. 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.

11. 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.

12. What is Selenium Grid?

Selenium Grid is a tool used together with Selenium RC 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.

13. 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

14. 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

15. What is a hub in Selenium Grid?

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

16. 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.

3. What are some advantages of selenium?

Following are the advantages of selenium-

Selenium is open source and free to use without any licensing cost.

It supports multiple languages like Java, ruby, python etc.

It supports multi browser testing.

It has good amount of resources and helping community over the internet.

Using selenium IDE component, non-programmers can also write automation scripts

Using selenium grid component, distributed testing can be carried out on remote machines possible.

17. What are some limitations of selenium?

Following are the limitations of selenium-

* We cannot test desktop application using selenium.
* We cannot test web services using selenium.
* For creating robust scripts in selenium webdriver, programming langauge knowledge is required.
* We have to rely on external libraries and tools forperforming tasks like – logging(log4J), testing framework-(testNG, JUnit), reading from external files(POI for excels) etc.

18. Which all browsers are supported by selenium webdriver?

Some commonly used browsers supported by selenium are-

Google Chrome – ChromeDriver

Firefox – FireFoxDriver

Internet Explorer – InternetExplorerDriver

Safari – SafariDriver

HtmlUnit (Headless browser) – HtmlUnitDriver

Android – Selendroid/Appium

IOS – ios-driver/Appium

EventFiringWebDriver

19. Can we test APIs or web services using selenium webdriver?

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

20. What are the testing type supported by Selenium WebDriver?

Selenium webdriver can be used for performing automated functional and regression testing.

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.So it is called as Headless Browser Testing.

HtmlUnitDriver driver = new HtmlUnitDriver();

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 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.

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 single forward slash(//) as shown below.

//input[@id='email']

29. How can we inspect the web element attributes in order to use them in different locators?

Using Firebug or developer tools we can inspect the specific web elements.

Firebug is a plugin of firefox that provides various development tools for debugging applications. From automation perspective, firebug is used specifically for inspecting web-elements in order to use their attributes like id, class, name etc. in different locators.

30.How can we locate an element by only partially matching its attributes value in Xpath?

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

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

The above statement will match the all the values of name attribute containing the word ‘user’ in them.

31.How can we locate elements using their text in XPath?

Using the text() method –

xPathExpression = //\*[text()='username']

32.How can we move to parent of an element using XPath?

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

33.How can we move to nth child element using XPath?

There are two ways of navigating to the nth element using XPath-

Using square brackets with index position-

Example – div[2] will find the second div element.

Using position()-

Example – div[position()=3] will find the third div element.

34.What is the syntax of finding elements by class using CSS Selector?

By .className we can select all the element belonging to a particular class e.g. ‘.red’ will select all elements having class ‘red’.

35.What is the syntax of finding elements by id using CSS Selector?

By #idValue we can select all the element belonging to a particular class e.g. ‘#userId’ will select the element having id – userId.

36.. How can we select elements by their attribute value using CSS Selector?

Using [attribute=value] we can select all the element belonging to a particluar class e.g. ‘[type=small]’ will select the element having attribute type of value ‘small’.

37. How can we move to nth child element using css selector?

Using :nth-child(n) we can move to the nth child element e.g. div:nth-child(2) will locate 2nd div element of its parent.

38. What is fundamental difference between XPath and css selector?

The fundamental difference between XPath and css selector is using XPaths we traverse up in the document i.e. we can move to parent elements. Whereas using CSS selector we can only move downwards in the document.

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

Assert: 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.

For detailed post check the below link.

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

Soft Assert: Soft Assertions are customized error handlers provided by TestNG. Soft Assertions do not throw exceptions when assertion fails, and they simply continue to the next test step. They are commonly used when we want to perform multiple assertions.

SoftAssert assert=new Softassert();

To mark a test as failed with soft assertions, call assertAll() method at the end of the test.

Hard Assert: Hard Assert throws an AssertException immediately when an assert statement fails and test suite continues with next @Test

41. 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.)

42.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();

43. Is the FirefoxDriver a Class or an Interface?

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

44. What is the super interface of WebDriver?

SearchContext.

45. 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.

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

WebDriver driver = new FirefoxDriver();

instead of creating

FirefoxDriver driver = new FirefoxDriver();

47.What is the purpose of doing this way?

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 = new FirefoxDriver();

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

NoSuchElementException – When no element could be located from the locator provided.

ElementNotVisibleException – When element is present in the dom but is not visible.

NoAlertPresentException – When we try to switch to an alert but the targetted alert is not present.

NoSuchFrameException – When we try to switch to a frame but the targetted frame is not present.

NoSuchWindowException – When we try to switch to a window but the targetted window is not present.

UnexpectedAlertPresentException – When an unexpected alert blocks normal interaction of the driver.

TimeoutException – When a command execution gets timeout.

InvalidElementStateException – When the state of an element is not appropriate for the desired action.

NoSuchAttributeException – When we are trying to fetch an attribute’s value but the attribute is not correct

WebDriverException – When there is some issue with driver instance preventing it from getting launched.

49.What is exception test in Selenium?

An exception test is an exception that you expect will be thrown inside a test class. If you have written a test case in such way that it should throw an exception, then you can use the @Test annotation and specify which exception you will be expecting by mentioning it in the parameters. Take a look at the example below: @Test(expectedException = NoSuchElementException.class)

Do note the syntax, where the exception is suffixed with .class

50.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>

51.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 –

• Explicit Waits –

• Fluent Waits –

• PageLoadTimeOut

• Thread.sleep() – static wait

52.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.

Driver.manage.timouts.implicitlyWait(10,TimeUint.SECONDS)

53.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.

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("someid")));

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

elementToBeClickable(WebElement element or By locator)

stalenessOf(WebElement element)

visibilityOf(WebElement element)

visibilityOfElementLocated(By locator)

invisibilityOfElementLocated(By locator)

attributeContains(WebElement element, String attribute, String value)

alertIsPresent()

titleContains(String title)

titleIs(String title)

textToBePresentInElementLocated(By, String)

54.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.

Wait wait = new FluentWait(driver) .withTimeout(30, SECONDS) .pollingEvery(5, SECONDS).ignoring(NoSuchElementException.class);

WebElement foo = wait.until(new Function() {

public WebElement apply(WebDriver driver) {

return driver.findElement(By.id("foo"));

}

});

55.How can we type text in a textbox element using selenium?

Using sendKeys() method we can type text in a textbox-

WebElement searchTextBox = driver.findElement(By.id("search"));

searchTextBox.sendKeys("searchTerm");

56. How can we clear a text written in a textbox?

Using clear() method we can delete the text written in a textbox.

driver.findElement(By.id("elementLocator")).clear();

57.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'");

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

By using getText() method

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

By using getAttribute(value);

driver.findElement(By.id(“elementLocator”)).getAttribute(“value”);

60. How to check a checkBox in selenium?

The same click() method used for clicking buttons or radio buttons can be used for checking checkbox as well.

61. How can we submit a form in selenium?

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

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

Also, the click() method can be used for the same purpose.

62.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);

63. 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.

64. 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");

65. 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”)

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

67. 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

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

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

driver.getCurrentUrl();

69. 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();

70. How to delete cookies in Selenium?

To delete cookies we use deleteAllCookies() method

driver.manage().deleteAllCookies();

71. 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

Explain the difference between close and quit command.

driver.close() – Used to close the current browser having focus

driver.quit() – Used to close all the browser instances

72. How to switch between multiple windows in selenium?

Selenium has driver.getWindowHandles() and driver.switchTo().window(“{windowHandleName}”) commands to work with multiple windows. The getWindowHandles() command returns a list of ids corresponding to each window and on passing a particular window handle to driver.switchTo().window(“{windowHandleName}”) command we can switch control/focus to that particular window.

for (String windowHandle : driver.getWindowHandles()) {

driver.switchTo().window(handle);

}

73. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in selenium?

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

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

74. How can we move to a particular frame in selenium?

The driver.switchTo() commands can be used for switching to frames.

driver.switchTo().frame("{frameIndex/frameId/frameName}");

For locating a frame we can either use the index (starting from 0), its name or Id.

75.What are the different keyboard operations that can be performed in selenium?

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

.sendKeys(“sequence of characters”) – Used for passing charcter sequesnce to an input or textbox element.

.pressKey(“non-text keys”) – Used for keys like control, function keys etc that ae non text.

.releaseKey(“non-text keys”) – Used in conjuntion with keypress event to simulate releasing a key from keyboard event.

76.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)

77.What are the different mouse actions that can be performed?

The different mouse evenets supported in selenium are

click(WebElement element)

doubleClick(WebElement element)

contextClick(WebElement element)

mouseDown(WebElement element)

mouseUp(WebElement element)

mouseMove(WebElement element)

mouseMove(WebElement element, long xOffset, long yOffset)

78. Write the code to double click an element in selenium?

Code to double click an element in selenium-

Actions action = new Actions(driver);

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

action.doubleClick(element).build().perform();

79. Write the code to right click an element in selenium?

Code to right click an element in selenium-

Actions action = new Actions(driver);

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

action.contextClick(element).build().perform();

80. How to mouse hover an element in selenium?

Code to mouse hover over an element in selenium-

Actions action = new Actions(driver);

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

action.moveToElement(element).perform();

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

We use Actions class to do Drag And Drop Action

Action.dragAndDrop(source,destination)

82.How can we fetch title of the page in selenium?

Using driver.getTitle(); we can fetch the page title in selenium. This method returns a string containing the title of the webpage.

83. How can we fetch the page source in selenium?

Using driver.getPageSource(); we can fetch the page source in selenium. This method returns a string containing the page source.

84. How to verify tooltip text using selenium?

Tooltips webelements have an attribute of type ‘title’. By fetching the value of ‘title’ attribute we can verify the tooltip text in selenium.

String toolTipText = element.getAttribute(“title”);

85. How to locate a link using its text in selenium?

Using linkText() and partialLinkText() we can locate a link. The difference between the two is linkText matches the complete string passed as parameter to the link texts. Whereas partialLinkText matches the string parameter partially with the link texts.

WebElement link1 = driver.findElement(By.linkText(“artOfTesting”));

WebElement link2 = driver.findElement(By.partialLinkText(“artOf”));

86. What are DesiredCapabilities in selenium webdriver?

Desired capabilities are a set of key-value pairs that are used for storing or configuring browser specific properties like its version, platform etc in the browser instances.

87.How can you redirect browsing from a browser through some proxy?

Selenium provides a PROXY class to redirect browsing from a proxy. Look at the example below:

String PROXY = “199.201.125.147:8080”;

org.openqa.selenium.Proxy proxy = new.org.openqa.selenium.Proxy();

proxy.setHTTPProxy(Proxy).setFtpProxy(Proxy) .setSslProxy(Proxy)

DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();

capabilities.setCapability(CapabilityType.PROXY, proxy);

capabilities.setCapability(CapabilityType.BROWSER\_NAME, "IE");

capabilities.setCapability(InternetExplorerDriver.

INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS,true);

System.setProperty("webdriver.ie.driver", "C:\\IEDriverServer.exe");

//it is used to initialize the IE driver

WebDriver driver = new InternetExplorerDriver(capabilities)

88.How to handle dropdowns in selenium?

Using Select class-

Select countriesDropDown = new Select(driver.findElement(By.id(“countries”)));

dropdown.selectByVisibleText(“India”);

//or using index of the option starting from 0

dropdown.selectByIndex(1);

//or using its value attribute

dropdown.selectByValue(“Ind”);

89. How to check which option in the dropdown is selected?

Using isSelected() method we can check the state of a dropdown’s option.

Select countriesDropDown = new Select(driver.findElement(By.id(“countries”)));

dropdown.selectByVisibleText(“India”);

//returns true or false value

System.out.println(driver.findElement(By.id(“India”)).isSelected());

90. How can we check if an element is getting displayed on a web page?

Using isDisplayed method we can check if an element is getting displayed on a web page.

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

91. How can we check if an element is enabled for interaction on a web page?

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

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

93.How can we capture screenshots in selenium?

Using getScreenshotAs method of TakesScreenshot interface we can take the screenshots in selenium.

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

FileUtils.copyFile(scrFile, new File(“D:\\testScreenShot.jpg”));

94. How can we handle window UI elements and window POP ups using selenium?

Selenium is used for automating Web based application only(or browsers only). For handling window GUI elements we can use AutoIT. AutoIT is a freeware used for automating window GUI. The AutoIt scripts follow simple BASIC lanaguage like syntax and can be easily integrated with selenium tests.

95. What is Robot API?

Robot API is used for handling Keyboard or mouse events. It is generally used to upload files to the server in selenium automation.

Robot robot = new Robot();

//Simulate enter key action

robot.keyPress(KeyEvent.VK\_ENTER);

96. How to do file upload in selenium?

File upload action can be performed in multiple ways-

Using element.sendKeys(“path of file”) on the webElement of input tag and type file i.e. the elements should be like –

<input type=”file” name=”fileUpload”>

Using Robot API.

Using AutoIT API.

Using Sikuli

97.. How to handle HTTPS website in selenium? or How to accept the SSL untrusted connection?

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

FirefoxProfile profile = new FirefoxProfile();

profile.setAcceptUntrustedCertificates(true);

profile.setAssumeUntrustedCertificateIssuer(false);

WebDriver driver = new FirefoxDriver(profile);

98.How to handle alerts in selenium?

In order to accept or dismiss an alert box the alert class is used. This requires first switching to the alert box and than using accept() or dismiss() command as the case may be.

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

//To accept the alert

alert.accept();

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

//To cancel the alert box

alert.dismiss();

99.How to execute javascript in selenium?

JavaScript can be executed in selenium using JavaScriptExecuter. Sample code for javascript execution-

WebDriver driver = new FireFoxDriver();

if (driver instanceof JavascriptExecutor) {

((JavascriptExecutor)driver).executeScript(“{JavaScript Code}”);

}

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

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

100.How to handle hidden elements in Selenium webDriver?

Using javaScript executor we can handle hidden elements-

(JavascriptExecutor(driver)) .executeScript(“document.getElementsByClassName(ElementLocator).click();”);

101.why my javascript execution return null?

You need to return from java script snippets

(JavascriptExecutor(driver)) .executeScript.(document.title); will return null

(JavascriptExecutor(driver)) .executeScript.(return document.title); will title

### 102.**How to scroll down a page using JavaScript in Selenium?**

### We can scroll down a page by using window.scrollBy() function. Example:

|  |  |
| --- | --- |
|  | ((JavascriptExecutor) driver).executeScript("window.scrollBy(0,500)"); |

**103. How to scroll down to a particular element?**

To scroll down to a particular element on a web page, we can use the function **scrollIntoView()**. Example:

|  |  |
| --- | --- |
|  | ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", element); |

104.How to scroll down a page using JavaScript in Selenium?

We can scroll down a page by using window.scrollBy() function. Example:

((JavascriptExecutor) driver).executeScript("window.scrollBy(0,500)");

105. How to scroll down to a particular element?

To scroll down to a particular element on a web page, we can use the function scrollIntoView(). Example:

((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", element);

106.What is Page Object Model or POM?

Page Object Model(POM) is a design pattern in selenium. A design pattern is a solution or a set of standards that are used for solving commonly occuring software problems.

Now coming to POM – POM helps to create a framework for maintaining selenium scripts. In POM for each page of the application a class is created having the web elements belonging to the page and methods handling the events in that page. The test scripts are maintained in seperate files and the methods of the page object files are called from the test scripts file.

107. What are the advantages of POM?

The advantages are POM are-

Using POM we can create an Object Repository, a set of web elements in seperate files along with their associated functions. Thereby keeping code clean.

For any chnage in UI(or web elements) only page object files are required to be updated leaving test files unchanged.

It makes code reusable and maintable.

108. What is Page Factory?

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

public class SamplePage {

WebDriver driver;

@FindBy(id="search")

WebElement searchTextBox;

@FindBy(name="searchBtn")

WebElement searchButton;

//Constructor

public samplePage(WebDriver driver){

this.driver = driver;

//initElements method to initialize all elements

PageFactory.initElements(driver, this);

}

//Sample method

public void search(String searchTerm){

searchTextBox.sendKeys(searchTerm);

searchButton.click();

}

}

109. What is an Object repository?

An object repository is centralized location of all the object or WebElements of the test scripts. In selenium we can create object repository using Page Object Model and Page Factory design patterns.

110. What is a data driven framework?

A data driven framework is one in which the test data is put in external files like csv, excel etc separated from test logic written in test script files. The test data drives the test cases, i.e. the test methods run for each set of test data values. TestNG provides inherent support for data driven testing using @dataProvider annotation.

111. What is a keyword driven framework?

A keyword driven framework is one in which the actions are associated with keywords and kept in external files e.g. an action of launching a browser will be associated with keyword – launchBrowser(), action to write in a textbox with keyword – writeInTextBox(webElement, textToWrite) etc. The code to perform the action based on a keyword specified in external file is implemented in the framework itself.

In this way the test steps can be written in a file by even a person of non-programming background once all the identified actions are implemented.

112. What is a hybrid framework?

A hybrid framework is a combination of one or more frameworks. Normally it is associated with combination of data driven and keyword driven frameworks where both the test data and test actions are kept in external files(in the form of table).

113.Name an API used for reading and writing data to excel files.

Apache POI API and JXL(Java Excel API) can be used for reading, writing and updating excel files.

114. Name an API used for logging in Java.

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

115. What is the use of logging in automation?

Logging helps in debugging the tests when required and also provides a storage of test’s runtime behaviour.

116.How to set the size of browser window using Selenium?

To maximize the size of browser window, you can use the following piece of code:

driver.manage().window().maximize(); – To maximize the window

To resize the current window to a particular dimension, you can use the setSize() method. Check out the below piece of code:

System.out.println(driver.manage().window().getSize());

Dimension d = new Dimension(420,600);

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

To set the window to a particular size, use window.resizeTo() method. Check the below piece of code:

((JavascriptExecutor)driver).executeScript("window.resizeTo(1024, 768);");

117.Explain how you will login into any site if it is showing any authentication popup for username and password?

Since there will be popup for logging in, we need to use the explicit command and verify if the alert is actually present. Only if the alert is present, we need to pass the username and password credentials. The sample code for using the explicit wait command and verifying the alert is below:

WebDriverWait wait = new WebDriverWait(driver, 10);

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

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

118.Explain eventfiringfriver?

EventFiringWebDriver is a class and is used to wrap your webdriver around to throw events and WebDriverEventListner is an interface which you have to implement to catch the webdriver events.

FirefoxDriver driver = new FirefoxDriver();

EventFiringWebDriver eventDriver = new EventFiringWebDriver(driver);

EventHandler handler = new EventHandler();

eventDriver.register(handler);

eventDriver.get("http://toolsqa.wpengine.com/automation-practice-switch-windows/");

WebElement element = eventDriver.findElement(By.id("target"));

element.click();

read once from toolsqa.

119.how to handle webtable,calender,broken links,google search using selenium

120.Waht is dom?

121.Explain architecture of selenium

122.how to handle bootstrap elements using selenium?

123)how to run Java testcase using Maven?

124)Maven surfire plugin information

125)Explain POI?

126)How to execute Failed testcases?

Using IretryAnalyzer

127)how to read config .properties file

128)which methods are used in TestUTIl in your project?