**Interview Questions**

**What are the**

**1. WAP java input: kirti**

**Output: itrik**

**Ans:**

public class Reverse

{

public static void main(String[] args)

{

**//This line creates a string called** name **with the value** "kirti"

String name = "kirti";

**//This creates an empty string called** reversed**. We will add characters to this string one by one in reverse order.**

String reversed = "";

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

{

reversed += name.charAt(i);

}

System.out.println("The name string is: " + name);

System.out.println("The reversed string is: " + reversed);

}

}

**Output:**

**the name string is: kirti**

**the reversed string is:itrik**

**2.where we use finally block in real time?**

 Closing database connections

 Closingfile streams or network sockets

 Releasingresources (memory, locks, etc.)

 Loggingsystem clean-up

 Rollingback transactions

**3. What is authentication and authorization in api?**

**Ans:** Authentication is the process of verifying the identity of a user or application that’s trying to access your API.

**Examples:**

A user logs in with email and password.

A mobile app sends an Auth token to access user data.

A service makes a request with an API key.

Authorization happens after authentication, and it determines what permissions the authenticated user or application has.

**Examples:**

An admin can delete a user, but a regular user cannot.

A user can access their own data but not someone else's.

A read-only token can't create or delete resources.

**4. What is exception? Exception types?**

**Ans**: java is a problem that happens during the execution of the program.

**Exception Types:**

1. **Checked Exceptions (Compile-time exceptions)**

**The exception which we got during the compile time.**

**Ex:**

IOException (which will get when we try to read the invalid file)

SQLException (database query expection)

FileNotFoundException

1. **Unchecked Exceptions (Runtime exceptions)**

**The exception is caught during the runtime**

**Ex:**

ArithmeticException (divide by 0)

NullPointerException (accessing null)

ArrayIndexOutOfBoundsException (wrong array index)

NumberFormatException (wrong string to number)

**5. Explain your framework in detail?**

Currently we are using Cucumber framework with TestNG implementations.

\* Our project structure looks like:

UnderProject:

\*Reports

**->**extentReport

->htmlReport

->sparkReport

->pdfReport

**Under src/test:**

src/test

\*context

\*pageInitializer

\*pageObjects

\*stepDefinitions

->hooks

\*utilities

->configReader

->driverFactory

->retryListeners

->testNGRetry

\*resources

->features

->config.properties

->extent.properties

\*xmls

->pom.xml

->testNG.xml

->extengConfig.xml

->log4J.xml

**Under src/test: We have packages like**

pageObjects stepDefinitions  hooks-->runnerClass-->utilitypageInitializerContext

pageObject- Object repository. We will store all the locators(xpaths,id,name,css), getters and setters(constructors) method for each xpaths.

stepDefinitions- Here we will define the java code for each step written in feature file (Give, when, then, and).

We wil extend the pageInitializer class in stepDefinition class.

hooks- we will add pre conditions and post conditions for scenarios.

We will use annotations to define these pre conditions and post conditions (@Before- setup, configReader(qa,dev) @After- tearDown)

We will add screenshot code for failed steps.

runnerClass- we will insert @CucumberOptions, @RunWith tags

@CucumberOptions- features (path of feature files), glue(path of the step definitions),

plugin ={"pretty",

"html: target/Cucumber.html",

"json: target/Cucumber. Son",

"junit: target/Cucumber.xml",

"com. aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"

},

dryRun = false,(to check whether all steps are defined or any steps are missed defining, it will not execute the actual code, it will just look for undefined steps if any)

monochrome = true,(for readable format output/report)

tags="@smoke"(which tags do we need to run out of many scenarios and tags)

runnerTestNG- we will not use @RunWith tag bcz we will be having testNG.xml file

rerun plugin- "rerun:target/failedRerun.txt"

runnerFailed-

@CucumberOptions(

plugin ={"pretty",

"html:target/Cucumber.html",

"json:target/Cucumber.json",

"junit:target/Cucumber.xml",

"com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:",

"timeline:target/parallel-threads",

"rerun:target/failedRerun.txt"

},

dryRun = false,

monochrome = true,

features = {"@target/failedRerun.txt"} ,

glue= {"org.example.StepDefinition"},

tags="@smoke"

)

Utility: utility folder holds these classes:

configReader- we have created a object for propertied file from where we will fetch the data.

driverFactor- getDriver(initializing the driver object), initDriver(switch statement to select the browser from diff browsers) , quitDriver( quitting the driver instance)

retryListener- retryListerners implements IAnnotationTransformer(interface)

testNGRetry- implements IRetryAnalyzer(interface)

ThreadLocal<Integer>retryCount=ThreadLocal.withInitial(()->0);

private static final int maxRetryCount = 1;

pageInitializer- we have created

\*First- we have mentioned each page objects pages by protected keyword.

\*Second- we have created driver object for each step definitions files.

context- we have used methods like- set (), get(), remove(),contains(), clear() to manipulate contextData.

We have added hash map code with key value pair.

Under resource**:**

\*Features- for each stepDefinition file, one feature will be present. Which holds multiple scenarios.and each scenario holds multiple steps like Given , When, Then , And keyword.

\*Config.properties(dev, QA)- Set of user data which we will be fetching for the execution.(Username, password, url, expected output)

\*extent.properties- related to extent report generation(what type- PDF,SPARK,HTML). And screenshot path/dir, base64, system Info(Windows)

xmlFiles

pom.xml-->dependencies

extent-config.xml-->extent report generation(Chart location, base64, report name, theme)

log4j.xml--> related to logs configuration(date, time)

testNG.xml--> listeners, parallel execution(thread count),runnerClass(testNGRunner)

**6. What is group id and artifact id in maven?**

We can use pom.xml

**Group id:** we can use Project's organization/group name reverse domin format

**Ex:** com.google, org.selenium

Artifactid: Name of the actual project/module,unique

**Ex:** selenium-java, testing

Version: Version of the project

**7. What are the different types of error code in api Explain?**

**1**. 1xx — Informational Errors

* Meaning: The server has received the request but is still thinking.
* Examples:
  + 100 Continue: OK so far, keep going.
  + 101 Switching Protocols: Server will change to a different protocol.

2. 2xx — Success Responses

* Meaning: Everything worked!
* Examples:
  + 200 OK: The request succeeded.
  + 201 Created: Something new was created (like a new user or record).
  + 204 No Content: Request succeeded but no content to return.

3. 3xx — Redirection

* Meaning: The client must do something else to complete the request.
* Examples:
  + 301 Moved Permanently: Resource has moved to a new URL.
  + 302 Found: Resource temporarily at a different URL.

4. 4xx — Client Errors (User/Client made a mistake)

* Meaning: Something is wrong in the request (wrong data, bad token, missing parameter).
* Examples:
  + 400 Bad Request: The server couldn't understand the request (bad syntax).
  + 401 Unauthorized: Authentication failed (wrong or missing token).
  + 403 Forbidden: You are not allowed to access this resource.
  + 404 Not Found: Resource not found (wrong URL or ID).
  + 409 Conflict: Request conflicts with current state (like trying to create a user that already exists).

5. 5xx — Server Errors (Server made a mistake)

* Meaning: You did everything right, but the server failed.
* Examples:
  + 500 Internal Server Error: Server crashed or failed unexpectedly.
  + 502 Bad Gateway: Server got a bad response from another server.
  + 503 Service Unavailable: Server is down or overloaded.
  + 504 Gateway Timeout: Server took too long to respond.

**8. WAP find max and second max using array**

Ex: public class MaxAndSecondMax {

public static void main(String[] args) {

int[] arr = {10, 25, 40, 15, 5};

int max = Integer.MIN\_VALUE;

int 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);

}

}

**9. Explain testNg annotations?**

**Ans**:

@BeforeSuite: Runs **once before or after** the entire suite (useful for setup and cleanup).

@BeforeTest: Runs before any test methods in a <test> tag in your testng.xml.

@BeforeClass: Runs **once before** the first test method in the class.

@BeforeMethod**: Runs before each** @Test **method**

**@Test:**

@AfterMethod: Runs **after each** @Test method.

@AfterClass: Runs **once after** all test methods are done.

@AfterTest: Runs after all the test methods in the <test> tag.

@AfterSuite:

**10. How to handle dynamic elements?**

Using **Explicit Waits** to wait for elements to be visible or clickable.

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

WebElement element = wait.until(ExpectedConditions.refreshed(

ExpectedConditions.elementToBeClickable(By.id("submit"))

));

#### **Using Relative XPath (Avoid Absolute XPath)**

**Relative XPath** is more flexible and won't break if the structure of the page changes slightly.

**Using JavaScript Executor**

**Fluevent wait**

**11. How to avoid stelreferenceElementException?**

"The element you are trying to interact with is no longer attached to the DOM."(right click inspect html)

This can happen when:

* The page is refreshed.
* The DOM changes after an action (like Ajax or JavaScript reload).
* The element is removed and added again.
* The script will run the page is refreshed,page structure change ,element refrences will change
* Try to attempt try,catch .retry the operation again find the xpath

**12. How handle error code in api?(the question is wrong)**

**13. What is difference between selenium 3 and selenium 4?**

Selenium 4 we introduced relative locaters like above below,right left,near. Selenium 3 supports **full-page screenshots**, while Selenium4 particular screen shots

Selenium 4 allows opening a **new browser tab or window** easily using driver.switchTo().newWindow(), which was more difficult in Selenium 3.selenium 4 explixit wait ,implicit wait

**14. WAP input: abc123 Output: 123 extra number?**

public class AlpabetsNumbers

{

public static void main(String[] args)

{

// Step 1: Create Scanner object to take input

Scanner sc=new Scanner(System.in);

System.out.println("Enter a String:");

String str=sc.nextLine(); // e.g., str: "abc123"

// Step 3:Extract only numbers by removing non-digits String numbers=str.replaceAll("\\D","");//removes everything that is not a digit

//Step 4: Print the resulte

System.out.println("the number is :"+numbers);

}

}

**Output:**

Enter a String:

abc123

the number is :123

**15. Explain your project? Which domain?**

**16. What is methods in java?**

**methods** are blocks of code that perform a specific task.

**Why Use Methods?**

* Reusability: Write once, use many times.
* Organization: Break big tasks into smaller pieces.
* Readability: Easier to read, debug, and maintain.
* Code Management: Makes programs modular.

**Basic Syntax of a Method:**

returnTpe methodName(parameters)

{

//method body

//code to be executed

}

Types of Methods

Static method

Constructor

Instances

Abstract method

**17. What is method overriding and method overloading?**

Metod overloading: method overloading is nothing but method having the same name but differ in no of parameters, type of parameters, order of parameters. Method overloading is defined as same calss. method overloading is called compile time polymorphism or static methods,private ,final overload

Method overriding: method overriding is nothing but method having the same name, same no of parameters and type of data, same return type. Method overloading is defined as different class.mo is using inheritances IS A relastinonship.is called dynamic or run time polymorphism.final private not overriding

**18. Explain oops concept where we use in framework?**

| **OOP Concept** | **Where It's Used** |
| --- | --- |

|  |  |
| --- | --- |
| **Encapsulation** | - In **Page Object Model (POM)**: Each web page is a class with private elements and public methods  - Keeps element locators and actions safe and organized |

|  |  |
| --- | --- |
| **Inheritance** | - Common **base classes** for WebDriver setup, configuration, hooks (@Before, @After)  - Test classes inherit from base class to reuse code |

|  |  |
| --- | --- |
| **Polymorphism** | - Step Definitions in Cucumber can be **overloaded** or **overridden** for different scenarios  - Useful in utilities (e.g., click() method behaves differently for buttons vs links) |

|  |  |
| --- | --- |
| **Abstraction** | - Use of **interfaces** or **abstract classes** for WebDriverManager, browser utilities, etc.  - Hides complex browser handling logic |

**19. Write a java program for occurrence of each character.**

package Stringexamples;

import java.util.HashMap;

import java.util.Map;

public class Charavcterstring

{

public static void main(String[] args)

{

String sentence = "java";

Map<Character, Integer> name = new HashMap<>();

for (char ch:sentence.toCharArray())

{

name.put(ch,name.getOrDefault(ch,0)+1);

}

for (Map.Entry<Character,Integer>entry: name.entrySet())

{

System.out.println("the name is:"+entry.getKey()+ " = "+entry.getValue());

}

}

}

**20.Write a java program to find duplicate elements in a string array**

public class Dublicatestr

{

public static void main(String[] args)

{

String arr[]={"Apple","Banana","Apple"};

HashMap<String,Integer> map=new HashMap<>();

for(String str:arr)

{

map.put(str,map.getOrDefault(str,0)+1);

}

System.out.println(map);

for(Map.Entry<String,Integer> m1: map.entrySet())

{

if(m1.getValue()>1)

{

System.out.println("this element is dublicate :"+m1.getKey());

}

}

}

}

**Output:**

{Apple=2, Banana=1}

this element is dublicate :Apple

**21. Difference between set and list**

**List** maintains insertion order, allows duplicate values, and supports multiple null values.

**Set** does not maintain insertion order, allows only unique values (no duplicates), and permits only one null value

**22. Where we use collection in framework**

|  |  |  |
| --- | --- | --- |
| List | WebElements, Excel/CSV rows | Multiple buttons/links |

|  |  |  |
| --- | --- | --- |
| Set | Unique runtime values | Store unique error messages or IDs |

|  |  |  |
| --- | --- | --- |
| Map | Key-value test data, config properties | Username/password, settings |

|  |  |
| --- | --- |
| List<Map> | Cucumber DataTable |

**23. Difference between array and arrayList**

**Array:** array is fixed size data structure, array is index-based access, array methods available limited,faster for fixed size data,array can store primitives&non primitives(objects).

Arraylist:resizable or growable/shrink,only objects(Integar,String) wrapper class,arraylist also index based access,many methods are available like add(),remove(),removeall(),addall(),contains,slightly slower.

**24. What is basic authentication in api?**

Username and password as input they convert into base64 using postman.

**25. Explain Agile methodologies**

Scram,daily standup

Sprint planning

Refinement meeting

Can can board

Sprint retro

We can create Jira store

Epic ,story ,defect,bug work under agile

Reports srs summery,pdoduct design review,software test reivew

http test plan:

| **Section** | **What it means** |
| --- | --- |
| **1. Introduction** | Short info about the project and goals of testing |
| **2. Scope** | What features will be tested and what will NOT be tested |
| **3. Test Strategy** | How the testing will be done (manual, automation, tools) |
| **4. Test Environment** | Setup needed (browsers, OS, test data, servers) |
| **5. Test Deliverables** | What testing documents will be created (test cases, bug reports, etc.) |
| **6. Schedule** | Timeline for when testing activities will happen |
| **7. Resources** | Who will be testing (team members) |
| **8. Risks and Mitigation** | What problems might happen and how to solve them |
| **9. Entry and Exit Criteria** |  |

Throughput (performance testing) different types of throughput

Enlight is the product example:in throughput

Regression create module ,how much time will take performance test

We are following agile methodology,sprint will be for 2 weeks

**26. If I want to run particular test cases multiple times so it is possible to achieve it using testNg ?**

Using @DataProvider to Run Test with Different Inputs.

Test casee failed Iretryanalyzer

listners

Using invocationCount in the @Test Annotation

**27. What is data provider in testing**

A **DataProvider is annaotation dataprovider** in TestNG is a way torun the **same test multiple times with different sets of data.**data is read from excel

**28. What are the various listeners you have used in your project?**

ITestListener for Extent Report integration and screenshots

IRetryAnalyzer for retrying flaky tests

ISuiteListener if logging at suite level

**29. What is interface**

Interfaces is is like a class but it will have only the method declaration on in it, in interface we will not have the concrete method it we will have only the method declaration in the interface.it is 100% abstraction.

**30. Difference between interface and abstract class**

Interface class are declared using the keyword called “implements”

Abstract class are declared using the keyword called “abstract”.

Interface does not have conceret methods

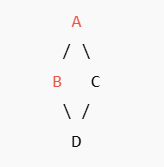
Abstract class can have concrete methods and non-concert methods.

Interface supports multiple inheritances

Abstract does not support multiple inheitances

**31. What diamond problem**

The diamond problem happens when a class inherits from two classes that both inherit from a common superclass — causing ambiguity about which path to take.  
It’s called a *"diamond"* because the inheritance diagram looks like a diamond shape:

Here, B and C inherit from A, and D inherits from both B and C.

**Classes cannot inherit from multiple classes**.

Java avoids the diamond problem by not supporting multiple class inheritance — a class can only extend one class (extends keyword).

**32. Which is best CSS Selector or xpath and why?**

Css selector

Speed is faster

Syntax is simple and clean

Readability -easier to read and write

**33. If have to mention specific condition or element to be visible** Which wait will be use?

Explicit wait

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

WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("loginButton")));

**34. Why constructor was used in framework?**

Constructors are mainly used for **initializing web elements**, especially when using **Page Object Model (POM).**

**Initialize WebDriver**: Page Object class can use the same WebDriver instance as your tests

**Avoid Code Duplication:** Initialize once and reuse.

**Maintain Clean Structure:** Keeps your Page Object classes independent and reusable.

**35. Explain pom.xml and where you use it**

pom.xml stands for Project Object Model

It's an XML file used by Maven

It manages the entire configuration of your project:  
 **dependencies:** Add libraries like Selenium, TestNG, Cucumber, WebDriverManager

**Plugins:** Use plugins for reports (Extent, Allure), surefire for running tests, etc.

**build settings:** Customize compile, test, and package steps

**Version control:** Manage versions of all libraries in one place

**Where we use it**

located at the root of your Maven project.

Used in Java frameworks (like Selenium with TestNG, Cucumber, etc.)

**36. How to find broken links and write code for it**

Does not return 200 reponse code,if a link return 200 reponse code working link(sucesss)

400,500(failure)

Code

public class BrokenLinks

{

public static WebDriver driver;

public static String Browser = "Chrome"; // Congif file

public static WebElement element;

public static void main(String[] args) throws InterruptedException

{

if (Browser.equals("Chrome"))

{ driver = new ChromeDriver();

}

else if (Browser.equals("Edge"))

{

driver = new EdgeDriver();

}

else if (Browser.equals("Firefox"))

{

driver = new FirefoxDriver();

}

driver.get("https://testautomationpractice.blogspot.com/");

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

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

for(WebElement e:linkelement)

{

String link=e.getDomAttribute("href");

if(link==null||link.isEmpty())

{

continue;

}

Try

{

URL url=new URL(link);

HttpURLConnection connection=(HttpURLConnection) url.openConnection();

connection.connect();

if (connection.getResponseCode()>400)

{ System.out.println(link+"broken link");

}

Else

{

System.out.println(link+"working link");

}

}

catch (Exception ex)

{

throw new RuntimeException(ex);

}

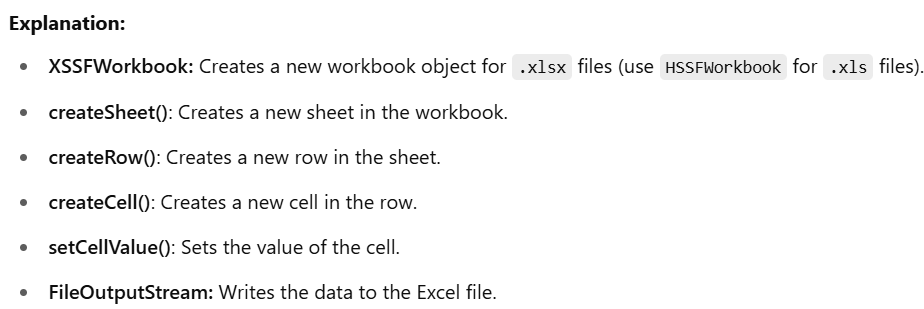
}

}

}

**37. How Read and write data in excel sheet?**

Apache POI is a Java library used to **read/write Excel files** (both .xls and .xlsx formats).



**38. There is a text box in which you have to type particularly any data into using selenium automation you can't use sendKeys() which are the other types into a particular textbox using selenium?**

Using JavaScriptExecutor

Using Actions Class

**39. How to run your automation?**

From TestNG XML File (in IDE like Eclipse/IntelliJ)

Using Maven Command (Terminal) Mv clean test

Using Cucumber Runner Class (**Cucumber with JUnit/TestNG,** just run the runner class:)

Through Jenkins or CI/CD Tool

**40. What are the challenges faced in automation?**

Suppose you're automating a shopping cart checkout:

* **Page loads slowly** → your script clicks “Add to Cart” before the button even appears → test fails.
* **Cart icon’s ID changes every time** → XPath built today will break tomorrow.
* **Popup window for payment** → your WebDriver needs to switch to the new windo

Some rules:

* Always build robust locators.
* Always use explicit waits.
* Keep your tests small, modular, and independent.
* Expect that scripts will break someday → easy maintenance is *essential*.

**41. What do you if you have a large test suit to execute in less time?**



**42. Complie time and run time polymorphism**

**Method Overloading** is when methods have the same name but differ in the number of parameters, order of parameters, or types of parameters, and are defined in the same class.

**Method Overriding** is when a method in a child class has the same name, same number of parameters, same types, and same order of parameters as a method in the parent class. It occurs in different classes with an inheritance (ISA) relationship.

**43. How to handle multiple windows?**

Get the main window handle (parent).

Click something that opens a new window.

Get all window handles.

Switch to the child window.

Do actions in the child window.

Switch back to the parent if needed.

**44. Explain 401 and 500 error code in api**

401 Unauthorized: Missing/invalid login info.

|  |  |  |  |
| --- | --- | --- | --- |
| 500 | Internal Server Error: Bug/server crash |  |  |

**45. What page object model?**

page Object Model is a design pattern used in test automation to create a separate Javaclass for each web page of your application.

**Each class:**

* Represents a single page.
* Stores all locators (elements) and methods (actions) related to that page.

**Advantages**

Clean and readable code  
Easy to maintain (change in UI = update only one class)  
 Reduces code duplication  
Helps you follow good programming practices (like OOP)

**46. How do you deal with flaky test cases in automation?**

A test that sometimes **passes** and sometimes **fails**, even when there’s no change in the app or code.

**47. SQL joins**

Joins are used to combine rows from **two or more tables** based on a related column

.

**48. Exception handling**

**49.How to reverse the words in the string? eg: I like program language java to java language program like i**

public class Charreverse

{

public static void main(String[] args) {

String name = "I like program language java";

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

System.out.println("The original string is: " + name);

System.out.print("Reversed words: ");

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

System.out.print(words[i] + " ");

}

}

}

**Output:**

the string is:I like program language java

the reverse word is:java language program like I

**50.Using the hashmap, how to find no of occurrences of a character in the string?**

import java.util.HashMap;

import java.util.Map;

public class CharFrequency

{

public static void main(String[] args) {

String str = "hello world";

// Create an empty HashMap to store char -> count

HashMap<Character, Integer> map = new HashMap<>();

// Loop through each character

for (char ch : str.toCharArray()) {

map.put(ch, map.getOrDefault(ch, 0) + 1);

}

// Print character frequencies

for (Map.Entry<Character, Integer> entry : map.entrySet()) {

System.out.println(entry.getKey() + " = " + entry.getValue());

}

}

}

**Output:**

**the name is:a = 2**

**the name is:v = 1**

**the name is:j = 1**

**51.Can the main method be overloaded,**

Yes: Java allows multiple methods with same name but different parameters

**52.can the main method override**

**No:** Because main is static, and static methods can’t be overridden

**53.Which access modifiers can be used in an interface?**

Interface can be public or default (no static).  
 **Inside interface:** methods can be public, default, static, private (since Java 8).

**54.Access modifiers in Java and its scope?**

The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

**There are four types of Java access modifiers:**

**Private:** The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

**Default:** The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

**Protected:** The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.

**Public:** The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

**55.What is the difference between scenario and scenario outline?**

| Feature | Scenario | Scenario Outline |
| --- | --- | --- |
| Purpose | Run test one time with fixed data | Run test multiple times with different data sets |
| Data Input | Hardcoded inside the steps | Uses placeholders (< >) and provides data through Examples table |
| Execution | Single execution | Executes once for each row in Examples table |
| Syntax Keyword | Scenario: | Scenario Outline: (or Scenario Template:) |
| When to Use | When you have only one test case | When you need to test same flow with different inputs (Data-driven testing) |
| Examples Table | Not used | Must be used |
| Example | Test login with specific user | Test login with multiple users and passwords |

**56.Can the main method be overloaded?(repeated)**

**57.Can the main method be overridden?(repeated)**

**58.The client comes and asks you to automate a feature. So how will you proceed? Should you use an existing framework or create a new one if you do not have one in place?(confusing)**

**59.What is a feature file in the Cucumber framework?**

A **feature file** is a **plain text file** (with. feature extension) that **contains high-level test scenarios** written in **Gherkin language.**

**60.What is the step definition file used for cucumber**

A step definition file is a Java class where you writethe actual code to execute the steps defined in the. feature file.

It acts as a bridge between Gherkin steps (Given/When/Then) and the automation code (like Selenium/WebDriver logic).

**61.How can you achieve that if you are required to make changes to your selenium files in a maven project?(confusing)**

**62.The syntax for mouse over text in Selenium**

To perform mouse hover in Selenium,used Actions class with moveToElement() and perform() methods.

/ Create an Actions object

Actions actions = new Actions(driver);

// Find the web element to hover over

WebElement element = driver.findElement(By.id("menu"));

// Perform mouse hover

actions.moveToElement(element).perform();

**63.Can I write: WebDriver driver = new WebDriver()?**

No, you **cannot** write WebDriver driver = new WebDriver(); in Selenium because **WebDriver is an interface**, not a class, and interfaces cannot be instantiated directly.

**64.Can I access a parent class method using a child class object?**

Yes, you **can access a parent class method using a child class object** in Java, as long as the method is **not private** and is either **public,** **protected,** or **default** (package-private) within the same package.

**65.can you execute failed test cases without manual intervention**

Yes, you can execute failed test cases without manual intervention by using features like TestNG’s IRetryAnalyzer, leveraging CI tools like Jenkins for automatic retries, and ensuring a robust retry mechanism is in place

**66.Architecture for Selenium3 and Selenium4**

**67.What are the factors that we consider when choosing an automation tool?(confusing)**

**68.What is an Action class & asked to mention a few methods of action class**

### **Common Actions that can be performed using Action Class:**

1. **Mouse Movements**: Moving to an element, dragging, or clicking and holding.
2. **Keyboard Actions**: Pressing specific keys, key combinations, or releasing keys.
3. **Mouse Events**: Clicking, double-clicking, right-clicking, etc.

**69.Was I asked to write Java code for handling the child window, performing some operations, and switching back to the parent window?**

To handle child windows in Selenium and switch between parent and child windows, you can follow these steps:

**Steps:**

1. **Get the handle of the parent window.**
2. **Click on the element that opens the child window.**
3. **Get the handles of all open windows.**
4. **Switch to the child window using its handle.**
5. **Perform operations on the child window.**
6. **Switch back to the parent window using its handle.**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.Iterator;

import java.util.Set;

public class HandleChildWindow {

public static void main(String[] args) {

// Set up WebDriver (assuming ChromeDriver is set in the system path)

WebDriver driver = new ChromeDriver();

// Navigate to the parent window's page

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

// Get the parent window handle

String parentWindowHandle = driver.getWindowHandle();

System.out.println("Parent Window Handle: " + parentWindowHandle);

// Perform an action that opens a new child window (e.g., clicking a link)

WebElement link = driver.findElement(By.id("childWindowLink"));

link.click();

// Get the handles of all open windows

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

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

// Switch to the child window

String parent = iterator.next(); // Parent window

String child = iterator.next(); // Child window

driver.switchTo().window(child); // Switch to the child window

System.out.println("Switched to Child Window");

// Perform operations on the child window (e.g., find an element and interact)

WebElement childElement = driver.findElement(By.id("childElement"));

childElement.click();

// Switch back to the parent window

driver.switchTo().window(parent);

System.out.println("Switched back to Parent Window");

// Perform operations on the parent window if needed

WebElement parentElement = driver.findElement(By.id("parentElement"));

parentElement.click();

// Close the driver

driver.quit();

}

}

* Get Window Handles: driver.getWindowHandles() returns a set of all open window handles.
* Switch to a Window: Use driver.switchTo().window(windowHandle) to switch between windows.
* Window Handle Management: You can use the parent handle to switch back once you are done with the child window.
* Parent window handle: This is stored before the child window is opened, and you switch back to it after interacting with the child window.
* Window handle switching: The switchTo() method is used to switch between the parent and child windows.
* Window handles set: The set of window handles returned by getWindowHandles() needs to be iterated because the order is not guaranteed.

**70. What is Velocity in Agile Testing?**

In Agile Testing, Velocity is a key metric used to measure the amount of work a team can complete during a single Sprint (a fixed-length iteration of work, typically 1 to 4 weeks). It helps track the team's performance and progress by quantifying the effort required to complete user stories or tasks.

**What Does Velocity Measure?**

Velocity measures the number of story points (or effort estimates) a team completes in a sprint. It provides an indication of the team’s capacity and is typically calculated based on user stories.

**Story Points**

Story points represent the relative complexity or effort required to complete a user story. These points are assigned during sprint planning and are used to gauge how much work can be handled in a sprint.

**Velocity** = The number of story points (or user stories) completed in a sprint.

It helps teams understand their **capacity**, forecast future sprints, and plan releases more accurately.

**Why Is Velocity Important?**

* Helps in **sprint planning** – estimate how much work to take on
* Aids in **release planning** – predict when all backlog items can be delivered
* Tracks **team performance** over time
* Detects **bottlenecks** or productivity dips

**71.What is a static keyword in Java?**

**72.Purchase order in amazon-selenium code**

**73.All waits-synchronization**

There are three main types of waits in Selenium:

1. Implicit Wait
2. Explicit Wait
3. Fluent Wait