

Selenium Questions and Answers:

1. Difference between Assert and Verify

Assert: When an assert fails, the test will be terminated, rest of the test cases are skipped

Eg: Assert.assertEquals ("Selenium", driver.findElement (By.id ("txtSelenium")).getText ());

Verify: If Verify fails, the test will be continue executing rest of the test cases and logging the failure

2. How will you take screenshot & write a code ?

```
TakesScreenshot t=(TakesScreenshot) driver;  
File f1 = t.getScreenshotAs(OutputType.FILE);  
File f2=new File("D:/screenshot.png");  
FileUtils.copyDirectoryToDirectory(f1, f2);
```

3. How will you scroll down and write a code?

Code:

```
JavaScriptExecutor j=(JavaScriptExecutor) driver;  
WebElement w = driver.findElement(By.xpath("//*[@text()='Live Demo']")); j.executeScript("arguments[0].scrollIntoView(true)", w);
```

4. Explain Robot class and write a code?

We can't upload a file using selenium.

So we have one class is there for uploading file (i.e.) **Robot class**

public static void

uploadFiles(File path) { **try** {

Robot robot =

new Robot();

robot.setAutoD

elay(3000);

StringSelection selection = **new**

StringSelection(

path.getAbsolutePath());

Toolkit.getDefaultToolkit().getSystemClipboard()

.setContents(selection,

cton, **null**); //

press **ctrl+v**

robot.keyPress(KeyEvent.VK_

```

CONTROL);
robot.keyPress(KeyEvent.VK_
V); robot.setAutoDelay(3000);
// release ctrl+v
robot.keyRelease(KeyEvent.VK_
CONTROL);
robot.keyRelease(KeyEvent.VK_
V);
// press enter

robot.setAutoDelay(3000);
robot.keyPress(KeyEvent.V
K_ENTER);

robot.keyRelease(KeyEvent.V
K_ENTER); } catch
(AWTException e) {
e.printStackTrace();
}
}
}

```

Here ,

- Robot □ class
- StringSelection □ class
- Using this method we can upload any file.

5. How will you handle windows based popup , other than robot class?

We can handle window based popup by using AutoIt

6. Assume you launching IE browser you facing protected mode Exceptions, how you solve that?

Go to Internet option in IE □ security □ unchecked “Enable Protected mode”

7. WebDriver driver=new Firefoxdriver() Explain?

WebDriver –Interface.

FirefoxDriver-class.

Driver-Object.

8.FirefoxDriver is a class or interface, why you don’t use interface here?

- FirefoxDriver is a class
- If we use interface we can’t create object.

9.Write a code for web table to print all data?

```

public class Dummy {
public static void main(String[] args) {
System.setProperty("webdriver.gecko.driver",
"C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
WebDriver driver = new FirefoxDriver();

```

```

driver.get("http://toolsqa.com/automation-practice-table/");
List<WebElement> tRows =
driver.findElements(By.tagName("tr")); for(WebElement
rows:tRows){
List<WebElement> tData =
driver.findElements(By.tagName("td")); for(WebElement
data:tData){
System.out.println(data.getText());
}
}
}
}
}

```

10. Write a code for dropdown?

```

public class Dummy {
    public static void main(String[] args) {
        System.setProperty("webdriver.gecko.d
river",
"C:/Users/siva/workspace/Selenium/driver/geckodriver
.exe"); WebDriver driver = new FirefoxDriver();
driver.get("http://ironspider.ca/forms/dropdowns.htm");
WebElement w =
driver.findElement(By.name("coffee")); Select
s=new Select(w);
s.selectByIndex(3);
    }
}

```

11. Write a code for maven dependency?

```

<dependency>
    <groupId>org.seleniumhq.selenium</gr
oupId> <artifactId>selenium-
java</artifactId>
    <version>3.4.0</version>
</dependency>

```

12. Write a Junit dependency code for maven?

```

<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
</dependency>

```

13. Write a code for multiple windows handling?

```
public class Dummy5 {  
    public static void main(String[] args) throws InterruptedException {  
        System.setProperty("webdriver.gecko.driver",  
"C:/Users/siva/workspace/Selenium/driver/geckodriver.exe")  
        ; WebDriver driver =new FirefoxDriver();  
        driver.get("https://www.hdfcbank.com/");  
        driver.findElement(By.xpath(".*[@id='cee_closeBtn']/img")).click()  
        ; String p = driver.getWindowHandle();  
        System.out.println(p);  
        driver.findElement(By.xpath(".*[@id='loginsubmit']")).click();  
        Set<String> all = driver.getWindowHandles();  
        for (String x : all) {  
            System.out.println(x);  
            if(!p.equals(x)){  
                driver.switchTo().window(x)  
                ; Thread.sleep(3000);  
  
                driver.findElement(By.xpath("html/body/div[4]/div[2]/div[1]/a")).click();  
                Thread.sleep(2000);  
                driver.close();  
            }  
        }  
    }  
}
```

14. Uploading a file from local

We can't upload a file using selenium.

So we have one class is there for uploading file (i.e.) **Robot class**

```
public static void uploadFiles(File path) {  
    try {  
        Robot robot = new Robot();  
        robot.setAutoDelay(3000);  
        StringSelection selection = new  
            StringSelection(  
                path.getAbsolutePath());  
        Toolkit.getDefaultToolkit().getSystemClipboard()  
            .setContents(selection, null);  
  
        // press ctrl+v  
        robot.keyPress(KeyEvent.VK_CONTROL);  
        robot.keyPress(KeyEvent.VK_V);  
    }  
}
```

```

        robot.setAutoDelay(3000);
        // release ctrl+v
        robot.keyRelease(KeyEvent.VK_CONTROL);
        robot.keyRelease(KeyEvent.VK_V);
        // press enter
        robot.setAutoDelay(3000);
        robot.keyPress(KeyEvent.VK_ENTER);

        robot.keyRelease(KeyEvent.VK_ENTER);

    } catch (AWTException e) {

        e.printStackTrace();

    }

}

```

15. If u are joined newly here, what are all software and tools u need for automation

1. Eclipse
2. JDK
3. Selenium
4. Drivers for browsers
5. Git url
6. Maven

16. Automate gmail account check the number o messages in inbox

```

String inboxDetails = driver.findElement(By.id("inbox"));

System.out.println(inboxDetails.getText());

```

17. why we go for jenkins

Jenkins triggers a build for every change made in the source code repository for example Git repository. Once the code is built it deploys it on the test server for testing. Concerned teams are constantly notified about build and test results. Finally, Jenkins deploys the build application on the production server.

18. If I am having 100 test cases I want to execute 50 ly how do u do

Eg: @Test(enable = false)

By using enable = false attribute, we can skip some test cases in TestNG

19. How do u order the test

cases Eg: @Test(priority = 1)

@Test(priority = 2)

By using priority attribute, we can order the test cases in TestNG

19. Where do you keep your input data

we keep our input data in excel and database

```
List<HashMap<String, String>> mapDatasList = new
ArrayList(); try {
    File excelLocaltion = new File("./Excel/Adactin.xlsx");

    String sheetName = "Adact";

    FileInputStream f = new FileInputStream(
        excelLocaltion.getAbsolutePath());

    Workbook w = new XSSFWorkbook(f);
    Sheet sheet = w.getSheet(sheetName);
    Row headerRow = sheet.getRow(0);
    for (int i = 0; i < sheet.getPhysicalNumberOfRows(); i++)
    { Row currentRow = sheet.getRow(i);
    HashMap<String, String> mapDatas = new HashMap<String,
String>();
        for (int j = 0; j < headerRow.getPhysicalNumberOfCells(); j++)
        { Cell currentCell = currentRow.getCell(j);

            switch (currentCell.getCellType()) {
            case Cell.CELL_TYPE_STRING:

mapDatas.put(headerRow.getCell(j).getStringCellValue()
, currentCell.getStringCellValue());

                break;
            case Cell.CELL_TYPE_NUMERIC:

mapDatas.put(headerRow.getCell(j).getStringCellValue(),
                String.valueOf(currentCell
.getNumericCellValue()));

                break;
            }
        }
    }
```

```

        }

        mapDatasList.add(mapDatas);
    }

    // System.out.println(mapDatasList);
    String s = mapDatasList.get(1).get("Username");
    String s1 = mapDatasList.get(1).get("Password");
    System.out.println(s);
    System.out.println(s1);

    } catch (Throwable e) {
        e.printStackTrace();
    }

}

}

```

20. When you use list and set in selenium

While handling multiple windows I will use set

When am using findElements for radiobutton,checkbox I will use return type

List

21. Agile methodology

Backlog

grooming Sprint

Planning

Scrum meeting

Retrospective

22. Did u attend standup call what u discuss

S I attend, we will discuss about like

- a. What did yesterday
- b. What have planned today
- c. Is any obstacles in project

23. Regression testing

REGRESSION TESTING is to confirm that a recent code change has not affected existing features or functionalities.

24. Create issues in JIRA

1. Title: Mention the title of an Application

2. Issue Type: Bug

3. Related To: Sprint 5

4. Description: Explain entire flow

details Eg: I launched the browser

I enter the URL and then

I entered Username and password

then I click the login button

Expected: Need to navigate Home

page Actual: Got the error

5. Priority: High

6. Severity: Major

7. Assigned To: Manager

25.If I am having 2 scenarios, first two steps are same in both scenarios now how do I handle effectively

Using background we can handle this, mention the first two repeated steps in a background

26. If I am having TestNG annotations like this @BeforeTest

@BeforeMethod

@Test

@Test1

@Test2

Tell me the execution order

@BeforeTest

@BeforeMethod

@Test

@BeforeMethod

@Test1

@BeforeMethod

@Test2

26. Severity and priority

Priority

Priority is defined as the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.

Priority is categorized into three types

- ☐ Low
- ☐ Medium
- ☐ High

Severity

Defect Severity is defined as the degree of impact that a defect has on the operation of the product i.e. its not going to affect the functionality

Severity are categorized into five types

- o Critical
- o Major
- o Moderate
- o Minor
- o Cosmetic

27. Tell me one example which is very low severity with a high priority

A very low severity with a high priority: A logo error for any shipment website, can be of low severity as it not going to affect the functionality of the website but can be of high priority as you don't want any further shipment to proceed with wrong logo.

28.`INput[Id="123"]>/`

`L1='541'`

`L2='541'`

`L3='541'`

`L4='541'` find the xpath of last li?

```
//input[@id='123']//following::l4
```

29. What framework u worked in your project ?

POM

JUnit

Data Driven

Cucumber

30. Windows handling scenario: Like if I am currently in main window , first I have to close 2nd window then move to 3rd window and need to print "we are in 3rd window" again move to 4th window and print "4th window" then switch back to main window ?

```
String parentId = driver.getWindowHandle(); Set<String>
allWindows = driver.getWindowHandles(); int windowCount =
0;
for (String x : allWindows) {
    windowCount++;
    if (!parentId.equals(x)) { if
        (windowCount == 2) {
            driver.switchTo().window(x);
            driver.close();
            continue;
        }
        driver.switchTo().window(x);
        System.out.println("I am in " + windowCount +
"window");
    }
    driver.switchTo().defaultContent();}
```

31. Difference between window and alerts?

ALERTS:

If we click a login or any button one pop-up will show like OK or Cancel.

If we click OK or Cancel, then only it will move to home window

It has 3 types,

1. Simple Alert
2. Confirm pop-up
3. Prompt pop-up

WINDOW:

if we click login button, one separate window will open.

We have to switch from parent window to child window then only we can automate

Using window handling method we can switch

32.What are all the possible locators in selenium?

Id
Class
Name
Xpath
Link
Partiallinktext
Linktext

33.Difference between partial link text and link text?

linkText:

In linktext we take all text in the locator

```
WebElement web = driver.findElement(By.linkText("Product Category"));
```

partialLinkText:

In the partial link text we take partially in the text

```
WebElement web = driver.findElement(By.partialLinkText("Category"));
```

34.What are the selenium commands?

1. Browser commands
2. Browser navigation commands
3. Web element commands
4. Find element commands
5. Check box & radio button
6. Drop down and multiple select

35.How will you launch browser?

Add jar (selenium)

Browser driver

- ii. We first add the selenium jar in library

For ex, Firefox launching

```
public class FacebookAccount {  
    public static void main(String[] args) {  
        System.setProperty("webdriver.gecko.driver"  
        "  
        "C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");  
    }  
}
```

```

        WebDriver driver = new
        FirefoxDriver();
        driver.get("https://www.facebook.com/"
        );

        driver.quit();
    }
}

```

36. What is the difference between get() and navigate()?

1. If we use get() method, it only launch the url
2. But if we use navigate() method, we can perform back(), refresh() and forward() action

37.How will you handle the compile time exception?

- Using try-catch block or using throws keyword we can handle this exception

38.Difference between close() and quit()?

- 1.Close() □ it will close the current browser
- 2.Quit() □ it will destroy the webdriver object

39.Explain about implicit wait and

explicit wait

Implicit wait:

It applicable for all elements

The implicit wait will tell to the web driver to wait for certain amount of time before it throws a "No Such Element Exception"

Once we set the time, web driver will wait for that time before throwing an exception.

Syntax:

```
driver.manage().timeouts().implicitlyWait(time, TimeUnit.SECONDS);
```

Ex:

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

Explicit wait:

1. The explicit wait is used to tell the Web Driver to wait for particular

element to certain conditions (Expected Conditions) or the maximum time exceeded before throwing an exception.

2.The explicit wait is an intelligent kind of wait, but it can be applied only for specified elements.

3.Explicit wait gives better options than an implicit.

4.Once we declare explicit wait we have to use "ExpectedConditions" or we can configure how frequently we want to check the condition using Fluent Wait.

Syntax:

```
WebDriverWait wait = new WebDriverWait(driver,time);
```

40.Explain about JDBC

connection? JDBC

Connection Steps:

Load driver

Connect to database

Write sql query

Prepare the statement

Set the values

Execute query

41. Advantages of TestNG over Junit ?

1.TestNG allows us to execute of test cases based on group. Let's take a scenario where we have created two set of groups "Regression" & "Sanity". If we want to execute the test cases under Sanity group then it is possible in TestNG framework.

2.Parallel execution of Selenium test cases is possible in TestNG.

3.As method name constraint is present in JUnit, such method name constraint is not present in TestNG and you can specify any test method names.

4.TestNG supports following three 3 additional setUp/tearDown level: @Before/AfterSuite, @Before/AfterTest and @Before/AfterGroup.

5.TestNG do not require extend any class.

6.TestNG allows us to define the dependent test cases each test case is independent to other test case.

42.You are not working parallel execution in junit

Parallel execution of Selenium test cases is possible in TestNG only.

43.Where you used Interface in selenium

- ☐ WebDriver
- ☐ JavascriptExecutor

- Action

44 . What is broken link and how do you find it?

- Broken links are links or URLs that are not reachable.

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.

45.What is Selenium WebDriver?

- Selenium WebDriver is a web automation framework that allows you to **execute your tests against different browsers**,
- WebDriver supports programming languages like java,.net, PHP , python etc.
- WebDriver **can support HtmlUnit** browser

46. Assume one web page, in this page we have to check font color, font size and font format. How will you automate

1.Sometimes you need to **read font properties** like font size, font color, font family, font background color etc.. during **WebDriver test** case execution. **Selenium WebDriver** Is very vast API and It has many built In methods to perform very small operations on web page.

2.We can read the font properties in selenium webdriver using **.getCssValue()** method

Example program :

```
public class fontTest {  
    WebDriver driver = null;  
  
    @BeforeTest  
    public void setup() throws Exception  
    {  
        driver = new FirefoxDriver();  
        driver.manage().window().maximize();  
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);  
        driver.get("http://only-testing-blog.blogspot.in/2014/05/login.html");  
    }  
  
    @Test  
    public void readFontProperty(){  
        //Locate text string element to read It's font properties.  
        WebElement text = driver.findElement(By.xpath("//h1[contains(.,'Example Login Page')]"));  
  
        //Read font-size property and print It In console.
```

```
String fontSize = text.getCssValue("font-size");
System.out.println("Font Size -> "+fontSize);

//Read color property and print It In console.
String fontColor = text.getCssValue("color");
System.out.println("Font Color -> "+fontColor);

//Read font-family property and print It In console.
String fontFamily = text.getCssValue("font-family");
System.out.println("Font Family -> "+fontFamily);

//Read text-align property and print It In console.
String fonttxtAlign = text.getCssValue("text-align");
System.out.println("Font Text Alignment -> "+fonttxtAlign);
}
}
```

47. Define Bug

Lifecycle? New:

When a defect is logged and posted for the first time. It's state is given as new.

Assigned:

After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. Its state given as assigned.

Open:

- At this state the developer has started analyzing and working on the defect fix.

Fixed:

When developer makes necessary code changes and verifies the changes then he/she can make bug status as „Fixed“ and the bug is passed to testing team.

Pending retest:

After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.

Retest:

At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.

Verified:

The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to “verified”.

Reopen:

If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “reopened”. The bug goes through the life cycle once again.

Closed:

- Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to “closed”.

This state means that the bug is fixed, tested and approved.

Duplicate:

If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to “duplicate”.

Rejected:

If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to “rejected”.

Deferred:

The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.

Not a bug:

The state given as “Not a bug” if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and field of the application like change of colour of some text then it is not a bug but just some change in the looks of the application.

48.Selenium WebDriver version?

- We are using Selenium Standalone 3.4.0

49.What is the tool used for defect management

- We are using defect management tool JIRA, TFS

50.How will you do parallel browser checking?

- 1.Browsers will open almost simultaneously and your test will run in parallel.
- 2.Now just set the '**parallel**' attribute to '**tests**' in the testng.xml
- 3.As well as we set parameter as 'browser' and value as 'firefox' or 'chrome' in the testng.xml
- 4.Here we can set what are all the browsers we need to check parallelly, we can set.

51.Explain about Autoit?

- 1.Selenium is an open source tool that is designed to automate web-based applications on different browsers but to handle window GUI and non HTML popups in application. AutoIT is required as these window based activity are not handled by Selenium.
2. AutoIt v3 is also freeware. It uses a combination of mouse movement, keystrokes and window control manipulation to automate a task which is not possible by selenium webdriver.
- 3.Moving ahead we will learn how to upload a file in selenium web driver using autoIT. Here we need three tools in order to this.

- 1.Selenium Webdriver
- 2.AutoIT editor and element identifier
- 3.The window that you want to automate

52.What is absolute and realative xpath

Absolute path

- / represents absolute path
- Absolute path (shows Full HTML DOM structure)

Relative path(

- // represents relative path
- Relative path(shows particular locators)
- ➤In finding Xpath, matching node should be only one

53.Dropdown SelectbyVisibleText sample code

```
WebElement w = driver.findElement(By.name("coffee"));
Select s=new Select(w);
s.selectByVisibleText("With cream & sugar");
```

54.If I click the button one alert is open how to switch over alert, then need to click OK , cancel and getText

```
driver.findElement(By.xpath("//input[@type='submit']")).click();
Alert a = driver.switchTo().alert();
System.out.println(a.getText());
```

```

a.accept();
a.dismiss();
driver.switchTo().defaultContent();
t();

```

55. Do you know synchronization concepts

Yes, In Selenium we have implicit Wait and Explicit Wait conditional statements

56. How to run one group of tests

- In TestNG we can execute one set of groups
- In test method we should mention like this `Test(groups={"facebook"})`
- Groups are specified in `testng.xml` file .

57. What is Data Provider

- parameters can be passed using Dataproviders.
- A Data Provider is a method annotated with `@DataProvider`.
- A Data Provider returns an array of objects.
- Ex: `@Test(dataProvider="getData")`

58. I have click the button then I write it for switch to alert but alert is not present how do u handle

- We can handle it by using try catch block
- Then we can wait until alert is present by using `alertIsPresent()`, then we can switch to alert

Ex:

```

Public void checkAlert(){
try{

    WebDriverWait wait = new WebDriverWait(driver,
2);
    wait.until(ExpectedConditions.alertIsPresent());
    Alert alert =
driver.switchTo().alert();
    alert.accept();
} catch (Exception
e) { //exception
handling
}
}

```

59. Mouse Over action code

Sample Code:

```

WebElement web = driver.findElement(By.xpath("//*[text()='Product Category']"));
Actions a=new Actions(driver);
a.moveToElement(web).perform();

```

60. If 4 numbers of window named with w1 as parent window and 3 child window are w2,w3,w4 are there. how will you switch from parent window to directly w4 child window

without using index?

```
public class NewUserActivationlv {  
    public static void main(String[] args) {  
  
        System.setProperty("webdriver.chrome.driver", "C:/Users/Desktop/Selenium/Driver/chromedriver.exe");  
  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.example.in/RetailLogin.html");  
        driver.manage().window().maximize();  
  
        String PID = driver.getWindowHandle();  
        System.out.println(PID);  
  
        driver.findElement(By.linkText("New User Login")).click();  
  
        Set<String> s1 =  
        driver.getWindowHandles(); List<String>  
        s = new ArrayList<String>();  
        s.addAll(s1);  
        System.out.println(s);  
        driver.switchTo().window(s.get(  
        4));  
    }  
}
```

61. What test case can be automated? in regression testing how will you choose which test cases need to be automated?

What are the criteria to automate test case?

Regression test case and smoke test cases can be automated.

Criteria for automate test case:

1. Repetitive tests that run for multiple builds.
2. Tests that tend to cause human error.
3. Tests that require multiple data sets.
4. Frequently used functionality that introduces high risk conditions.
5. Tests that are impossible to perform manually.
6. Tests that run on several different hardware or software platforms and configurations.
7. Tests that take a lot of effort and time when manual testing.

choose test cases for Automated regression testing:

1. Include the test cases which have frequent defects
2. Include the test cases which verify core features of the product
3. Include all Complex Test Cases

62. id = "1234_loginto_76201" write xpath for this?

Ans: `//*[@id='loginto']`

63. What are the frameworks u have worked

- POM Design Pattern
- BDD(Cucumber)
- DataDriven
- Junit
- TestNG

64. What are all the tools you have worked

- Maven – Build tool
- Jenkins – Continuous Integration tool
- JIRA,TFS – Defect Management tool

65. What is Statement, prepared statement and callable statement

Statement	PreparedStatement	CallableStatement
It is used to execute normal SQL queries.	It is used to execute parameterized or dynamic SQL queries.	It is used to call the stored procedures.
It is preferred when a particular SQL query is to be executed only once.	It is preferred when a particular query is to be executed multiple times.	It is preferred when the stored procedures are to be executed.
You cannot pass the parameters to SQL query using this interface.	You can pass the parameters to SQL query at run time using this interface.	You can pass 3 types of parameters using this interface. They are – IN, OUT and IN OUT.
This interface is mainly used for DDL statements like CREATE, ALTER, DROP etc.	It is used for any kind of SQL queries which are to be executed multiple times.	It is used to execute stored procedures and functions.
The performance of this interface is very low.	The performance of this interface is better than the Statement interface (when used for multiple execution of same query).	The performance of this interface is high.

Statement: Example

//Creating The Statement Object

```
Statement stmt = con.createStatement();
```

//Executing The Statement

```
stmt.executeUpdate("CREATE TABLE STUDENT(ID NUMBER NOT NULL, NAME VARCHAR)");
```

Preared Statement: Example

```
//Creating PreparedStatement object
PreparedStatement pstmt = con.prepareStatement("update STUDENT set
NAME = ? where ID = ?");
//Setting values to place holders using setter methods of PreparedStatement
object
```

```
pstmt.setString(1, "MyName"); //Assigns "MyName" to first place
holder
pstmt.setInt(2, 111); //Assigns "111" to second place holder
//Executing PreparedStatement
pstmt.executeUpdate();
```

Callable Example:

```
//Creating CallableStatement object
CallableStatement cstmt = con.prepareCall("{ call anyProcedure(?, ?, ?)}");
//Use cstmt.setter() methods to pass IN parameters
//Use cstmt.registerOutParameter() method to register OUT parameters
//Executing the CallableStatement
cstmt.execute();
//Use cstmt.getter() methods to retrieve the result returned by the stored procedure
```

66. What is Stale state element exception

- The element is no longer attached to the DOM.
- If webpage get refreshed at the middle of any operation with that particular element, after refresh element reference get refreshed.

67. Write a code to launch chrome browser

```
WebDriver driver=new ChromeDriver();
WebDriver is an interface
```

FrameWork Questions and Answers:

1. Explain Cucumber Framework

- a. Cucumber is a BDD(Behavior Driven Development) tool
- b. Cucumber has three things:
 - i. Feature File
 - ii. Step Definition and
 - iii. Test Runner
- c. We will use Gherkin language in feature file to write user stories in more efficient way using Gherkin keywords like
 - i. Scenario
 - ii. Given
 - iii. When
 - iv. Then
 - v. And
 - vi. But
- d. Step Definition file contains the actual code to execute.
- v. In Test Runner class, we will use JUnit annotations like @RunWith, @CucumberOptions

@ CucumberOptions mention the feature file in feature and step Definition in glue.

24. J unit Annotations and explain?

1. **@Test:** Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
2. **@Before:** Method annotated as @Before lets the system know that this method shall be executed every time before each of the test method.
3. **@After:** Method annotated as @After lets the system know that this method shall be executed every time after each of the test method.
4. **@BeforeClass:** Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test method.
5. **@AfterClass:** Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test method.
6. **@Ignore:** Method annotated as @Ignore lets the system know that this method shall not be executed.

3.Briefly explain pom?

POM:

- Page Object Model
- POM is an object repository design pattern in selenium webdriver
- POM creates our testing code maintainable and reusable
- Page factory is an optimized way to create object repository
- In POM have to create separate class for every web page

POM Steps & Rules:

- ☐ Create a maven project/java project
- ☐ We have to create 3 source folders(packages)
 1. Src/main/java
- ☐ It contains POM information (i.e) locators of every page in separate class
- ☐ 2. Src/test/java (JUnit,TestNG)
- ☐ It contains login, asserts & registration
- ☐ 3. src/resources/java
- ☒ It contains reusable methods/codes. Ex, browser coding, radio button, scroll down and etc.

4.What is framework, why we go for framework

- ☐ Framework is a code structure that helps to make code maintenance easy.
- ☐ Without frameworks, we will place the “code” as well as “data” in the same place which is neither re-usable nor readable.
- ☐ Using Frameworks, produce beneficial outcomes like increased code re-usage, higher portability, reduced script maintenance cost, higher code readability, etc.

5.What is the main method of junit

- ☐ No need to write main method in junit, because you can always call a JUnit runner to run a test case class as a system command.
- ☐ But If you want to know how to call a JUnit runner in a main() method,

Sample code:

```
public static void main(String[] args) {  
    junit.textui.TestRunner.run(DirListerTest.class);  
}
```

6.Explain about TestNG?

TestNG

- ☐ TestNG is a framework, it is an advance of JUnit
- ☐ TestNG (NG - New Generation), we can overcome all the disadvantages of JUnit by using TestNG
- ☐ TestNG has various class, interface and methods

Advantages

- ☐ Order is possible, i.e. we can prioritize tests by using priority attribute
- ☐ It will give default reports like html, xml format
- ☐ Cross browser testing and Parallel browser execution is possible
- ☐ Less configuration
- ☐ We can control execution by using annotations

7.Parallel test execution on TestNG.?

- ☐ Browsers will open almost simultaneously and your test will run in parallel.
- ☐ Now just set the ‘parallel’ attribute to ‘tests’ in the xml

8.Difference between @BeforeMethod, @BeforeClass and @BeforeTest

@BeforeMethod: The annotated method will be run before each test method.

@BeforeClass: The annotated method will be run before the first test method in the current class is invoked.

@BeforeTest: The annotated method will be run before any test method belonging to the classes inside the <test> tag is run.

9. Difference between JUnit and TestNG

- Configuration of parameterized test is very easy in TestNG where as it is a tough in JUnit.
- Group test is supported by TestNG and JUnit does not support it.
- Dependency test configuration is not possible in JUnit but it is possible in TestNG
- @Before test, @After test, @Before suit, @After suit, @before groups, @After groups are supported by TestNG and not by JUnit.
- TestNG supports test prioritization and parallel tests but Junit does not support it.

10. Where is the test report saved in testng

- You have to first run the tests by running the `testng.xml` as TestNG suite.
- Then after that refresh the project.
- Now goto the **test-output folder** -->html-->index.html (Open with Web Browser)

11. Difference between keyword driven and data driven

- Keyword driven is also called action driven. It is a framework which tester do not need to write java code to implement a test cases in selenium automation.
- Testers can use keyword driven framework to create many test cases that need to be tested automatically.
- Data driven is just used to run one automation test script with huge amount of testing data.
- Segregation of code from data input where code stays the same and data input keeps changing. Primarily useful in regression kind of scenarios.
- In practice, they are combined together to make a hybrid testing framework. So you can benefit both advantages.

12. How will you execute parallel browsing execution, where it is mentioned

Browsers will open almost simultaneously and your test will run in parallel.

- To execute parallel browsing we should set the '**parallel**' attribute to '**tests**' in the xml
- It is mentioned in testing.xml

13. What is groups

- Using TestNG we can execute one set of groups and excluding another set.
- This gives us the maximum flexibility in divide tests and doesn't require us to recompile anything if you want to run two different sets of tests back to back.
- Groups are specified in testng.xml file and can be used under the tag.

14. Difference between before and after hooks

- Before hooks will be run before the first step of each scenario. They will run in the same order of which they are registered.

- After hooks will be run after the last step of each scenario, even when there are failing, undefined, pending or skipped steps. They will run in the *opposite* order of which they are registered.

15. Difference between Scenario and Scenario Outline

Scenario:

- Copying and pasting scenarios to use different values quickly becomes tedious and repetitive:

```
Scenario: eat 5 out of 12
  Given there are 12 cucumbers
  When I eat 5 cucumbers
  Then I should have 7 cucumbers
```

```
Scenario: eat 5 out of 20
  Given there are 20 cucumbers
  When I eat 5 cucumbers
  Then I should have 15 cucumbers
```

- **Scenario outlines** allow us to more concisely express these examples through the use of a template with placeholders, using Scenario Outline, Examples with tables and < > delimited parameters:

```
Scenario Outline: eating
  Given there are <"start"> cucumbers
  When I eat <"eat"> cucumbers
  Then I should have <"left"> cucumbers
```

Examples:

start	eat	left
12	5	7
20	5	15

The Scenario Outline steps provide a template which is never directly run. A Scenario Outline is run once for each row in the Exam

16. What is background

It's a keyword used in featured class for declare the common step as once.

Eg: "Given" is common for all scenarios ...so we declare under the background keyword.

Tools:

1. What is the use of maven?

1. Project build tool
2. If one person (D) download jar file, it will automatically updated in maven repository. Then anyone can use
3. Maven is used to define project structure, dependencies, build, and test management.
4. Using pom.xml(Maven) you can configure dependencies needed for building testing and running code.

2. Explain about Jenkins?

Jenkins is an leading open source [continuous integration](#) server built with Java. It is used to build and test software projects continuously making it easier to integrate changes to the project. Support for scheduled builds & automation test execution.