**1)How do you convert string to Integer?**

The parseInt() method is used to convert a string to a primitive int. If the string cannot be parsed as an integer, it throws a NumberFormatException.

public class StringToInteger {

public static void main(String[] args) {

String str = "123"; // String to be converted to integer

int number = Integer.parseInt(str); // Converts String to int

System.out.println("Converted Integer: " + number);

}

}

**VICE VERSA:**

public class IntegerToString {

public static void main(String[] args) {

int num = 123; // Example integer

String str = String.valueOf(num); // Convert int to String

System.out.println("Converted String: " + str);

}

}

**If I declare a variable in an interface what are the default properties which variable has - Public, static and void**

In Java, when you declare a variable in an interface, it implicitly has certain default properties. These properties are:

public: The variable is accessible from anywhere.

static: The variable belongs to the interface itself (not to any instance).

final: The variable's value cannot be changed after it is initialized.

**Can you extend a final class**? - We can't extend a class

A final class in Java is a class that cannot be subclassed (inherited). When you declare a class as final, it means that no other class can extend it. This is often done for classes that should not be modified or for security reasons, to prevent subclassing and ensure that the class's behavior remains unchanged.

**Can we overwrite a static method** - No we can't overwrite a static class

Overriding happens when a subclass provides a specific implementation of a method that is already defined in its superclass, and the method is invoked on an object of the subclass. This is based on runtime polymorphism.

Static methods are associated with the class itself, not with instances of the class. Since static methods are resolved at compile-time (based on the reference type), they cannot be overridden in the same way instance methods can. Instead, if a subclass defines a static method with the same signature as a static method in the superclass, it will hide the superclass's method, but it is not true overriding.

**Can you access instance variable from Static method - That cannot be accessible**

No, you cannot directly access an instance variable from a static method in Java.

Reason:

Static methods belong to the class itself, not to any particular instance of the class. They are accessed using the class name and can be called without creating an instance of the class.

Instance variables, on the other hand, belong to instances of the class (i.e., objects). To access an instance variable, you need to have an object reference.

how can we access??- we have to create a object to class

**How to convert arrray to List?? - We have arrray through list**

Arrays.asList(): Converts a non-primitive array into a fixed-size list (not fully mutable).

new ArrayList<>(Arrays.asList()): Converts an array to a mutable ArrayList.

How to sort a method.

Have you worked on hashmaps

String -

apple,apple,apple,orange,orange,mango

Pick the most repitive word and how many times it is repeating?? (You can use any logic)

For Programming: -

Sort an array without using any inbuilt method

**What are the frameworks in Selenium**

Data-Driven Framework: Use external data sources (Excel, CSV) to drive test execution.

Keyword-Driven Framework: Use keywords to represent actions, allowing non-technical people to write tests.

Hybrid Framework: Combination of multiple frameworks (e.g., Data-Driven + Keyword-Driven).

Page Object Model (POM): Organize tests by creating page objects that represent different web pages.

BDD Framework: Use Gherkin syntax (Cucumber) for test scenarios written in natural language.

TestNG: A powerful test framework that integrates with Selenium and supports parallel execution.

JUnit: A simpler testing framework often used with Selenium.

Selenium Grid: Enable parallel execution of tests across multiple machines and browsers.

Have you in worked in POM

**What are the cucumber options we pass - Feature, step definition, dry run..**

Option Description Example

features Path to feature files features = "src/test/resources/features"

glue Path to step definitions glue = "com/example/stepdefinitions"

tags Filters scenarios based on tags tags = "@SmokeTest"

plugin Report formats (HTML, JSON, etc.) plugin = {"pretty", "html:target/cucumber-reports"}

monochrome Makes console output cleaner monochrome = true

strict Ensures all steps are defined (fails if any step is undefined) strict = true

dryRun Verifies that all steps have corresponding definitions (without running tests) dryRun = true

name Filters scenarios based on name or regex name = "Login Functionality"

snippetType Style for step definition snippets snippetType = SnippetType.CAMELCASE

stepNotifications Prints detailed step notifications to the console stepNotifications = true

**If you want to execute only few feature file .how will you do it in cucumber..**

Specify specific feature files using the features option in the @CucumberOptions annotation.

Use wildcards in the path to match multiple feature files (e.g., Login\*.feature).

Filter using tags to only execute specific scenarios or feature files with certain tags (e.g., @SmokeTest).

**what are the diff ways we can switch to I frame - By Index, By Name or By Element**

**How do you mouse over in selenium - we can use action class (Move to element)**

WebElement elementToHover = driver.findElement(By.id("menuItem"));

// Create an instance of Actions class

Actions actions = new Actions(driver);

// Perform mouse hover action

actions.moveToElement(elementToHover).perform();

Worked on TestNG:

**How you do parallel execution?**

test level,suite level

parallel="classes": Runs each test class in parallel.

thread-count="2": Specifies the number of threads to be used for parallel execution (2 threads in this case).

API Automation?

**What is an object mapper?**

An ObjectMapper is a class provided by the Jackson library, which is a widely used library for processing JSON in Java. The main role of ObjectMapper is to convert Java objects to their JSON representation and vice versa. This process is called serialization (Java object to JSON) and deserialization (JSON to Java object).

Key Functions of ObjectMapper:

Serialize (Java Object to JSON): Converts a Java object into a JSON string.

Deserialize (JSON to Java Object): Converts a JSON string into a Java object.

Reading and Writing JSON: It can read JSON from various sources (like a file, a string, or an InputStream) and can write JSON to different output destinations (like a file, a string, or an OutputStream).

**Have you worked on hampcrust matchers?**Hamcrest is a library for writing matchers (or assertions) that allow you to write readable and expressive tests in Java. It provides a set of matchers to validate various conditions in your code, such as equality, null checks, collections, etc. Hamcrest is often used in conjunction with JUnit for asserting test conditions.

A matcher is essentially a condition that evaluates whether a specific object or value meets some criteria (like "is equal to", "is not null", "contains a specific value", etc.).

**Can abstact class can implement a interface**

Yes, an abstract class can implement an interface. When an abstract class implements an interface, it is required to provide implementations for all the methods defined in the interface unless the abstract class itself is abstract (which means it can leave the implementation of those methods to its subclasses).

**Does Java Support Multiple inheritance??**Ans - Java disallows multiple inheritance to avoid the complexity and ambiguity associated with it, particularly the "diamond problem," where a class inherits from two classes that have a common ancestor, leading to conflicts in the inheritance of methods.

**Can a interface extend multiple interfaces**

An interface in Java can extend multiple interfaces, which allows for more flexible and modular design of your code. It is an example of Java's support for multiple inheritance through interfaces.

Common String Functions

**What is Split Function Written**

The split() method in Java is used to split a string into an array of substrings based on a specified delimiter (or regular expression). It's a commonly used method for breaking down a string into smaller parts.

Regular Expressions: The delimiter passed to the split() method can be a regular expression. So, special characters (like ., \*, +, etc.) must be escaped with a backslash \\ (e.g., \\. for a dot).

Limit: When you provide a limit, it can affect the resulting array:

If limit > 0: The split will stop after limit parts.

If limit = 0: The string will be split as much as possible, but trailing empty strings will be discarded.

If limit < 0: The string will be split as much as possible, and empty strings will be included in the result.

How do u convert a String to Integar

How do u convert a Array to List

Write a Program where method should take a number and say whether it is a amstrong number or not

Collections

How do you sort a list, What does Sort method take as parameter

How can you call a method into List can you pass into it

what are generics in Java

How do you loop a list

Frameworks in Selenium

How do you rewrite excel using Test Ng

Which Anotation used to rewrite excel

How do u do parllel execution using Test NG

How do u do a mouse over using Selenium

What Kind of Reporting Implemented in your Project

What type of Xpath you have written; Examples of xpath you have written

Suppose you are not able to click a button in Xptah, what is the other way to do it.

How will you create a itarator

1. Difference between set and list

2. If .click() method is not working, how will you handle the situation.

3. How do you retrieve data from web table

**4. Return type of @data provider**

Object[][] - A two-dimensional array where each element represents a set of test data for a single invocation of the test method.

Iterator<Object[]> - An Iterator that returns the test data in the form of an Object[].

Iterable<Object[]> - An Iterable that returns the test data in the form of an Object[].

5. How will you handle frames

6. How will you handle multiple windows

**7. Return type of window handles--set of Strings**

The Set<String> represents a collection of unique elements, so it ensures that the window handles are unique.

8. How do you do parallel execution using testng--

9. Can an interface extend multiple interface

10. Can an abstract class implement interface

**11. What does instance Of() used for**

In Java, the instanceof keyword is used to test whether an object is an instance of a specific class, subclass, or implements a particular interface.

It is a type comparison operator that checks if an object is of a given type (i.e., class or interface) at runtime.

12. How do you send data from testng--

13. How do you sort a list

14. What is Wrapper class

15. How will you convert a string to an integer

**16. Any other way to convert string to an       integer other than parseInt()**

Integer.parseInt(): The most common method that returns a primitive int.

Integer.valueOf(): Returns an Integer object.

new Integer(): Deprecated constructor (avoid using it).

Scanner: Can parse strings to integers using nextInt().

17. Methods in list

18. Methods in string

**19. What actually is a webdriver**

In Selenium, a WebDriver is an interface that provides a way for developers and testers to interact with web browsers programmatically. It allows automation of web applications for testing purposes. WebDriver provides a simple, object-oriented interface to interact with the browser, simulating user actions like clicking, typing, navigating, and verifying page elements.

**20. What actually is a Chromedriver()**

ChromeDriver is a class in Selenium that provides the WebDriver interface to control the Google Chrome web browser. It is an implementation of the WebDriver interface specifically designed for Google Chrome.

**21. How will you retrieve key value pair using Hashmap--get() method.**

22. Do you know iterator.How will you use iterator

23.How to Switch to Frame ?

24.How to Handle multiple window ?

25.xpaths in Selenium

26.String functions

27.HashMap

28.Collection

29.List and Set methods

30.Frameworks in Selenium

31.POM

32.How to run Testng

33.What is split method do?

**34.What is parameters?**

In programming, parameters are variables used to pass information into functions or methods. They allow you to provide data to a function or method so it can perform its task based on that input.

**35.What is listeners?**

In TestNG, listeners are used to track and modify the behavior of tests. They are implemented using interfaces provided by TestNG. You can use listeners to perform operations such as logging, taking screenshots, updating reports, or performing clean-up tasks before or after a test is executed.

36.How to handle web tables?

37.What are the List methods available?

38.What is Maven?

**39.How to run testng.xml through pom.xml?**

To run testng.xml through pom.xml (Maven), you need to configure the Maven Surefire Plugin in your pom.xml file. The Surefire Plugin is responsible for running tests in a Maven project, and you can configure it to use TestNG and specify your testng.xml file for running the tests.

**40.How to access values in Hashatable?--get() method**

41. Description of framework

**42. Alternate of .click()**

Actions class: A flexible way to simulate clicks.

JavaScriptExecutor: Executes a click action via JavaScript, bypassing the WebDriver click behavior.

sendKeys(Keys.ENTER): Simulates pressing the Enter key on the element.

Robot class: A lower-level way to simulate mouse clicks.

43.Methods of String Class

44. TestNg Plugin for Devops

45. automation for web table

46. wrapper class

47. Multiple inheritance in interface

48. can abstract class extend Interface ?

**49. Dynamic xpath--functions**

**50. Return type of data provider--object[][]**

51. How to handle Frame

52. Hash Map