Selenium Java Interview Questions and Answers Part-2

1) What are the different types of navigation commands in Selenium?

Different navigation commands in selenium are:

– navigate().to();

-navigate().forward();

-navigate().back();

-navigate().refresh();

2) How to fetch the current page URL in Selenium WebDriver?

We can use the getCurrentUrl() method to get the current page URL.

```
driver.getCurrentUrl();
```

3) How can we maximize browser window in Selenium WebDriver?

We can use the maximize() method to maximize browser window.

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

4) How to delete cookies in Selenium?

We can use deleteAllCookies() method to delete cookies in selenium.

```
driver.manage().deleteAllCookies();
```

5) What are the different ways for refreshing the page using Selenium WebDriver?

Browser refresh operation can be performed using the following ways in Selenium:

– Refresh method

```
driver.manage().refresh();
```

– Get method

```
driver.get(“https://www.google.com”);
driver.get(driver.getCurrentUrl());
```

-Navigate method

```
driver.get("https://www.google.com");
driver.navigate.to(driver.getCurrentUrl());
```

-SendKeys method

```
driver. findElement(By.id("username")).sendKeys(Keys.F5);
```

6) What is the difference between driver.getWindowHandle() and driver.getWinowHandles() in Selenium WebDriver and their return type?

driver.getWindowHandle() – To get the window handle of the current window. Returns a string of alphanumeric window handle

driver.getWinowHandles() – To get the window handle of all current windows. Return a set of window handles

7) How to handle hidden elements in Selenium WebDriver?

We can use the JavaScriptExecutor to handle hidden elements.

```
JavascriptExecutor js = (JavascriptExecutor)driver;
js.executeScript("document.getElementById('displayed-text').value='text123'");
```

8) How can you find broken links in a page using Selenium WebDriver?

```
  List<WebElement> elements = driver.findElements(By.tagName("a"));
   List finalList = new ArrayList();
   for (WebElement element : elementList){
  if(element.getAttribute("href") != null){
  finalList.add(element);
   }
}
return finalList;
```

9) How to find more than one web element in the list?

We can find more than one web element by using the findElements() method in Selenium.

```
List<WebElement> elements = driver.findElements(By.tagName("a"));
```

10) How to read a JavaScript variable in Selenium WebDriver?

```
//Creating the JavascriptExecutor interface object by Type casting
JavascriptExecutor js = (JavascriptExecutor)driver;

//Perform Click on LOGIN button using JavascriptExecutor
js.executeScript("arguments[0].click();", button);
```

11) What is JavascriptExecutor and in which case JavascriptExecutor will help in Selenium automation?

JavaScriptExecutor is an Interface that helps to execute JavaScript through Selenium Webdriver.

In case, when selenium locators do not work you can use JavaScriptExecutor. You can use JavaScriptExecutor to perform a desired operation on a web element.

12) How to handle Ajax calls in Selenium WebDriver?

The best approach would be to wait for the required element in a dynamic period and then continue the test execution as soon as the element is found/visible. This can be achieved with WebDriverWait in combination with ExpectedCondition , the best way to wait for an element dynamically, checking for the condition every second and continuing to the next command in the script as soon as the condition is met.

```
WebDriverWait wait = new WebDriverWait(driver, waitTime);
wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
```

13) List some scenarios which we cannot automate using Selenium WebDriver?

-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.

14) How you build object repository in your project framework?

We can build object repository using Page Object Model or Page Factory.

15) What is Page Object Model (POM) and its advantages?

Page Object Model is a design pattern for creating an object repository for web UI elements. Each web page in the application is required to have its own corresponding page class. The page class is thus responsible for finding the WebElements in that page and then perform operations on those web elements.

The advantages of using POM are:

-Allow us to separate operations and flows in the UI from verification – improves code readability

-Since the Object Repository is independent of test cases, multiple tests can use the same object repository

-Reusability of code

16) What is Page Factory?

Page Factory class in Selenium is an extension to the Page Object Design pattern. It is used to initialize the elements of the page object or instantiate the page objects itself.

Annotations in Page Factory are like this:

```
@FindBy(id = "userName")
WebElement txt_UserName;
```

OR

```
@FindBy(how = How.ID, using = "userName")
WebElement txt_UserName;
```

We need to initialize the page object like this:

```
PageFactory.initElements(driver, Login.class);
```

17) What is the difference between Page Object Model and Page Factory?

Page Object Model is a design pattern to create an Object Repository for web UI elements. However, Page Factory is a built-in class in Selenium for maintaining object repository.

18) What are the advantages of Page Object Model?

The advantages of using Page Object Model are:

-Allow us to separate operations and flows in the UI from verification – improves code readability

-Since the Object Repository is independent of test cases, multiple tests can use the same object repository

-Reusability of code

19) How can we use Recovery Scenario in Selenium WebDriver?

We can develop Recovery scenarios using exception handling i.e. By using “Try Catch Block” within your Selenium WebDriver Java tests

20) How to upload a file in Selenium WebDriver?

Uploading files in WebDriver is done by simply using the sendKeys() method on the file-select input field to enter the path to the file to be uploaded.

```
driver.get(baseUrl);
WebElement uploadElement = driver.findElement(By.id("uploadfile_0"));
// enter the file path onto the file-selection input field
uploadElement.sendKeys("C:\\newhtml.html");
```

21) How to download a file in Selenium WebDriver?

Step 1- Create a firefox Profile.

Step 2- set Preferences as per requirement.

Step 3- Open Firefox with firefox profile.

```
public class DownloadFiles {
public static void main(String[] args) {
// Create a profile
FirefoxProfile profile=new FirefoxProfile();
// Set preferences for file type
profile.setPreference("browser.helperApps.neverAsk.openFile", "application/octet-stream");
// Open browser with profile
WebDriver driver=new FirefoxDriver(profile);
// Set implicit wait
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
// Maximize window
driver.manage().window().maximize();
// Open APP to download application
driver.get("http://www.file.com/download_file");
// Click on download
driver.findElement(By.xpath("path")).click();
}
}
```

22) How to run Selenium WebDriver tests from command line?

```
java -jar selenium-server.jar -htmlSuite "*firefox" "http://10.8.100.106" "C:\mytestsuite\m
```

23) How to switch to frames in Selenium WebDriver?

For switching between frames, use driver.switchTo().frame(). First locate the frame id and define it in a WebElement.

Ex:

```
WebElement fr = driver.findElementById("theIframe");
driver.switchTo().frame(fr);
```

24) How to connect to a database in Selenium?

```
Connection con = DriverManager.getConnection(dbUrl,username,password);
```

25) How to resize browser window using Selenium WebDriver?

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