**CORE JAVA**

1. **throw and throws keyword in Java** – **throw** an exception for business need this is used – **throws** declare an exception in method signature, code have any exception it is handled by try and catch block or declare using throws keyword – indication that code might get exception
2. **what, why and advantages of using POM in your project –** Page Object Model is a Design Pattern in Selenium – basic foundation to build a framework – Increase productivity, reduce redundancy, supports Multi Lingual application, supports for better reporting, and maintain uniformity.
3. **Why main method static in Java** – Static are belongs to the classes not for an individual object – if main method will not be a static it is applicable for all the object and it is not acceptable by JVM to calls.
4. **Constructor** – it is a method which initialize the object – same name as class name and object get initialize when a class is instantiates with a new keyword.
5. **Constructor Chaining** – process of calling one constructor of a class from another constructor – **this** keyword to call the current class – **super** keyword to call from parent class
6. **Difference between Constructor and Method –** initializing the object state – no return type – invoked implicitly – if no constructor java will compile with default constructor – constructor name should be same as class name / exposing the objects behavior – method have return type – invoked explicitly
7. **Set –** set is an interface that extends the collection interface – does not allows the duplicate – Array list is the implementation class of the list.

Implementation class of the set are

**HashSet** – maintain the random order

**TreeSet** – maintains the ASCII order

**LinkedHashSet** – maintains the insertion order

1. **String in Java –** String is a nonprimitive data type – final class – storing the text with double quotes – contains methods that can perform certain operations on strings.
2. **Create a String object –** to create a String object using new operator or using String Literal
3. **String Pool –** special type of memory maintained by JVM – store unique string objects – assign same string literal to different String variables JVM stores only one copy of the string object – String variable will start referring the String object.
4. **Access Specifier –** Java gives various access specifier to set access level for classes, variables, methods, constructors – public(everywhere), private(with in the class), default(with in the package), protected – access in control depending on our requirements
5. **Object oriented Programming –** based on object rather than functions and procedures. Individual objects grouped in to classes. It implements real world entities like Inheritance, polymorphism, etc in to programming. It is also binding data and code together.

* Simplicity in solving complex problems
* Reducing the redundancy
* Allows data hiding
* Problems can be divided in to different parts making it to simple to solve
* Concept of polymorphism gives flexibility to the program by allowing the entities to have multiple forms

1. **Inheritance –** It is a special object oriented concept where a sub class inherit from the parent class (methods, fields) – Single, Multiple and Multilevel – Reusability and maintenance – **Extends**
2. **Polymorphism -**  object oriented concept where single action can be transformed into multiple form – **Static** Method Overloading (Compile time) – same multiple method name with different arguments – **Dynamic** Method Overriding (Run time) – child class can redefine method present in the base class, same name as well as the same signature
3. **Encapsulation** – Binding the data and the code in to a single unit. It is also allows data hiding as the data specified in one class is hidden from the other classes.
4. **Abstract class -** It consists of a abstract methods. Methods are declared but not defined. It these methods are exclusively used in sub class, they need to be exclusively defined.
5. **ArrayStoreException -**  Occurs when try to store non- suitable element in an array object.
6. **ArrayOutOfBoundException –** Program tries to access invalid index of an array

**SELENEIUM**

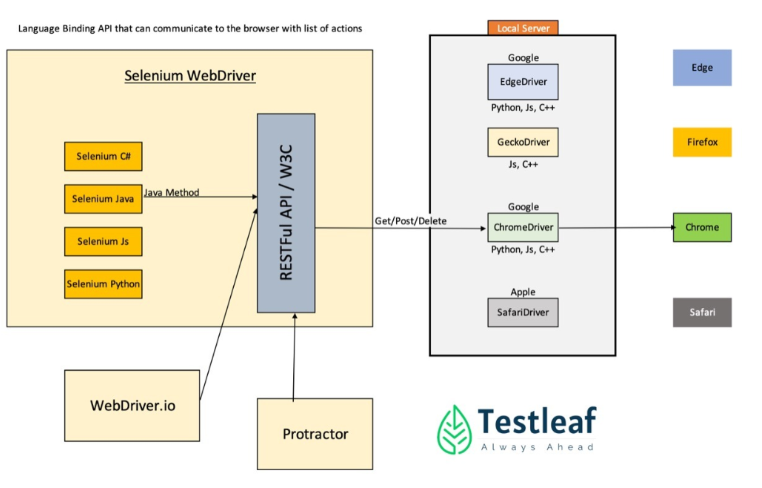
Annotation:

Tag that provides additional information about the class or method. It makes strong framework

TestNG annotation makes the test script more manageable, sophisticated, and effective. It extremely useful for testers and make the lives easier.

**Interview Questions**

**How selenium works internally?** Selenium is an open-source library which can be written in any language Each language has its own binding

While working in a Java language binding - “ChromeDriver driver = new ChromeDriver();” – this java code internally calls the selenium java implementation and it internally calls w3c protocols based request.

Selenium 4 completely used W3C whereas Selenium 3 and older version use JSON Protcols Technically its more of a rest api calls goes and reaches the chromedriver which is native driver running on the machines and those api hits local server, the request is taken by the servers, which get executed in the attached browser

**Common exception that you face in selenium**

I faced lot of different Selenium Webdriver exception. Some of the standard exception are

**TimeoutException** – Expected condition does not met with the specified time

**ElementNotClickableException**, ClickInterceptedException – When I do click() action

**StaleElementReferenceException** – dom element does not found after I found it

**NoSuchElementException** – Because of Bad Xpath

**NullPointerException** – From java perspective, If driver is closed or crashed for some reasons

**ArrayIndexOutOfBoundException** – Data and arrays, size is not well manipulated

**What is cucumber? Explain the need of using the cucumber**

Cucumber is a behavior - driven development testing tool. The BDD frameworks major goal is to bring together a variety of project responsibilities, such as quality assurance, developers and business analysts to understand application without diving too deeply into the technical components.

Testers use cucumber to create test cases for evaluating program behavior. It is an important tool to automate acceptance tests written in logical language that customer can understand. Its primarily used to develop acceptance tests for web apps based on their features behavior.

**What is a page object model**

It is a design pattern in selenium that is also used for enhancing test maintenance and reducing code duplication page object model is used in a framework like data driven, modular, hybrid framework etc It is a class that is also for serving as an interface to a page of our application under test

Anything that is present on the web page is a **WebElement (**HTML element)

**Why do we need Waits in Selenium?**

Waits are very much important in Selenium automation. Without Waits, our tests won't work as expected. wait enables us to perform our operations effectively without breaking the tests.

**Waits in Selenium**

Why do we get exception when interacting with Web Elements?

* The Element we try to interact / find is not at all available in the DOM
* Element is present in the DOM but not visible
* Element is present in the DOM but not interactable (enabled)

Types of Wait in Selenium:

|  |  |  |
| --- | --- | --- |
| * Implict | * Explict | * Thread.Sleep |

**Implict Wait**: Every 0.5 Sec it will interact with DOM -> **Polling** (the continuous checking, what state they in)

**Explicit Wait**: It is a type of Selenium waits which allows us to command the web driver to wait until several conditions are met before performing an action on the web element.

How to use? - > 1) create a WebDriverWait instace (Object) => driver, seconds 2) Wait until(Condition to met)

Some Conditions:

1) elementToBeClickable(); 2) elementToBeSelected(); 3)presenceOfElementLocated() 4)textToBePresentInElement() 5) alertIsPresent()

**Fluent wait:** it is used to tell the web driver to wait for the particular condition, as well as the frequency with which we want to check the condition before throwing an exception – reducing the burden of default pooling

**Difference between driver.close and driver.quit**

**driver.Close()-**This method is used to close the current open window. It closes the current open window on which **driver has focus on.**

**driver.Quit()-**This method is used to destroy the instance of WebDriver. It **closes all Browser Windows associated with that driver** and safely ends the session.

**Difference between get and Navigate method**

So the main difference between get() and navigate() is, both are performing the same task but with the use of navigate() you can move back() or forward(), refresh in your session's history.

navigate() is faster than get() because navigate() does not wait for the page to load fully or completely.

**How to take Screenshot in selenium?**

* Using selenium TakesScreenshot
* Call GetScreenShotAs Method and store it in a file
* Create a image file in any location
* Copy the screenshot in destination file (image) file using fileunits or we can use selenium filehandler as well to copy the file.

**How to do refresh the webpage?**

Navigate.refresh(); you can refresh the page when navigate().to() method is used

driver.get(driver.getCurrentUrl();

**Different ways to maximize the window?**

* driver.manage().window().maximize();
* Dimension dimension = new Dimension(1366, 800);

driver.manage().window().setSize(dimension);

* using chromeoptions is a class

ChromeOptions ch = new ChromeOptions();

ch.addArguments("—start-maximized");

ChromeDriver driver = new ChromeDriver(ch);

**Difference between Thread.Sleep() and Selenium.setSpeed()?**

|  |  |
| --- | --- |
| Thread.Sleep(2000);  Operation1;  Thread.Sleep(2000);  Operation 2;  Thread.Sleep(2000);  Operation 3; | Selenium.setSpeed(2000);  Operation1;  Operation 2;  Operation 3;  **This method is deprecated no longer available** |

**Print the google suggestion for a given search term**

**Different ways to perform a search in google**

* Using Send keys
* Robot class

Robot robot = new Robot();

robot.keyPress(KeyEvent.VK\_ENTER);

robot.keyRelease(KeyEvent.VK\_ENTER);

* . submit();
* /n - search.sendKeys("Game of thrones \n");

**Write a script without using find element**

driver.switchTo().activeElement().sendKeys("Madurai \n");

**How to Rerun the failed TestNg cases**

We can run using run testing-failed.xml

**How to Scroll the page in Horizontal and Vertical Axis**

Scroll to some position

JavascriptExecutor executor = (JavascriptExecutor) driver;

executor.executeScript("window.Scroll(0,450)","");

Thread.sleep(5000);

executor.executeScript("window.Scroll(0,-450)","");

//Go to Bottom of the page

executor.executeScript("window.scroll(0,document.body.scrollHeight)","");

Thread.sleep(5000);

executor.executeScript("window.Scroll(0,0)","");

//Go to a specific element position

executor.executeScript("arguments[0].scrollIntoView(true);","");

Robot robot = new Robot();

robot.keyPress(KeyEvent.VK\_PAGE\_DOWN);

robot.keyRelease(KeyEvent.VK\_PAGE\_UP);

**Assertion vs verify**

Assertion is like School Exam Verify is like Semester Exam

@Test

public void assertTest() {

// Hard Assertion

System.out.println("Before Assertion");

Assert.fail();

System.out.println("After Assertion");

@Test

public void assertTest1() {

// Soft Assertion

SoftAssert assert1 = new SoftAssert();

System.out.println("Before Assertion");

assert1.fail();

System.out.println("After Assertion");

**Give Input without using SendKeys**

JavascriptExecutor executor = (JavascriptExecutor) driver;

executor.executeScript("document.getElementsByName('q')[0].value='Amazon'", "");

executor.executeScript("arguments[0].value='Riya'", searchBox);

driver.switchTo().activeElement();

Robot robot =new Robot();

robot.keyPress(KeyEvent.VK\_R);

robot.keyRelease(KeyEvent.VK\_R);

robot.keyPress(KeyEvent.VK\_I);

robot.keyRelease(KeyEvent.VK\_I);

robot.keyPress(KeyEvent.VK\_Y);

robot.keyRelease(KeyEvent.VK\_Y);

robot.keyPress(KeyEvent.VK\_A);

robot.keyRelease(KeyEvent.VK\_A);

**Check all checkboxes in Webpages**

Write Xpath for checkboxes and tore it in a if statement and use click Method.

**TestNG Annotations and Hierarchy**

To create a TestNG Class -> Right Click -> Other -> TestNg Class -> Next

Suite -> Test -> Class -> Method

**Execute a test case multiple time with TestNg invocation count**

@Test(invocationCount=3)

**How to set timeouts for a test case in TestNg**

@Test(timeOut=3000); - In this time period the test will be executed or else it will throw TimeoutException instead of arrayIndexOutofBoundException

**How to handle Exception in TestNg**

@Test(timeOut=3000, expectedException=ArrayIndexOutofBoundsException.class);

**Significance of AlwaysRun attribute in TestNG**

@Test(timeOut=2000)

public void parentsPermission() throws InterruptedException {

Thread.sleep(3000);

System.out.println("Permission Given");

}

@Test(dependsOnMethods="ParentsPermission",alwaysRun=true)

public void loveMarraige() {

System.out.println("Marraige Happened");

}