SELENIUM

What is Selenium? What are the different Selenium components?

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.
Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.

The suite package constitutes of the following sets of tools:

 Selenium Integrated Development Environment (IDE) –

Selenium IDE is a record and playback tool. It is distributed as a Firefox Plugin.

 Selenium Remote Control (RC) –

Selenium RC is a server that allows user to create test scripts in a desired programming language. It also allows executing test scripts within the large spectrum of browsers.

 Selenium WebDriver –

WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.

 Selenium Grid –

Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

What is an Xpath?

 Xpath is used to locate a web element based on its XML path. XML stands for Extensible Markup Language and is used to store, organize and transport arbitrary data. It stores data in a key-value pair which is very much similar to HTML tags. Both being markup languages and since they fall under the same umbrella, Xpath can be used to locate HTML elements.

1] Types of the assertion.

Selenium Assertions can be of three **types**: "**assert**", "verify", and " waitFor". When an "**assert**" fails, the test is aborted. When a "verify" fails, the test will continue execution, logging the failure. A "waitFor" command waits for some condition to become true.

2] Do you have created framework from scratch, or you have maintained that?

I **have not created Framework from scratch** by myself but yes, I was part of framework creation and created some part of it.

The difference between sleep and wait in Java? (answer)

Though both are used to pause currently running thread, sleep is actually meant for short pause b ecause it doesn't release lock,whilewaitis meant for conditional wait and that's why it release loc k which can

then be acquired by another thread to change the condition on which it is waiting.

==========

3] Can you tell me about difference between Throw and Throws keyword?

Throw is a keyword used **inside a body of function**. And **Throws used while initializing any method**. By using Throw we can **throw only one exception while** for **Throws we can declare multiple exceptions** which might occur in that particular function.

Throws keyword **followed by instance name** and **Throw keyword is followed by class name** of that exception.

==========

4] will handle multiple window in selenium.

We have windowhandle & windowhandles function for handling Multiple windows. Windowhandle will give the string value of only the active window that is open whereas windowhandles will give set of all the windows that are open in browser.

5] What is the difference between FINDELEMENT & FINDELEMENTS?

FINDELEMENT will give the first appearance of that element which matches our locator, whereas **findelements will give us list of all the elements** which is present over the webpage and matching our locator**. And if we don't find the element findelement will give us nosuchelementexception whereas findelements will return NULL/Empty list.**

6] DIfference between this and super?

this keyword mainly represents the **current instance of a class**. On other hand **super** keyword represents the **current instance of a parent class**. this keyword used to call default constructor of the same class.

What are the different types of locators in Selenium?

Locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have different types of locators in Selenium:

- ID
- ClassName
- Name
- TagName
- LinkText
- PartialLinkText
- Xpath
- CSS Selector
- DOM

7] How you will move from one window to another?

First we will check what all windows are open by using driver.getwindowhandles, to get set of opened windows , then **I use iterator to iterate over each of the pages** and inside for loop will check like Current URL matches with the excepted page, if match then switch to that window by using driver.switchTo(Destination window) -> to return back to main parent window use driver.defaultwindow.

==========

8] Tell me the difference between Implicit & Explicit wait?

Implicit wait applies for all the elements **and all the tests like** if we give 10 sec of implicit wait it will wait for 10 sec for each element before giving nosuchelement exceptions.

While Explicit wait can be applied for **any particular step** for which you want extra wait time so we can use explicit wait. We can use mix of both waits to depend on the situation of the step.

==========

9] Can you tell me some exceptions in selenium?

NoSuchElementException
NoSuchWindowException
NoSuchframeException ,
StaleElementReferenceException,
TimeoutException.

==========

10] Difference between break and continue statement.

Break statement **resumes the control of the program to the end of loop** and made executional flow outside that loop.
Continue statement **resumes the control of the program to the next iteration of thatloop** enclosing 'continue' and made executional flow inside the loop again

==========

11] Have you used the action class and where it is used?

Using the **Actions class** in Selenium, we can implement the **sendKeys() method** to type specific values in the application. That is how **you use** the **actions class** in Selenium with sendKeys() method. .....
Theperform() method is **used** to perform the series of **actions** that **are** defined.

==========

12] Difference between Actions and Action?

Actions is a class that **is** based on a builder design pattern. This **is** a user-facing API for emulating complexuser gestures. Whereas **Action is** an Interface which represents a single user-interaction **action**

==========

13] What are hashmap and HashSet? Explain?

HashMap and HashSet both are one of the most important classes of Java Collection framework.**HashMap** Stores elements in form of key-value pair i.e each element has its corresponding key which isrequired for its retrieval during iteration. **HashSet** stores only objects no such key value pairs maintained.

==========

14] Where do you use a hashmap?

Maps are **used** for when you want to associate a key with a value and Lists are an ordered collection. Map is an interface in the Java Collection Framework and a **HashMap** is one implementation of the Map
interface. **HashMap** are efficient for locating a value based on a key and inserting and deleting values based on a key. HashMap<String, Integer> map = new HashMap<>();

==========

15] How many types of WebDriver API 's are available in selenium?

Chrome, Geko, Chromium, Edge, html, android,

==========

16] How can you make sure that page is loaded via wed driver or selenium?

Via first apply wait , is element present, then get text.

17] How do you run selenium webdriver test from the command line?

For that go to cmd-> java-class path(of the selenium project) ->hit enter

==========

18] What are the different exception you faced in selenium webdriver?

Webdriver exc, noalertpresent exc, nosuchwindow exc, nosuchelement exc, timeout exc.

==========

19] How do you scroll down a page using javascript in selenium?

Windows.scrollby function

==========

20] How do you scroll down to a particular element?

Windows.scroll.intoview function

==========

21] Which all files can be used as a data source for different frameworks?

.csv, .xml, .text etc

==========

22] What are listeners in selenium?

Listeners actually is an interface, that modifies the behavior of the system. It is used for customization ofreports. 2 types webdriver listeners, TestNg Listeners.

23] How do you take screenshots in selenium webdriver?

takescreenshot function

==========

24] What do you mean by assertions in selenium?

Assert, verify, wait for

==========

25] How many phases are there in maven build cycle?

6 validate-compile-test-package-install-deploy

==========

26] How will you handle keyboard and mouse related action using selenium?

By action class, robot class, venium driver

==========

27] What do you mean by WebDriver?

Webdriver is an interface which is used to automate api of browser for testing.

==========

28] How do you handle drag and drop option?

Using action classes

How do you avoid NullPointerException, while comparing two Strings in Java

• Call equals() and equalsIgnoreCase() method on known String literal rather unknown object

• Prefer valueOf() over toString() where both return same result

• Using null safe methods and libraries

• Avoid returning null from a method, instead, return an empty collection or an empty array.

• Use of annotation @NotNull and @Nullable

• Avoid unnecessary autoboxing and unboxing in your code

• Follow Contract and define reasonable default value

• Use Null Object Pattern

: When do Double and BigDecimal give different answers for equals() and compareTo() == 0.

• new Double(0.0).equals(0) returns false, while new Double(0.0).equals(0.0) returns true.

• BigDecimal.ZERO.equals(BigDecimal.valueOf(0.0)) returns false, while BigDecimal.ZERO.equals(BigDecimal.valueOf(0)) returns true.

==========

When InvalidMonitorStateException is thrown? why?

This exception is thrown when you try to call wait()/notify()/notifyAll() any of these methods for an Object from a point in your program where u are NOT having a lock on that object.(i.e. u r not executing any synchronized block/method of that object and still trying to call wait()/notify()/notifyAll())

wait(), notify() and notifyAll() all throw IllegalMonitorStateException. since This exception is a subclass of RuntimeException so we r not bound to catch it (although u may if u want to). and being a RuntimeException this exception is not mentioned in the signature of wait(), notify(), notifyAll() methods.

==========

29] How you handle java pop-ups in selenium?

Using alert, switch to alert, accept, dismiss, get text.

30] What does means Public static void main(variable,value)

Public/private/protected/default-Access specifier Static- modifier
Void- return type Main-
class name

==========

31] How to input text into a text box without Sendkeys?

JavascriptExecuter executor = (JavascriptExecutor)driver;
executor.excuteScript("document.getElementById("<<inputbox_id>>").value='new value');

==========

32] What are the open source frameworks supported by selenium webdriver?

TestNG, Junit, Cucumber, Robot Framework, Appium, Protractor.

==========

33] How to handle hidden elements in selenium webdriver?

JavascriptExecuter js = (JavascriptExecutor)driver;
js.excuteScript("document.getElementById("<<displayed_text>>").value='Hiddentext);

==========

34] How to handle iframes in selenium webdriver?

driver.switchTo().frames(via index value, name, webelement);
driver.findeElement(by.id("value")).getText();
driver.switchTo().defaultContent();-To get back from iframe

==========

35] How you handle dropdown values?

From select class, via visible text, value, index

==========

36] How to get color of webelement using selenium webdriver?

First get the locator of webElement , then get
String color= object.getCssValue("background-color")
String HexbackColor= color.fromString(color).asHex();
It will give you RGb codes , you need to convert them into back color using HEX function

==========

37] How you handle alert in selenium webdriver?

Simple alert(one option), Confirm Alert(Y/N), Prompt alert(enter any value) Alert a= driver.switchTo().alert();
a.getText();
a.accept(), a.dismiss(), a.sendKeys("name");

==========

38] How you handle multiple windows tabs in selenium webdriver?

String PID=driver.getWindowHandle();

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

Apply for loop on allWindowHandle -> switchTo().window(Id); ->if(!(Id.equals(PID)) ->driver.close();

==========

39] How to compare two array list?

Via Collection.sort(); and equal

==========

40] Difference between list and set.

The main **difference between List and Set is** that **Set is** unordered and contains different elements, whereas the **list is** ordered and can contain the same elements in it

==========

41] Use of constructor.

The purpose of constructor is **to initialize the object of a class** while **the purpose of a method is to perform a task** by executing java code. **Constructors** cannot be abstract, final, static and synchronised while methods can be. **Constructors** do not have return types while methods do.

TESTNG

1] Can you explain me TestNG?

TestNG is **advanced version of Junit only.** It is mainly used by Dev/QA **for maintain the code easily** and **for unit testing**. It provides lots of benefits to us like **we can create a suite** and we can write all the required Tc in one go only using that suite. We can group our Tc we can set priority we can run our tc in parallel mode, We can **generate good reports via TestNG**. We can write functionality depends on methods, depends on group. We can run single tc multiple time with single set of data of multiple set of Data.

==========

2] Can you tell me what is assert in TestNG?

Assert is like **verification** where we check like expected thing and actual thing are same or not.

==========

3] Which assert you have used in TestNg?

We have used **Hard assert and Soft Assert**, while applying Hard assert if we found any glitch in expected and actual then it will through exception and move to next @test while Soft assert it won't give exception and move to next step of that test. And to get all the exceptions in console we need to write at the end assert.all.

==========

4] Can you tell me about the order of TestNG annotations?

@BeforeSuite
@BeforeTest
@BeforeClass
@BeforeMethod
@Test
@AfterMethod
@AfterClass
@AfterTest
@AfterSuite

==========

5] Do you heard about Priority in TestNg can we set -ve priority?

Yes, like priority is 0, -1, TestNg will run -1 then 0 then 1. And if we have any @test which is not having any priority set, then in that case it will search via alphabetic order whichever comes first and execute test respectively.

==========

6] Can you tell me how you will re-run failed scenario in cucumber?

For that we can **use re-run attribute in our test runner file.** After that we can write one file location. Where all the test cases which failed while execution get stored. So next time while running execution we can give this file location and run the failed TC.

7] Do you work in cucumber, can you tell me what all files required in cucumber?

In cucumber we have **Feature file, Step Definition file and Test Runner file**.
In **feature file we used to write scenario** in gherkin language which is most like in plain English language. Here we use some of the keywords like feature, scenario, scenario outline, given, when, then, and, example, background keywords for writing our test scenarios steps.
In **Step Definition file we write mapping code for all the scenario of feature file.**
In test Runner file we provide the address of the feature file, step definition file, and all-important Tags,Plugin, Listeners in that.

8] You have worked in Cucumber & TestNG according to you which one is best?

I will **consider Cucumber** as it is most likely understood by Laymen people which is English plain language. Because in order to understand the functionality flow no need to go look and script/code. Via Scenario steps lines only we can get clear understanding about the functionality.
It helps to come all the QA members Dev, Client, Product Owner on same page.

==========

9] How to run single method multiple time in TestNG?

We have invocation **count attribute in @test annotiation.** We can write invocation count as 3 if we want to run it 3 times. Apart from that we can write threadpull.size if we want to run that case in multiple thread.

What exactly is REST?

REST is basically an architectural style of the web services that work as a channel of communication between different computers or systems on the internet. The term REST API is something else.

Those application programming interfaces that are backed by the architectural style of REST architectural system are called REST APIs. REST API compliant web services, database systems, and computer systems permit requesting systems to get robust access and redefine representations of web based resources by deploying a predefined set of stateless protocols and standard operations.

By these protocols and operations and redeploying the manageable and updatable components without causing the effect on the system, REST API systems deliver fast perf…

What Is a SOAP API?

SOAP is a **standard communication protocol system that permits processes using different operating systems like Linux and Windows to communicate via HTTP and its XM**L. SOAP based APIs are designed to create, recover, update and delete records like accounts, passwords, leads, and custom objects.

These offers over twenty different kinds of calls that make it easy for the API developers to maintain their accounts, **perform accurate searches** and much more. These can then be used with all those languages that support web services.

SOAP APIs take the advantages of making web based protocols such as HTTP and its XML that are already operating the all operating systems that are why its developers can easily manipulate web services and get responses without caring about language and platforms at all.

REST API	SOAP API
REST API has no has no official standard at all because it is an architectural style.	SOAP API, on the other hand, has an official standard because it is a protocol.
REST APIs uses multiple standards like HTTP, JSON, URL, and XML	while SOAP APIs is largely based on HTTP and XML.
As REST API deploys multiple standards, so it takes fewer resources and bandwidth as compared to SOAP that uses XML for the creation of Payload and results in the large sized file.	
. REST API takes advantage of URL exposure like @path("/WeatherService")	while SOAP API use of services interfaces like @WebService.
REST API, on the other hand, don't make emphasis on too many standards and results in corrupt API in the end.	SOAP API defines too many standards, and its implementer implements the things in a standard way only. In the case of miscommunication from service, the result will be the error.
REST API uses Web Application Description Language	SOAP API used Web Services Description language for describing the functionalities being offered by web services.
REST APIs are more convenient with JavaScript and can be implemented easily as well.	SOAP APIs are also convenient with JavaScript but don't support for greater implementation.

Streams:

Every program needs to write its data to a place or channel, every program needs to read data from a channel as well. In Java these channels from where a program writes or reads it data are known as Stream.

Streams are of two types basically:

Byte Stream-Classes named (*Streams)

Character Stream-Classes named (*Reader and *Writer)

Every stream that has the ability to write data contains a set of write methods. And every stream that can read data from any source has a similar set of read methods. Once the stream is created all these methods must be invoked.

Serialized objects are nothing but objects converted into Streams and sent over to files or through network for storage and retrieval.

Object class methods

Which are the various methods of Object class …?

• **protected Object clone()** –

Used to create and return a copy of this object.

• **boolean equals(Object obj)** –

Used to indicate whether some other object is "equal to" this one.

• **protected void finalize()** –

garbage collector calls this method on an object when it determines that there are no more references to the object.

• **Class<?> getClass() –**

Used to get the runtime class of this Object.

• **int hashCode() –**

Used to get a hash code value for the object.

• **void notify() –**

Used to wake up a single thread that is waiting on this object's monitor.

• **void notifyAll() –**

Used to wake up all threads that are waiting on this object's monitor.

• **String toString() –**

Used to get a string representation of the object.

• **void wait() –**

marks the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object.

• **void wait(long timeout) –**

marks the current thread to wait until either another thread invokes the notify() method or the notifyAll() method for this object, or a specified amount of time has elapsed.

• **void wait(long timeout, int nanos) –**

marks the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object, or some other thread interrupts the current thread, or a certain amount of real time has elapsed.

Provide an example how inheritance can break encapsulation?

It's all about the protected access modifier: it makes members private to everybody but derived classes. It's a "illusion of privacy", to sum it up.

The point about encapsulation is being able to change the private guts of a class having to worry only of making the public interface work as before. You know that changes to the private members of a class are affecting only the members of that class.

Why should you make an Object Immutable

• immutable objects are thread-safe so you will not have any synchronization issues.

• Immutable objects are good Map keys and Set elements, since these typically do not change once created.

• Immutability makes it easier to write, use and reason about the code (class invariant is established once and then unchanged)

• Immutability makes it easier to parallelize your program as there are no conflicts among objects.

• The internal state of your program will be consistent even if you have exceptions.

• References to immutable objects can be cached as they are not going to change.

How HashMap works in Java??

HashMap in Java works on hashing principle. It is a data structure which allows us to store object and retrieve it in constant time O(1) provided we know the key.

In hashing, hash functions are used **to link key and value in** HashMap. Objects are stored by calling put(key, value) method of HashMap and retrieved by calling get(key) method. When we call put method, hashcode() method of the key object is called so that hash function of the map can find a bucket location to store value object, which is actually an index of the internal array, known as the table. HashMap internally stores mapping in the form of Map.

Entry object which contains both key and value object. When you want to retrieve the object, you call the get() method and again pass the key object. This time again key object generate same hash code (it's mandatory for it to do so to retrieve the object and that's why HashMap keys are immutable e.g. String) and we end up at same bucket location. If there is only one object then it is returned and that's your value object which you have stored earlier. Things get little tricky when collisions occur. It's easy to answer this question if you have read good book or course on data structure and algorithms like this one. If you know how hash table data structure works then this is a piece of cake.

Since **the internal array of HashMap is of fixed size**, and if you keep storing objects, at some point of time hash function will return same bucket location for two different keys, this is called collision in HashMap. In this case, a linked list is formed at that bucket location and a new entry is stored as next node.

How will you retrieve Value object if two Keys will have the same hashcode??

We will call get() method and then HashMap uses Key Object's hashcode to find out bucket location.

After finding bucket location , we will call keys.equals() method to identify correct node in LinkedList and return associated value object for that key in Java HashMap.

==========

What happens On HashMap in Java if the size of the HashMap exceeds a given threshold defined by load factor?

The threshold of an HashMap is the product of current capacity and load factor.

Threshold = (Current Capacity) * (Load Factor)

For example, if the HashMap is created with initial capacity of 16 and load factor of 0.75f, then threshold will be,

Threshold = 16 * 0.75 = 12

That means, the capacity of the HashMap is increased from 16 to 32 after the 12th element (key-value pair) is added into the HashMap.

10] What is the difference between checked and unchecked exceptions?

There are two types of **exceptions**: **checked exception** and **unchecked exception**
The main **difference between checked and unchecked exception** is that the **checked exceptions** are **checked** at compile-time while **unchecked exceptions** are **checked** at runtime

checked exceptions – SQLException,IOException,ClassNotFoundException,InvocationTargetException

unchecked exceptions –

NullPointerException,ArrayIndexOutOfBoundsException,ArithmeticException,IllegalArgumentException

==========

11] What is the use of Dry Run in cucumber?

Dry run is not running our whole application it will check whether all features are mapped with Step definition.

==========

12] Which locator you are using in your framework and why?

Mostly we **used ID and Xpath** because Id is the fastest and unique one and after that we prefer Xpath. Anyways we have other locators as well like **css , class name, tag name, Link text, Partial Link text**

1What Maven Architecture and explain pom.xml?

POM is an acronym for Project Object Model. The **pom**. **xml** file contains information of project and configuration information for the **maven** to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc. **Maven** reads the **pom**.

13] What is dry run in Cucumber?

Dry-run is used to compile feature files and step definitions in **cucumber**. It is specially used in the stage when you will have to see if there are any compilation errors, to check that you can use **dry-run**. **Dry- run** options can either set as true or false.

14] Annotations in Cucumber

Total 11 Annotations -Feature, Scenario, Background, given, when , then, and, but, example, scenario outline, scenario template.

15] How do you handle if XPath is changing dynamically?

Option 1: Look for any other attribute which Is not changing every time In that div node like name, class etc. So If this div node has class attribute then we can write xpath as bellow.

//div[@class='post-body entry-content']/div[1]/form[1]/input[1]

Option 2: We can use absolute xpath (full xpath) where you do not need to give any attribute names In xpath.
/html/body/div[3]/div[2]/div[2]/div[2]/div[2]/div[2]/div[2]/div/div[4]/div[1]/div/div/div/div[1]/div/div/di v/div[1]/div[2]/div[1]/form[1]/input[1]

Option 3: We can use starts-with function. In this xpath's ID attribute, "post-body-" part remains same every time. //div[starts-with(@id,'post-body-')]/div[1]/form[1]/input[1]

Option 4: We can use contains function. Same way you can use contains function as bellow.div[contains(@id,'post-body-')]/div[1]/form[1]/input[1]

TOP SELENIUM COMMANDS WITH DETAILS

1) get() Methods

- driver.get("https://google.com");
- driver.getClass();
- driver.getCurrentUrl();
- driver.getPageSource();
- driver.getTitle();
- driver.getText();
- driver.findElement(By.id("findID")).getAttribute("value");
- **driver.getWindowHandle();**

==========

2) Locating links by linkText() and partialLinkText()

- *driver.findElement(By.linkText("Google")).click();*
  driver.findElement(By.partialLinkText("abode")).click();

==========

3) Selecting multiple items in a drop dropdown

- // select the multiple values from a dropdown
- Select selectByValue = new Select(driver.findElement(By.id("SelectID_One")));
- selectByValue.selectByValue("greenvalue"); - By Value
- selectByValue.selectByVisibleText("Red"); - By Visible Text
- selectByValue.selectByIndex(2); - By Index

==========

4] close() and quit() methods

- *driver.close();* - closes only a single window that is being accessed by the WebDriver instancecurrently

- *driver.quit();*- closes all the windows that were opened by the WebDriver instance

==========

5) Exception Handling

WebElement saveButton = driver.findElement(By.id("Save"));
- try{
- if(saveButton.isDisplayed()){
- saveButton.click();
- }
- }
- catch(NoSuchElementException e){
- e.printStackTrace();

6] isEnabled()

- **isEnabled()** to Check Whether the Element is Enabled Or Disabled in the Selenium WebDriver.

- findElement(By, by) with sendKeys() to type in the form fields.
- findElement(By, by) with getText() to store value of targeted web element.
  - **Submit()** to submit a web form.
  - **findElements(By, by)** to get the list of web elements**.**
  - List<WebElement> allChoices = dropDown.findElements(By.xpath(".//fruitoption"));
  - **findElements(By, by) with size()** to verify if an element is present**.**
  - Boolean checkIfElementPresent= driver.findElements(By.xpath("//input[@id='checkbox2']")). size()!= 0;
  - **pageLoadTimeout(time,unit)** to set the time for a page to load
  - driver.manage().timeouts().pageLoadTimeout(500, SECONDS);
  - **implicitlyWait()** to set a wait time before searching and locating a web element.

==========

navigate() to navigate between the URLs**.**

- driver.navigate().to("https://www.softwaretestinghelp.com");
- driver.navigate().back();
- driver.navigate().forward();

==========

getScreenshotAs() to Capture the entire page screenshot in Selenium WebDriver.

- File shot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
- FileUtils.copyFile(shot, new File("D:\\ shot1.jpg"));

==========

moveToElement() from the Actions class to simulate mouse hover effect.

- Actions actions = new Actions(driver);
- WebElement mouseHover = driver.findElement(By.xpath("//div[@id='mainmenu1']/div"));
- actions.moveToElement(mouseHover);
- actions.perform();

==========

dragAndDrop() from Actions class to drag an element and drop it on another element.

- WebElement sourceLocator = driver.findElement(By.xpath("//*[@id='image1']/a"));
- WebElement destinationLocator = driver.findElement(By.xpath("//*[@id='stage']/li"));
- Actions actions=new Actions(driver);
- actions.dragAndDrop(sourceLocator, destinationLocator).build().perform();

==========

switchTo() and accept(), dismiss() and sendKeys() methods from Alert class to switch topopup alerts and handle them.

- Alert alert = driver.switchTo().alert();
- alert.sendKeys("This Is Softwaretestinghelp");
- alert.accept()

==========

getWindowHandle() and getWindowHandles() to handle Multiple Windows in SeleniumWebDriver.

- String handle= driver.getWindowHandle();
- Set<String> handle= driver.getWindowHandles();
- for (String handle : driver.getWindowHandles()){
- driver.switchTo().window(handle);
- }

==========

getConnection() from DriverManager to start Database Connection.

- DriverManager.getConnection(URL, "username", "password")
- driver.manage().timeouts().implicitlyWait(1000, TimeUnit.SECONDS);

==========

Asserts using assertEquals(),assertNotEquals(), assertTrue() and assertFalse() to comparethe results.

- Assert.assertEquals(message, “This text”);
- Assert.assertNotEquals(message, “This text”);
- Assert.assertTrue(result<0);
- Assert.assertFalse(result<0);

==========

What are the technical challenges that you faced with selenium?

 There are quite a few technical challenges we faced in the initial stages of implementation; as my insurance application has a restriction - it opens only in Internet Explorer. With this said, locating the UI elements (test objects) via., developer tools & view source which was really challenging.

 Also there are many of the objects which doesn’t have the id, name; also the application is mostly in form tables, so we had to go for Xpath-Relative using developer tools post that we found an Fire-IEBrowser1.4 (VB Scripted excel) that helped in getting the Xpath-absolute & relative very easily.

 We had many obstacles in locating few UI Elements, for which we have used AUI.

 It took almost 20 working days to sort out all the issue and finish an end to end business transaction; with all the solutions found to the major problems we had automated all the 21 live stream products within a short span of time.

 Also mention about other challenges in action builder, selenium crashes, non-support to window handles, file uploads, IDE to webdriver conversion challenges etc

What is the difference between driver.close() and driver.quit command?

close():

WebDriver's close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does is return any value.

quit():

Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does is return any value.

Difference between findElement/findElements ?

findElement () will return only single WebElement and if that element is not located or we use some wrong selector then it will throw NoSuchElement exception.

findElements() will return List of WebElements – for this we need to give locator in such a way that it can find multiple elements and will return you list of webelements then using List we can iterate and perform our operation.

Does java supports multiple inheritance ?

No, but You can achieve it through interface

Inheritance:

The concept of getting properties of one class object to another class object is known as inheritance.

Here properties mean variable and methods.

Types of Inheritance:

Multiple inheritance.

Multilevel inheritance

Difference between assert and verify in selenium Webdriver

When an "assert" fails, the test will be aborted. Assert is best used when the check value has to pass for the test to be able to continue to run log in.

Where if a "verify" fails, the test will continue executing and logging the failure.Verify is best used to check non critical things. Like the presence of a headline element.

What is the difference between "GET" and "NAVIGATE" to open a web page in selenium web driver?

Get method will get a page to load or get page source or get text that's all whereas navigate will guide through the history like refresh, back, forward.Forexample if we want to move forward and do some functionality and back to the home page this can be achieved through navigate() only. driver.get will wait till the whole page gets loaded and driver.navigate will just redirect to that page and will not wait

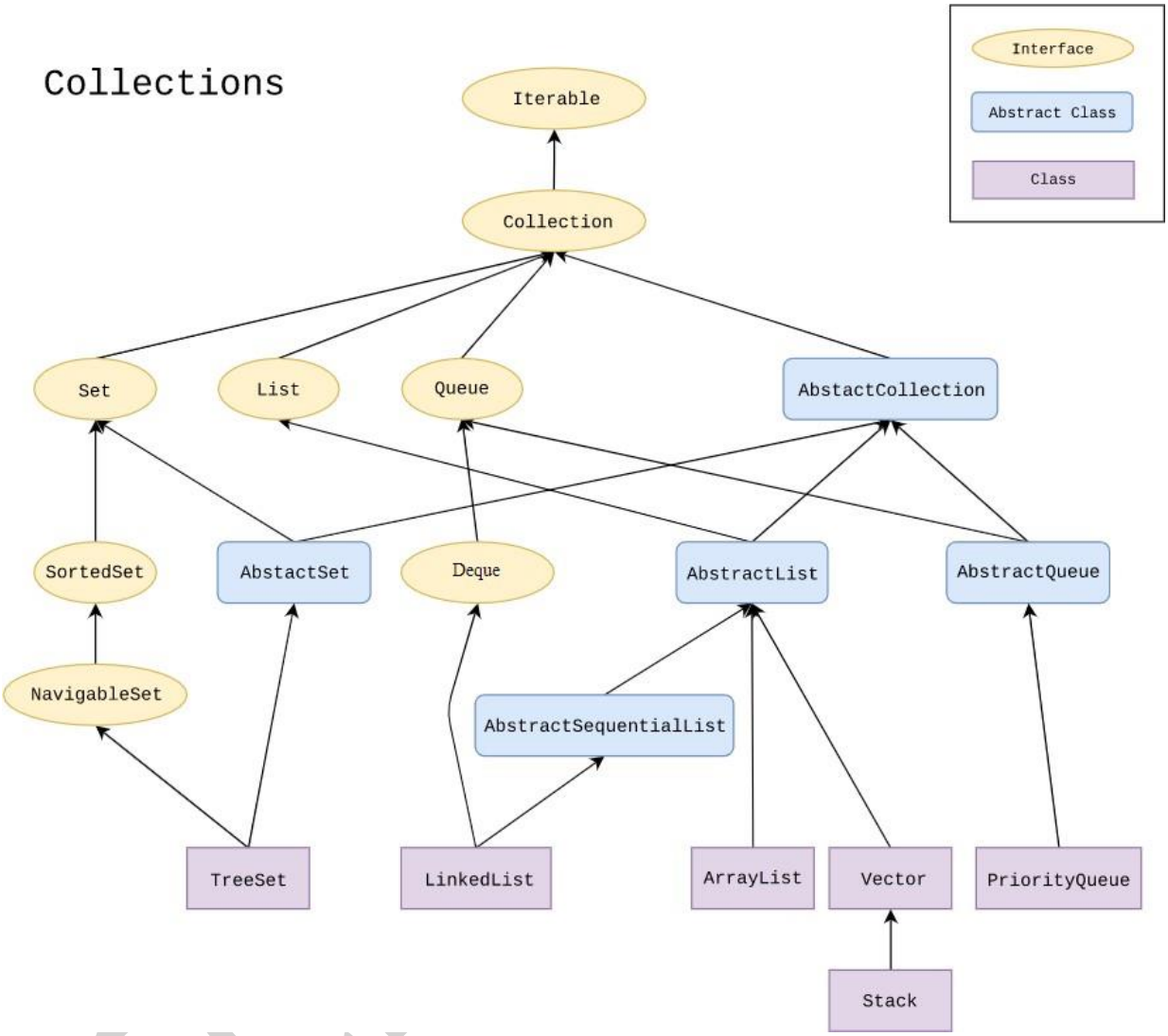
What is an epic, user stories and task?

Epic: A customer described software feature that is itemized in the product backlog is known as epic. Epics are sub-divided into stories

User Stories: From the client perspective user stories are prepared which defines project or business functions, and it is delivered in a particular sprint as expected.

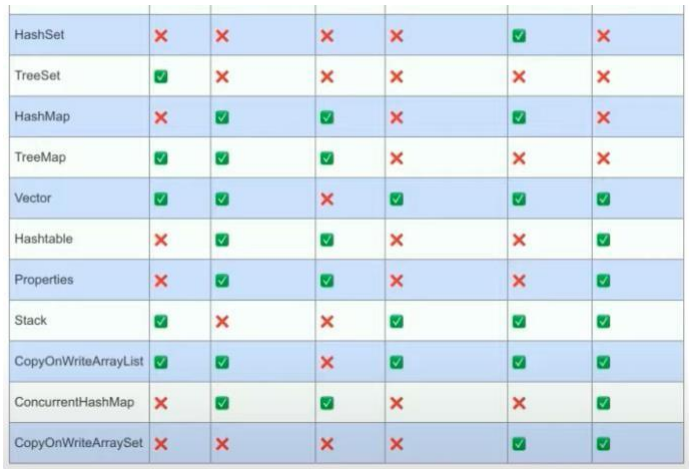
Task: Further down user stories are broken down into different task

COLLECTION FRAMEWORK



- A collection represents a group of objects.
- Java collections provide classes and Interfaces for us to be able to write code.
- We need collections for efficient storage and better manipulation of data in java.
- Collection reduces programming effort, provide in-build methods and classes.
- **ArrayList** -> For variables size collections
- **Set** -> For distinct collection
- **Stack**(queue)-> A LIFO (Last In First Out) data structure
- **HashMap** -> For strong key - value pairs
- **Iterator**- To iterate the element from collection.

- Collections class is available in java util package collection class also provides static methods forsorting, searching.
- **Common methods** available in Collection are add(), addAll(), remove(), removeAll(), size(),clear(), contains(), containsAll(), retain(), retainAll()
- **Common exception** is collections are NullPointerException, ClassCastException, IllegalargumentException, IllegalStateException, UnsupportedOperationException
-



HashSet	✗	✗	✗	✗	✓	✗
TreeSet	✓	✗	✗	✗	✗	✗
HashMap	✗	✓	✓	✗	✓	✗
TreeMap	✓	✓	✓	✗	✗	✗
Vector	✓	✓	✗	✓	✓	✓
Hashtable	✗	✓	✓	✗	✗	✓
Properties	✗	✓	✓	✗	✗	✓
Stack	✓	✗	✗	✓	✓	✓
CopyOnWriteArrayList	✓	✓	✗	✓	✓	✓
ConcurrentHashMap	✗	✓	✓	✗	✗	✓
CopyOnWriteArraySet	✗	✗	✗	✗	✓	✓

- **Thread Safety**-When multiple threads are working on same data, and the value of our data is
  changing, that scenario is not thread-safe and we will get inconsistent results. When a thread is already working on an object and prevent another thread on working on the same object is known as thread safety. We can achieve Thread safety via Synchronization, Volatile Keyword, Atomic variable, Final Keyword.

==========

Array List:

- ArrayList<Object Type> ar = new ArrayList<Object Type>();
- ArrayList is Dynamic in nature.
- Virtual Capacity of ArrayList by default is 10 but Physical capacity if we did not add any object is
  0. Once we start adding Physical objects Virtual Capacity got decreased by same.

==========

Hashmap:

- Hashmap<String, String>capitalmap = new Hashmap<String, String>();
  capitalmap.put("India", "New Delhi");

INTERVIEW QUESTIONS

1] What is Automation Testing ?

2] What are the role and Responsibilities of Automation Test Engineer ?

4] Advantage of Automation Testing?

5] Advantage of Selenium Testing

6] Dis advantage of Selenium Testing

7] Explain selenium Architecture ?

8] Explain Selenium Flabous ?

9] What is web driver and its Method ?

10] What is web elements and its Method ?

11] What is difference between Close and Quit ?

12] What is Difference between Get and Navigate ?

13] How to Maximize a Browser?

14] How to change position of browser

15] How to open a browser and write a code for this.

16] Explain Locators and its type

17] What are the different synchronization

18] Explain Absolute and Relative x path

19] Handling of Auto suggestion

20] Difference between Find Element and Find Elements

21] How to handle list boxes

22] How to handle customize list boxes

23] How to do drag and drop and write a code for this

24] How to handle pop ups

24] How to handle Iframes

25] How to work with excel sheet

26] How to take screen shoot

27] Mention the types of web locators?

28] What are the types of waits supported by WebDriver?

29] Mention the types of Navigation Commands

30] What is the major difference between driver.close() and driver.quit()?

31] How to click on a hyperlink in Selenium?

32] How to scroll down a page using JavaScript?

33] How to Mouse over a web element?

34]What is POM (Page Object Model)?

35]Can Captcha be automated?

36]How to take screenshots in WebDriver?

37] How to upload a file in Selenium WebDriver?

38]What is the difference between single and double slash in Xpath?

39] Get method

40]What is web driver?

41] Different types of locators?

42] What is the function of web element?

43] Any idea about POP-UP and its types?

44] What is syntax?

45] Explain framework in details

46] Exp collection framework

47] On which frame work you have work?

48] What is array ?

49] Diff between array and array list ?

50] Diff between list and Set ?

51] Differencebetween Array list and Linked list ?

52] Difference between Hashmap and Hashtable ?

53] Different between Collection and Collections ?

54] How to swap two numbers without using third variable

55] Explain Mutable and Inmutable in nature?

56] Can we handle Exception without Catche block?

POM

1] Explain pom

Page Object Model | POM

Creating Selenium test cases can result in an unmaintainable project. One of the reasons is that too many duplicated code is used. Duplicated code could be caused by duplicated functionality and this will result in duplicated usage of locators. The disadvantage of duplicated code is that the project is less maintainable. If some locator will change, you have to walk through the whole test code to adjust locators where necessary. By using the page object model we can make non-brittle test code and reduce or eliminate duplicate test code. Beside of that it improves the readability and allows us to create interactive documentation. Last but not least, we can create tests with less keystroke. An implementation of the page object model can be achieved by separating the abstraction of the test object and the test scripts.

Note: We will follow the same example which we have used in ***First Test Case***. Let's assume it our base test case and implement the Page Object Model (POM) in it.

How to do it…

1. Create a '***New Package***' file and name it as '**pageObjects'**, by right click on the Project and select **New** > **Package**. We will be creating different packages for Page Objects, Utilities, Test Data, Test Cases and Modular actions. It is always recommended to use this structure, as it is easy to understand, easy to use and easy to maintain.

2. Create a ***'New Class'*** file and refer the name to the actual page from the test object, by right click on the above created Package and select **New** > **Class**. In our case it is **Home Page** and **LogIn Page**.

```
package pageObjects;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
public class Home_Page {
private static WebElement element = null;
public static WebElement lnk_MyAccount(WebDriver driver){
element = driver.findElement(By.id("account"));
return element;
}
public static WebElement lnk_LogOut(WebDriver driver){
element = driver.findElement(By.id("account_logout")); Selenium Page 82
```

return *element*;
}
}

3. Now create a **Static Method** for each **Element** (Object) in the Home Page. Each method will have an**Argument** (driver) and a **Return** value (element).
Driver is being passed as an **Argument** so that Selenium is able to locate the element on the browser (driver).
Element is returned, so that an **Action** can be performed on it.
Method is declared as **Public Static**, so that it can be called in any other method without **instantiate**the class.
Follow the same rule for creating **LogIn Page** class. **package** pageObjects;
import org.openqa.selenium.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
public class LogIn_Page {
private static WebElement *element* = **null**;
public static WebElement txtbx_UserName(WebDriver driver){
element = driver.findElement(By.*id*("log"));
return *element*;
}
public static WebElement txtbx_Password(WebDriver driver){
element = driver.findElement(By.*id*("pwd"));
return *element*;
}
public static WebElement btn_LogIn(WebDriver driver){
element = driver.findElement(By.*id*("login"));
return *element*;
}
}

2] Dis advantage of pom

3] Advantage of Pom

4] Explain page factory

5] Diff between Page Factory and Pom

TEST NG

1] Advantage of TEST NG

2]What are the Annotation used in TEST NG

3] What is sequence of using annotation in TEST NG

4] What are the keyword in TEST NG

5] What is the importance of testng.xml file

6] How to do parallel execution using test ng

7] What is Data provider