**Interview Questions**

1. **CA Technologies**

1. Which is the parent of Web Driver Wait ?

Wait interface

2. Eliminate duplicates from an array

A[] = {15,2,8,5,3,1,2,8}

**public** **class** RemoveDuplicatesFromArray {

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

**int** a[] = {15,2,8,2,5,3,1,8};

HashSet<Integer> hs = **new** HashSet<Integer>();

**for**(**int** i=0;i<a.length;i++){

hs.add(a[i]);

}

System.*out*.println(" "+hs);

}

}

3.

Public class Employee

{

String name;

int age;

long salary;

}

Employee e = new employee();

e.add(“Punith”,22,2000);

e.add(“Basu”,16,3000);

e.add(“Manju”,18,2000);

Sort an array based on name.

4. Wap to convert each third character to upper case

“Hi hello how are you”

5.What is the output of the following

Int i=3;

switch(i)

{

case 0: S.o.p (“hi”);

case 1: S.o.p (“hello”);

case 2: S.o.p (“how”);

case 3: S.o.p (“are”);

case 4: S.o.p (“you”);

default: S.o.p (“test”);

}

5. Query to find the last but one record from a table.

select \* from Trn\_ATN where ATN\_No=(select max(ATN\_No)-1 from Trn\_ATN)

6. Basic Unix commands

7.Table contains age,name,address,phone no and state

Wap to read data from excel file using data provider and pass data as argument to @Test method for the page using data provider in POM

8. Close all opened windows except current window.

String originalHandle = driver.getWindowHandle();

//Do something to open new tabs

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

if (!handle.equals(originalHandle)) {

driver.switchTo().window(handle);

driver.close();

}

}

driver.switchTo().window(originalHandle);

9.How signature passed in rest api ?

10.What are all different methods in rest api and why we used?

**2.JC Penny Telephonic**

**1. What is difference between build management tool and test management tool**

**Build Management Tool**

Build tools are programs that automate the creation of executable applications from source code(eg. .apk for android app). Building incorporates compiling,linking and packaging the code into a usable or executable form.

Basically build automation is the act of scripting or automating a wide variety of tasks that software developers do in their day-to-day activities like:

Downloading dependencies.

Compiling source code into binary code.

Packaging that binary code.

Running tests.

Deployment to production systems.

Why do we use build tools or build automation?

In small projects, developers will often manually invoke the build process. This is not practical for larger projects, where it is very hard to keep track of what needs to be built, in what sequence and what dependencies there are in the building process. Using an automation tool allows the build process to be more consistent.

Various build tools available(Naming only few):

For java - Ant,Maven,Gradle.

For .NET framework - NAnt

c# - MsBuild.

**Test management tools**

Test management tools are used to store information on how testing is to be done, plan testing activities and report the status of quality assurance activities. The tools have different approaches to testing and thus have different sets of features.

Tools – TestRail, Test Complete etc

**2. What is difference between version control and source control**

Version control systems are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

3. What is difference between webdriver and firefox driver

WebDriver is an interface and all the methods which are declared in Webdriver interface are implemented by respective driver class. But if we do upcasting,we can run the scripts in any browser .i.e running the same automation scripts in different browsers to achieve [Runtime Polymorphism](https://www.softwaretestingmaterial.com/polymorphism-in-java/).

General information: Selenium WebDriver is an Interface which contains different methods (eg., *get(), getTitle(), close() etc.,*). All the third party Browser vendors implement these methods in addition to their browser specific methods. This would in-turn help the end-users to use the exposed APIs to write a common code and implement the functionalities across all the available Browsers without any change. Selenium developers don’t know how all these browsers work. So Selenium developers just declare methods whatever they required and leave the implementation part to the browser developers.

The FirefoxDriver class (and the ChromeDriver, and the OperaDriver, and the...) all attempt to adhere to the contract laid out by the WebDriver and WebElement interfaces. From a programmer's point of view they should be functionally identical. This being the real world, sometimes they're not. This can be because sites behave differently depending on which browser is accessing them (for example, think of the sites that serve H264 video to mobile clients, but flash to desktop browsers), but can be because of implementation differences between the drivers.

**4. What is polymorphism**

polymorphism is the ability by which, we **can create functions or reference variables which behaves differently in different programmatic context**

**Polymorphism in Java** is a concept by which we can perform a *single action in different ways*. Polymorphism is derived from 2 Greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.

There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism. We can perform polymorphism in java by method overloading and method overriding.

**3.Optum**

1. Wap to find duplicate characters

2. Write a script for Login and Home Page using POM

3. Use of Auto It and write a code to upload a file

4. Questions on String manipulation

5. Advantages of POM

Page Object Model is a design pattern to create **Object Repository** for web UI elements. Under this model, for each web page in the application, there should be corresponding page class. This Page class will find the WebElements of that web page and also contains Page methods which perform operations on those WebElements.

**Advantages of POM**

1- avoid to write the duplicate locators for same WebElement which is the big issue in other frameworks.  
2- Maintenance of the test script which becomes very easy.  
3- improves readability.

**4.Mphasis**

Most of the questions on Cucumber BDD

1.What is Cucumber ?

2.Difference between BDD and TDD?

3.How do you run multiple test data?

4.What are the challenges you have faced?

5.What are hooks in cucumber?

**5. CGI**

1. Explain the Framework architecture

**2.Why do we use BeforeTest in TestNG ?**

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

You have 1 Testclass, with 3 Testmethods:@BeforeClass will be executed once, @BeforeTest will be executed 3 times. thats the difference. @BeforeClass can be used to set up the test environment,@BeforeTest can be used to clean data or setup data

3.How do you connect to Database ?

1. Load and Registering the Driver  
2. Establishing Connection.  
3. Creating Statement Object  
4. Execute the Statement  
5. Closing the connection.

#### 1. Load and Register the Driver:

For registering the Driver we Load the Driver class using forName() method.

forName() is the static factory method which is present in predefined class called "Class". This method loads the class which is mentioned as parameter.

Class.forName("com.mysql.jdbc.Driver");// **class**.**forName** **load** **the** **Driver** **class**

Internally this Driver class will register the driver by using static method called registerDriver().

#### 2. Establishing Connection:

For establishing connection with database we call static method called getConnection(...) present in DriverManager Class. This method contains three arguments of string type. i.e., url, username and password

DriverManager.getConnection("jdbc:mysql://localhost:3306/Employee","root","root");

URL contains "jdbc(main protocol):mysql(sub protocol for mySql)://localhost:3306(sub name for mysql (host:prot))/Employee(database)" and this method return type is Connection Object ie.,

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Employee","root","root");

#### 3. Creating Statement Object:

For creating statement object we need to call a method called createStatement() which is present in Connection Interface.

con.createStatement();

And this method returns Statement object and it is no argument method.

Statement st= con.createStatement();

#### 4.Executing Queries:

For executing queries there are different methods present in Statement Interface for retrieving records and for updating records.

**Retrieving records:**  
for executing select queries(for fetching records) we call a method called executeQuery(String qry) by taking string as parameter.

st.executeQuery("**Select** \* **from** Employee");

This method returns ResultSet object.

Resultset rs= st.executeQuery("Select \* from Employee");// once executeQuery() executes the query **and** stores the records **in** to ResultSet object.

#### 5. Closing the Connection:

Once execution of all statements were completed we need to close all the connections by using method called close() present in Connection interface

con.**close**();

**4.Difference between Synchronous and Asynchronous ?**

Synchronous means things are getting done in sequential order. This means if one process is going on, other one won’t start unless already running process is finished.

Asynchronous means things won’t wait for process to finish and can start running new process or proceed with further execution.

5. Bug life cycle

6.

**6. Synechron Technologies**

**Written and face to face questions**

1. Wap to add zeroes at the end and display as same order

Input – {1,4,5,0,6,0,8,0,3,0}

Output – {1,4,5,6,8,3,0,0,0,0}

**2.What is load factor and default size in HashMap ?**

The amount of capacity which is to be exhausted for the HashMap to increase its capacity.

load factor is a measure **"Till what load, hashmap can allow elements to put in it before its capacity is automatically increased".**

**The decision of "When to increase the number of buckets" is decided by Load Factor.   
  
Load Factor is a measure which decides when exactly to increase the hashmap capacity or you can say bucket capacity, so that get and put operation can still have O(1) complexity.**   
  
**Default, initial capacity of the HashMap is 16 and Load factor is 0.75**

**So, when to increase the hashmap size is decided by product of,   
(initial capacity of hashmap \* Load factor of hashmap).**

**Lets see, when initial size of hashmap will be increased based on above forumla,  
initial capacity of hashmap \* Load factor of hashmap =  16 \* 0.75 = 12.**  
 **This represents that uptil 12th key-value pair hashmap will keep its size to 16 and as soon as 13th item(key-value pair) will come into the Hashmap,  it will increase its size from default 2^4 = 16 buckets to 2^5 = 32 buckets.**

Load factor is by default 0.75 of the initial capacity (16) therefore 25% of the buckets will be free before there is an increase in the capacity & this makes many new buckets with new hashcodes pointing to them to exist just after the increase in the number of buckets.

3.What is difference between LinkedList and ArrayList ?

4.ArrayList {“121 Vijay 2000, 122 Ajay 5000, 123 Sujay 8000”}

Sort an array list based on name in ascending order

5.WAP to sort an array using an array list {6,5,3,8,0}

6.What is an interface ?

7.When do we use interface and abstraction ?

8.Suppose any new requirement come in future ,will you add requirement in interface or abstraction and if you add in interface ,will it affect to already implemented sub classes ?

9.WAP to sort an array {6,5,3,8,0}

10.Draw an Exception architecture

11.What is Stack and Heap Memory

12.Can we implement interface if interface contains default method ?

Is below program correct ?

Interface TestA{

default public void Testing()

{

System.out.println(“Hi”);

}

Interface TestB{

default public void TestingMethod()

{

System.out.println(“Hello”);

}

Class TestC implements TestA,TestB

{

default public void Testing()

{

System.out.println(“Hi”);

}

default public void TestingMethod()

{

System.out.println(“Hello”);

}

}

13.What is WebDriver ?

14.What is FirefoxDriver ?

15.WebDriver driver = new FirefoxDriver();

Which oops concept and why ?

16. WebDriver driver = new WebDriver();

What above line states ?

17.what is the output of the below code ?

Class Test {

public void testMethod{

Try{

System.out.printf(1);

int I = 5/0;

}catch(Exception e){

Throwable th = new Throwable(“Simple”);

}}

Try{

throw b;

}

Finally()

{

System.out.println(2);

}

}

18.Can we add try without catch ,if yes then where will catch exception ?

19.What is the difference between throw and throws ?

20.What is BDD ?

21.What is Scenerio and Scenerio Outline ?

22.What is the difference Scenerio and Scenerio Outline ?

23.What is Data Table in Cucumber ?

24.What is Background in Cucumber?

25.What is Gherkins ?

26.What is Given, When and Then ?

**7. Global Logic**

1. What are different exceptions in Selenium ?

2. How many ways we can handle frames?

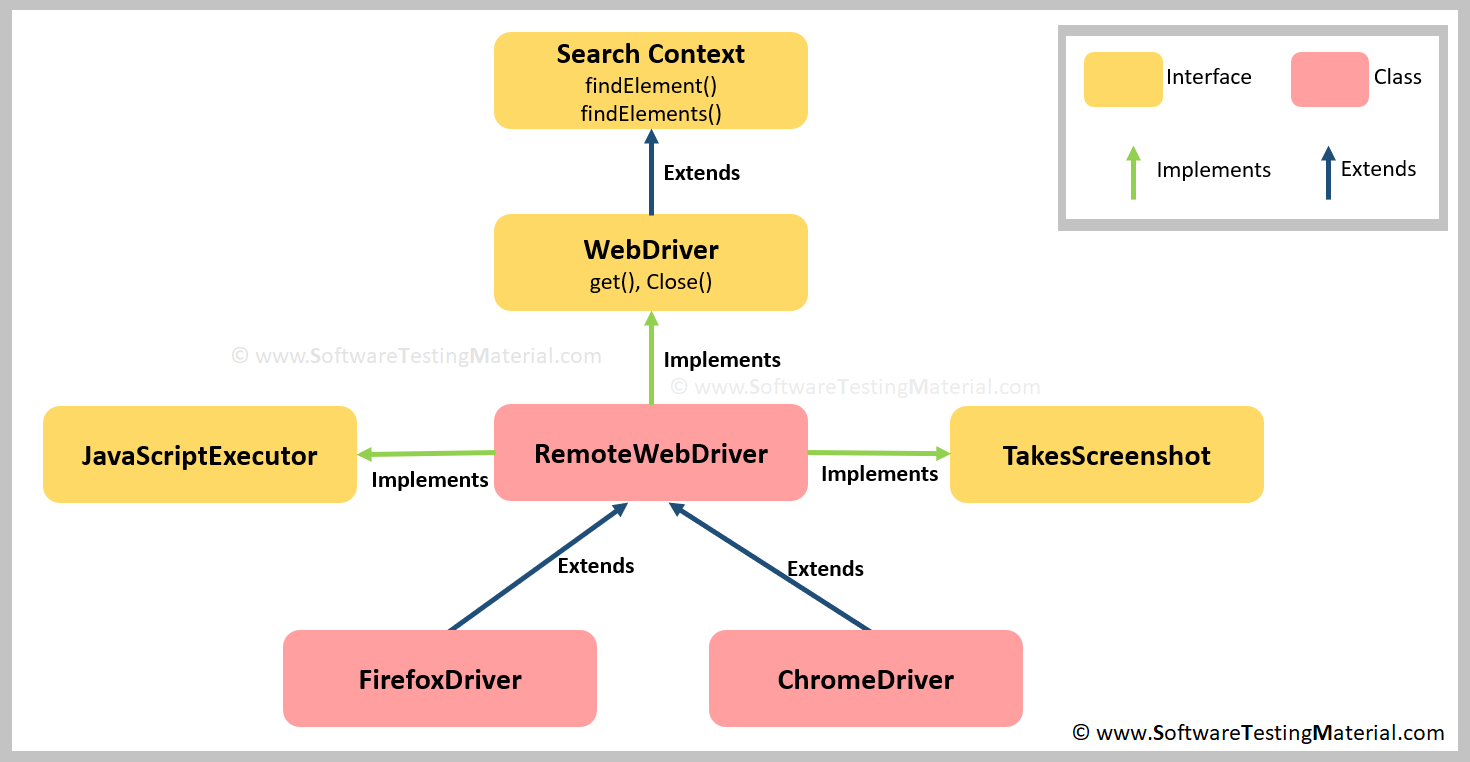
3. How many ways we can select drop down values?

**4. WebDriver driver = new FirefoxDriver() – Why we write in Selenium Scripts**

1. Why not WebDriver driver = new WebDriver()  
2. Why we wont prefer FirefoxDriver driver = new FirefoxDriver()  
3. Why WebDriver driver = new FirefoxDriver()  
4. What is RemoteWebDriver and where we use it

[Java Tutorial for Selenium Testers](https://www.softwaretestingmaterial.com/java-tutorial/)

Before going into the details lets see the below image.



Following are the few points based on the above image.

* SearchContext is the super most interface in selenium, which is extended by another interface called WebDriver.
* All the abstract methods of SearchContext and WebDriver interfaces are implemented in RemoteWebDriver class.
* All the browser related classes such as FirefoxDriver, ChromeDriver etc., extends the RemoteWebdriver class.

[Must learn: Selenium WebDriver Architecture](https://www.softwaretestingmaterial.com/selenium-webdriver-architecture/)

WebDriver defines common methods which all browser classes (such as Firefox, Chrome etc.,) use. All these class methods are derived from WebDriver interface.

All the [abstract](https://www.softwaretestingmaterial.com/abstraction-in-java/) methods of both the [interfaces](https://www.softwaretestingmaterial.com/interface-in-java/) are implemented in RemoteWebDriver class which is extended by browser classes such as Firefox Driver, Chrome Driver etc.

Let’s see why can’t we use the following statement.

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

We cannot write our code like this because we cannot create Object of an Interface. WebDriver is an interface.

But we can use any of the following statements in our script

*FirefoxDriver driver = new FirefoxDriver();*

or

*WebDriver driver = new FirefoxDriver();*

Let’s see both of them in detail.

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

The FirefoxDriver instance which gets created based on above statement will be only able to invoke and act on the methods implemented by FirefoxDriver and supported by Firefox Browser only. We know that FirefoxDriver is a class and it implements all the methods of WebDriver interface. Using this statement, we can run our scripts only on Firefox Browser.

To act with other browsers we have to specifically create individual objects as below:

*ChromeDriver driver = new ChromeDriver();*  
*InternetExplorerDriver driver = new InternetExplorerDriver();*

We don’t just run our scripts only on single browser. We use multiple browsers for [Cross Browser Compatibility](https://www.softwaretestingmaterial.com/cross-browser-testing/).  We need the flexibility to use other browsers like ChromeDriver() to run on Chrome Browser and InternetExplorerDriver() to run on IE Browser and so on.

So, once you initiate a Firefox browser using *FirefoxDriver driver = new FirefoxDriver();*same object cannot be used to initiate Chrome Browser (you have to rename it)

*ChromeDriver driver = new ChromeDriver();*

To solve this we use “*Webdriver driver = new FirefoxDriver();*”

Lets see this now.

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

We can create Object of a class FirefoxDriver by taking reference of an interface (WebDriver). In this case, we can call implemented methods of WebDriver interface.

As per the above statement, we are creating an instance of the WebDriver interface and casting it to FirefoxDriver Class. All other Browser Drivers like ChromeDriver, InternetExplorerDriver, PhantomJSDriver, SafariDriver etc implemented the WebDriver interface (actually the RemoteWebDriver class implements WebDriver Interface and the Browser Drivers extends RemoteWebDriver). Based on this statement, you can assign Firefox driver and run the script in Firefox browser (any browser depends on your choice).

We will see RemoteWebDriver in later section below.

If you define driver as a WebDriver, switching will be very easy. If we use this statement in our script then the WebDriver driver can implement any browser. Every browser driver class implements WebDriver interface and we can get all the methods. It helps you when you do testing on multiple browsers.

Example:

*WebDriver driver = new FirefoxDriver();*  
*driver.quit();*  
*driver = new ChromeDriver();*

WebDriver is an interface and all the methods which are declared in Webdriver interface are implemented by respective driver class. But if we do upcasting,we can run the scripts in any browser .i.e running the same automation scripts in different browsers to achieve [Runtime Polymorphism](https://www.softwaretestingmaterial.com/polymorphism-in-java/).

*WebDriver driver = new FirefoxDriver();*

Here, *WebDriver* is an interface, *driver* is a reference variable, *FirefoxDriver()* is a [Constructor](https://www.softwaretestingmaterial.com/java-tutorial/#constructor), *new*is a keyword, and *new FirefoxDriver()*is an Object.

General information: Selenium WebDriver is an Interface which contains different methods (eg., *get(), getTitle(), close() etc.,*). All the third party Browser vendors implement these methods in addition to their browser specific methods. This would in-turn help the end-users to use the exposed APIs to write a common code and implement the functionalities across all the available Browsers without any change. Selenium developers don’t know how all these browsers work. So Selenium developers just declare methods whatever they required and leave the implementation part to the browser developers.

**5.Difference between WebDriver and RemoteWebDriver:**

As per Java’s interface concept, Interface contains only Method’s signature and it doesn’t contain the Method’s definitions.

WebDriver is the Interface which contains all the Selenium Method signatures (Eg findElement(),switchTo(),get() etc.) where as definition for those methods are in RemoteWedDriver Class.

If we want to run our automation scripts on the local machine’s browser then we can use any class (such as Firefoxdriver, iedriver, chromedriver, htmlunitdriver) except RemoteWebDriver. WebDriver will start up a web browser on the computer where the code instantiates it.

If we want to run our automation scripts on the remote machine’s browser then we use RemoteWebDriver. RemoteWebDriver requires the Selenium Standalone Server to be running but to use other drivers Selenium Standalone Server is not required.

If you want to work with Grid then you have to stick with RemoteWebDriver. The only requirement is that for a RemoteWebDriver to work, you would always have to have it pointing to the URL of a Grid.

If you are using any of the drivers other than RemoteWebDriver then the communication will happen on the local machine’s browser.

Ex: *Webdriver driver = new FirefoxDriver();*

driver will access Firefox on the local machine, directly.

If we use RemoteWebDriver then we have to mention where the Selenium Server is located and which web browser you want to use.

Ex: *WebDriver driver = new RemoteWebDriver(new URL Desired capabilities.firefox());*

We can use RemoteWebDriver the same way we would use WebDriver locally. The primary difference is that remote webdriver needs to be configured so that it can run your tests on a remote machine.

WebDriver is an interface that you should be using throughout your tests. RemoteWebDriver is a concrete implementation of that interface. In general, its always a good idea to code against interfaces wherever possible.

6.How many Design Patterns are there and which are those ?

7.Convert values from Set to List

8. Explain Database connection steps

# 9. Where you have applied OOPS in Automation Framework

### ****ABSTRACTION****

In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,) in a Page Class. We utilize these locators in tests but we can’t see these locators in the tests. Literally we hide the locators from the tests.

Abstraction is the methodology of hiding the implementation of internal details and showing the functionality to the users.

### ****INTERFACE****

Basic statement we all know in Selenium is WebDriver driver = new FirefoxDriver();

WebDriver itself is an Interface. So based on the above statement WebDriver driver = new FirefoxDriver(); we are initializing Firefox browser using Selenium WebDriver. It means we are creating a reference variable (driver) of the interface (WebDriver) and creating an Object. Here WebDriver is an Interface as mentioned earlier and FirefoxDriver is a class.

An interface in Java looks similar to a class but both the interface and class are two different concepts. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. We can achieve 100% abstraction and multiple inheritance in Java with Interface.

### ****INHERITANCE****

We create a Base Class in the Framework to initialize WebDriver interface, WebDriver waits, Property files, Excels, etc., in the Base Class.

We extend the Base Class in other classes such as Tests and Utility Class. Extending one class into other class is known as Inheritance.

### ****POLYMORPHISM****

Combination of overloading and overriding is known as Polymorphism. We will see both overloading and overriding below.

Polymorphism allows us to perform a task in multiple ways.

**METHOD OVERLOADING**

We use implicit wait in Selenium. Implicit wait is an example of overloading. In Implicit wait we use different time stamps such as SECONDS, MINUTES, HOURS etc.,

A class having multiple methods with same name but different parameters is called Method Overloading

### ****METHOD OVERRIDING****

We use a method which was already implemented in another class by changing its parameters. To understand this you need to understand Overriding in Java.

Declaring a method in child class which is already present in the parent class is called Method Overriding. Examples are get and navigate methods of different drivers in Selenium .

### ****ENCAPSULATION****

All the classes in a framework are an example of Encapsulation. In POM classes, we declare the data members using @FindBy and initialization of data members will be done using Constructor to utilize those in methods.

Encapsulation is a mechanism of binding code and data together in a single unit.

### ****WEB ELEMENT:****

Web element is an interface used to identify the elements in a web page.

### ****WEBDRIVER:****

WebDriver is an interface used to launch different browsers such as Firefox, Chrome, Internet Explorer, Safari etc.,

### ****FIND BY:****

FindBy is an annotation used in Page Object Model design pattern to identify the elements.

### ****FIND ELEMENT:****

Find Element is a method in POM to identify the elements in a web page.

**8. Aricent**

1. How are you maintaining Object Repository ?

2.Write Mysql Database connection steps to get records.

3.Write a program to remove duplicates from file.

4.How to handle windows pop ups ?

5.Do you have experience on DevOps?

**9.Infinite Computer Solutions**

1. Difference between Abstraction and Interface

2. Difference between throw and throws

3.What is the output for below code

Class Test {

Public void TestMethod(){

Try{

Code;

Throw new IOException();

}catch(Exception e){

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

} catch(IOException e){

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

}

} }

4. How will you execute Feature files ?

5. How will you execute different tags (Sanity, Regression) ?

6.Wap to display values from any Map ?

7.Difference between StringBuffer and StringBuilder

8.How do you add estimation for the User Stories (Manual and Automation)?

9.How do you add estimation for testing the Defect ?

10.If many defects are assigned to you for testing at the end of the sprint and have the high priority then when will you test these defects ?

11.What would you do as Test Engineer (Daily activities QA process)

**10.Cognizant**

1.What is TestNG ?

2.What is Maven?

3.Why should we use Maven ?

4.If I add Selenium Stand alone server 2.5 dependency in pom.xml

Will it update latest version or update only given version ?

5.Can I run TestNG method without @Test tag ?

6.Does below code execute ?

Class Test

{

                Public void launchApplication()

                {

                                WebDriver driver = new ChromeDriver();

                                Driver.get(“[www.google.com](http://www.google.com/)”);

                }

@Test

                Public void login()

                {

                                Login page code

                }

}

7.Does below code execute ?

Class Test {

                @BeforeTest

Public void launchApplication()

                {

                                WebDriver driver = new ChromeDriver();

                                Driver.get(“[www.google.com](http://www.google.com/)”);

                }

                Public void login()

                {

                                Login page code

                }

@AfterTest

Public void launchApplication()

                {

                                Driver.close();

                }

}

8.

- Open Google website

- Enter “Testing” in Google Search text field

   It will list different types of testing

- Get the size of listed links

- Click on “Mobile Testing” link

 It will open in new window

- Get the all the links and display

9.Check boxes dynamically added, how would you select last but one check box.

Note - Page might contain other check boxes.

div<input type=”check box”></div>

div<input type=”check box”></div>

div<input type=”check box”></div>

div<input type=”check box”></div>

10.How to switch back to Parent window ?

**11.Mindtree**

1.What is diff between FindBys and FindByAll ?

2.Write a Program to separate digits and characters from given string and add it in separate strings

String st = “ab24cdg45klr10”;

o/p – > st1 = “abcdgklr”;

st2 = “244510”;

3.How do you identify the broken links on the web page ?

4.How to move image from one place to another place ?

5.What does Data Table return in Step Definition class ?

6. Will step execute If I replace \* with Given in feature file ?

7. Write Step Definition methods for below Steps

Given: User is on Login page

When: User enters username and password

Then: Click on Login button

8.Whar are Cucumber components used in Feature file ?

9. Identify file extension(.xls or xlsx) and write a program to read specific cell value from excel file.

10.What are different locators available in Selenium ?

**12. Clarivate Analytics**

1.Program

i/p – 10010

o/p – 11000

2.Have you used Wrapper classes?

3.Have you used abstract class in your framework?

4.Why you used abstract class in your framework?

5.How do you use driver instance in Cucumber Step Definition class ?

6.Have you used POM in Cucumber ?

7.How does Cucumber initialize in Runner Class ?

**8.How do you do Parallel testing ?**

testNG.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="Suite" parallel="tests">

<test name="ChromeTest">

<parameter name="browser" value="chrome" />

<classes>

<class name="com.qa.test.ParallelTest" />

</classes>

</test>

<test name="FirefoxTest">

<parameter name="browser" value="firefox" />

<classes>

<class name="com.qa.test.ParallelTest" />

</classes>

</test>

<test name="IETest">

<parameter name="browser" value="ie" />

<classes>

<class name="com.qa.test.ParallelTest" />

</classes>

</test>

</suite>

package com.qa.test;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class ParallelTest {

WebDriver driver;

@Test

@Parameters("browser")

public void launchBrowser(String browser){

if(browser.equalsIgnoreCase("chrome")){

System.setProperty("webdriver.chrome.driver","D:/AutomationJars/chromedriver.exe");

driver = new ChromeDriver();

}else if(browser.equalsIgnoreCase("firefox")){

System.setProperty("webdriver.gecko.driver","D:/AutomationJars/geckodriver.exe");

driver = new FirefoxDriver();

}else if(browser.equalsIgnoreCase("ie")){

System.setProperty("webdriver.ie.driver","D:/AutomationJars/IEDriverServer.exe");

driver = new InternetExplorerDriver();

}

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

driver.manage().deleteAllCookies();

driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);

driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS);

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

System.out.println(driver.getTitle());

driver.close();

}

}

9.How do you initialize driver instance for different browsers like chrome, firefox and IE in parallel testing ?

10.What java library used for excel operations?

11.What does Data Provider declares in method?

12.How do you use Data Provider to retrieve data from excel?

13.What are different ways to initialize driver instance in Base Class?

14. What are different ways to initialize driver instance in Page Classes?

15.How do you get driver instance in @Test Class?

16.We have two testng xml files

Ex: testing.xml, testing\_smoke.xml

If you want execute specific xml through command prompt, how will you do ?