Selenium Automation Testing Interview Question 2025

1. Tell me about yourself and your experience. 👨💻🧑‍🎨

2. Abstraction vs Encapsulation 🤔💡

Explain the difference between **Abstraction** (hiding complex details and showing only the essential features) and **Encapsulation** (bundling data and methods into a single unit or class).

3. Different types of Asserts ✅🔍

What are the different types of asserts in Selenium, such as **assertEquals**, **assertTrue**, **assertFalse**, **assertNotNull**, and their use cases in automation testing?

4. Different types of Waits ⏳🕰️

Explain the different types of waits in Selenium, such as **Implicit Wait**, **Explicit Wait**, and **Fluent Wait**, and how each one is used to handle dynamic web elements in automation.

5. Different Exceptions ⚠️❌

What are some common exceptions you might encounter in Selenium, such as **NoSuchElementException**, **TimeoutException**, **StaleElementReferenceException**, and how do you handle them?

6. Code to traverse through multiple windows. 🌐🔄

Explain or write a code to switch between multiple browser windows using Selenium WebDriver.

Example Code:

```
// Store the window handle
String parentWindow = driver.getWindowHandle();

// Switch to child window
Set<String> allWindows = driver.getWindowHandles();
for (String windowHandle : allWindows) {
    if (!windowHandle.equals(parentWindow)) {
        driver.switchTo().window(windowHandle);
    }
}
```

7. Code to find whether the number is palindrome. 🔢🔄

Write a piece of code to check if a number is a palindrome (reads the same forward and backward).

Example Code:

```java
public class Palindrome {
    public static void main(String[] args) {
        int num = 121;
        int reversed = 0, remainder, original;
        original = num;

        while (num != 0) {
            remainder = num % 10;
            reversed = reversed * 10 + remainder;
            num /= 10;
        }

        if (original == reversed) {
            System.out.println(original + " is a palindrome.");
        } else {
            System.out.println(original + " is not a palindrome.");
        }
    }
}
```

8. findElement vs findElements 🔍📋

Explain the difference between **findElement** (returns a single element) and **findElements** (returns a list of elements) methods in Selenium, and when to use each one.

1. What are the different types of waits in Selenium, and when would you use each one? ⏳

This question tests your understanding of **Implicit Wait**, **Explicit Wait**, and **Fluent Wait** in handling dynamic web elements during automation.

2. How do you handle browser alerts in Selenium? ⚠️

Explain the process of handling **Alerts**, **Confirmations**, and **Prompts** in Selenium, and the methods like alert.accept(), alert.dismiss(), etc.

3. How do you identify and handle broken links in a website using Selenium? 🚫🔗

This question tests your knowledge of detecting and handling **broken links** in an automated test. Describe the approach for checking the HTTP status of links.

4. What's the difference between findElement and findElements in Selenium, and when would you use each? 🔍📋

Explain the distinction between these two methods in Selenium, focusing on how **findElement** returns a single element and **findElements** returns a list of elements.

5. Can you explain the Object-Oriented Programming (OOP) concepts you've applied in your Selenium automation projects? 💻💡

This question examines your understanding of **OOP concepts** such as **Encapsulation**, **Inheritance**, **Polymorphism**, and **Abstraction** in the context of automation testing.

6. How do you handle pagination in Selenium when testing a web application with multiple pages of data? 📄🔄

Describe how to navigate through **pagination** in a web application, handling dynamic page numbers and ensuring that you can test across multiple pages.

7. How do you implement parallel test execution in Selenium, and what are the benefits? ⚡🚀

Explain how to run tests in parallel using **TestNG** or **JUnit**, and discuss the advantages of parallel execution in speeding up the testing process.

8. Can you provide a code example for detecting and handling broken links in a web page using Selenium? 🔗💻

Provide a code snippet to detect and report **broken links** by checking their HTTP response status during automation.

9. How do you handle multiple windows or tabs in Selenium, and can you provide an example of switching between them? 🌐🔄

Describe how to manage and switch between multiple windows or tabs using **window handles** in Selenium WebDriver.

10. Can you explain the role of annotations in Selenium test frameworks, such as TestNG or JUnit? 📝⚙️

Discuss the different **annotations** like @Test, @BeforeMethod, @AfterMethod in **TestNG** or **JUnit**, and how they help in organizing and managing test execution.

1. How do you approach debugging, scripting, designing, and execution?

2. Can you write Selenium code for handling alerts?

3. How would you handle windows in Selenium automation?

4. Write a Java program to reverse a string and calculate its length without using reverse

5. Explain the syntax for performing a mouseover action.

6. Are you familiar with database testing? If so, please explain.

7. Could you describe your experience with mobile testing and provide examples?

8. Write Selenium code to extract table and row values from a website.

9. What are the common exceptions encountered in Selenium?

10. What issues have you faced while running Selenium scripts in Java?

11. What is tooltip.

12. Can you explain the Git and Bitbucket ?

13. Explain the concept of XPaths and their importance in automation testing.

14. Which framework(s) are you currently using for test automation?

15. How would you upload a file using Selenium automation?

16. Write code to retrieve the color of a link using Selenium.

17. What is Maven, and how do you use it in test automation?

18. What is POM (Page Object Model), and how does it help in test automation?

19. Explain the POM structure

1.How to handle production issues in testing

2.what is your approach if developer says the issue which was raised is not a bug

3.What are the step you will consider for building automation framework from scratch,

4.What is approach when UI is not loading/not working

5.How will you check whether the build is old or not?

6.What is your approach to start the project if you don't have any single requirement docs or data?

7.What kind of testing will be done in Production?

8.How to do failure analysis in automation

9.Write a program to find longest repeating chatacter in a string

10.suppose u have mp.put("1","Rasika,Cognizant"); data in map how u will replace Cognizant by VISA and suppose mp.put("1","Cognizant ,Cognizant"); in this u need to replace 2nd Cognizant

11.Java snippet to find the Product of two concecutive elements in an array

12.what will you do if build is deployed in production is not working

13.In Sign page even after providing correct credentials it shows popup with some error msg what might be the reason but working fine in manual

14.When some persons are not available and we need to finish that task immediately, how did you handled

15.In cucumber I never want use login details again an again then what would be my approach

Top 50 basic Selenium questions:

Section 1: Selenium Basics (1-10)

1. What is Selenium?
2. What are the different types of Selenium tools?
3. What is Selenium WebDriver?
4. What is the difference between Selenium IDE and Selenium WebDriver?
5. How do you launch a browser using Selenium WebDriver?
6. What are the different types of browsers supported by Selenium?
7. How do you handle alerts and pop-ups in Selenium?
8. What is the use of Selenium Grid?
9. How do you run Selenium tests in parallel?
10. What are the advantages of using Selenium?

Section 2: WebDriver (11-20)

11. What is the difference between driver.get() and driver.navigate().to()?
12. How do you maximize a browser window using Selenium?
13. How do you handle cookies in Selenium?
14. What is the use of WebDriverManager?
15. How do you switch between multiple windows in Selenium?
16. How do you handle frames in Selenium?
17. What is the use of Actions class in Selenium?
18. How do you perform mouse hover actions in Selenium?
19. How do you perform drag-and-drop actions in Selenium?
20. What is the use of WebDriverWait class in Selenium?

Section 3: Locators (21-30)

21. What are the different types of locators in Selenium?
22. How do you use ID locator in Selenium?
23. How do you use name locator in Selenium?

24. How do you use XPath locator in Selenium?
25. How do you use CSS selector locator in Selenium?
26. How do you use linkText locator in Selenium?
27. How do you use partialLinkText locator in Selenium?
28. How do you use tagName locator in Selenium?
29. How do you use className locator in Selenium?
30. What is the difference between absolute and relative XPath?

Section 4: TestNG and Selenium (31-40)

31. What is TestNG?
32. How do you integrate TestNG with Selenium?
33. What are the different annotations in TestNG?
34. How do you use @BeforeMethod annotation in TestNG?
35. How do you use @AfterMethod annotation in TestNG?
36. How do you use @BeforeClass annotation in TestNG?
37. How do you use @AfterClass annotation in TestNG?
38. How do you use @Test annotation in TestNG?
39. How do you use @DataProvider annotation in TestNG?
40. What is the use of ITestResult interface in TestNG?

Section 5: Advanced Selenium (41-50)

41. What is the use of Page Object Model (POM) in Selenium?
42. How do you use POM in Selenium?
43. What is the use of Data-Driven Testing in Selenium?
44. How do you use Data-Driven Testing in Selenium?
45. What is the use of Hybrid Framework in Selenium?
46. How do you use Hybrid Framework in Selenium?
47. What is the use of Selenium Grid for parallel execution?
48. How do you use Selenium Grid for parallel execution?
49. What is the use of Docker for Selenium testing?
50. How do you use Docker for Selenium testing?

These questions cover the basics of Selenium, WebDriver, locators, TestNG, and advanced Selenium concepts.

🚀 ***Java Most Important Coding Interview Questions***

1. Reverse a String: Write a Java program to reverse a given string.

2. Find the Largest Element in an Array: Find and print the largest element in an array.

3. Check for Palindrome: Determine if a given string is a palindrome.

4. Factorial Calculation: Write a function to calculate the factorial of a number.

5. Fibonacci Series: Generate the first *n* numbers in the Fibonacci sequence.

6. Check for Prime Number: Write a program to check if a given number is prime.

7. String Anagrams: Determine if two strings are anagrams of each other.

8. Array Sorting: Implement sorting algorithms like bubble sort, merge sort, or quicksort.

9. Binary Search: Implement a binary search algorithm in a sorted array.

10. Duplicate Elements in an Array: Find and print duplicate elements in an array.

11. Linked List Reversal: Reverse a singly-linked list.

12. Matrix Operations: Perform operations like addition, multiplication, or transpose.

13. Implement a Stack: Create a stack data structure with basic operations (push, pop).

14. Implement a Queue: Create a queue data structure with basic operations (enqueue, dequeue).

15. Inheritance and Polymorphism: Implement class hierarchy and demonstrate polymorphism.

16. Exception Handling: Demonstrate the use of try-catch blocks to handle exceptions.

17. File I/O: Read from and write to a file using Java's file I/O capabilities.

18. Multithreading: Create a multithreaded program and demonstrate thread synchronization.

19. Lambda Expressions: Use lambda expressions to implement functional interfaces.

20. Recursive Algorithms: Solve problems using recursion, like computing factorial or Fibonacci sequence.

21. Swap two Numbers: Swap two numbers without using a third variable.

Automation Selenium Question:

1. What were your roles and responsibilities in your previous Selenium Automation projects? 👨‍💻

2. Can you explain the pom.xml file in your automation project? 📂⚙️

3. I see you have used the TestNG framework. Can you explain how you configured TestNG in your project? 🛠️📊

4. How do you write test scenarios for searching in a search box? Can you give an example? 🔍💡

Describe the process of creating **test scenarios** for verifying **search functionality**, including input validation, result verification, and edge case handling.

Example Test Scenario:

- **Test Case:** Verify search functionality on a search engine.
    - Step 1: Open the website (Google).
    - Step 2: Enter a search term (e.g., “Java”).
    - Step 3: Verify that the search results are displayed correctly.
    - Step 4: Validate the result page contains relevant links.

5. Can you explain the various annotations in TestNG? 📋📝

Describe the important **TestNG annotations** such as **@Test**, **@BeforeMethod**, **@AfterMethod**, **@BeforeClass**, **@AfterClass**, **@BeforeSuite**, and **@AfterSuite**, and explain when each of them is used in your automation framework.

6. Where do you write reusable components in your automation project? 🔄💻

Explain the structure of your **automation framework** and where you place reusable components (e.g., **utility classes**, **functions**, or **Page Objects**) for easier maintenance and better test scalability.

7. What are the different Selenium locators, and which one do you prefer in your projects? 🔍⚡

Discuss the different types of **Selenium locators** such as **ID**, **Name**, **Class Name**, **Tag Name**, **Link Text**, **Partial Link Text**, **CSS Selector**, and **XPath**, and explain when and why you would use each.

8. Write a Selenium test script to search for the "Java" keyword in Google search bar, count the number of links on the page, open each link in a new window, get the title of the opened webpage, and close the browser. 💻🔗

9. What is the difference between an Agile Retrospective and an Agile Review? 🔄✅

1st Round:

1. Tell me about yourself.
2. What are the various selenium components, and which components are you using?
3. How do you inspect elements?
4. What are the various locators you are using in your day-to-day life?
5. Write syntax for XPath?
6. Write a syntax to initialize the webdriver.
7. Explain absolute and relative XPath with an example
8. Explain your project and framework
9. How do you get the data from the Excel sheet?
10. Where do you use java in selenium webdriver?
11. What is oops?
12. Explain the abstraction concept with an example.

13. Explain the inheritance concept with an example.
14. Explain method overloading and Method overriding with an example.
15. What is an interface? Explain with an example.
16. Write a basic selenium script.
17. Explain the defect life cycle.
18. Write a syntax to select a value from the drop-down.
19. Write syntax to get the text from the table.
20. What are Hashmap and HashSet? Explain?
21. Where do you use Hashmap?
22. What is the exception you get in selenium?
23. What is implicitly wait, explicitly wait, webdriver wait, Thread.sleep
24. Write syntax for the webdriver wait.
25. The difference between implicitly, explicitly, webdriver wait and thread.sleep.
26. Which framework are you using in your project?
27. Are you using any integration tools?
28. What is Jenkins?
29. Where do you upload test results?
30. What are retesting and regression testing?
31. What is smoke testing?
32. How do you get to know if the developer has fixed the issues or not?
33. What are the roles and responsibilities?
34. Which java & selenium versions are you using?
35. How do you handle if XPath is changing dynamically?
36. Have you ever written test cases in your projects?
37. They asked Some selenium questions.

2nd round:

1. Tell me about yourself.

2. What is performance testing? (I have mentioned this in my resume)

3. For which functionality module have you performed performance testing in your project?

4. What is the response time?

5. Which language do you use for Appium testing? (I have mentioned this in my resume)

6. Did you write test scripts for mobile automation?

7. Can we use the same code in the Appium, the test script developed for a web app?

8. What is Jenkins?

9. What is the layout of Jenkins?

10. What is GIT & SVN?

11. Difference between GIT & SVN

12. Does Jenkins require a local system for CI?

13. They also asked some basic java questions

---

1. When to use an abstract class and when to use an interface? 🔄

Abstract Class:

- An **abstract class** is used when you want to provide common functionality to subclasses while leaving some methods to be implemented by the subclasses. It can have both **abstract methods** (methods without a body) and **concrete methods** (methods with a body).

- **Use abstract class** when:
    - You want to provide a common base class for related objects.
    - You need to share code among classes.
    - You want to define both abstract and concrete methods.
    - You need to maintain state via instance variables.

Interface:

- An **interface** is used when you want to define a contract for classes to implement. It only contains method signatures (no implementation), and classes that implement the interface are required to provide the implementation.
- **Use interface** when:
    - You need to ensure that certain methods are implemented by classes, but the classes don't need to inherit any common behavior.
    - You need to define multiple behaviors that can be applied to different classes.
    - You want to achieve **multiple inheritance** in Java, which is not possible with abstract classes.

Key Differences:

- **Abstract Class** can have method implementations, fields, and constructors.
- **Interface** cannot have method implementations (prior to Java 8), and all methods are abstract by default (except default methods and static methods in Java 8 and above).

2. Why can't we create the framework with all static methods? ⚠

While it's technically possible to create a framework with all **static methods**, it's not advisable for several reasons:

1. **Limited Extensibility**: Static methods are tied to a particular class and cannot be overridden. This makes the framework hard to extend or customize, as you cannot replace or change the behavior dynamically.

2. **Testability Issues**: Static methods make it difficult to mock or test components, as they don't rely on instance-based behaviors. It's hard to isolate dependencies when everything is static.
3. **Code Reusability**: Static methods cannot benefit from polymorphism, which is key for achieving reusability. You won't be able to easily change implementations of certain functions based on the context.
4. **Global State**: Static methods hold state globally, which may lead to unexpected results, especially in multi-threaded applications or when state changes are not controlled properly.

Best Practice: It's better to design your framework in an object-oriented manner, using instance methods where required, to allow for flexibility, inheritance, and better unit testing.

---

3. Int arr = {1, 2, 3, 4, 5, 7, 8, 9}; Skip 3 and all elements after 3, then skip 7 and then give some output. 🔢

Solution:

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5, 7, 8, 9};
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == 3) {
                break;    // Skip 3 and all elements after 3
            }
            System.out.print(arr[i] + " ");
        }
        System.out.println();    // Moving to the next line
        // Skipping 7
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == 7) {
```

```
                continue;    // Skip 7
            }
            System.out.print(arr[i] + " ");            }
    } }
```

Output:

1 2

1 2 4 5 8 9

- First, the program prints the elements before **3**.
- Then, it skips **7** and prints the remaining elements.

4. Implement HashMap in an interface and use it in the main. 🔑

Solution:

```
import java.util.HashMap;
interface DataStore {
    HashMap<Integer, String> map = new HashMap<>();
    // Method to add data to the map
    void addData(Integer key, String value);
    // Method to get data from the map
    String getData(Integer key);
}
class DataStoreImpl implements DataStore {
    @Override
    public void addData(Integer key, String value) {
        map.put(key, value);
    }
    @Override
```

```
    public String getData(Integer key) {
        return map.get(key);
    }
}
public class Main {
    public static void main(String[] args) {
        DataStore dataStore = new DataStoreImpl();

        // Adding data
        dataStore.addData(1, "Apple");
        dataStore.addData(2, "Banana");
        // Retrieving data
        System.out.println("Data for key 1: " + dataStore.getData(1));
        System.out.println("Data for key 2: " + dataStore.getData(2));
    }
}
```

Explanation:

- The **interface** DataStore defines methods for adding and retrieving data from a **HashMap**.
- The **DataStoreImpl** class implements the interface and provides the actual implementation for adding and getting values from the map.
- The **main method** demonstrates adding data and retrieving it using the implemented methods.

5. Remove redundancy for a string. 🔠

Solution:

Here's a solution that removes **duplicate characters** in a string:

```
import java.util.LinkedHashSet;

public class RemoveRedundancy {

    public static void main(String[] args) {

        String input = "programming";

        // Using LinkedHashSet to remove duplicates while maintaining order

        LinkedHashSet<Character> set = new LinkedHashSet<>();

        for (char c : input.toCharArray()) {

            set.add(c);

        }

        // Convert LinkedHashSet back to string

        StringBuilder result = new StringBuilder();

        for (char c : set) {

            result.append(c);

        }

        System.out.println("String after removing redundancy: " + result.toString());

    }
}
```

String after removing redundancy: progamin

- We use a **LinkedHashSet** because it removes duplicates while maintaining the original order of characters.
- The **StringBuilder** is used to convert the **LinkedHashSet** back into a string.

JAVA

1)how to reverse the words in the string ?

2)Using hash map, how to find no of occurrences of a character in the string?

3) Strings programs: Reverse a string, Uppercase letter retrieve, lowercase.

4) Retrieve words from sentence -- Regex

5) Unique Substring - Strings

6) Array Manipulation - add array elements etc.

7) Reverse Number - first explain with methods then move with loop and if conditions.

8) Basic HashMap manipulations like put and get.

9) Count number of letters in String

10) Count number of elements in array

11) Calculate Duplicate Characters.

12) Fibonacci Series

13) Even and Odd. Prime numbers 🟢🔴

14) Palindrome number/string. 🔄

15) Check If array contain element String and Number 🔍

16) Swap numbers without 3rd variable. 🔄

17) Track duplicate - if element already exist then pop error. 🚨

18) Count only Alphabets/Special Characters/Digits

19) Count Vowels.

20) Split Strings.

21) Largest Value/ Second largest and smallest digits.

22) Extract last characters in a String.

Theory for Java

1) String Objects and String literals creation in memory

2) String buffer and String builder

3) Static variables-methods-classes

4) Interface and Abstract Classes and use with example

5) Instance variable, local variable and Class variable.

6) Constructors cannot be overridden because it's should name the same class name.

7) Why interface cannot have constructors

interface cannot have constructor in java, because interface not have any instance member so nothing to construct.

Methods present in the interface are only declared not defined, As there is no implementation of methods,

there is no need of making objects for calling methods in the interface and thus no point of having constructor in it

8) HashMap, Sets, Lists, and Arrays (HashMap-Sets-Lists-Array proper understanding.

Lists >

ArrayList = it is not fixed value, it is an ordered collection, it accepts duplicate values. No Thread Safe

==========

> **Linked List** = it is not fixed value, changes can be done with this linked list. No Thread Safe

addFirst() Adds an item to the beginning of the list.

addLast() Add an item to the end of the list

removeFirst() Remove an item from the beginning of the list.

removeLast() Remove an item from the end of the list

getFirst() Get the item at the beginning of the list

getLast() Get the item at the end of the list

==========

>**Vector List** = Insertion order and also thread safe

Sets < HashSet = Duplicates not allowed, unordered collection.

The HashSet class has many useful methods. For example, to add items to it, use the add() method:

< LinkedHashSet = Duplicactes not allowed, insertion order

< TreeSet = Duplicates not allowed, sorted order (asscending or decending)

MAP > Map default provides key, pair value.

HashMAP= Key, pair values, null values allowed, No thread safety

HashTable= thread safety and null values not allowed

9) Primitive types and Non Primitive types - (Wrapper class)

Integer and Character

when you using collections that time you must be using Wrapper classes

Another useful method is the toString() method, which is used to convert wrapper objects to strings.

In the following example, we convert an Integer to a String, and use the length() method of the String class to output the length of the "string":

10) Typecasting - implicit and explicitly

11) Autoboxing and unboxing

12) Why Java not complete OOPS. - because it support Primitives types.

13) This and Super keywords with code understanding

The this keyword is used to refer to the current object, while the super keyword is used to refer to the parent object.

The this keyword can be used to access the current object's instance variables and methods, while the super keyword can be used to access the parent object's instance variables and methods.

14) OOPS concepts

15) Final – Finally

1) Get and Navigate

- How do you use get() and navigate() methods in Selenium? What is the difference between them?

2) Close and Quit

- What is the difference between driver.close() and driver.quit() in Selenium? When would you use each?

3) Assertions and Types in TestNG

- What are assertions in Selenium? Can you explain the different types of assertions available in TestNG?

4) Frameworks - POM and BDD

- What is POM (Page Object Model) and how does it work in Selenium?
- Can you explain BDD (Behavior-Driven Development) and its integration with Selenium for test automation?

5) Localization and Globalization Testing Using Automation Code

- How would you perform localization and globalization testing using Selenium?

6) Scenario and Scenario Outline Diff

- Can you explain the difference between a **Scenario** and a **Scenario Outline** in Cucumber? When would you use each?

7) Hooks

- What are hooks in Cucumber? Can you explain the different types of hooks and when they are executed?

8) WebDriver driver = new ChromeDriver() - Explain Each Keyword

- Can you break down the statement WebDriver driver = new ChromeDriver() and explain the role of each keyword?

9) Why WebDriver Used? Why Not Firefox or Chrome, etc.

- Why do we use WebDriver for automation? Why not other browsers like Firefox or Safari? What are the advantages of WebDriver over other tools?

10) Script Working in Night But Not the Next Day - Reasons

- Your Selenium script runs fine at night but fails the next day. What could be the reasons for this?

11) Cross Browser Testing Differences

- What are the differences when performing cross-browser testing in Selenium? How would you approach running tests on different browsers?

12) Thread.sleep, Fluent Wait, Implicit Wait, and WebDriver Wait (Explicit Wait)

- Can you explain the difference between Thread.sleep(), FluentWait, ImplicitWait, and ExplicitWait in Selenium?
- Why and when should you use each of these waits in Selenium?

Example:

  - **Before:** driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
  - **After:** driver.manage().timeouts().scriptTimeout(Duration.ofMinutes(2));
  - driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(10));
  - **Explicit Wait Example:**

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector(".classlocator")));
```

13) Actions, Select, Alert - Learn All

- Can you explain the usage of **Actions**, **Select**, and **Alert** classes in Selenium?
    - How do you handle mouse actions, keyboard actions, and drag-and-drop actions using the **Actions** class?
    - How do you interact with dropdowns using the **Select** class?
    - How do you handle browser alerts using the **Alert** class?

14) Chrome Options - Disable Notifications, etc.

- How can you disable browser notifications in Chrome using ChromeOptions in Selenium? Can you give an example of setting ChromeOptions for other purposes like headless mode?

15) Xpath - Axes Like Preceding, Following Siblings, Parent

- Can you explain XPath axes such as preceding, following, sibling, and parent? How would you use them in Selenium for locating elements?

1) Descendant in XPath

- Can you explain what a descendant is in XPath? How is it used to locate elements in a web page?

2) Why Captchas and OTPs Cannot Be Automated?

- Why can't Captchas and OTPs be automated using Selenium? What are the challenges involved?

3) Desired Capabilities Uses in Framework

- What are Desired Capabilities in Selenium? How are they used in a test automation framework?

4) Switch to 3rd Child Window

- How can you switch to the 3rd child window in Selenium WebDriver? Can you explain the code snippet for this?

```
Set<String> windows = driver.getWindowHandles();

Iterator<String> iterator = windows.iterator();

String firstWindow = iterator.next();
```

String secondWindow = iterator.next();

String thirdWindow = iterator.next(); // Switch to 3rd window

driver.switchTo().window(thirdWindow);

5) Switch to Frames

- How do you switch between frames in Selenium WebDriver? Can you provide an example?

6) Refresh Page in Different Ways

- What are the different ways to refresh a page in Selenium? Explain with examples.
    - driver.navigate().refresh();
    - driver.navigate().to(driver.getCurrentUrl());
    - driver.findElement(By.id("username")).sendKeys(Keys.F5);

7) Types of Exceptions with Examples

- What are the different types of exceptions in Selenium? Can you provide examples of how they occur and how to handle them?

8) Checked and Unchecked Exceptions in Java

- What is the difference between checked and unchecked exceptions in Java? Provide examples of both.

9) JavaScript Executor

- What is the JavascriptExecutor interface in Selenium? How would you use it for:
    - Getting an element
    - Clicking an element
    - Highlighting an element
    - Scrolling an element within a page

Example:

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,500)"); // To scroll

```
Thread.sleep(2000);

js.executeScript("document.querySelector('.tableFixHead').scrollTop=500");
```

10) Take Screenshot for Failed Test Cases

- How can you take a screenshot of a failed test case using Selenium? Can you explain the implementation and the role of TakesScreenshot interface? Example code:

```
TakesScreenshot ts = (TakesScreenshot) driver;

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

File file = new File(System.getProperty("user.dir") + "//reports//" + testCaseName + ".png");

FileUtils.copyFile(source, file);

return System.getProperty("user.dir") + "//reports//" + testCaseName + ".png";
```

11) DOM and Stale Element Exception

- What is the DOM (Document Object Model)? What is a Stale Element Reference Exception in Selenium, and how can you avoid it?

12) WebDriver Architecture - Latest vs Old

- Can you explain the difference between the old WebDriver architecture and the latest WebDriver architecture?

13) SSL Security and Auth Pop-Up Handling

- How do you handle SSL security certificate pop-ups and authentication pop-ups in Selenium? Can you provide an example of handling authentication via http://username:password@url?

14) TestNG Priorities (Including Negative Values)

- What are TestNG priorities, and how can you assign priorities to test methods? Can a priority be negative? If yes, how does it affect test execution order?

15) TestNG Parameters (Invocation Count, DependsOnMethods, DataProvider, etc.)

- Can you explain TestNG parameters such as InvocationCount, dependsOnMethods, Parameters, DataProvider, and their usage in test cases?

16) TestNG XML and Integration with POM

- How do you integrate TestNG XML with a Page Object Model (POM) framework? Can you provide an example?

17) Maven Understanding

- What is Maven, and how is it used in a Selenium framework? Can you explain the pom.xml file in detail?

18) Jenkins (Code Execution and Project Creation)

  - How does Jenkins execute test cases?
  - Can you explain the project creation process in Jenkins and integration with Git?

19) MetaGroups in TestNG

- What are MetaGroups in TestNG? How do they help in test execution and reporting?

20) How to Handle Ajax Calls in Selenium

21) Context Node in Selenium

🔍 **Refresh page in different ways**

driver.navigate.refresh();

driver.navigate.to(driver.getCurrentURL());

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

🔍**) Types of exceptions with examples.**

🔍**) Checked and Unchecked Exceptions - JAVA**

🔍**) Javascript Executor - Get element**

click element

highlight element

Scroll element within a element

```
// to scroll it down we use JavaScript
    JavascriptExecutor js=(JavascriptExecutor)driver;
```

```
    js.executeScript("window.scrollBy(0,500)");// from console to scroll it we use this window.scrollBy based on how much scroll to give we use that much value( 200|| 500)
    Thread.sleep(2000);
    js.executeScript("document.querySelector('.tableFixHead').scrollTop=500");
```

🔍) **TakeScreenshot - Explain interface and for failed testcases**

```
  TakesScreenshot TS= (TakesScreenshot)driver;
  File source= TS.getScreenshotAs(OutputType.FILE);
  File file= new File(System.getProperty("user.dir") +"//reports// "+testCaseName+ ".png");
  FileUtils.copyFile(source, file);
```

FIleUtils is class consists the copyfile method

```
  return System.getProperty("user.dir" )+"//reports// " +testCaseName+ ".png";
```

explain ITestListeners - with methods.

🔍) **What is DOM and Stale Element exception**

🔍) **WebDriver architecture latest and old.**

🔍) **SSL security and auth pop up - handle**

using http://username:password@url_without_https in

driver.get()

🔍) **TestNG priorties (can be negative also)**

🔍) Paramters like - Invocation count, dependsOnMethod,

Paramters, Data-Provider etc.

🔍) **TestNG XML and integration with POM.**

🔍) **Maven understanding**

🔍) **Jenkins (depends on exp and Resume)**

explain code execution and if experience tell about project

creationa nd GIT.

🔍) **MetaGroups - TestNG**

🔍) **How to handle Ajax Calls**

🔍) **Context Node**

coding-related questions strings and arrays for automation testers:

#Strings

1. Reverse a String
2. Check if a String is a Palindrome
3. Determine if Two Strings are Anagrams
4. Find the Longest Substring Without Repeating Characters
5. Compress a String by Counting Consecutive Characters
6. Check if One String is a Rotation of Another
7. Tokenize a String into Words
8. Generate All Permutations of a String
9. Count Vowels and Consonants in a String
10. Check if a String is a Pangram

#Arrays

1. Find Pairs of Integers in an Array that Sum up to a Target
2. Remove Duplicates from a Sorted Array
3. Rotate an Array to the Right by a Given Number of Steps
4. Merge Two Sorted Arrays into a Single Sorted Array
5. Find the Missing Number in an Array from 1 to N
6. Find the Contiguous Subarray with the Largest Sum

7. Check if an Array Contains Duplicates

8. Calculate the Product of Array Except Self

9. Move Zeroes to the End of an Array

10. Find a Peak Element in an Array

1) Why do we prefer TestNG over JUnit for the testing framework?

2)What are the different ways to input text other than sendkeys()?

3)What are the Relative Locators with respect to Selenium 4?

4)Is it possible to read PDF files using Selenium WebDriver? If yes, then how?

5)How would you automate a CAPTCHA using a Selenium WebDriver?

6)How can you read Barcode if it is present on a Web Page?

7)Why are we getting StaleElementReference Exception in Selenium? Give Reasons.

8) For Angular/React/Node.js Based Application which automation tool do you prefer -Selenium, Cypress or Playwright and Why?

9) What kind of tests should not be automated?

10) Which XPath Axes do you know?

11) Compare Selenium WebDriver v3 vs v4 - Key features.

12) How can you disable images on a Web Page as it takes more to load a page?

13) How would you handle the alert popups in Selenium WebDriver?

14)How will you handle dynamic elements by using XPath?

15) What is the Test Automation Pyramid?

16) Suppose a file is 100mb, how can I wait for the file to be downloaded completely?

17) Explain Soft Assertions & Hard Assertions in the Automation Framework.

18) Can you access the OTP code from your email through Selenium Automation? If yes then how?

19) How to handle Psuedo-Elements by Selenium WebDriver?

20)How to handle a Browser Based Authentication pop-up in a Selenium WebDriver?

21) How can you handle cookies in a Selenium WebDriver? Will you be able to use the same cookies to restore the session of Web application?

22)How would you automate a scenario where you need to verify specific colors -Theme testing?

23) What are the different Chrome Options you have used so far in your automation framework?

24) The Automation script needs to run on multiple browsers and platforms. How do you handle this?

25)How will you give the total time required for the test case to automate?

26) Are you able to automate the below WebElements:

<canvas data-type="lower" width="489" height="211" class=""></canvas>

<mat-select role="combobox" aria-autocomplete="none" aria-haspopup="listbox" class="mat-mdc-select"><!---></mat-select>

27)How to select a good test automation tool for a project?

28) What is SikuliX?

29)What are the common Selenium Exceptions you faced so far and how have you handled them?

30)What is the CI/CD pipeline & how will you implement it in your automation framework?

INTERVIEW QUESTIONS

Project specific Questions

1. What was the duration of your project?
2. Explain about your project.
3. How many testers were their on this project.
4. According to you which was the complex part of the project from testing point of view.
5. How did you do the testing of your project?
6. How many test cases have you designed? How many you wrote in a day?
7. How many bugs did you find?
8. Tell me any high Severity bugs that you found.
9. What happens when the client changes requirements?
10. Which tool you used for defect reporting?
11. What is the database used for your project.
12. In which technology this application is developed?

Company specific Questions

1. Where are you working currently?
2. How that Seed has sent you here for an interview when you are working in the same company.
3. Where it is located?
4. How many testers and developers are their?
5. Are testers and developers sitting at same location?
6. What is the hierarchy in testing team?
7. Whom do you report to?

8. Is this a dummy project?

9. How do you come to know about your tasks?

10. Are you a permanent employee?

11. How long you are working in Seed?

12. Can you name some clients of your company?

HR Questions

1. Introduce yourself.

2. What are your strengths?

3. What are your weaknesses?

4. What you do not like in this world?

5. How do you commute?

6. Are you a team player?

7. How long will you take to join?

8. Are you ready to work on contract position?

9. Where would you like to see yourself 5 years down the line.

10. Are you a permanent employee?

1. Define what is Software Testing?

2. What is difference between Severity and Priority?

3. What is difference between Regression and Retesting?

4. What is the Role of the Tester?

5. How will you convince a developer who is not ready to resolve the bug you have raised?

6. What is difference between Debugging and Unit testing?

7. Have you done White Box Testing? Do you know what type of bugs you can trace through this type of testing?

8. What is the important column in Test case?

9. What is the use of any Test Case Management tool?

10. Why Testing is necessary?

11. What is the Bug Life Cycle?

12. How do you decide which test cases to consider for Regression Testing?

13. Is testing QA or QC?

14. Differentiate between Integration Testing and System Testing.

15. Have you participated in User Acceptance Testing (UAT)?

16. Have you seen a Test Plan? Who prepares it ?

17. What are the various contents of a Test Plan?

18. What methods of Black Box Testing you have used?

19. Do you know Exploratory Testing?

20. What is Prototyping? What are the various types?

21. Define phases of SDLC.

22. Which SDLC model your company follows?

23. When do we write Stubs?

24. What is the use of Driver script?

25. What kinds of reviews are conducted during SDLC phase?

26. What is the role of Metrics in Testing?

27. What are key challenges of Software Testing field?

28. Explain your views about Quality.

29. What is the advantage of using any bug tracking tool.

30. Explain about website Usability testing.

31. What are defect attributes?

32. What is pesticide paradox?

33. Explain V model.

34. What is Load and Stress testing, explain with example

35. Differentiate between Verification and Validation?

36. Explain Boundary Value Analysis with an example.

37. What is the difference between Web application and Client Server Application?

38. What is Localization testing?

39. What is difference between Black box testing and White box testing?

40. What is the difference between Authentication and Authorization give an example.

41. Other than functionality what else you should test in Web Application?

42. How you know that testing is enough?

43. How will you test the reports?

44. What is compatibility testing?

45. If we test the application on IE 8 then is it necessary to test it on IE 7?

***Sapient Interview Questions*:**

java :

1) what is polymorphism?

2) Exception handling ?

3) Difference b/w throw and throws?

4) what is file in and File out ?

5) How to write the data for notepad ? write the code ?

6) What is abstract and interface with example ?

7) write a program fact using recursion ?

8) How will you handle multiple exceptions ?

Selenium :

1) explain selenium architecture?

2) write a xpath or make my trip application? write the xpath for Holiday and no need to change the existing xpath .By using the same xpath you need to identify the hotels as well.

3) Explain selenium grid how to setup?

4) How are you maintaining the logs in your framework?

5) Explain framework how you design?

TestNG

1) How will you execute the dependency test cases ?

2) How will you execute the 3 test cases at time ?

3) difference b/w Data provider and parameters?

4) what is the difference b/w Before Test and Before method ? How will it work? Give me an example ?

5) write the testng xml format ?

BBD:

1) What is BDD, why do people can prefer BDD nowadays ?

2) in BDD why before and after using ? Give me an example ?

Manual Testing :

What is a test strategy ?

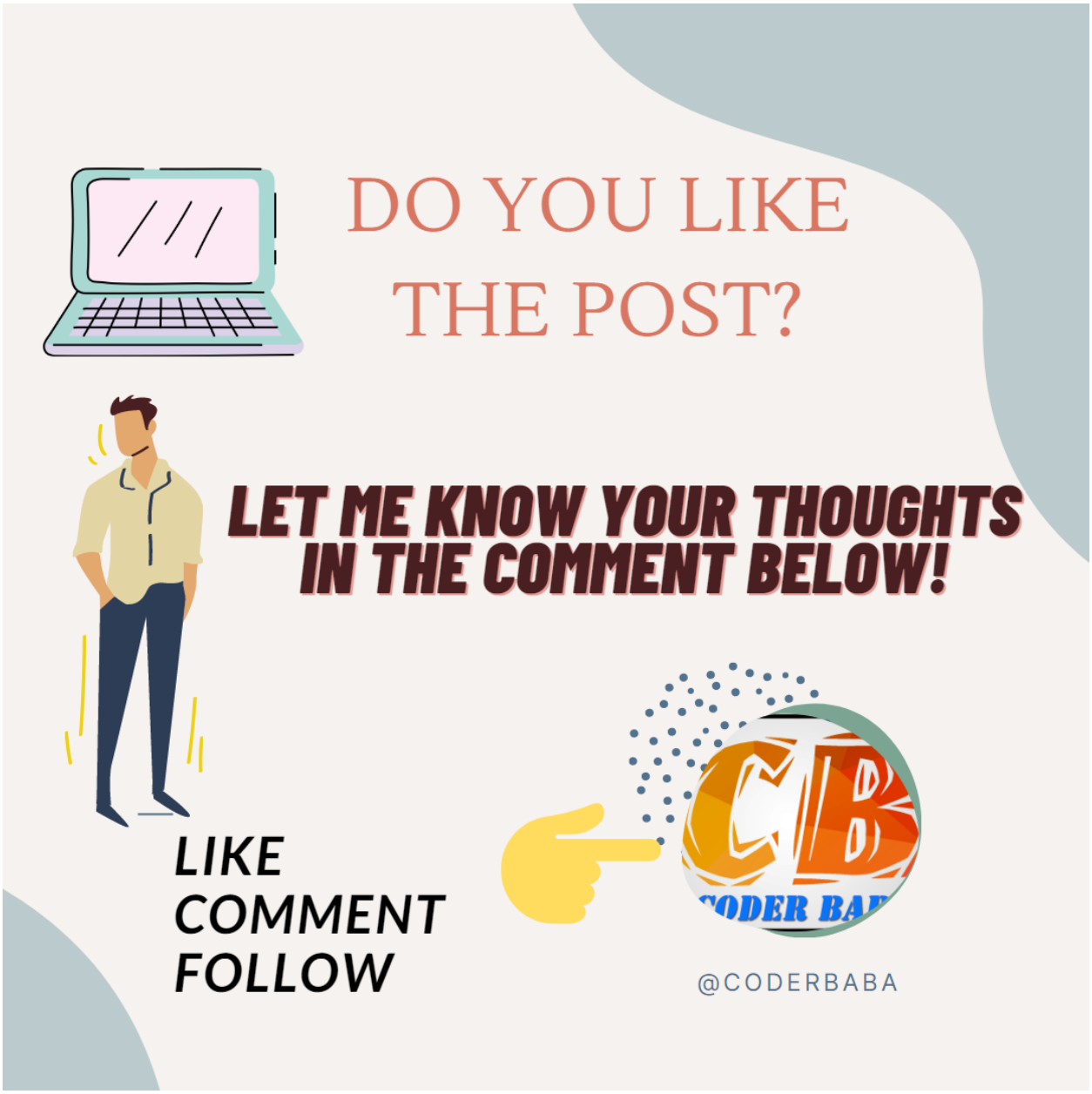
What is the test plan ? How will you create?

how will you follow the agile process in your company ?

What is estimation?

Jenkins :

how you will create the job and how it will execute ?


DO YOU LIKE THE POST?
LET ME KNOW YOUR THOUGHTS IN THE COMMENT BELOW!
LIKE
COMMENT
FOLLOW
@CODERBABA