**Interview Questions**

1. **Differences between Static and Non Static**

Static:

Any member which is prefixed with static keyword is known as Static member of a class

Non Static:

Any member which is not prefixed with static keyword is known as Non Static member of a class

1. **Differences between Static and Non Static members/block**

Static:

1.Any member which is prefixed with static keyword is known as Static member of a class

2.Static is a single copy

3.Static is always associated with class

4.All static members will get stored inside "Static Pool Area"

5.We can access static members with in a same class directly and for different class we can access with the help of class name.

Non Static:

1.Any member which is not prefixed with static keyword is known as Non Static member of a class

2.Non Static is a multiple copy

3.Non Static is always associated with object

4.All non static members will get stored inside "Heap Area"

5.We can access non static members inside static member of same class with the help of object creation

1. **Selenium coding level Architecture**

Search context

TakesScreenshot Web Driver JavaScriptExecutor

Remote WebDriver

ChromeDriver EdgeDriver SafariDriver FirefoxDriver

Web Driver: It is an Interface present in selenium Package. In this interface we have 11 abstract methods used to perform Browser related actions. It is core interface of Selenium.

get(String Url)

getTitle()

getCurrentUrl()

close()

quit()

manage()

navigate()

getWindowHandle()

getWindowHandles()

getPageSource()

switchTo()

Search Context: : It is an Interface present in selenium Package. In this interface we have 2 abstract methods used to identify web elements.

findElement(By Class obj)

findElements(By Class obj)

TakesScreenshot: It is an Interface present in selenium Package. In this interface we have 1 abstract method which is used to take screenshot.

getScreenshotAs()

JavasciptExecutor: It is an Interface present in selenium Package. In this interface we have 2 abstract methods which is used to write java script code.

executeScript()

executeAsyncScript()

Remote WebDriver: It is the only child implementation class for all the Interfaces

Browser Specific Classes: ChromeDriver, EdgeDriver, SafariDriver, FirefoxDriver

**4.Program for Missing Elements in an Array**

public class MissingNumbersInAnArray {

public static void main(String[] args) {

int a[]= {11, 13, 2, 15, 5, 7, 9, 10, 16, 19};

int[] arr= new int[20];

for (int i : a) {

arr[i]=1;

}

for(int i=1;i<arr.length;i++)

{

if(arr[i]==0)

{

System.out.println(i);

}

}

}

}

**5.How to Handle the Window**

With the help of switchTo(), getwindowHandle() and getwindowHandles() I can able to handle the window.

**6. Do you know SQL**

SQL is also known as Sequal Query Language. When user Interface want to communicate with Database Server, we go for SQLlanguage.

**7. Program for Static and Non Static block**

public class BlockExample {

static {

// Static block

System.out.println("Static block initialized");

// Can initialize static variables here

}

{

// Non-static block (instance initializer block)

System.out.println("Non-static block initialized");

// Can initialize instance variables here

}

public BlockExample() {

System.out.println("Constructor called");

// Constructor code

}

public static void main(String[] args) {

System.out.println("Main method called");

BlockExample obj1 = new BlockExample();

BlockExample obj2 = new BlockExample();

}

}

8. **If you want to do cross-browser**

**in xml, what is your approach**

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

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

<suite thread-count="2" parallel="tests" name="Suite">

<test name="TestRunner\_1">

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

<classes>

<class name="com.vtiger.crm.orgtest.CreateOrgTest"/>

<!-- <class name="com.vtiger.crm.contacttest.CreateContactTest"/> -->

</classes>

</test> <!-- Test -->

<test name="TestRunner\_2">

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

<classes>

<class name="com.vtiger.crm.orgtest.CreateOrgTest"/>

<!-- <class name="com.vtiger.crm.contacttest.CreateContactTest"/> -->

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

And in script level we do changes

@Parameters("BROWSER")

@Test

Public void createConTest(String browser)

{

if(browser.equalsIgnoreCase("chrome"))

{

WebDriverManager.chromedriver().setup();

driver=new ChromeDriver();

}

if(browser.equalsIgnoreCase("edge"))

{

WebDriverManager.edgedriver().setup();

driver=new EdgeDriver();

}

}

9. **Write a selenium script to search your name in Google search text and get all the suggestion sort in ascending order and then get the 3rd Index**

public class EnterNameInGoogleSearchAndFindThirdIndex {

public static void main(String[] args) {

WebDriver driver=new ChromeDriver();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

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

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

driver.findElement(By.xpath("//textarea[@aria-label='Search']")).sendKeys("Kalpana");

List<WebElement> alldata = driver.findElements(By.xpath("//div[@id=\"Alh6id\"]/descendant::ul/li"));

ArrayList<String> al=new ArrayList<String>();

for (WebElement data : alldata) {

al.add(data.getText());

System.out.println(data.getText());

}

System.out.println("-------------------------------");

Collections.sort(al); // sort in ascending order

for (String info : al) {

System.out.println(info);

}

System.out.println("-------------------------------");

System.out.println("Third Index Data "+al.get(3));

driver.quit();

}

}

10.**Wrie a Java program to Reverse a string “Java is Wonderful Language**”

public class ReverseTheString {

public static void main(String[] args) {

String s1=" Java is Wonderful Language ";

String s2="";

for(int i=s1.length()-1; i>=0; i--)

{

s2=s2+s1.charAt(i);

}

System.out.println(s2); // egaugnaL lufrednow a si avaJ

}

11. **Suppose you are have written 100 test scripts and in that 30 test cases get failed so how you will learn all the failed testscripts using particular testng .XML file**

In the test-output Folder we have testng-failed.xml file

In this file all the failed test cases will be appeared

If we want to run only failed test cases we just run this testng-failed.xml file

12. **Can we override Static methods**

No

13. **Can we overload static method**

Yes we can

14. **How you will create new Tab**

driver. switch To().newWindow(windowType.TAB);

15. **Difference between this and super keyword**

|  |  |
| --- | --- |
| **this** | **super** |
| The current instance of the class is represented by this keyword. | The current instance of the parent class is represented by the super keyword. |
| In order to call the default constructor of the current class, we can use this keyword. | In order to call the default constructor of the parent class, we can use the super keyword. |
| It can be referred to from a static context. It means it can be invoked from the static context. | It can't be referred to from a static context. It means it cannot be invoked from a static context. |
| We can use it to access only the current class data members and member functions. | We can use it to access the data members and member functions of the parent class. |

16. Difference between implicit wait and explicit wait.

**ImplictWait:**

It is a dynamic wait, implicit wait is also called as global wait, which means we will use one time and it will effect or work throughout the project.

* Imlicit wait work with findElements() & findElement().
* If element is found then it will return the address,but if element is not found then it will again return back to findElement () & findElements() with every 0.5 seconds .
* If Impicit time is over and element is not found then it will give NoSuchElement Exception.

**ExplicitWait:**

An explicit wait can be implemented using the WebDriverWait class in conjunction with ExpectedConditions. The WebDriverWait waits for a certain condition to occur before continuing with the execution of the script.

### Components of Explicit Wait

1. **WebDriverWait**: This is used to specify the maximum amount of time to wait.
2. **ExpectedConditions**: A set of predefined conditions provided by Selenium, such as visibility, clickability, presence, etc.

### Common Expected Conditions

Here are some commonly used expected conditions:

* presence\_of\_element\_located(locator): Waits until the element is present in the DOM.
* visibility\_of\_element\_located(locator): Waits until the element is visible on the page.
* element\_to\_be\_clickable(locator): Waits until the element is clickable.
* text\_to\_be\_present\_in\_element(locator, text): Waits until the text is present in the specified element.

17. **What is fluent wait and when we go for it**

Fluent wait is another type of wait provided by Selenium WebDriver that is more flexible than explicit wait. It allows you to define the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition. Additionally, it allows you to specify the types of exceptions to ignore while waiting for the condition.

**When to Use Fluent Wait**

Fluent wait is particularly useful in the following scenarios:

1. **Dynamic Content**: When dealing with elements that load at different times, or content that changes dynamically.
2. **Polling at Regular Intervals**: When you need to repeatedly check for a condition within a specified timeframe, rather than continuously.
3. **Ignoring Specific Exceptions**: When you want to handle specific exceptions (like NoSuchElementException) while waiting for an element to be present or interactable.
4. **Complex Conditions**: When you have more complex conditions that need to be met for an element to be ready for interaction.

18. What review commands have you given for the test scripts

**Commands for Reviewing Test Scripts**

1. **Check Element Locators**:
   * Ensure that element locators (ID, XPATH, CSS selectors, etc.) are correctly identified and used.
   * Verify the stability of the locators to avoid breakages with minor UI changes.

**Validate Wait Strategies**:

* Ensure appropriate wait strategies (explicit, implicit, fluent) are used based on the scenario.
* Avoid using arbitrary sleep statements; prefer explicit or fluent waits.

**Verify Exception Handling**:

* Ensure proper exception handling is in place to deal with unexpected issues.
* Check that specific exceptions are caught and handled gracefully.

**Check for Assertions**:

* Validate that appropriate assertions are in place to verify the expected outcomes of the test.
* Ensure the assertions provide meaningful error messages for better debugging.

**Review Code Structure and Readability**:

* Ensure the code is well-structured and follows consistent naming conventions.
* Check for the use of functions or classes to modularize the code

19.How do you perform keyboard functions and mouse action

Performing keyboard functions and mouse actions in Selenium WebDriver can be accomplished using the Actions class. The Actions class provides methods to handle complex user interactions such as clicking, double-clicking, drag and drop, and sending keyboard inputs.

### Keyboard Functions

#### Sending Keys to an Element

To send keys to an element (like filling out a text field), you can use the send\_keys method:

#### Using Actions Class for Keyboard Events

For more complex keyboard interactions, use the Actions class:

### Mouse Actions

#### Clicking and Double-Clicking

You can perform click and double-click actions using the Actions class:

#### Right-Clicking (Context Click)

To perform a right-click action:

#### Drag and Drop

To perform drag and drop actions:

#### Clicking and Holding

To click and hold an element:

#### Hovering Over an Element

To hover over an element:

19. Write an xpath for suggestions in google browser

//ul[@role='listbox']/li[@role='presentation']//span

20. In how many ways exception can be handle

### . ****Try-Catch Block****

The most common way to handle exceptions is by using a try-catch block. This involves wrapping the code that might throw an exception in a try block and handling the exception in the catch block.

### ****Multiple Catch Blocks****

You can use multiple catch blocks to handle different types of exceptions separately.

### ****Finally Block****

The finally block is used to execute code that must run regardless of whether an exception is thrown or not. It is typically used for cleanup activities.

### ****Try-With-Resources****

Introduced in Java 7, the try-with-resources statement ensures that each resource is closed at the end of the statement. It is used with resources that implement the AutoCloseable interface.

### ****Throwing Exceptions****

You can throw exceptions using the throw keyword. This is useful for custom exception handling and propagating exceptions

### ****Catching Multiple Exceptions in a Single Catch Block****

Java 7 introduced multi-catch blocks, allowing you to catch multiple exception types in a single catch block.

16. **Difference between functional and exploratory testing**

**Functional testing**: Testing each and every module following with the components with multiple sets of data is known as functional testing.

Documentation is required.

Here we do perform only component level testing.

**Exploratory testing:** when requirements are not clear and there in no requirements, will follow the similar kind of application is known as exploratory testing.

Documentation is not required.

Here we do perform complete testing

17. **When you will do exploratory testing**

**Exploratory testing:** when requirements are not clear and there in no requirements, will follow the similar kind of application is known as exploratory testing.

Documentation is not required.

Here we do perform complete testing.

**18. What all exceptions gone through in your project**

Stale element reference exception, NoSuchElementException, NullPointerException, IllegalStateException, IllegalArgumentException, SQLException, FileNotFoundException,TestNGException, MethodMatcherException,AlerUnhandledException..etc,

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

**Abstract Class:** In abstract class contains complete methods as well as incomplete methods and also have constructor, Any method is incomplete in class then class is declared as abstract class, if the class is declared as abstract class we cannot create the object of that class, we can complete the incomplete methods in any level of subclass as per sub class requirement.

**Interface:** In interface by default methods are public abstract and variables are an achieve public static final, in interface we can achieve hundred percent abstraction. Interface is a type definition block here we cannot create the object of it because by default methods are public abstract.

**20. Can we create object in abstract class and interface**

**Abstract class:** In abstract class we cannot create the object

**Interface:** In interface also we cannot the create the objects