1. where we use finally block in real time?

The finally block is used for cleanup operations, such as closing files, releasing resources, or handling other post-execution tasks. It is guaranteed to execute after the try-catch block, regardless of whether an exception occurs.

### 2. Authentication and Authorization in API

* Authentication verifies the identity of the user (e.g., username, password, token).
* Authorization determines what actions the authenticated user can perform, based on permissions or roles.

3. What is an exception? Types of exceptions?

An exception is an event that disrupts the normal flow of execution. Types include:

* Checked exceptions (e.g., IOException)
* Unchecked exceptions (e.g., NullPointerException)
* Error exceptions (e.g., OutOfMemoryError)

4. Explain your framework in detail?

This will be specific to the framework you have built. A typical explanation includes:

* Type: Data-driven, Keyword-driven, BDD, etc.
* Languages: Java, Python, etc.
* Tools used: Selenium, TestNG, Maven, Jenkins, etc.
* Architecture: POM (Page Object Model), TestNG for running tests, data providers for test data, reports for test execution logs.

5. Group ID and Artifact ID in Maven

* Group ID is the unique identifier for a group of projects (e.g., com.example).
* Artifact ID is the unique identifier for a specific project or module (e.g., my-app).

6. Types of Error Codes in API

* 2xx: Success (e.g., 200 OK)
* 3xx: Redirection (e.g., 301 Moved Permanently)
* 4xx: Client Errors (e.g., 404 Not Found)
* 5xx: Server Errors (e.g., 500 Internal Server Error)

7. Java Program to Find Max and Second Max in Array

java

CopyEdit

public class MaxSecondMax {

public static void main(String[] args) {

int[] arr = {10, 20, 4, 45, 99};

int max = Integer.MIN\_VALUE, secondMax = Integer.MIN\_VALUE;

for (int num : arr) {

if (num > max) {

secondMax = max;

max = num;

} else if (num > secondMax && num != max) {

secondMax = num;

}

}

System.out.println("Max: " + max);

System.out.println("Second Max: " + secondMax);

}

}

8. TestNG Annotations

* @Test: Marks a method as a test case.
* @BeforeSuite, @AfterSuite: Runs before/after the entire suite.
* @BeforeTest, @AfterTest: Runs before/after each test.
* @BeforeMethod, @AfterMethod: Runs before/after each test method.
* @DataProvider: Provides test data.

9. What is a static keyword in Java?

The static keyword in Java signifies that a member (variable, method, block, or nested class) belongs to the class itself rather than to any specific instance (object) of that class. It's primarily used for memory management and defining members that are shared among all instances of the class.

10. How to handle dynamic elements?

You can use dynamic XPath with contains() or use CSS Selectors with partial matching (^, $, \*).

Example:

driver.findElement(By.xpath("//\*[contains(@id, 'dynamicID')]"));

11. Avoid StaleElementReferenceException

* Re-locate the element before interacting with it.
* Use WebDriverWait to ensure the element is still attached to the DOM.

12. Handling Error Codes in API

You can handle error codes by checking the HTTP status codes and implementing logic to deal with failures.

For example:

if(response.getStatusCode() == 500) {

// Retry or handle server error

}

13. Difference Between Selenium 3 and Selenium 4

* Selenium 4 supports the W3C WebDriver standard, while Selenium 3 used the JSON Wire Protocol.
* Selenium 4 has improved browser interaction and DevTools Protocol support.

14. Java Program for Output: 123 (Extra number detection)

public class ExtractNumbers {

public static void main(String[] args) {

String input = "abc123";

String result = input.replaceAll("[^0-9]", "");

System.out.println(result); // Output: 123

}

}

15. Explain Your Project and Domain

The semiconductor domain encompasses materials and devices that can control the flow of electrical current, acting as an intermediary between conductors and insulators. These materials are fundamental to modern electronics, powering everything from smartphones to computers.

Key aspects of the semiconductor domain:

* **Material Properties:**

Semiconductors have electrical conductivity between that of conductors (like copper) and insulators (like glass). They can be manipulated to control current flow, making them ideal for building electronic components.

* **Applications:**

Semiconductors are the foundation of integrated circuits (IC), also known as chips, which are used in a vast array of electronic devices.

* **Control of Conductivity:**

Semiconductors' conductivity can be adjusted by adding impurities (doping), enabling precise control over electrical current flow.

* **Examples:**

Silicon, germanium, and gallium arsenide are common semiconductor materials.

* **Industry Significance:**

The semiconductor industry is a major driver of technological advancement, with companies like Intel and TSMC playing key roles in designing and manufacturing these chips.

* **Sub-domains within Materials Science:**

The semiconductor domain is a sub-area of materials science, involving both simulation and experimentation to study and optimize materials.

16. Difference between Set and List.

* + Set: Contains unique elements with no defined ordering (unless using a SortedSet).
  + List: Allows duplicate entries and maintains insertion order, supporting indexed access.

17.Where do we use collections in a framework?  
Use Cases:

* + Storing test data (lists, maps for dynamic data storage).
  + Managing locators and page objects.
  + Caching test results or logs before reporting.

18.Difference between array and ArrayList.

* + Array: Fixed in size and can hold primitives as well as objects.
  + ArrayList: A resizable, dynamic collection that holds objects only and provides methods for easy manipulation of data.

19.What is basic authentication in API?  
Explanation:

* + It uses a simple Base64 encoding scheme where the username and password are combined into a single string (e.g., "username:password") and then encoded. This encoded string is sent as part of the Authorization header (in the format "Basic <encoded\_value>").

20.Explain Agile methodologies.  
Core Ideas:

* + Iterative and incremental development.
  + Emphasis on collaboration, adaptability, and customer feedback.
  + Frameworks such as Scrum, Kanban, and Extreme Programming (XP) which promote continuous planning and improvement.

21.Running particular test cases multiple times in TestNG?  
Answer:

* + You can achieve this using the invocationCount attribute on the @Test annotation. For example:

@Test(invocationCount = 5)

public void runMultipleTimes() {

// test code

}

22.What is a DataProvider in TestNG?  
Explanation:

* + A DataProvider is an annotation that supplies a method with multiple sets of data so that the same test method can be executed with different input values. This supports data-driven testing.

23.What are the various listeners you have used in your project?  
Examples:

* + ITestListener: To log test start, success, failure, and skip events.
  + ISuiteListener: To capture events before and after the test suite execution.
  + Other listeners (e.g., for capturing screenshots on failure).

24.What is an interface?  
Explanation:

* + An interface is a reference type in Java that can contain abstract methods (and default methods since Java 8). It specifies what a class must do but not how.

25.Difference between interface and abstract class.

* + Interface: All methods are abstract (unless default or static) and does not maintain state; a class can implement multiple interfaces.
  + Abstract Class: Can have both abstract and concrete (implemented) methods as well as member variables; a class can extend only one abstract class.

26.What is the diamond problem?

* + It refers to an ambiguity that can occur in multiple inheritance (as seen in some languages) when two classes inherit from the same interface or class and a child class inherits from both. Java avoids this by not supporting multiple inheritance of classes.

27.Which is best – CSS Selector or XPath and why?

* + CSS Selectors: Generally faster and easier to read but might lack advanced traversal capabilities.
  + XPath: More powerful for navigating complex DOM structures (e.g., locating parent or preceding-sibling elements).
  + The choice depends on the specific test scenario and element structure.

28.Which wait condition to use when an element must be visible?

* + Use an explicit wait with condition such as ExpectedConditions.visibilityOfElementLocated.

29.Why is a constructor used in a framework?

* + Constructors initialize objects. In an automation framework they are commonly used to set up objects such as WebDriver instances, initialize page elements, or load configurations required for tests.

30.Explain pom.xml and where you use it.

* + The pom.xml file is the Project Object Model file used by Maven. It defines project dependencies, plugins, build configurations, and other settings. It is used to manage project builds and dependencies.

1. How to find broken links and write code for it.  
   Overview:
   * Write code to iterate over all <a> tags on a webpage, extract the href attribute, and then use an HTTP client (such as HttpURLConnection in Java) to check the response code.  
     Pseudocode:

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

for (WebElement link : links) {

String url = link.getAttribute("href");

// Make HTTP request to 'url' and check response code.

// If response code is 400 or 500 series, mark as broken.

}

31.How to read and write data in an Excel sheet?  
Answer:

* + Use Apache POI (for XLS and XLSX files) which provides classes to read from and write to Excel files. For example, using the Workbook, Sheet, and Row classes to manipulate Excel data.

32.Entering text into a textbox if sendKeys() is not allowed?  
 Alternate Approaches:

* + Use the JavaScript Executor to set the value directly. For example:

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].value='your text';", element);

33.How to run your automation?  
Answer:

* + Typically, automation tests are executed using a build tool (like Maven) with TestNG or JUnit, using commands such as mvn test or via a Continuous Integration server (e.g., Jenkins) which triggers the suite automatically.

34.What are the challenges faced in automation?  
Common Challenges:

* + Handling dynamic elements and synchronization issues.
  + Dealing with flaky tests or environmental inconsistencies.
  + Maintaining large test suites and managing test data.
  + Integrating with CI/CD pipelines and managing cross-browser/cross-platform issues.

35.hat do you do if you have a large test suite to execute in less time?  
Approaches Include:

* + Implementing parallel test execution using TestNG’s parallel capabilities.
  + Utilizing a grid (e.g., Selenium Grid) or cloud-based testing platforms to distribute tests.
  + Optimizing tests by prioritizing or grouping similar test cases.

36.Compile Time and Run Time Polymorphism.

* + Compile Time (Static) Polymorphism: Achieved by method overloading.
  + Run Time (Dynamic) Polymorphism: Achieved by method overriding.

37.How to handle multiple windows?

* + Use driver.getWindowHandles() to get all window IDs.
  + Iterate through the handles, switch to the desired window with driver.switchTo().window(windowHandle), perform actions, and then switch back.

38.Explain 401 and 500 error codes in API.

* + 401 Unauthorized: The client request lacks valid authentication credentials.
  + 500 Internal Server Error: The server encountered an unexpected condition that prevented it from fulfilling the request.

39.What is the Page Object Model (POM)?

* + POM is a design pattern that creates an object repository for web UI elements. Each webpage is represented by a separate class, encapsulating the elements and behaviors, leading to cleaner and more maintainable code.

40.How do you deal with flaky test cases in automation?  
Techniques:

* + Implement retry mechanisms.
  + Use explicit waits to handle timing issues.
  + Refactor locators to be more stable.
  + Log and track intermittent issues and fix the underlying causes.

41.SQL Joins.  
Brief Explanation:

* + INNER JOIN: Retrieves matching rows in both tables.
  + LEFT (OUTER) JOIN: Retrieves all rows from the left table and matched rows from the right.
  + RIGHT (OUTER) JOIN: Retrieves all rows from the right table and matched rows from the left.
  + FULL (OUTER) JOIN: Retrieves rows when there is a match in one of the tables.

42.Exception Handling.

* + Use try-catch blocks to catch exceptions.
  + Optionally include a finally block for cleanup.
  + Throw exceptions when necessary using the throws keyword in method signatures.

43.How to reverse the words in a string?  
Example: Convert "I like program language java" to "java language program like I".

public class ReverseWords {

public static void main(String[] args) {

String sentence = "I like program language java";

String[] words = sentence.split(" ");

StringBuilder reversed = new StringBuilder();

for (int i = words.length - 1; i >= 0; i--) {

reversed.append(words[i]).append(" ");

}

System.out.println(reversed.toString().trim());

}

}

44.Using a HashMap, how to find the number of occurrences of each character?  
Explanation: (Similar to question 19 above.)

* + Loop through each character in the string, update its count in a HashMap, and then print the frequency.

45.Can the main method be overloaded?

* + Yes, you can overload the main method with different parameter lists. However, the JVM only calls the standard public static void main(String[] args) method as the program entry point.

46.Can the main method be overridden?

* + No. Since the main method is static, it belongs to the class rather than an instance, and static methods cannot be overridden.

47.Which access modifiers can be used in an interface?

* + All methods declared in an interface are implicitly public (and abstract, unless declared as default or static).
  + Fields in an interface are implicitly public, static, and final.

48.Access modifiers in Java and their scope.

* + public: Accessible from any other class.
  + protected: Accessible within the same package and subclasses in other packages.
  + default (no modifier): Accessible only within the same package.
  + private: Accessible only within the same class.

49.Difference between scenario and scenario outline (Cucumber).  
Explanation:

* + Scenario: A single test case with hard-coded values.
  + Scenario Outline: A template for multiple scenarios where the same steps are run with different sets of data provided in an Examples table.
  + Clarify requirements and the scope.
  + Identify test scenarios.
  + Decide on the tool set (e.g., Selenium, API testing libraries).
  + Propose a scalable framework (e.g., POM for UI tests or data-driven approaches for API tests).
  + Estimate timelines and discuss reporting and maintenance aspects.

50.How can you execute failed test cases without manual intervention?

* + Use TestNG’s IRetryAnalyzer or similar mechanisms to re-run failed tests automatically.
  + Integrate with Continuous Integration tools so that a build automatically triggers a re-run of failed tests.

51.Factors to consider when choosing an automation tool.  
Considerations:

* + Cost and Licensing: Open-source vs. commercial licenses.
  + Community Support: Availability of help and updates.
  + Technology Stack Compatibility: How well the tool integrates with your technology environment (web, mobile, desktop).
  + Ease of Use and Learning Curve: Developer familiarity.
  + Reporting & Maintenance: Ability to generate detailed reports and ease of updating tests.

52.What is the Action class and a few methods of it?

* + The Actions class in Selenium is used to build complex user interactions.
  + Common methods:
    - clickAndHold(), dragAndDrop(), doubleClick(), moveToElement(), contextClick().

53.Handling child windows – Java code example.  
Approach:

* + Retrieve all window handles.
  + Iterate to find the child window.
  + Switch to it, perform operations, then switch back.  
    Sample Code Fragment:

String parentWindow = driver.getWindowHandle();

Set<String> handles = driver.getWindowHandles();

for (String handle : handles) {

if (!handle.equals(parentWindow)) {

driver.switchTo().window(handle);

// Perform operations on child window

driver.close(); // Close the child window if needed

}

}

driver.switchTo().window(parentWindow);

54.What is Velocity in Agile Testing?

* + Velocity is a metric that measures the amount of work a team completes during a sprint.
  + It is used to predict future capacity and helps in sprint planning.

55.What is the static keyword in Java?

* + The static keyword is used to denote that a member (variable or method) belongs to the class rather than any instance.
  + Static members are shared among all instances of the class.

56.Purchase order in amazon-selenium code

* + This likely refers to an automation test script for simulating a purchase order process on an e-commerce platform like Amazon.
  + Key steps would include searching for a product, adding it to the cart, proceeding through the checkout process, and verifying the order summary.

### 57. How to solve merge conflict in Git?

* When: Merge conflict happens when two branches have changes in the same part of a file.
* How:
  + Pull latest code first (git pull).
  + Merge the feature branch (git merge feature\_branch).
  + Git will show conflict markers (<<<<<<< HEAD, =======, >>>>>>> branch).
  + Manually edit and resolve conflicts.
  + After resolving:

git add <filename>

git commit -m "Resolved merge conflict"

* + Then push:

git push

58. What is Git Stash?

* Git stash is used to save uncommitted changes temporarily without committing.
* Useful when you need to switch branches but don’t want to commit incomplete work.
* Example:

git stash // Save changes

git stash list // See stashed changes

git stash pop // Retrieve the last stash

59. How to integrate test cases through Jenkins? (Explain Flow)

* Steps:
  1. Create a Maven/Gradle project for automation.
  2. Jenkins → Create New Job → Select Freestyle Project or Pipeline.
  3. Source Code Management: Connect your GitHub/Bitbucket repo.
  4. Build Section:
     + Add command to build/run tests:

mvn clean test

* 1. or use a pom.xml goal.
  2. Configure reporting (Extent, Allure).
  3. Schedule (optional) using Triggers (e.g., build every night or on commit).
  4. Execute the job → View results in Jenkins.

60. Where do we use Polymorphism in Selenium framework?

* Usage:
  + Overriding: We override WebDriver methods using custom classes sometimes.
  + Overloading: Reusing the same method name with different parameters (ex: clickButton(WebElement) and clickButton(String locator)).
  + In Framework:
    - In Page Object Model (POM) - multiple page classes implementing same interface.
    - In Base classes where methods are reused and modified as per page needs.

61. How to handle dynamic elements?

* Strategies:
  + Use relative XPath with contains(), starts-with(), ends-with(). Example:

driver.findElement(By.xpath("//input[contains(@id, 'search')]")).sendKeys("Laptop");

* + Use waits (Explicit Wait):

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

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//button[contains(text(),'Submit')]")));

* + Use JavaScriptExecutor when normal locators fa

62. Selenium Code for Amazon Search Box (Enter 'laptops' and print first 5 laptop names)

WebDriver driver = new ChromeDriver();

driver.get("https://www.amazon.in/");

driver.findElement(By.id("twotabsearchtextbox")).sendKeys("laptops");

driver.findElement(By.id("nav-search-submit-button")).click();

List<WebElement> laptopNames = driver.findElements(By.cssSelector("span.a-size-medium.a-color-base.a-text-normal"));

for (int i = 0; i < 5; i++) {

System.out.println(laptopNames.get(i).getText());

}

driver.quit();

63. Java Program to Count Unique Elements (Input: SSSRJTEH)

import java.util.HashSet;

public class UniqueCharacters {

public static void main(String[] args) {

String input = "SSSRJTEH";

HashSet<Character> uniqueChars = new HashSet<>();

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

uniqueChars.add(c);

}

System.out.println("Total unique characters: " + uniqueChars.size());

System.out.println("Unique Characters: " + uniqueChars);

}

}

64. What is POM (Page Object Model)?

* It is a design pattern in Selenium Automation Framework.
* Each web page is represented by a Java class.
* In each class:
  + Define page locators.
  + Create reusable methods (actions) for the page.
* Advantages:
  + Clean code.
  + Reusable and maintainable.
  + Easy to update when UI changes.

Example structure:

* + LoginPage.java
  + HomePage.java
  + CartPage.java

65. Java Program to calculate 35% of 5 lakh

public class PercentageCalculation {

public static void main(String[] args) {

double amount = 500000;

double percentage = 35;

double result = (percentage / 100) \* amount;

System.out.println("35% of 5 lakh is: " + result);

}

}

Output:

35% of 5 lakh is: 175000.0

Top of Form

Bottom of Form

Bottom of Form