194. Selenium Framework Interview Questions - Part 1

**1. what is the design pattern you have used in writing the tests in the Framework ?**

So when interviewers ask you about design patterns, so that means now this is your test(submit order test), what is the design? You have used to drive the objects and the data ?

we use the page object model.

So we clearly say that our design pattern is on page object model and we should also stress that our is mix of page object. That is the main design pattern.

But we have also used page a factory pattern to create objects so that very interviewer will have clear idea about how we have used this annotations.

So page factory pattern is subset of the page object model. See, page object model is the main pattern what we used. But inside that again we have used page factory pattern to create or derive locators or objects.

And from there, you might get questions like

how you have initialized elements for the page factory ?

we create init elements method in the constructor and we use annotations find by.

have you use a page factory annotation for list of elements? - So in that case you can explain that the return type, what you will choose is the list.

how the objects are being created ?

 instead of creating objects for the classes in that test, we are smartly driving the objects by writing them in the action methods. So that means if you are sure that if you go to the next page from the log in application method, then the last step of that method gives the object of the next page class. So you can explain this piece maybe with an example, saying that we incorporate objects in the action methods if we are sure that if that action method takes us to new page.

**2. how are reusable utilities handled within the framework?**

So there are two different ways we are handling it.

First, for all Selenium related methods, we are making sure that we have one class called abstract component where we are writing all reusable methods like wait for elements to appear. Go to cart page. These are common methods, what we identified.

So we are driving them in one class.

How are you reusing them in all classes?

So simply, we are inheriting that reusable method class into all of our page object files so that you can access those reusable methods across every page object class as we are inheriting them.

So like that, reusability is happening without creating code redundant .Without creating multiple times we are placing in one file and inheriting that into all page object

files. This is for methods of reusable selenium methods.

We also have another reusable method from the test perspective that is nothing but a base test. So all test related initialization also reusable here because this initialize driver is something where I use in every test. And I also use launch application.

They are down. These are commonly used in every test. So test related reusable methods. Also, we are driving it in the base test and all of our test cases will inherit that base test. So that means we have two types of reusable classes. One is core selenium methods where we are handling in abstract component and page objects, utilize

that reusable and another one is test utility is to start up, close the test or do some basic refactor. And our test cases reuse that base test by inheriting like this, you can project them.

**3. where did you use inheritance OOPS (object oriented ) concept in your framework?**

You know, inheritance is one of the Java object oriented concept.

you can explain that we have some reusability classes and we don't want to create objects for each and every reusable class. So we are simply inheriting them to our child classes so that where we are using inheritance effectively to use reusable methods in classes.

Best Ex : here is in your test.

So you have a setup that you have to first launch an application and go to your URL, everything you have to do.

And finally, you have to close the browser, but you don't want to repeat that code in each and every test.

You have to keep your test clean by separating it from the pre request and post request.

So for that what you did, you give those annotations in the parent class, but still you give before method and after method.

So those will be applicable if you try to inherit it.

So if you inherit a parent class, then your test will also consider the annotations which are different not only in your class, but also in the parent class.

So all the pre request post request code you encapsulated in the parent class without showing that in the test, keeping the test clean exactly to what requirements is, and then you are inheriting it. So that way the scanning is happening in the parent class itself.

So like that, you can talk a lot about how you use inheritance in your framework.

4. How did you drive the data from external files in the framework?

So that is something we have developed very advanced way in our framework, right.

So you can simply say that we use JSON files to parse the data.

So based upon number of entries you created in your JSON, so you are testable run that many times. So here I created two entries in my JSON.

So that means my test here is parameterized with the two different data sets and it will run two times.

when you explain like this, the immediately next question, what do you get is, okay, how do you do that?

You are telling that in JSON your test will read and do it but how that is possible and then you have to explain from scratch that starting from test engine data provider so you can explain them that there is a concept called the data provider in testing which will help you to parameterized test data.

So data provider also accepts hash map as an object to send that data.

So I'll create multiple hash map objects and I send from my data provider to iterate them in multiple times.

So I created one JSON file and I return one utility which converts my JSON content to hash map.

And once it is converted, that hash map object I am passing into my data provider.

And then they'll again ask, how do you convert your JSON to hash map?

there is a dependency called Jackson databind, which you need to externally get it and put it in your project.

So using Jackson data by utilities, there are classes like Object Mapper which will convert your JSON string content into a list of hash maps like that if you tell they can definitely correlate on the process of you are automating.

**5. Do you use interfaces in the framework? If so, what is the scope of it?**

If you go to a Listeners class, see that we are using keyword called implements.

So that means we are implementing ITestlisteners here. we are using listener's implement in our framework. So here Itestlistener is an interface.

So where did you use interface means? – ITest listener is one of the interface we are implementing in our framework.

Why are you implementing I test listener interface?

Because it provides a methods which are really helpful for us to write some code where we can avoid writing. In our test case

for example, for every test, we have to create one entry in the report. And if we know that, that has to be done as a prerequisite.

So we are using Itestlistener where in our on test start method I'll write the code.

lly, I have a requirement that I have to take a screenshot on test of failure.

So that is possible only if I have any listener where.

I can handle that on test failure, which is provided by Itest listener interface and where I can write the code inside on test failure.

So like that, we are using test engine interfaces and implementing in our framework to handle all this code to avoid writing them in the test directly.

And you know, there is another space interface we are using here. And you can also tell them Web driver itself is an interface. sSo here we are using different classes to implement the methods in this web driver interface.

Chrome driver is the one class where it is it implements to run in Chrome browser.

Similarly, Firefox driver is one class which implements to run in Firefox and Edge as well.

But overall, on higher level, web driver itself is an interface.

If somebody asks you, do you use the interface in selenium, then you can say, first step.

What we use web driver itself is interface and top of it, we use testNGinterface is also good.

**195. Selenium Framework Interview Questions - Part 2**

**6. How are you achieving encapsulation in the framework?**

Def of encapsulation - that it mainly says that hiding the implementation details of your class from another class.

Now let's say there is one class called checkout page.

So if anybody want to access this method, submit order, they can create object of this class and call this method. These are the public action methods where it will be available to anyone in your framework to access. But we really don't want to expose these variables.

So if you have a requirement like that, then how will you hide them so that another class cannot use them?

Now you might get a question Why do I want to hide them? Let it be there.

Why? Or What is the reason of hiding them and achieving encapsulation here?

Yes, there is one possibility. Now let's say suborder means submit dot click.

So here another class can, this is checkout.

If another class want to access this method, submit order.

Let’s say I want to access in the confirmation page and in that get confirmation message method.

I'm just creating object of checkout page class here.

when you are creating object of that class,not only methods.You are also able to access fields of that class.

Now you can say you create an object for this class, Checkout page. And this is, submit is your web element. But here, the problem is instead of calling action method, you are directly calling this field. That's fine, it'll still work.

So from your perspective, submitting order button, clicking means you have to do some set of actions like wait for element to new page to load,send out just all this. But shortcut ways directly clicking on this button by calling this web element.

So what people say is you have to use only action methodsin your script.You should not call web elements directly into another classes, okay? Web elements should be used internally

only by action methods. Nobody else should use it.

So how can you stop that somebody is using thatin the test instead of calling action methods.In that case, you will mark that field as a private.

If you mark it as a private, then they are strictly accessible only within that class.

As long as you set private, nobody can illegally access these elements outside your class.

Only these guys can access it, So like that, you are hiding your fields, and only exposing your action methods to public.

So this is one of the way to make your code more optimized and more strict in terms of encapsulation. That is what we say here that hiding the implementation details and only showing them as action methods as visible by marking them public.

So when somebody asks you about encapsulation, you can describe like this.

We are achieving that by keeping our fields private and action methods public.

And if they ask why do you do that? Then explain this use case

that we don't want them to access fields, which needs to be done only with action methods.

**7. does your framework support parallel runs? How are you writing thread safe code?**

Yes. For our framework to support parallel run.

First thing, what we are doing is in our testing XML file. We are making sure that attribute of parallel is set to tests.

So that means we were asking these tests to run parallelly and each test constitutes one class name.

So two classes are running parallel with its own thread. So there is no conflict here because at a time only one test in each class will run. And here also one test in each class will run.

So two tests will run. Parallelly. Why am saying there is no conflict? Bcze each test starts in its own thread. We are saying run test parallelly so there's no way they are overriding.

Now let's say this test how opens driver.

See it initialize driver but you might get doubt that will another test cannot override this driver object and make it mess. No, it won't override.

The reason is each test starts in its own thread. So if you start another test here, that also will start in its own thread. So it will not talk to the driver objects, whatever you created in the another test thread because that started in separate thread, so it won't collide with them.

If you want to run test in parallel, We were wrapping each class in one test block like this and we are simply saying a run parallel equal to test. So that way we don't really face any issue.

But at one point of time we really came across this overriding concept where.

While generating a report.

Test's driver Everything is fine, but if you see the reporting, if you complete my reporting sections there, I'm sure that you already know that we encountered an error that reporting is being replaced. So here there is no thread safe concept.

So we ran into issues of one test overriding another test report, right, if you remember.

So we solved that problem using thread local class.

So if you go to listener's, if you recall our previous concept when we ran into some overriding and thread synchronization problem, then we smartly came up with a class called Thread Local where it stores that thread information as a thread ID.

So like this we have avoided a concurrency issue and implemented thread safe mechanism for our extent reports.

So all this concept we did not do for any driver objects because in build we have started that in separate threads. So no issues.

But in Listeners we don't have a separate thread here as our tests from this XML file are reporting to this one single report there. The concurrent problem arrives to solve that concurrent problem.

We smartly brought a new class in Java which is thread local, which can identify them as a with unique thread ID and made sure that we achieve the thread safe when we are reporting all test results into one single report and that we're done with the help of getters and setters.

You create an object of thread local and you set all these concurrent objects and you retrieved it smartly using the unique thread.Just focus on this thread. Local class.

That's it.

If they ask How do you achieve thread safe, then say that - thread local class is one which can get to your unique thread and you store them using set method and retrieve them using get method.

And if they ask Where did you run into this concurrent issue?

Explain them that in while generating one single report, consolidated report for all test cases, we ran into issue and we solve the problem using thread local class

**8.  Do you have static keywords in the framework? If so it’s usage ?**

This is a tricky question which may ask with the combination of parallel testing.

So basically, if you keep your driver objects static, then you cannot achieve parallel testing in

your framework Why?

Static keyword is something which is commonly shared across all the threads.

Now, previously I told you that these tests will start in parallel in the separate threads, right?

So when it is accessing driver, it creates a fresh instance.

And then another thread is also accessing driver parallelly to create new browser.

It again creates a fresh instance for its own thread. All the variables are freshly created for their own thread, so there is no colliding.

But the danger with static is if you mark them as a static, then variables will be shared across the threads.

So that means if you run two tests parallel here, same driver variable will be shared for two threads if you mark them as static. That's the problem with static.

definition of static in Google, like where static is used in Java, static variables

use the common space location for all the threads.

So wherever parallel testing comes into picture there, there is no scope for static keywords.

I recommend you to avoid it wherever necessary.

You can write it in some places, like taking screenshots which are independent of your test runs so

you can have it for a product name.

For some text variables, you can have it, but don't make static for main initialization configuration variables like driver or so teardown methods, please don't use it if you want, you can use it here properties because let both threads use same file that is not

harming us because both files are not overwriting anything. They are simply reading this dot properties file.

n number of people can read at a time, but n number of people should not modify.That is our problem.

Can't you use same variable across multiple threads here?

If you get question like this, yes, this is readable. Nobody is really writing anything inside this. They were just reading that file.

**9. how are you sending global properties to your test at runtime?**

So this also we are doing in multiple ways.

So one way is that we create a Java Properties file which we just saw and we are reading this properties in our base test with properties class.

You can explain them that there is a class called properties which have ability to read any file which have a dot properties extension.

And we are loading that file and we are reading it at runtime like get property and if it matches with whatever you have, then based upon that you are setting your browser name.

So you might get a question like what are different variables where you can declare globally, you can declare globally about which browser you want to execute, that you can declare an properties file and write from there, and you can also give the URL because when you work in real time project for QA environment,

you will have one URL for UAT for prod, all the URLs you can mention there and you can decide at runtime on which environment to run by setting environment name environment equals to Q So that gives you URLs will execute and you can also set timeouts.

So here I put 10 seconds timeout that also you can drive it from the properties file.

So based on the value you set here, it will take that value.

So like this you can discuss about environments, URLs, browser, name and timeouts.

All these can be driven from the global properties.

This is one way you can also drive the global properties from and terminals, right?

If you remember then when we are giving MAVEN commands, we have used hyphen D and given parameter value.

So here it is. So browser name here. I'm not driving from global data.

You can also get that at runtime from your commands if your project is on Maven C for even node based projects.

Also you have how to drive from runtime, but Java is exclusively on Maven, so obviously you can discuss them that. I can also drive from Maven.

Then they will ask you a question like how you can catch that maven variable in your code.

Yes, you can do it with the help of system dot, get property and provide your maven parameter name.

So that way you can catch it and you can explain them very clearly that you can use ternary operator to decide if you want to drive it from Maven or if you want to drive it from your properties file.

So like that you can talk very much about global properties and how you are driving it in your test.

**10. what is the mechanism you use to run only selected set of tests inside the framework ?**

So let's say you have 100 test cases and you identify only 25 as smoke tests, which you want to run on a daily basis.

How we can do that?

Yes, we are doing that using testing groups concept.

You can explain them that whatever test we identified as a smoke we are going to mark them as a group equals to smoke as one of the helper attribute for our test annotation.

And once you mark them, then we will create one separate XML file and clearly tell them that only run the tests which are identified as a smoke with the group stack.

So like this, I'll trigger this testng xml file.

So only those tests which are identified with groups as smoke will trigger.

And you should also carefully explain them that not only just creating your separate group XML in the pom. XML also will create a separate profile for that group.

There are multiple things we are doing here.

One Mark the test case as groups by giving one name, and with that group, go ahead and create separate XML file and include that group name and then take that XML file, come to pom.XML and create a new profile for that group.

And finally from Maven, I would just say MVN test hyphen P and give that profile ID, that's it.

That will internally take care of only running those test cases.

So this way people will get this guy have clear picture about just not only identifying also how to run test with profiling with the selected set of group tests.

**11.  How are you handling flaky tests in the framework?**

So the flaky test means tests which are inconsistent and might fail randomly, but pass in the second attempt. In testng there is another interface called IRetryAnalyzer.

This is testng the interface which we are implementing.

So when any class implement this interface, we are forced to call this method.

So whenever a test fails testNG comes to this method.

If this retry returns true, then test will keep retrying it continuously until this method written

false.

So for that we applied one logic up to how many times it has to retry and we are incrementing the count.

So here in this case, maximum retry is one.

So after one time and this condition will become false and this method return false, right?

If retry method return false then testNG won't retry anymore. It will stop there.

So like this internally we implemented one logic and we are using this return just not only creating this class and implementing which test we want to retry that particular test, we want to add another helper attribute like this calling retry analyzer and give that which class you have written logic to retry like this you have to connect and this is really important.

If your test is flaky and it fails in the first attempt and passes in second attempt, and if you observe frequently then implement this retry to stingy analyzer and add it to your test so that it will make sure that it returns all flaky test and it will give you the exact output results

**12. Does your framework Take screenshots on test failures? How did you implement it ?**

**In listeners class**

So whenever a test fails, we definitely know that it comes to this block and here where we are handling screenshot code as simple as that.

So for that again we are relying on Itest listener interface which gives us all the overwriting methods.

using TestNG listener. whenever test is fail we can catch it and the test failure method will execute that. We will smartly read the screenshot code.

and here again there is so much drama, right, that attaching screen to the report.

All that also you can explain as an extension but simply in the catch block.

I mean in test failure, this is where I'm taking screenshot, but we have some more code about how to attach the screenshot to report that if you want, you can talk or you can simply tell that I'll handle that in on test failure method.

So this way it only takes screenshots only when test fails and skips when test passes.

**196. Selenium Framework Architecture Diagram - Explained**

So your test case is something which will be in the middle and your test case will have all the step to execute. And these steps we need locators.

And for locators we are relying on page objects,

All these are the three page objects(P1,P2,P3) where you are using to pull up the locators for your test.

If your application have ten pages, then you have ten page locators.

And now on top of this page object, we also realized that, you know, there is one abstract class we have where it takes care all reusable stuff.

Now wait for page two visible or go to court page header section, footer section and anything. If you think that these page object classes can reuse the code.

Now, if there is any four lines of code which you think you can reuse in P1class, P2 class, P3

class, then you can just make sure that is driven from one common class, which is an abstract component, and all your classes inherits from abstract component.

So this is how you are maintaining common objects of all page objects to one abstract component.

we are not duplicating the code in the files with the help Abstract component class.

Let's say TC1,TC2,TC3

Right now, even these test cases meet some common core, like initializing the browser, closing the browser, and selecting the browser to execute.

All that information needs to be reused in each and every test for this Common Core to reuse.

We are having base test as the another class where we declare all the common code which belongs to all these test cases(TC1,TC2,TC3).

And again, we are using inheritance concept here.

So all these classes will extend this parent class and use the reusable utility.

So our page object(P1,P2,P3) utilities uses abstract component and

our test cases(TC1,TC2,TC3) uses base test as a parent and segregate all the common code.

So you write a test cases, you take a page objects, and your test needed data from where it is coming. So data is something coming from. Json files.

And before it reaches your test middle, we have a parser. In the base test, we wrote a parser called get data to JSON.

So that method basically reads you JSON file converts into a hash map and our test cases use that hash map.

And here we have something called data provider for each and every test.

So basically your JSON files will be read by this method and hash map will be received by data provider. And then data provider sends data for you by utilizing it.

JSON FILE -> BASE TEST -> DATA PROVIDER -> FINAL TC

So this is how you are sending the data to your test.

Now capturing screenshots all that are externally taken care by listeners.

So in listeners you have like before test and after test success after test failure, you have a lot of things here

So all your test will start the journey from the laziness only.

It executes scored in the listener that create one entry for extent report and then it goes to the test. So extend reporting is something HTML report is getting generated with the help of this listener's bcze every test when it is trying to execute first, it will come here and execute a method where it creates entry to extent report and then go and execute it.

Once the execution is done again, it comes back to the listeners and then say test failure, pass and send that entry to extent report.

So we are using listeners as a middleware between our excellent reporting and testing.

So once TC execution is done, the results are reported here(Listeners). From here it goes to this point(Extent Report). So this is how we handle excellent reports.

And finally we have something called XML files which are test run.

So these will decide which test runner (TR1,TR2,TR3)

These will decide which tests to execute. So first you will trigger this test.

And what happens is XML files will see what test needs to be executed and then it gets connected to your test block.

All this is a test cases, And your XML is what we trigger,

how we trigger our XML - We triggered it from our pom dot XML file here where we provide entries for those testNG XML in terms of profiles.

And here is your terminal where you provide maven command.

So execution starts from here.

You run your maven command, it will go to pom.XML, see which profile this guy is asking to execute. And that particular profile XML will be triggered.

And in that xml, all the information will be there on which test to execute.

So it will go here and while executing test cases it need page object. So it goes to page object gets it and page object.

If needs any reusable thing, then they go to abstract.

And again it depends upon base test to set up configurations for your test to run.

And if it needs data, then it goes to Jason files, it connects to Jason files and then go to base

test and do data provider hash map and finally test will be fitted and while process of execution,

it reports everything to listeners and

listeners report to extend has HTML reporting.

This is what our entire framework is doing.

And top of it, if you use any cucumber feature files as a wrapper, then you can still have another wrapper here as a feature files and explain them that step definitions, same thing.

And also thread local group is something we are using inside listeners to achieve parallel execution.