**Bài 6: Xpath/CSS**

Automation testing level/ selenium component

Một số platform: Web Browser, Desktop application, Mobile Application. Native app: 1 nền tảng như android hoặc ios. Hybird app:

Kiểm thử phần mềm gồm 4 level:

Vào trang art of testing/ toolsqa

1. **Level of testing:**

* Functional Testing:
* Non-Functional testing: Security, performance testing such as Tiki, load testing

1. **Software testing level**

* Unit Test: bên dev làm, mình chỉ hỗ trợ kĩ thuật. viết code để xem hàm đó chạy có đúng không
* Integration test: kiểm thử tích hợp, ví dụ tích hợp 2 màn hình hoặc 2 chức năng với nhau. Business rules API…, workflow test, test theo các flow. Server testing, test những ứng dụng
* System test: test trên môi trường thực với dữ liệu thực. Có mỗi kĩ thuật là API testing.
* Acceptance test: Dành cho khách hàng kiểm thử anphal và beta.

1. **Một số môi trường để test:**

* App server
* Local
* Testing -> Tester
* Dev -> Developer
* Staging -> Demo cho Client/ PO
* Pre-Production: dữ liệu gần như khách hàng sử dụng
* Production/ Live: Người dùng khách hàng sử dụng

1. **Automation testing Level**

* 3 level chính

**+ Unit:** Dùng Jnit Test TestNG

**+ API:** test tầng service. ***API là gì?*** Là giao diện lập trình ứng dụng (Application Programming Interface)

Ví dụ về API để giải thích:

1 mô hình client và server/ backend, dữ liệu lưu vào database

Ví dụ đăng nhập, Client request gửi đến thông tin để đang kí đến server sau đó lưu vào database và response truyền về client

Sau này để phát triển hơn thì nó sinh ra 1 API xem như là 1 layer hoặc nhiều layer để xử lý nhiều tác vụ khác nhau ví dụ như website Tiki khá là nặng nên cần sử dụng APi. Nó là thằng để giao tiếp các layer với nhau.

Hoặc nó đại diện cho 1 thằng thứ 3, nó lấy thông tin từ 1 app khác để cung cấp cho 1 bên này sử dụng. Càng ngày ứng dụng càng phức tạp và lớn, ví dụ ứng dụng muốn kết nối với gg hoặc fb thì cần phải dung ggAPI hoặc FB API.

Hoặc có thể dụng API này qua API kia

Dùng Postman, return status 200.

**+ Functional UI:** Nắm được cấu trúc của dự án, selenium (làm trên web) Appium (mobile)

1. **Selenium**

* Là một tool để làm automation trên web, lịch sử từ thoughtworks
* Hỗ trợ cho rất nhiều ngôn ngữ: Java, C#, Python, JS, Ruby
* Hỗ trợ trên FireFox, Chrome, IDE, Opera, Safari
* Nó không bao gồm testing framework. Bình thường chúng ta sử dụng framework WebDriveIO để kết hợp nó.
* Ngôn ngữ chương trình được hỗ trợ thông qua Selenium drivers.

**WebDriverIO**: là 1 frameworks sử dụng base trên nền tảng ngôn ngữ JavaScript

1. **Selenium Components**

* Selenium IDE: cũ trên trình duyệt Firefox và Chrome
* Selenium RC: remote Control
* Selenium driver:
* Selenium grid: Phân tán nhiều máy, ví dụ dự án của mình cần chạy trên cả window, IOS .. thì cần phải dung cái này, số lượng testcase cần chạy nhiều rồi tính cả version. Build Cloud

**Bài 7+8+9: Xpath, CSS**

Cú pháp của locator

Có bao nhiêu maching node: có bao nhiêu element

1. **Web elements**

* Textbox/TestArea/Dropdown List
* Button/Checkbox/Radio Button
* Link/Date Time Picker
* Menu/Sub-menu
* Image/Tooltip/ Slider
* Table/Header/Label/Icon
* Video/TimeLine

1. **Locator in selenium**

* ID: Fastest
* Class: Fastest
* Name: Fastest
* Link text
* Partial Link text
* TagName
* CSS (Có thể cover 6 loại locator trên) - Faster
* Xpath (Có thể cover 7 loại locator trên) Slower

1. **Get element**

Selenium nó sẽ luôn thao tác với phần tử đầu tiên

* **Xpath và các kỹ thuật lấy xpath**

+ Cú pháp: **//tagname[@attribute=’value’]**

**Cấp độ ưu tiên**

|  |  |
| --- | --- |
| Locator phải là duy nhất | Để nó thao tác được đúng với element mình cần |
| Nên dùng id/class/name trước | được selenium hỗ trợ (3 locator) nên chạy nhanh hơn các att khác |

**Xpath có 2 loại:**

* Absolute: tuyệt đối, không bỏ qua bất kì node nào dùng /
* Relative: tương đối, dùng //

+ **Lấy Xpath theo attribute value = unique**

**+ Parent node:** lấy từ parent xuống

**+ Lấy tương đối dùng contains() nếu giá trị attribute hoặc value quá dài**

Cú pháp:

Theo text: //tagname[contains(text(),’value’)]

The attribute: //tagname[contains(@attribute,’value’)]

+ Lấy tuyệt đối thì cú pháp

Theo text(): //tagname[text()=’value’]

Theo attribute: //tagname[attribute()=’value’]

**+ Starts-with: lấy attr hoặc text lấy ở đầu**

Cú pháp: //tagname[starts-with(@attribute,’value’)]

**+ string():**

* Text nó nằm ở trên chính node đó hoặc nằm trong child node ở bất kì vị trí nào hoặc nested nó cũng work
* Text này có khoảng trắng/ xuống dòng/ tab ở đầu hoặc cuối chuỗi vẫn work đc

Cú pháp //tagname[contains(string(),’value’)]

Dùng để lấy text trong thẻ

string(Chuỗi Xpath)

**+ concat():** Dùng trong trường hợp text vừa có nháy đơn vừa có nháy đôi

**Cú pháp:** Hello "Join" What's happened

‘Hello "Join", What’, ”'s happend?”

**Có cả dấu nháy đơn và dấu nháy đôi**

**+ index of text()**

**//tagname[@attribute]**

* **CSS** bỏ dấu // và @ **tagname[attribute=’value’]**

**Topic 10 + 11: Xpath**

Thời gian chờ elements: **implicityWait**

**+ AND (Tuyệt đối) vs OR (Tương đối)**

**+ last() and position()**

Last(): muốn lấy ra một cái ngang cấp nhau. Lấy ca cái cuối cùng. Chú ý lấy đúng cái cuối cùng

Position() lấy ra cái vị trí muốn lấy, cái này là hàm, vì vậy muốn lấy ta có thể lấy //li[n]

Hoặc li[position()=n]

**AXES XPATH**

* **ancestor (tổ tiên):**
* **parent (cha): //tagnam[@att=’value’]//parent::div[@attr=’value’]**
* **preceding (bác): //tagnam[@att=’value’]//preceding::div[@attr=’value’]**
* **following (chú)**
* **preceding-sibling (anh của node hiện tại): //tagnam[@att=’value’][2]/preceding-sibling::li**
* **following-sibling (em của node hiện tại)://tagnam[@att=’value’][2]/following-sibling::li**
* **child (con)**
* **descendant (cháu của node hiện tại):**

***Nếu dùng CSS:***

Dấu . thay cho class

Dấu + đại diện cho following-sibling

[ đại diện cho and

Ví dụ li[4] thì dùng nth-Child(4)

Contains(@title,’value’) => [title\*=’value’]

^ là start-with

Bên xpath ko có ends-with nhưng bên css thì có ta dùng $

, là or

**Sự khác nhau giữa CSS và Xpath:**

CSS:

* không làm việc với text
* không đi ngược lên được

Xpath: không có ends-with nhưng css thì có và thay bằng dấu $

Verify: Xpath: $x; CSS: $$ and Jquery = $

Perfomance

**ID, Class, Name**

* run fastest

**CSS**

* Chạy nhanh hơn Xpath
* Làm việc với IE browser tốt hơn so với Xpath
* Không làm việc với text
* Không lấy ngược được parent, ancestor
* Áp dụng cho dự án cơ bản tốt

**Xpath**

* Chạy chậm hơn so với các locator khác
* Tìm được tất cả các loaltor
* Làm việc được với text, preceding-sibling, dynamic locator
* Có thể đi ngược lên để tìm các node parent, ancestor
* Xử lí các ca phức tạp: data-table, table, grid

Ưu tiên Xpath

* Xpath chạy có duy nhất
* Ưu tiên các attribute
* Xpath có thể chạy cho được nhiều step
* Chạy được

**Topic 12: Selenium Ecosystem**

**Các câu hỏi về kiến thức Automation Testing**

1. **Can you brief me about yourself?**

Hi, my name is Anh Tester.

I started my career as a Testing Executive 4 years back with Infosys currently I  am working as Test Engineer.

My responsibility is to understand Business Requirement Specification and High Level scenarios and to convert them into test cases & Automation scripts if required. Execution of test cases and reporting of defect to the developer if there any and  get them fixed. I have experience on Functional, Automation, Regression,  Smoke, Sanity, Web accessibility, Web Analytics, Mobile Testing.

In my previous project I have worked on Automation testing where we have used  Selenium with java and TestNG Cucumber framework for BDD approach. We have  used Page object model where we have separated our test cases with page objects,  and we performed testing on the same. For build management tool we are using Maven  for version controlling we are using Git and for automating our jobs for nightly run or  any schedule we are using Jenkins.

For defect management & test case management we have used JIRA, TEST RAIL & HP  ALM. I have worked on tools like BrowseStack, DeviceAnywhere, Toadsql, I am working on Agile environment we have daily standup call and we have 2-week  sprint cycle. I am part of 8-member team out of which we are 3-Tester, 2- dev, 1- manager, 1-scrum master.

1. **Tell me your Day to Day activities as QA?**

First thing I do after login in my system. I check the active sprint in Jira for our project  code. There I can see my assigned open tasks. After that I will check my mail if there is  any important mail I need to take action on. Then we have our daily scrum meeting  where we used to tell our previous day actions what we did, what we are planning for  today and if we have any blocker to discuss. Product owner and scrum master help us  to resolve that blocker. After that I need to take the pending task and do needed action  whether creating test case, Execution, Defect retesting if any.

1. **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.
2. **How much you rate yourself in Java out of 10?**

Out of 10 I will rate myself 8 in java as QA Automation engineer.

1. **Can you tell me Oops concepts and relate it with your Framework?**We have Polymorphism, Inheritance, Encapsulation and Abstraction in Oops. So, we will  start with

1) **DATA ABSTRACTION**

Data Abstraction means to handle complexity by hiding unnecessary details from  the user. In java, abstraction is achieved by interfaces and abstract classes. We  can achieve 100% abstraction using interfaces.

In Selenium, WebDriver itself acts as an interface. Consider the below statement:

WebDriver driver = new ChromeDriver();

We initialize the Chrome Browser using Selenium Webdriver. It means we are creating  a reference variable (driver) of the interface (WebDriver) and creating an Object. Here  WebDriver is an Interface and ChromeDriver is a class.

We can apply Data Abstraction in a Selenium framework by using the Page Object  Model design pattern. We define all our locators and their methods in the page class.  We can use these locators in our tests but we cannot see the implementation of their  underlying methods. So we only show the locators in the tests but hide the  implementation. This is a simple example of how we can use Data Abstraction in our  Automation Framework.

2)**ENCAPSULATION**

Encapsulation is defined as the wrapping up of data under a single unit. It is the  mechanism that binds together code and the data it manipulates. Encapsulation can be  achieved by: Declaring all the variables in the class as private and writing public  methods in the class to set and get the values of variables.

All the classes in an Automation Framework are an example of Encapsulation. In  Page Object Model classes, we declare the data members using @FindBy and  initialization of data members will be done using Constructor to utilize those in  methods.

3)**INHERITANCE**

Inheritance is the mechanism in java by which one class is allowed to inherit the  features (fields and methods) of another class.

We can apply Inheritance in our Automation Framework by creating a Base Class to  initialize the WebDriver interface, browsers, waits, reports, logging, etc. and then we can  extend this Base Class and its methods in other classes like Tests or Utilities. This is a  simple example of how we can apply Inheritance in our framework.

**4) POLYMORPHISM**

Polymorphism allows us to perform a single action in different ways. In Java  polymorphism can be achieved by two ways:

– **Method Overloading**: When there are multiple methods with same name but different  parameters then these methods are said to be overloaded. Methods can be overloaded  by change in number of arguments or/and change in type of arguments.

In Selenium Automation, Implicit wait is an example of Method Overloading. In  Implicit wait we use different time stamps such as SECONDS, MINUTES, HOURS  etc.

– **Method Overriding**: It occurs when a derived class has a definition for one of the  member functions of the base class. That base function is said to be overridden. In Selenium Automation, Method Overriding can be achieved by overriding any  WebDriver method. For example, we can override the findElement method In assertion we have used overload because in assertion we used to like  asset.true(actual, expected) and second time we can use same assert.true(actual,  expected, message).

1. **How can you use interface and how it is different from Abstract class?**

Abstract class may have Abstract and concrete methods, and there is not any  compulsion in adding abstract method in abstract class. But in Interface, we do have  only abstract methods and we don’t need to write abstract keyword in Interface this is  by default public and abstract.

1. **What do you mean by Static keyword in Java?**

Static means it is at class level not at instance level, we have static method, static  variable & static inner class. When we have any variable as static so it will remain  same for all the instance of our classes, and static/Private/Final methods can’t be  over-ridden like if we have initialized any method as Static so we cannot override it in  any child class.

1. **How to call static method and variable in java?**

Direct calling, Calling by class name.

1. **Can I access Static method by using object reference?**

Yes we can, but we got one warning that you need to access it via Direct or By class  name.

1. **How to call non-static method and variable in java?**

For calling non static method we need to create object first.

1. **Can we overload & override main**

**method?**Overload-Yes, Override

No

1. **What do you mean by wrapper class and how will you do data conversion?**Wrapper class in java are used for data conversion. In data conversion if user  wants to convert Int to string, String to int, Boolean, double then we use Wrapper  class.

integer.parseInt(); - To convert string to

Integer Double.parseDouble(); - To convert

string to Double Boolean.parse Boolean(); -

To convert string to Boolean String.valueof(); -

To convert Integer to String.

1. **Can you convert string a =”110a” in integer?**

No we got NumberFormatException while converting the above string.

1. **What do you mean by Call by Value & Call by Reference in Java?**Call by value means suppose we have created one sum method with input parameter  int a, int b. So while calling the creating the object and running we provide values that is  know as call by value.
2. **What do you mean by Exceptions in Java?**

Exception is like any interruption in our normal flow. Like if we are running anything and  we got issues in our script this is we called exception, we have 2 types of exception  Run Time & Compile Time. (checked & Unchecked exceptions)

1. **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.
2. **How much you rate yourself in**

**selenium out of 5?**Out of 5 I will rate

myself 3.5 in selenium.

1. **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.

1. **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.

1. **Why declaring the driver object as static in base class is not a good practice in  selenium? What are the challenges we might face if we do the same?**If you perform sequential test case execution, then it will work perfectly. But when you  do the same thing for parallel execution it will stop the execution because your first test  case will use your driver object but at the same another testcase running parallelly they  will also grab the driver object so it may through null pointer exception or may get some  weird result and your script will fail. But if you want to declare driver object as static you  can do. By defining local variable of driver object in each class.
2. **What are the conditions that throws ClassCastException?**

Integer to string, string is not a subclass of Integer, so a Class Cast  Exception will be thrown. Object I = Integer.valueOf(42);

String s = (String)i;

1. **Can you tell me how you will handle multiple windows 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.
2. **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.defaultContent().

1. **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.

1. **Can you tell me some exceptions in selenium?**

NoSuchElementException, NoSuchWindowException

NoSuchframeException, StaleElementReferenceException,

TimeoutException.

1. **Can you tell me about StaleElementReferenceException?**

Stale means old or decayed, here it sounds like element which was present on that  page is no longer there or decayed. To handle this, we can refresh the webpage  before pointing to that element. We can write script for waiting via explicit wait by  writing expected condition.refresh. Or we can go with page object model in that we  can over come this stale element exception.

1. **What do you mean by User Defined Exception?**

User Defined Exception or custom exception is creating your own exception class and  throws

that exception using 'throw' keyword. This can be done by extending the class  Exception. ... The keyword “throw” is used to create a new Exception and throw it to the  catch block.

1. **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.

1. **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.

1. **Can you tell me about the order of TestNG**

**annotations?**@BeforeSuite

@BeforeTest

@BeforeClass

@BeforeMethod

@Test

@AfterMethod

@AfterClass

@AfterTest

@AfterSuite

1. **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.

1. **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.

1. **What is the difference between scenario & scenario outline?**

When we have single scenario and we need to run it one time at that place we use  Scenario.

If you want some parametrization or Data Driven testing at that time, we can use  scenario outline where we have to use Example keyword like if we are running this  scenario for 3 different data set like username & pass. so, it will run 3 times.

1. **Can you tell me more about Background Keyword?**

Background is used when we have some common Given part. Suppose we have  pre-condition that we have to check this before each scenario. so in order to avoid  rewriting same step we can write it in Background.

1. **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.

1. **What is hooks in cucumber?**

In cucumber we use hooks for common functionalities, hooks are like we want to run  before & after each of the scenario. In hooks we have 2 different @before, @ after  which run before and after of each scenario. Also @beforestep, @afterstep which run  before and after each step.

1. **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.
2. **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.

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.

1. **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.

1. **Have you used GIT in your project can you explain about it?**

Yes I have used GIT, It is a version control tool. Where we can maintain our central  repo. we used to manage our code via GIT only. We use Git to maintain our project in  our local system. So, if someone like to work on that project I need to send complete  update copy to him and after that he can work on that. There are chances that single  project is handled by multiple teams across the globe. So, it will be difficult if we won’t  use GIT.

1. **Can you give me some GIT commands which you used on daily basis?**Git status- which shows status of all the files,if we have some files which is not yet  added to our repo so it will give us untracked file.

After that we can use GIT add command after adding it will added to particular index  and we can commit this file using Git Commit-(Message) we can commit this untracked  file. Also we have Git Merge, Git Post, Git Pull, Git It in etc.

1. **How to solve Merge conflict in GIT?**

As we are only 2 tester working on this project, if we have any merge conflict I used to  pull all the latest file/scripts to my local system. Then I will analyze the difference  between that particular file and merge file. After that I will check with my team member  whether all his imp things are covered then I will add my steps and push the script to the  central repo.

1. **You have worked in Jenkins can you tell me how you have created jobs in  Jenkins?**

We have separate Dev-Ops Team to create Jenkins jobs at broad level but we also  have access to jenkins, so we have created jobs for our internal purpose. For creating any job we have click on create new job->inside that give name of your job- >select freestyle project->then add. Beside that we can provide description of our project  and in source code management we can choose Git-> provide repo url ->after that  provide some schedule if you want to run the job on any specific schedule time.-> select  window batch command-file location-save-click on build now for running. After triggering  we can check log in console.

1. **What is the difference between Smoke & Sanity Testing?**

Smoke and Sanity we can are like same thing because both are checking important  functionality. Smoke testing is done on first stable build from developer to check like  whether it is stable enough to move further or not. While Sanity testing is subset of  regression test which we perform on stable build and here also we used to check all  the imp functionality.

1. **What is Agile ceremony?**

We have 4 Agile ceremony -Sprint planning, Sprint review, Sprint Retrospective, Daily  scrum meeting.

1. **Why the main method is static?**

Java **main**() **method**is always **static**, so that compiler can call it without the creation of  an object or before the creation of an object of the class. ... **Static method**of a class can  be called by using the class name only without creating an object of a class.

1. **What is Run time polymorphism**

**Run**-**Time Polymorphism**: Whenever an object is bound with the functionality at **run  time**, this is known as **runtime polymorphism**. The **runtime polymorphism**can be  achieved by method overriding. Java virtual machine determines the proper method to  call at the **runtime**, not at the **compile time**.

1. **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.

1. **Method overloading and overriding.**

**Method overriding**is used to provide the specific implementation of the **method**that  is already provided by its super class. **Method overloading**is performed within class.  **Method overriding**occurs in two classes that have IS-A (inheritance) relationship. In  case of **method overloading**, parameter must be different.

1. **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.

1. **Difference between static and non-static methods**

**Static method**uses complie time binding or early binding. **Non**-**static method**uses  run time binding or dynamic binding. A **static method**cannot be overridden being

compile time binding. A **non**-**static method**can be overridden being dynamic  binding.

1. **Explain Git workflow.**

**Step**1: Set up a Github Organization. ...

**Step**2: Fork Organization Repository to Your Personal GitHub. ...

**Step**3: Clone the Repository to Your Local Machine. ...

**Step**4: Create a Branch for your Working Files. ...

**Step**5: Set Remote Repository to the GitHub Organization. ...

**Step**6: Get Coding!

**Step 7:**Pull the Most Recent Files From the Organization Repo

**Step 8**: Merge the Master Branch Into the Feature Branch

**Step 9:**Push Your Code to your GitHub Repo

**Step 10**: Make a Pull Request to the Organization Repo

1. **How to set up Jenkins?**

**Step 1**− Go to the Jenkins dashboard and Click on New Item

**Step 2**− In the next screen, enter the Item name, in this case we have named it Helloworld.  Choose the ‘Freestyle project option’

**Step 3**− The following screen will come up in which you can specify the details of the job. **Step 4**− We need to specify the location of files which need to be built. In this example, we  will assume that a local git repository(E:\Program) has been setup which contains a  ‘HelloWorld.java’ file. Hence scroll down and click on the Git option and enter the URL of  the local git repository.

**Note**− If you repository if hosted on Github, you can also enter the url of that repository  here. In addition to this, you would need to click on the Add button for the credentials to add  a user name and password to the github repository so that the code can be picked up from  the remote repository.

**Step 5**− Now go to the Build section and click on Add build step → Execute Windows batch  command

**Step 6**− In the command window, enter the following commands and then click on the Save  button.

Javac HelloWorld.java

Java HelloWorld

**Step 7**− Once saved, you can click on the Build Now option to see if you have successfully  defined the job.

**Step 8**− Once the build is scheduled, it will run. The following Build history section shows  that a build is in progress.

**Step 9**− Once the build is completed, a status of the build will show if the build was  successful or not. In our case, the following build has been executed successfully. Click on  the #1 in the Build history to bring up the details of the build.

**Step 10**− Click on the Console Output link to see the details of the build

1. **Can we declare many interfaces object class inside the interface class.**

Yes, **you can**define a **class inside**an **interface**. **In**general, if the methods  of the **interface**use this **class**and if **we**are not using it anywhere else **we  will declare**a **class**within an **interface**.

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.

1. **Abstraction in java and exp?**

In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,)  in a Page Class. We utilize these locators in tests but we can’t see these locators in the  tests. Literally we hide the locators from the tests.

1. **What is a super keyword in java?**

The **super keyword**refers to superclass (parent) objects. It is used to call  superclass methods, and to access the superclass constructor. The most common  use of the **super keyword**is to eliminate the confusion between superclasses and  subclasses that have methods with the same name.

1. **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 that loop enclosing '**continue**' and made executional  flow inside the loop again

1. **Diff between Abstract class & interface?**

**Abstract class**can inherit another **class**using extends keyword and implement an **interface**. **Interface**can inherit only an inteface. **Abstract class**can be  inherited using extends keyword. **Interface**can only be implemented using  implements keyword.

1. **What is a static keyword in Java?**

In the **Java**programming language, the **keyword static**indicates that the particular  member belongs to a type itself, rather than to an instance of that type. This means that  only one instance of

that **static**member is created which is shared across all instances of the class. 62. **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. ... The perform() method is **used**to perform the  series of **actions**that **are**defined.

1. **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,IllegalArgume ntException

NumberFormatException

1. **Apart from sendkeys, are there any different ways, to type content onto the  editable field?**

WebDriver driver = new FirefoxDriver(); JavascriptExecutor executor =  (JavascriptExecutor)driver;

executor.executeScript("document.getElementById("textbox\_id").value=

'new value';);

1. **What is static and non-static?**

In non-static method, the method can access static data members and static methods  as well as non-static members and method of another class or same class. Binding  process. Static method uses compile time or early binding. Non-static method uses  runtime or dynamic binding. Overriding.

1. **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.

1. **What is the difference between length and length() in Java?**

The **length**is an instance variable of an array in **Java**whereas **length()**is a method of  String class

1. **What is an abstract class?**

**Abstract Classes**and Methods **Abstract class**: is a restricted **class**that cannot be  used to create objects

(to access it, it must be inherited from another **class**). **Abstract**method: can only be  used in an **abstract class**, and it does not have a body. The body is provided by the  subclass (inherited from).

1. **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 complex user gestures. Whereas **Action is**an Interface which represents  a single user-interaction **action.**

1. **How do you handle keystrokes in Selenium?**

**Using Actions Class**: Actions action = new Actions(driver); action. keyDown(**Keys**. ... **Using SendKeys**Chord: driver. findElement(By. ...

**Using Robot Class**: // Create Robot class Robot rb = new Robot(); // Press control  keyboard key rb.

1. **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.

1. **Annotations in Cucumber**

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

1. **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 is required for its retrieval during iteration. **HashSet**stores  only objects no such key value pairs maintained.

1. **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<>();

**// Add elements to the map**

map.put("vishal", 10);

map.put("sachin", 30);

map.put("vaibhav", 20);

**// Print size and content**

System.out.println("Size of map is:- " + map.size());

System.out.println(map);

**// Check if a key is present and if present, print value**

if (map.containsKey("vishal")) {

Integer a = map.get("vishal");

System.out.println("value for key"+ " \"vishal\" is:- " + a);

1. **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]

1. **Does Jenkins require a local system for CI?**

It is a server-based application and requires a web server like Apache Tomcat 77. **What is a singleton class?**

The Singleton's purpose is to control object creation, limiting the number of objects to  only one. Since there is only one Singleton instance, any instance fields of a Singleton  will occur only once per class, just like static fields. Singletons often control access to  resources, such as database connections or sockets.

For example, if you have a license for only one connection for your database or your  JDBC driver has trouble with multithreading, the Singleton makes sure that only one  connection is made or that only one thread can access the connection at a time.

1. **When finally block get executed?**

The **finally block**always **executes**when the try **block**exits. This  ensures that the **finally block**is **executed**even if an unexpected  exception occurs.

1. **How many times you can write catch block?**

maximum **one catch block will**be executed. No, **we can write multiple catch  block**but only **one**is executed at a **time.**

1. **What 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**.

**REAL TIME SELENIUM + JAVA INTERVIEW QUESTIONS**

1. **How many types of WebDriver API ’s are available in selenium?**

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

1. **How can you make sure that page is loaded via wed**

**driver or selenium?**Via first apply wait , is element present,

then get text.

1. **How can we launch a batch file in selenium webdriver project?**

Take path of batch file->process.batchjob= runtime.get(runtime.executable”Path”)); 4. **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

1. **What are the different exception you faced in selenium webdriver?**Webdriver exc, noalertpresent exc, nosuchwindow exc, nosuchelement exc, timeout exc. 6. **How do you scroll down a page using**

**javascript in selenium?**Windows.scrollby

function

1. **How do you scroll down to a**

**particular element?**

Windows.scroll.intoview function

1. **Which all files can be used as a data source for different frameworks?**.csv, .xml, .text etc
2. **What are listeners in selenium?**

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

1. **How do you take screenshots in selenium**

**webdriver?**takescreenshot function

1. **What do you mean by assertions in**

**selenium?**Assert, verify, wait for

1. **How many phases are there in maven build cycle?**

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

1. **How will you handle keyboard and mouse related action**

**using selenium?**By action class, robot class, venium driver

1. **What do you mean by WebDriver?**

Webdriver is an interface which is used to automate api of browser for testing. 15. **How do you handle drag and drop option?**Using action classes

1. **How you handle java pop-ups in selenium?**
2. Using alert, switch to alert, accept, dismiss, get text.
3. **What does means Public static void main(variable,value)**

Public/private/protected/default-Access specifier Static- modifier

Void- return type Main-class name

1. **How to input text into a text box without Sendkeys?**JavascriptExecuter  executor = (JavascriptExecutor)driver;  executor.excuteScript(“document.getElementById(“<<inputbox\_id>>”).value =’new value’);

1. **What are the open source frameworks supported by**

**selenium webdriver?**TestNG, Junit, Cucumber, Robot

Framework, Appium, Protractor.

1. **How to handle hidden elements in selenium webdriver?**JavascriptExecuter js = (JavascriptExecutor)driver;  js.excuteScript(“document.getElementById(“<<displayed\_text>>”).value

=’Hiddentext);

1. **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

1. **How you handle dropdown values?**

From select class, via visible text, value, index

1. **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 25. **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”);

1. **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();

=================================================================== ==================

**JAVA ONE LINE CONCEPTS**

1. java file mai ek hi public class hoti hai, uske alava aur classes v ho sakti hai par  public ek hi.
2. If we need to create one variable for multiple values, we need to use Array concept. 3. Int marks[] = new int[5]
3. Array can store only homogenous data, int for int array, string for string, 5. If we need to add heterogeneous data in array, we need to create object array. 6. Object a[] = new Object[5]; now we can add different data type objects. 7. Array is fixed in size, which we define while creating.
4. If we try to access index value >= given index value, we got  arrayOutOfBundException.
5. Arrays are not defined by any data layer structure so we can’t run readymade  methods on it.
6. To overcome this, we have collection framework under which there are  ArrayList, List, HashMap, HashTable, Tree, Stack.
7. We can add new elements in run time under collections while in array we cannot. 12. Collection is a group of objects. To represent this we need certain interfaces and  classes.
8. Common operations we generally do on collections are **adding objects, removing objects & finding object.**
9. **Collection (I)**is called collection interface having methods which are common  throughout all collections.
10. **Collections**is basically a class from java.util package which contains some  methods which we can use for collection objects.
11. Collection – 1. List, 2. Set, 3. Queue
12. **List (I)**is child of collection(I). In list Insertion order is preserved and duplicates are  allowed.
13. **ArrayList, LinkedList, Vector**these are different classes which implements **List  Interface**.
14. **Set(I)**is child of collection(I). Insertion order is not preserved & duplicates not  allowed.
15. **HashSet, Linked Hashset**these are different classes which implements **Set  Interface**.
16. **Queue(I)**is child of collection(I). We used it when we need prior to processing  means first in first out concept.
17. **priorityQueue**is class which implements **Set Interface.**
18. There is one independent interface known as **Map(I).**In Map(I) objects are created  with **key and value**pair. Key cannot be duplicate, but Value can be.
19. **Hashmap, Linked Hashmap, Hash Table**these are different classes which  implements **Map**

**Interface.**

1. Whatever methods present in Collection(I) are also present in their child interface  i.e List, Set, Queue.

1. **add(object o), addAll(Collection c), remove(Object o),  removeAll(Collection c), retainAll(Collection c)**these are some  methods of Collection Interface.
2. **clear(), isEmpty(), size(), contains(), conatinsAll(), toArray()**are also some  methods.
3. In List **index**play an important role because with the help of index only we  can find duplicates elements.
4. **add(index , object), get(index), set(index, object)**are methods of List Interface. 30. **ArrayList al= new ArrayList(),**it allows heterogenous objects also.
5. **ArrayList<Str> al= new <str>(),**now it can store objects of string only. 32. **Collections.Sort(al) , Collections.Shuffle(al)**This will sort & shuffle the objects of  arraylist.
6. We can read the data with for loop, for each loop, iterator () method.
7. JVM have 2 types of memories **Static pool -static data, heaps-Non static data,**

**COLLECTION FRAMEWORK**

1. A collection represents a group of objects.
2. Java collections provide classes and Interfaces for us to be able to write code. 3. We need collections for efficient storage and better manipulation of data in java. 4. Collection reduces programming effort, provide in-build methods and classes. 5. **ArrayList**-> For variables size collections
3. **Set**-> For distinct collection
4. **Stack**(queue)-> A LIFO (Last In First Out) data structure
5. **HashMap**-> For strong key - value pairs
6. **Iterator**- To iterate the element from collection.
7. Collections class is available in java util package collection class also provides static  methods for sorting, searching.
8. **Common methods**available in Collection are add(), addAll(), remove(),  removeAll(), size(), clear(), contains(), containsAll(), retain(), retainAll()
9. **Common exception**is collections are NullPointerException,  ClassCastException, IllegalargumentException, IllegalStateException,  UnsupportedOperationException

1. **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.
2. **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.

1. **Hashmap:**

* Hashmap<String, String>capitalmap = new Hashmap<String,

String>(); capitalmap.put(“India”, “New Delhi”);

**MAVEN INTERVIEW QUESTIONS**

1. **What is Maven:**

* Maven is a *project management tool*that is based on POM (project object model). It is  used for projects build, dependency and documentation.
* It simplifies the build process.

1. **Why Maven Required**

Maven project eliminates the Adding set of Jars in each

project. It Creates write Project Structure.

Building and Deploying the project is very simple.

1. **What is Build Tool**

A build tool takes care of everything for building a process. It does following: • Generates source code

* Compiles source code
* Packages compiled code into JAR of ZIP file
* Installs the packaged code in local repository, server repository, or central repository

1. **How to install Maven in Window machine**

* Download maven and extract it
* Add JAVA\_HOME and MAVEN\_HOME in environment variable
* Add maven path in environment variable
* Verify Maven apache

maven-3.5.0-bin.zip

1. **Verify maven**

To verify whether maven is installed or not, open the command prompt and write: mvn −version

Maven home: E:\apache-maven\apache-maven-3.3.9\bin

Java version: 1.8.0\_102, vendor: Oracle Corporation

Java home: C:\Program Files\Java\jdk1.8.0\_144\bin

1. **Maven Repository**

There are 3 types of maven repository:

* Local Repository
* Central Repository
* Remote Repository

Maven searches for the dependencies in the following order: **Local repository**then  **Central repository**then **Remote repository**. **Local Repository:**Means .m2 folder in  your system

**Central Repository:**Maven **central repository**is located on the web. It has been  created by the apache maven community itself

Remote Repository: Company Specific Library or Custom Library <project  xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com</groupId>

<artifactId>Cucumber</artifactId>

<version>0.0.1</version>

<packaging>jar</packaging>

<name>selenium</name>

<url>http://maven.apache.org</url>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.5.1</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

Project: It is the root element of pom.xml file

Model Version: It is the sub element of project. It specifies the model Version. It should  be set to 4.0.0

**Group Id:**will identify your project uniquely across all projects (com.test.selenium) **Artifact Id:**project name

**Packaging:**defines packaging type such as jar, war etc.

**Dependencies:**defines dependencies for this project.

**Dependency:**defines a dependency. It is used inside dependencies.

**Scope:**defines scope for this maven project. It can be compile, provided, runtime,  test and system.

1. **Maven Life Cycle**

* **validate**- validate the project is correct and all necessary information is available • **compile**- compile the source code of the project
* **test**- test the compiled source code using a suitable unit testing framework. These  tests should not require the code be packaged or deployed
* **package**- take the compiled code and package it in its distributable format, such as a  JAR.
* **verify**- run any checks on results of integration tests to ensure quality criteria are met • **install**- install the package into the local repository, for use as a dependency in other  projects locally
* **deploy**- done in the build environment, copies the final package to the remote  repository for sharing with other developers and projects.

mvn deploy:deploy-file -DgroupId=<group-id> \

-DartifactId=<artifact-id> \

-Dversion=<version> \

-Dpackaging=<type-of-packaging> \

-Dfile=<path-to-file> \

-DrepositoryId=<id-to-map-on-server-section-of-settings.xml> \

-Durl=<url-of-the-repository-to-deploy>

**===================================================================**

**JAVA HASHMAP INTERVIEW QUESTIONS**

1.How does put() method of HashMap works in Java? On hashing principle of key value  pair

1. What is the requirement for an object to be used as key or value in HashMap? 3. What will happen if you try to store a key which is already present in HashMap? 4. Can you store a null key in Java HashMap?
2. Can you store a null value inside HashMap in Java?
3. How does HashMap handle collisions in Java?
4. Which data structure HashMap represents?
5. Which data structure is used to implement HashMap in Java?
6. Can you store a duplicate key in HashMap? (answer)
7. Can you store the duplicate value in Java HashMap? (answer)
8. Is HashMap thread-safe in Java?

12.What will happen if you use HashMap in a multithreaded Java application? 13.What are the different ways to iterate over HashMap in Java?

1. How do you remove a mapping while iterating over HashMap in Java?
2. In which order mappings are stored in HashMap?
3. Can you sort HashMap in Java? (answer)
4. What is the load factor in HashMap? A load factor is a number that controls the resizing of HashMap when a number of elements in the HashMap cross the load factor as if the load factor is 0.75 and when becoming more than 75% full then resizing trigger  which involves array copy.
5. How does resizing happens in HashMap? (answer)
6. How many entries you can store in HashMap? What is the maximum limit? 20.What is the difference between the capacity and size of HashMap in Java? 21.What will happen if two different keys of HashMap return the same hashcode()? **===========================================================================**

**String one line que.**

java.lang.String class is used to create a string object.

**Different String methods:**

* **compareTo**- The Java String compareTo() method is used for comparing two strings  lexicographically.
* **boolean equals()**- The java string equals() method compares the two given strings  based on the content of the string (case sensitive)
* **String concat()**– concat two strings
* **boolean equalsIgnoreCase()**- The java string equals() method compares the two given  strings based on the content of the string (not casesensitive)
* **char charAt()**– index position - The java string charAt() method returns a char value  at the given index number.
* **boolean contains()**
* **toUpperCase()**– convert to upper case
* **toLowerCase()**– convert to lower case

* **trim()**– remove spaces from both sides of string
* **substring()**-- returns part of string
* **boolean endsWith()**
* **boolean startWith()**– ends with specified suffix or not
* **int length()**
* **replace()**
* int num = Integer.parseInt(str);- **Convert String to int using Integer.parseInt(String)**• int num = Integer.valueOf(str);- **Convert String to int using Integer.valueOf(String)**• **Convert int to String using**

**String.valueOf()**String int ivar = 123;

String str = String.valueOf(ivar);

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

System.out.println(555+str);

* **Convert int to String using Integer.toString()**

int ivar = 123;

String str = Integer.toString(ivar);

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

System.out.println(555+str);

* In java, string objects are immutable. Immutable simply means unmodified or  unchangeable.Once string object is created its data or state can't be changed but a new  string object is created.

**====================================================================== Array one line que.**

* **Write down syntax of iterator function?**

Iterator<String> it = studentList.iterator();  while(it.hasNext()){ System.out.println(it.next());

}

* **How to find max min of an unsorted array?**

**MAX**

public class maxmin {

public static void main(String[] args) { int [] arr = {1, 45, 67, 98, 455, 678};  int max = Integer.MIN\_VALUE;

for ( int element :

arr){

if(element>max)

{ max=element;

}

}

System.out.println(“Max element is " + max);

}}

**MIN**

public class maxmin {

public static void main(String[] args) { int [] arr = {1, 45, 67, 98, 455, 678,-6};

int min = Integer.MAX\_VALUE;

for ( int element : arr){

if(element<min){

min=element;

}

}

System.out.println(“Min element is " + min);

}}

=================================================================== 1. **How to reverse any array?**public class reverse array {

public static void main(String[] args) { int [] arr = {1, 45, 67, 98, 455, 678};  int l = arr.length;

int n =

Math.floorDiv(l,2); int

temp;

for(int i=0; i<n;i++){ temp= arr[i]; arr[i]= arr[l-i-1]; arr[l-i-1]= temp;

}

for(int element:arr)

{

System.out.print( element + " ");

}

}}

================================================================ public class reverse array {

public static void main(String[] args) { int [] Array ={7,8,9,3,4,6,11,67,98};  int k=Array.length-1; for(k=Array.length-1;k>=0;k--){

System.out.print( Array[k] + " ");

}

}}

===================================================================

1. **How to remove duplicate elements from ArrayList?**we can handle this scenario via  **LikedHashSet**

ArrayList<Integer> numbers = new

ArrayList<Integer>(Arrays.asList(1,2,2,4,6,7,2,3,5,4,3,8,2,8)); LinkedHashSet<Integer> linkedHashSet = new LinkedHashSet<Integer>(numbers); ArrayList<Integer>  numbersListWithoutDuplicate = new ArrayList<Integer>(LinkedHashSet));  System.out.println(numbersListWithoutDuplicate);

**Also we can handle this via stream**

ArrayList<Integer> marksList = new

ArrayList<Integer>(Arrays.asList(1,2,2,4,6,7,2,3,5,4,3,8,2,8)); List<Integer>  marksListUnique= marksList.stream().distnict().collect(Collectors.toList()); System.out.println(marksListUnique);

1. **How to compare two array list?**Via Collection.sort(); and equal
2. **How to find additional element in list while comparing 2 List?**If we have 2 list l1 & l2 , first we remove all element of l2

L1.removeAll(l2):

Sysout(L1) – you will get additional element.

1. **How to find common element in list while comparing 2 List?**L1.retainAll(L2);

Sysout(L1) – you will get common element.

1. **How will you print length of string without using length method.**String str = “Pankaj”

Sysout(str.toCharArray().length);

Sysout(str.lastIndexOf(“”));

1. **How to find missing element in integer array?**
2. **How to reverse a string?**

String str = “Pankaj”; int len = str.length();

String rev = ” ”

for(int i<len-1 , i>=0, i-

-){ rev = rev +

str.charAt(i);

}

Sysout(rev);



Create a string-> create new stringBuffer and here you can apply reverse fuction. String str = “Pankaj”;

StringBuffer sf = new

StringBuffer(s);

Sysout(sf.reverse());

1. **How will you remove special/junk char from string?**

We have to use regular expression [a-z, 0-9, A-Z]

String str = “Y^%^\*%&\*^\*(&\*(Pankaj”;

FPT Software Academ – #Code Your Life

Str= Str.replaceAll([^a-z, 0-9, A-Z], “”);

Sysout(str);

1. **How to reverse an Integer?**

int num = 12345; int rev = 0;

while(num !=0){

rev =rev \*10+ num %

10; num = num/10;

}

Sysout (rev)

}

======================================================================== **//How to handle alert in Selenium write the syntax.**

public boolean isAlertPresent() {

try

{

driver.switchTo().alert();

return true;

}

catch (Exception e)

{

return false;

}

}

----------------------------------------------------------------------------------------------------------------------------- **@Test // How can you switch to alert in selenium.**

public void test4() {

driver.switchTo().alert().accept();

driver.switchTo().alert().dismiss();

Alert text = driver.switchTo().alert();

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

}

------------------------------------------------------------------------------------------------------------------------------- **@Test // How can you switch to frame in selenium**

public void test5() {

driver.switchTo().frame(0);

driver.switchTo().frame("frame");

driver.switchTo().frame(driver.findElement(By.xpath("//input[@title='Search']"))); driver.switchTo().defaultContent();

}

---------------------------------------------------------------------------------------------------------------- **@Test // How can you handle windows in selenium**

public void test6() {

String pwindow = driver.getWindowHandle();

**// To get the window handle of single parent window**

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

// **To get the window handles of all open windows in**

**browser.**Iterator<String> allWindow =

allWindows.iterator();

// **Apply iterator method to iterate on open windows.**

String parentWindow = allWindow.next();

// **getting the next window handle by using .next()**

**method**

driver.switchTo().window(parentWindow

);

// **After getting window handle switching to that**

**window.**String childWindow =

allWindow.next();

driver.switchTo().window(childWindow)

; driver.switchTo().defaultContent();

// **this command will get us back to parent window from child window.**}

------------------------------------------------------------------------------------------------------------------------------- **@Test // How to use waits in selenium**

public void test7() {

**// below one is implicit wait which is applicable for all the we elements**driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);

**// below one is Explict wait syntax , first we need to create object of  WebDriverWait.**

WebDriverWait wait = new WebDriverWait(driver, 10);

**// here you can do your validation of your action whatever you want to check.**

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//input[@title='Searc h']")));

}

------------------------------------------------------------------------------------------------------------------------------- **@Test // How to mouse hover in selenium & drag drop**

public void test8() {

// **To do mouse hover & drag drop first we need to create object of  Actions class for web driver instance**

Actions act = new Actions(driver);

// **now we have to use moveToElement method of action**

**class. //And to complete the action we need to use build &**

**perform method**

act.moveToElement(driver.findElement(By.xpath("//input[@title='Search']"))).build().perform(); act.dragAndDrop(driver.findElement(By.xpath("Source")),driver.findElement(By.xpath("target"))); }

------------------------------------------------------------------------------------------------------------------------------- **@Test // How to select options from drop down**

public void test9() {

**// To select options from drop down first we need to create object of select  class**

Select sel = new Select(driver.findElement(By.xpath("")));

sel.selectByIndex(4); **// to select item by index**

sel.selectByValue(""); **// to select item by value**

sel.selectByVisibleText("text"); **// to select item by visible**

**text**

java.util.List<WebElement> allItems = sel.getOptions(); **// To get all items of drop down**

}

------------------------------------------------------------------------------------------------------------------------------- **@Test // How to get screenshots in selenium**

public void test10() {

// **Directly we cannot take screenshots in selenium, we have to cast the  driver first with takescreenshot class**

File shot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE); **// Now you have to store file in your drive location.**

FileUtils.copyFile(shot, new File("D:\\shot1.jpg"));

}

------------------------------------------------------------------------------------------------------------------------------- ---

**@Test // How to X & Y coordinates of window in selenium**

public void test11() {

driver.manage().window().getPosition().getX();

driver.manage().window().getPosition().getY();

}

-------------------------------------------------------------------------------------------------------------- **@Test // How you can count similar type of objects in web page.**

public void test12() {

int size = driver.findElements(By.xpath("")).size();

}

------------------------------------------------------------------------------------------------------------------------ -

**@Test // How to execute java script in selenium?**

public void test13() {

**//Directly we cannot use Javascript executer in selenium, we have to cast the  driver first with Javascript executer class**

JavascriptExecutor jse =((JavascriptExecutor)driver);

jse.executeScript("window.scrollBy(0,600)");

}

---------------------------------------------------------------------------------------------------------------------------- **@Test // How to read data from excel files using selenium?**

public void test14() {

Workbook workbook = WorkbookFactory.create(new FileInputStream(file)); Sheet sheet = workbook.getSheetAt(0);

}

**@Test // How to connect to database using selenium?**

public void test15() {

DriverManager.getConnection(URL, "username", "password");

}

------------------------------------------------------------------------------------------------------------------------------- **ABSTRACTION**

In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,) in a  Page Class. We utilize these locators in tests, but we can’t see these locators in the tests.  Literally we hide the locators from the tests.

Abstraction is the methodology of hiding the implementation of internal details and  showing the functionality to the users.

**INTERFACE**

Basic statement we all know in Selenium is WebDriver driver = new FirefoxDriver();

WebDriver itself is an Interface. So based on the above statement WebDriver driver = new  FirefoxDriver(); we are initializing Firefox browser using Selenium WebDriver. It means we are  creating a reference variable (driver) of the interface (WebDriver) and creating an Object. Here  WebDriver is an Interface as mentioned earlier and FirefoxDriver is a class.

An interface in Java looks like a class but both the interface and class are two different  concepts. An interface can have methods and variables just like the class, but the methods  declared in interface are by default abstract. We can achieve 100% abstraction and multiple  inheritance in Java with Interface.

**INHERITANCE**

We create a Base Class in the Framework to initialize WebDriver interface, WebDriver waits,  Property files, Excels, etc., in the Base Class.

We extend the Base Class in other classes such as Tests and Utility Class. Extending one  class into other class is known as Inheritance.

**POLYMORPHISM**

Combination of overloading and overriding is known as Polymorphism. We will see both  overloading and overriding below.

Polymorphism allows us to perform a task in multiple ways.

**METHOD OVERLOADING**

We use implicit wait in Selenium. Implicit wait is an example of overloading. In Implicit wait we use  different time stamps such as SECONDS, MINUTES, HOURS etc.,

A class having multiple methods with same name but different parameters is called Method Overloading **METHOD OVERRIDING**

We use a method which was already implemented in another class by changing its parameters. To  understand this you need to understand Overriding in Java.

Declaring a method in child class which is already present in the parent class is called Method  Overriding. Examples are get and navigate methods of different drivers in Selenium.

**ENCAPSULATION**

All the classes in a framework are an example of Encapsulation. In POM classes, we declare the data  members using @FindBy and initialization of data members will be done using Constructor to utilize  those in methods.

Encapsulation is a mechanism of binding code and data together in a single unit. I would like to discuss some other topics which we use in Automation Framework. **WEB ELEMENT:**

Web element is an interface used to identify the elements in a web page.

**WEBDRIVER:**

WebDriver is an interface used to launch different browsers such as Firefox, Chrome, Internet Explorer,  Safari etc.,

**FIND BY:**

FindBy is an annotation used in Page Object Model design pattern to identify the elements. **FIND ELEMENT:**

Find Element is a method in POM to identify the elements in a web page.

**How to find duplicate char using hashmap?**

**Find the count of char using hashmap?**

Top 7 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) Submitting a form***

// submit the form

driver.findElement(By.<em>id</em>("submit")).submit();

***#5) Handling iframes***

**Select iframe by id**

*driver.switchTo().frame(“ID of the frame“);*

**Locating iframe using tagName**

*driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));***Locating iframe using the index:**

**a)frame(index)**

*driver.switchTo().frame(0);*

**b)frame(Name of Frame)**

*driver.switchTo().frame(“name of the frame”);*

**c)frame(WebElement element)**Select

Parent Window

*driver.switchTo().defaultContent();*

***#6) close() and quit() methods***

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

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

***#7) Exception Handling***

WebElement saveButton = driver.findElement(By.id("Save"));

try{

if(saveButton.isDisplayed()){

saveButton.click();

}

}

catch(NoSuchElementException e){

e.printStackTrace();

}

***#4) 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. driver.manage().timeouts().implicitlyWait(1000, TimeUnit.SECONDS);

**untill() from WebdriverWait and visibilityOfElementLocated()**from ExpectedConditions to wait

explicitly till an element is visible in the webpage.

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated  (By.xpath("//input[@id=’name’]")));

**untill() from WebdriverWait and alertIsPresent() from ExpectedConditions to wait explicitly till  an alert appears.**

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.alertIsPresent()

);

**Select class for selecting and deselecting values from the drop-down in Selenium WebDriver.**WebElement mySelectedElement = driver.findElement(By.id("select"));

Select dropdown= new Select(mySelectedElement);dropdown.selectByVisibleText("Apple");

**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 to popup  alerts and handle them.

Alert alert = driver.switchTo().alert();

alert.sendKeys("This Is Softwaretestinghelp");

alert.accept()

**getWindowHandle() and getWindowHandles()**to handle Multiple Windows in Selenium WebDriver. 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" )

**POI to read from the excel files.**

Workbook workbook = WorkbookFactory.create(new FileInputStream(file));

Sheet sheet = workbook.getSheetAt(0);

**Asserts using assertEquals(),assertNotEquals(), assertTrue() and assertFalse() to compare the  results.**

Assert.assertEquals(message, “This text”);

Assert.assertNotEquals(message, “This text”);

Assert.assertTrue(result<0);

Assert.assertFalse(result<0);

**As a tester what should be your approach when requirements change continuously?**When requirement keeps changing, continuously agile tester should take following approach

* Write generic test plans and test cases, which focuses on the intent of the  requirement rather than its exact details
* To understand the scope of change, work closely with the product owners or business analyst • Make sure team understand the risks involved in changing requirements especially  at the end of the sprint
* Until the feature is stable, and the requirements are finalized, it is best to wait if  you are going to automate the feature
* Changes can be kept to a minimum by negotiating or implement the changes in the next sprint

**List out the pros and cons of exploratory testing (used in Agile) and scripted testing?**

|  |  |  |
| --- | --- | --- |
|  | **Pros** | **Cons** |
| Exploratory  Testing | - It requires less preparation- Easy to modify  when requirement changes- Works well when  documentation is scarce | - Presenting progress and Coverage to project management is difficult |
| Scripted  Testing | - In case testing against legal or regulatory  requirements it is very useful | - Test preparation is usually time-consuming Same  steps are tested over and again- When requirement  changes it is difficult to modify |

**Explain the difference between Extreme programming and Scrum?**

|  |  |
| --- | --- |
| Scrum | Extreme Programing (XP) |
| - Scrum teams usually have to work in iterations called sprints which  usually last up to two weeks to one month long | - XP team works in iteration that last for one or two weeks |
| - Scrum teams do not allow change into their sprints | - XP teams are more flexible and change their iterations |
| - In scrum, the product owner prioritizes the product backlog but the  team decides the sequence in which they will develop the backlog items | - XP team work in strict priority order, features developed are prioritized by the customer |
| - Scrum does not prescribe any engineering practices | - XP does prescribe engineering practices |

**4) 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

**5) Explain what is re-factoring?**

To improve the performance, the existing code is modified; this is re-factoring. During re factoring the code functionality remains same

**6) Explain how you can measure the velocity of the sprint with varying team capacity?**

When planning a sprint usually, the velocity of the sprint is measured on the basis of  professional judgement based on historical data. However, the mathematical formula used to  measure the velocity of the sprint are,

* first – completed story points **X**team capacity: If you measure capacity as a  percentage of a 40 hours weeks
* Second – completed story points **/**team capacity: If you measure capacity in man-hours For our scenario second method is applicable.

**7) Mention the key difference between sprint backlog and product backlog?**Product backlog: It contains a list of all desired features and is owned by the product owner.

Sprint backlog: It is a subset of the product backlog owned by development team and  commits to deliver it in a sprint. It is created in Sprint Planning Meeting

**8) In Agile mention what is the difference between the Incremental and Iterative development?**

**Iterative:**Iterative method is a continuous process of software development where the  software development cycles are repeated (Sprint & Releases) till the final product is  achieved.

Release 1: Sprint 1, 2… n

Release n: Sprint 1, 2….n

**Incremental:**Incremental development segregates the system functionality into increments or portions.  In each increment, each segment of functionality is delivered through cross-discipline work, from the  requirements to the deployment.

**9) Explain what is Spike and Zero sprint in Agile? What is the purpose of it?**

**Sprint Zero:**It is introduced to perform some research before initiating the first sprint. Usually  this sprint is used during the start of the project for activities like setting development  environment, preparing product backlog and so on.

**Spikes:**Spikes are type of stories that are used for activities like research, exploration,  design and even prototyping. In between sprints, you can take spikes for the work related to  any technical or design issue. Spikes are of two types Technical Spikes and Functional  Spikes.

**10) What is test driven development?**

Test driven development or TDD is also known as test-driven design. In this method, developer  first writes an automated test case which describes new function or improvement and then  creates small codes to pass that test, and later re-factors the new code to meet the acceptable  standards.

**11) Prototypes and Wireframes are widely used as part of?**

Prototypes and Wireframes are prototypes that are widely used as part of Empirical Design. **12) Explain what is Application Binary Interface?**

Across different system platforms and environments a specification defining requirements  for portability of applications in binary form is known as Application Binary Interface.

**13) Explain in Agile, burn-up and burn-down chart?**

To track the project progress burnup and burn down, charts are used.

Burnup Chart: It shows the progress of stories done over time.

Burndown Chart: It shows how much work was left to do overtime.

**14) Explain what is Scrum ban?**

Scrum ban is a software development model based on Scrum and Kanban. It is specially  designed for project that requires frequent maintenance, having unexpected user stories and  programming errors. Using these approach, the team’s workflow is guided in a way that allows  minimum completion time for each user story or programming error.

**15) What is story points/efforts/ scales?**

It is used to discuss the difficulty of the story without assigning actual hours. The most common  scale used is a Fibonacci sequence ( 1,2,3,5,8,13,….100) although some teams use linear  scale (1,2,3,4….), Powers of 2 (1,2,4,8……)

and cloth size (XS, S ,M,L, XL).

**16) Explain what is tracer bullet?**

The tracer bullet is a spike with the current architecture, the current set of best  practices, current technology set which results in production quality code. It is not a  throw away code but might just be a narrow implementation of the functionality.

**17) What is a test stub?**

A test stub is a small code that replaces an undeveloped or fully developed  component within a system being tested. Test stub is designed in such a way that  it mimics the actual component by generating specifically known outputs and  substitute the actual component.

**18) What are the differences between RUP (Rational Unified Process) and Scrum  methodologies?**

|  |  |
| --- | --- |
| RUP | SCRUM |
| - Formal Cycle is defined across four phases, but some workflows can  be concurrent | - Each sprint is a complete cycle |
| - Formal project plan, associated with multiple iterations is used. | - No end to end project plan. Each next iteration plan is determined at the end of the current iteration |
| - Scope is predefined ahead of the project start and documented in  the scope document. During the project, scope can be revised. | - It uses a project backlog instead of scope scrum |
| - Artifacts include Scope Document, formal functional requirements  package, system architecture document, development plan, test  scripts, etc. | - Operational software is the only formal artifacts |
| - Recommended for long term, large, enterprise level projects with  medium to high complexity | - Recommended for quick enhancements and organization that are not dependent on a deadline |

**19) Why Continuous Integration is important for Agile?**

Continuous Integration is important for Agile for following reasons.

* It helps to maintain release schedule on time by detecting bugs or integration errors • Due to frequent agile code delivery usually every sprint of 2-3 weeks, stable  quality of build is a must and continuous integration ensures that
* In helps to maintain the quality and bug free state of code-base
* Continuous integration helps to check the impact of work on branches to the main  trunk if development work is going on branches using automatic building and merging  function

**20)What testing is done during Agile?**

The primary testing activities during Agile is automated unit testing and exploratory testing.

Though, depending on project requirements, a tester may execute Functional and Non functional tests on the Application Under Test (AUT).

**21) Explain what is Velocity in Agile?**

Velocity is a metric that is calculated by addition of all efforts estimates related with user  stories completed in an iteration. It figures out how much work Agile can complete in a sprint  and how much time will it need to finish a project.

**22) What are the qualities of a good Agile tester should have?**

A good Agile tester should have following qualities

* It should be able to understand the requirements quickly
* Agile tester should know Agile principals and concepts well
* As requirements keep changing, tester should understand the risk involve in it • Based on the requirements Agile tester should be able to prioritize the work • Continue communication between business associates, developers and tester is must

**23)Who are all involved in the Agile team?**

In agile the two main leads are

* **Scrum Masters**: It coordinates most of the inputs and outputs required for an agile  program
* **Development Managers**: They hire right people and develop them with the team **24)Mention in detail what are the role’s of Scrum Master?**

Scrum Master key responsibilities involves

* Understand the requirements and turn them into working software
* Monitoring and Tracking
* Reporting and Communication
* Process Check Master
* Quality Master
* Resolve Impediments
* Resolve Conflicts
* Shield the team and performance feedback
* Lead all the meetings and resolve obstacles

**25) Mention what are the Agile quality strategies?**

Agile quality strategies are

* Re-factoring
* Non-solo development
* Static and dynamic code analysis
* Reviews and Inspection
* Iteration/sprint demos
* All hands demo
* Light weight milestone reviews
* Short feedback cycles
* Standards and guidelines

**26)Mention what are the Tools that can be useful for screenshots while working on Agile  projects?**

While working on Agile projects you can use tools like

* BugDigger
* BugShooting
* qTrace
* Snagit
* Bonfire
* Usersnap

**27)Mention what are the advantages of maintaining consistent iteration length  throughout the project?**

The advantages are

* It helps team to objectively measure progress
* It provides a consistent means of measuring team velocity
* It helps to establish a consistent pattern of delivery

**28)If a timebox plan needs to be reprioritized who should re-prioritise it?**

If a timebox plan needs to be reprioritized it should include whole team, product owner, and  developers.

**29) Mention what should a burndown chart should highlight?**

The burn-down chart shows the remaining work to complete before the timebox (iteration) ends. **30)Mention what is the difference between Scrum and Agile?**

* **Scrum**: In the scrum, a sprint is a basic unit of development. Each sprint is followed by a  planning meeting, where the tasks for the sprint are identified and estimated. During  each sprint, the team creates finished portion of a product
* **Agile**: In Agile, each iteration involves a team working through a full software  development cycle, including planning, design, coding, requirement analysis, unit  testing, and acceptance testing when a product is demonstrated to stakeholders

In simple words, Agile is the practice and scrum is the process to following this practice.

**31) Mention what are the challenges involved in AGILE software development?**Challenges involved in Agile Software development includes

* It requires more testing and customers involvement
* It impacts management more than developers
* Each feature needs to be completed before moving on to the next
* All the code has to work fine to ensure application is in working state
* More planning is required

**32)When not to use Agile?**

Before using Agile methodology, you must ask following questions

* Is functionality split-able
* Is customer available
* Are requirements flexible
* Is it really time constrained
* Is team skilled enough

**33)Explain how can you implement scrum in an easy way to your project?**These are the tips which can be helpful to implement scrum in your project.

* Get your backlog in order
* Get an idea of the size of your product backlog items
* Clarify sprint requirement and duration to complete the sprint backlog • Calculate the team sprint budget and then break requirements into tasks • Collaborate workspace- a center of all team discussion, which includes plans,

roadmaps, key dates, sketches of functionality, issues, log, status reports, etc. • Sprint- Make sure you complete one feature at a time before moving on to the next. A  sprint should not be abort unless if there is no other option

* Attend a daily stand-up meeting: In meeting you need to mention, what have been  achieved since the last meeting, what will they achieve before the next meeting and is  anything holding up their progress
* Use burndown chart to track daily progress. From the burndown chart, you can estimate  whether you are on track, or you are running behind
* Complete each features well before moving on to the next
* At the end of the sprint- hold a sprint review meeting, mention what is achieved or  delivered in the sprint.

**34) Explain what does it mean by product roadmap?**

A product roadmap is referred for the holistic view of product features that create the product  vision.

**Các câu hỏi về tính cách và định hướng trong tương lai**

Trong phần cuối cùng của buổi tuyển dụng Automation Tester, bạn có thể gặp các câu hỏi về tính cách và định hướng tương lai. Bạn hãy cho nhà tuyển dụng thấy mình là người cầu tiến, có khả năng làm việc trong môi trường cộng tác cũng như có ý định làm việc lâu dài với công ty.

Automation Tester là nghề nghiệp triển vọng cho các bạn sinh viên IT tại Việt Nam. Các bạn hãy chuẩn bị thật kỹ những tình huống có thể xảy ra để vượt qua các vòng **tuyển dụng Automation Tester**một cách tốt nhất.

***Và dưới đây là Anh Tester chọn ra 10 câu hỏi cần quan tâm cho các bạn:***

1. [***How to locate an element by partially matching the attribute’s value using Xpath?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#partial_match_xpath)
2. [***How can we move to the parent of an element using XPath?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#parent_of_element_xpath)
3. [***What is the fundamental difference between XPath and CSS selectors?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#difference_between_xpath_css_selector)
4. [***How to switch between multiple windows in Selenium?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#multiple_windows_selenium)
5. [***Write the code to double-click an element.***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#double_click_selenium)
6. [***What are some commonly encountered exceptions in Selenium?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#exceptions_in_selenium)
7. [***How can we capture screenshots using Selenium?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#screenshot_in_selenium)
8. [***What is the difference between driver.findElement() and driver.findElements()?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#findElement_findElements)
9. [***How to do drag and drop in Selenium?***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#drag_and_drop)
10. [***Explain the line of code Webdriver driver = new FirefoxDriver();***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#webdriver_driver_new_firefox_driver)

***Các phần chính trong 100 câu về Selenium gồm:***

1. [***Selenium Interview Questions***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#Selenium_Interview_Questions)
2. [***Selenium Java Interview Questions***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#Selenium_Java_Interview_Questions)
3. [***Selenium Framework Interview Questions***](https://anhtester.com/blog/100-cau-hoi-va-cau-tra-loi-phong-van-ve-selenium-b321.html#Selenium_Framework_Interview_Questions)

**Selenium Interview Questions**

**Ques.1. What is Selenium?**

Ans. Selenium is a robust test automation suite that is used for automating web-based applications. It supports multiple browsers, programming languages, and platforms.

**Ques.2. What are the different forms of Selenium?**

Ans. Selenium comes in four forms-

1. Selenium WebDriver – Selenium WebDriver is used to automate web applications by directly calling the browser’s native methods.
2. The Selenium IDE Plugin – Selenium IDE is an open-source test automation tool that works on record and playback principles.
3. Selenium RC component – Selenium Remote Control(RC) is officially deprecated by Selenium and it used to work using javascript to automate the web applications.
4. Selenium Grid – Allows Selenium tests to run in parallel across multiple machines.

**Ques.3. What are some advantages of Selenium?**

Ans. Following are the advantages of Selenium-

1. Selenium is open source and free to use without any licensing cost.
2. It supports multiple languages like Java, Ruby, Python, etc.
3. Selenium supports multi-browser testing.
4. It has vast resources and helping community over the internet.
5. Using Selenium IDE component, non-programmers can also write automation scripts.
6. Using the Selenium Grid component, distributed testing can be carried out on remote machines.

**Ques.4. What are some limitations of Selenium?**

Ans. Following are the limitations of Selenium-

1. We cannot test desktop applications using Selenium.
2. We cannot test web services using Selenium.
3. For creating robust scripts in Selenium Webdriver, programming language knowledge is required.
4. Also, we have to rely on external libraries and tools for performing tasks like – logging(log4J), testing framework-(TestNG, JUnit), reading from external files(POI for excels), etc.

**Ques.5. Which browsers/drivers are supported by Selenium Webdriver?**

Ans. Some commonly used browsers supported by Selenium are-

1. Google Chrome – ChromeDriver
2. Firefox – FireFoxDriver
3. Internet Explorer – InternetExplorerDriver
4. Safari – SafariDriver
5. HtmlUnit (Headless browser) – HtmlUnitDriver
6. Android – Selendroid/Appium
7. IOS – ios-driver/Appium

**Ques.6. Can we test APIs or web services using Selenium Webdriver?**

Ans. No. Selenium WebDriver uses the browser’s native method to automate the web applications. So, there is no support for testing web services using Selenium WebDriver.

**Ques.7. What are the various ways of locating an element in Selenium?**

Ans. The different locators in Selenium are-

1. Id
2. XPath
3. CSS selector
4. className
5. tagName
6. name
7. link text
8. partialLinkText

**Ques.8. How can we inspect the web element attributes in order to use them in different locators?**

Ans. In order to locate web elements, we can use the Developer tool and plugins like Firebug.  
The developer tool can be launched by pressing F12 on the browser. Users can easily hover over any element and find its different HTML properties.  
  
Firebug is a plugin of Firefox that provides various development tools for debugging applications. From an automation perspective, Firebug is used specifically for inspecting web elements in order to find their attributes like id, class, name, etc. in different locators.

**Ques.9. What is an XPath?**

Ans. Xpath or XML path is a query language that is used for selecting nodes from XML documents. Also, it is one of the locators supported by Selenium Webdriver.

**Ques.10. What is an absolute XPath?**

Ans. An absolute XPath is a way of locating an element using an XML expression, beginning from the root node i.e. HTML node in the case of web pages.  
  
The main disadvantage of absolute XPath is that even if there is a slight change in the UI or any element then also whole XPath will fail.  
Example – html/body/div/div[2]/div/div/div/div[1]/div/input

**Ques.11. What is a relative XPath?**

Ans. A relative XPath is a way of locating an element using an XML expression, starting from anywhere in the HTML document.  
  
In this way, there are different ways of creating robust relative XPaths that are unaffected by changes in other UI elements.  
Example – //input[@id=’username’]

**Ques.12. What is the difference between single slash(/) and a double slash(//) in XPath?**

Ans. In XPath, a single slash is used for creating absolute XPaths, beginning from the root node. Whereas double slash is used for creating relative XPaths.

**Ques.13. How can we locate an element by only partially matching the value of its attributes in Xpath?**

Ans. Using contains() method we can locate an element by partially matching its attribute’s value. This is particularly helpful in scenarios where the attributes have dynamic values with a certain constant part.

xPath expression = //\*[contains(@name,'user')]

Basically, the above statement will match all the values of the name attribute containing the word ‘user’ in them.

**Ques.14. How can we locate elements using their text in XPath?**

Ans. Using the text() method –

xPathExpression = //\*[text()='username']

**Ques.15. How can we move to the parent of an element using XPath?**

Ans. Using ‘/..’ after the XPath expression of the child element, we can move to the parent of an element.  
For example, the locator **//div[@id=”childId”]/..** will move to the parent of the div element with id value as ‘childId’.

**Ques.16. How can we move to the nth-child element using XPath?**

Ans. Basically, there are two ways of navigating to the nth element using XPath-

* Using square brackets with index position-  
  Example – div[2] will find the second div element.
* Using position()-  
  Example – div[position()=3] will find the third div element.

**Ques.17. What is the syntax of finding elements by class using CSS Selector?**

Ans. By using **.className** in the CSS locator, we can select all the elements belonging to a particular class e.g. ‘.red’ will select all elements having class ‘red’.

**Ques.18. What is the syntax of finding elements by id using CSS Selector?**

Ans. By using **#idValue** in the CSS locator, we can select all the elements belonging to a particular class e.g. ‘#userId’ will select the element having an id – userId.

**Ques.19. How can we select elements by their attribute value using the CSS Selector?**

Ans. Using **[attribute=value]** in the CSS locator, we can select all the elements belonging to a particular class e.g. ‘[type=small]’ will select the element having attribute type of value ‘small’.

**Ques.20. How can we move to the nth-child element using the CSS selector?**

Ans. Using**:nth-child(n)** in the CSS locator, we can move to the nth child element e.g. div:nth-child(2) will locate 2nd div element of its parent.

**Ques.21. What is the fundamental difference between XPath and CSS selectors?**

Ans. The fundamental difference between XPath and CSS selector is – using XPaths we can traverse up in the document i.e. we can move to parent elements. Whereas using the CSS selector, we can only move downwards in the document.

**Selenium Java Interview Questions**

**Ques.22. How can we launch different browsers in Selenium WebDriver?**

Ans. By creating an instance of the desired browser driver e.g. below command will initialize the Firefox browser.

WebDriver driver = new FirefoxDriver();

**Ques.23. What is the use of driver.get(“URL”) and driver.navigate().to(“URL”) commands? Is there any difference between the two?**

Ans. Both driver.get(“URL”) and driver.navigate().to(“URL”) commands are used to navigate to a URL passed as parameter.  
There is a minor difference between the two commands-

1. driver.navigate() allows moving back and forward in browser history with the help of driver.navigate().forward() and driver.navigate().back() commands.
2. In the case of single-page applications (where the URL is appended by ‘#’ to navigate to different sections of the page), driver.navigate().to() navigates to a particular section by changing the URL without refreshing the page whereas driver.get() refreshes the page also.  
     
   This refreshing of the page is also the primary reason because of which history is not maintained in the case of the driver.get() command.  
   **Reference –**[**Stack overflow**](https://stackoverflow.com/questions/33865618/difference-between-webdriver-get-and-webdriver-navigate-to-in-the-case-of-ur/33868976#33868976)

**Ques.24. How can we type text in a textbox element using Selenium?**

Ans. With the help of sendKeys() method we can type text in a textbox-

WebElement searchTextBox = driver.findElement(By.id("srch"));

searchTextBox.sendKeys("searchTerm");

Copy

**Ques.25. How can we clear a text written in a textbox?**

Ans. In order to delete the text written in a textbox, we can use the clear() method.

driver.findElement(By.id("elementLocator")).clear();

**Ques.26. How to check a checkBox in Selenium?**

Ans. The same click() method used for clicking buttons or radio buttons can be used for checking the checkbox as well.

**Ques.27. How can we submit a form in Selenium?**

Ans. Using the submit() method we can submit a form in selenium.

driver.findElement(By.id("form1")).submit();

Also, the click() method can be used for the same purpose.

**Ques.28. Explain the difference between close and quit command.**

Ans. The difference between close and quit command is-  
driver.close() – Used to close the current browser having a focus.  
driver.quit() – Used to close all the browser instances.

**Ques.29. How to switch between multiple windows in Selenium?**

Ans. Selenium has **driver.getWindowHandles()** and **driver.switchTo().window(“{windowHandleName}”)** commands to work with multiple windows.  
  
The getWindowHandles() command returns a list of ids corresponding to each window. If we pass a particular window handle to the **driver.switchTo().window(“{windowHandleName}”)** command then we can switch control/focus to that particular window.

for (String windowHandle : driver.getWindowHandles()){

driver.switchTo().window(handle);

}

Copy

**Ques.30. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in Selenium?**

Ans. The driver.getWindowHandle() returns a handle of the current window (a single unique identifier).  
Whereas driver.getWindowHandles() returns a set of handles of all the windows available.

**Ques.31. How can we move to a particular frame in Selenium?**

Ans. The driver.switchTo() commands can be used for switching to a particular iframe.

driver.switchTo().frame("{frameIndex/frameId/frameName}");

For locating a frame, we can either use the index (starting from 0), its name, or its Id.  
**Ques.32. Can we move back and forward in the browser using Selenium?**

Ans. Yes, using driver.navigate().back() and driver.navigate().forward() commands, we can move backward and forward in a browser.  
**Ques.33. What are the different ways to refresh a browser?**

Ans. There a multiple ways to refresh a page in Selenium-

* Using *driver.navigate().refresh()* command.
* Using *sendKeys(Keys.F5)* on any textbox on the webpage.
* Using driver.get(“URL”) on the current URL or using *driver.getCurrentUrl()*.
* Using driver.navigate().to(“URL”) on the current URL or driver.navigate().to(driver.getCurrentUrl());

**Ques.34. How can we maximize the browser window in Selenium?**

Ans. We can maximize the browser window using the following command-

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

**Ques.35. How can we fetch a text written over an element?**

Ans. Using the ***getText()*** method we can fetch the text over an element.

String text = driver.findElement("elementLocator").getText();

**Ques.36. How can we find the value of different attributes like name, class, value of an element?**

Ans. Using *getAttribute(“{attributeName}”)* method, we can find the value of different attributes of an element e.g.-

String valueAttribute = driver.findElement(By.id("locator")).getAttribute("value");

Copy

**Ques.37. How to delete cookies in Selenium?**

Ans. Using deleteAllCookies() method.

driver.manage().deleteAllCookies();

**Ques.38. What is an implicit wait in Selenium?**

Ans. An implicit wait is a type of wait that waits for a specified time while locating an element before throwing NoSuchElementException. By default, Selenium tries to find web elements immediately when required without any wait. So, it is good to use implicit wait. This wait is applied to all the elements of the current driver instance.

driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

**Ques.39. What is an explicit wait in Selenium?**

Ans. An explicit wait is a type of wait that is applied to a particular web element until the expected condition specified is met.

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("elementId")));

Copy

It is advisable to use explicit waits over implicit waits because higher timeout value of implicit wait (set for handling only some of the elements) gets applied to all the web elements. Thus increasing the overall execution time of the script. On the other hand, we can apply different timeouts to the different elements in case of explicit waits.

**Ques.40. What are some expected conditions that can be used in Explicit waits?**

Ans. Some of the commonly used expected conditions of an element that can be used with explicit waits are-

* elementToBeClickable(WebElement element or By locator)
* stalenessOf(WebElement element)
* visibilityOf(WebElement element)
* visibilityOfElementLocated(By locator)
* invisibilityOfElementLocated(By locator)
* attributeContains(WebElement element, String attribute, String value)
* alertIsPresent()
* titleContains(String title)
* titleIs(String title)
* textToBePresentInElementLocated(By, String)

**Ques.41. What is a fluent wait?**

Ans. A fluent wait is a type of wait in which we can also specify polling interval (the time intervals after which driver will try to find the elements when not located) along with the maximum timeout value.

Wait wait = new FluentWait(driver)

.withTimeout(20, SECONDS)

.pollingEvery(5, SECONDS)

.ignoring(NoSuchElementException.class);

WebElement textBox = wait.until(new Function<webdriver,webElement>()

{

public WebElement apply(WebDriver driver) {

return driver.findElement(By.id("textBoxId"));

}

});

Copy

**Ques.42. What are the different keyboard operations that can be performed in Selenium?**

Ans. The different keyboard operations that can be performed in Selenium are-

1. **.sendKeys(“sequence of characters”)** – Used for passing character sequence to an input or textbox element.
2. **.pressKey(“non-text keys”)** – Used for keys like control, function keys etc that are non-text.
3. **.releaseKey(“non-text keys”)** – Used in conjunction with keypress event to simulate releasing a key from keyboard event.

**Ques.43. What are the different mouse actions that can be performed using Selenium?**

Ans. The different mouse events supported in Selenium are-

1. click(WebElement element)
2. doubleClick(WebElement element)
3. contextClick(WebElement element)
4. mouseDown(WebElement element)
5. mouseUp(WebElement element)
6. mouseMove(WebElement element)
7. mouseMove(WebElement element, long xOffset, long yOffset)

**Ques.44. Write the code to double-click an element.**

Ans. Code to double click an element-

Actions action = new Actions(driver);

WebElement element=driver.findElement(By.id("elementId"));

action.doubleClick(element).perform();

Copy

**Ques.45. Write the code to right-click an element.**

Ans. Code to right-click an element in selenium-

Actions action = new Actions(driver);

WebElement element=driver.findElement(By.id("elementId"));

action.contextClick(element).perform();

Copy

**Ques.46. How to mouse hover an element in Selenium?**

Ans. Code to mouse hover over an element-

Actions action = new Actions(driver);

WebElement element=driver.findElement(By.id("elementId"));

action.moveToElement(element).perform();

Copy

**Ques.47. How to fetch the current page URL in Selenium?**

Ans. In order to fetch the current page URL, we can use the getCurrentURL() command.

driver.getCurrentUrl();

**Ques.48. How can we fetch the title of the page in Selenium?**

Ans. Using **driver.getTitle()** command, we can fetch the page title in Selenium. This method returns a string containing the title of the webpage.

**Ques.49. How can we fetch the page source in Selenium?**

Ans. Using the **driver.getPageSource()** command, we can fetch the page source in selenium. This method returns a string containing the page source.

**Ques.50. How to verify tooltip text using Selenium?**

Ans. Tooltips web elements have an attribute of type ‘title’. By fetching the value of the ‘title’ attribute, we can verify the tooltip text in selenium.

String toolTipText = element.getAttribute("title");

**Ques.51. How to locate a link using its text in Selenium?**

Ans. Using linkText() and partialLinkText() methods, we can locate a link. The difference between the two is – linkText() matches the complete string passed as a parameter to the link texts. Whereas partialLinkText() only matches the string parameter partially.

WebElement link1 = driver.findElement(By.linkText("artOfTesting"));

WebElement link2 = driver.findElement(By.partialLinkText("artOf"));

Copy

**Ques.52. What are DesiredCapabilities in Selenium WebDriver?**

Ans. Desired capabilities are a set of key-value pairs that are used for storing or configuring browser-specific properties. For example – browser’s version, platform, etc in the browser instances.

**Ques.53. How can we find all the links on a web page?**

Ans. All the links are of anchor tag ‘a’. So by locating elements of tagName ‘a’ we can find all the links on a webpage.

List<WebElement> links = driver.findElements(By.tagName("a"));

Copy

**Ques.54. What are some commonly encountered exceptions in Selenium?**

Ans. Some of the commonly seen exceptions in Selenium are-

* NoSuchElementException – When no element could be located by the locator provided.
* ElementNotVisibleException – When an element is present in the DOM but is not visible.
* NoAlertPresentException – When we try to switch to an alert box but the targetted alert is not present.
* NoSuchFrameException – When we try to switch to a frame but the targetted frame is not present.
* NoSuchWindowException – When we try to switch to a window but the targetted window is not present.
* UnexpectedAlertPresentException – When an unexpected alert blocks the normal interaction of the driver.
* TimeoutException – When a command execution gets a timeout.
* InvalidElementStateException – When the state of an element is not appropriate for the desired action.
* NoSuchAttributeException – When we are trying to fetch an attribute’s value but the attribute is not correct.
* WebDriverException – When there is some issue with the driver instance preventing it from getting launched.

**Ques.55. How can we capture screenshots using Selenium?**

Ans. In order to take screenshots in Selenium, we can use the getScreenshotAs method of the TakesScreenshot interface.

File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(scrFile, new File("D:\\AnhTester.jpg"));

Copy

**Ques.56. How to handle dropdowns in Selenium?**

Ans. Using Select class-

Select countriesDropDown = new Select(driver.findElement(By.id("countries")));

dropdown.selectByVisibleText("India");

//or using index of the option starting from 0

dropdown.selectByIndex(1);

//or using its value attribute =

dropdown.selectByValue("Ind");

Copy

**Ques.57. How to check which option in the dropdown is selected?**

Ans. Using is Selected() method, we can check the state of a dropdown’s option.

Select countriesDropDown = new Select(driver.findElement(By.id("countries")));

dropdown.selectByVisibleText("VietNam");

//returns true or false value

System.out.println(driver.findElement(By.id("vietnam")).isSelected());

Copy

**Ques.58. How can we check if an element is getting displayed on a web page?**

Ans. Using the isDisplayed() method we can check if an element is getting displayed on a web page.

driver.findElement(By locator).isDisplayed();

**Ques.59. How can we check if an element is enabled for interaction on a web page?**

Ans. Using the isEnabled method, we can check if an element is enabled or not.

driver.findElement(By locator).isEnabled();

**Ques.60. What is the difference between driver.findElement() and driver.findElements() commands?**

Ans. The difference between driver.findElement() and driver.findElements() commands is-

* findElement() returns a single WebElement (found first) based on the locator passed as a parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
* Syntax of findElement()-  
  WebElement textbox = driver.findElement(By.id(“textBoxLocator”));  
    
  Whereas the syntax of findElements()-  
  List <WebElement> elements = driver.findElements(By.id(“value”));
* Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of 0 elements.

**Ques.61. How can we handle window UI elements and window POP ups using selenium?**

Ans. Selenium is used for automating web-based applications only(or browsers only). If we want to handle window GUI elements then we can use tools like AutoIT.  
  
AutoIT is a freeware used for automating window GUI. The AutoIt scripts follow simple BASIC language like syntax. Also, it can be easily integrated with Selenium tests.

**Ques.62. What is Robot API?**

Ans. Robot API is used for handling Keyboard or mouse events.

Robot robot = new Robot();

//Simulate enter key action

robot.keyPress(KeyEvent.VK\_ENTER);

Copy

**Ques.63. How to do file upload in Selenium?**

Ans. File upload action can be performed in multiple ways-

1. Using element.sendKeys(“path of file”) on the webElement of input tag and type file i.e. the elements should be like –  
   <input type=”file” name=”fileUpload”>
2. With the help of Robot API.
3. Using the AutoIT API.

**Ques.64. How to handle the HTTPS website in Selenium or how to accept the SSL untrusted connection?**

Ans. Using profiles, we can handle accept the SSL untrusted connection certificate. Profiles are basically set of user preferences stored in a file.

FirefoxProfile profile = new FirefoxProfile();

profile.setAcceptUntrustedCertificates(true);

profile.setAssumeUntrustedCertificateIssuer(false);

WebDriver driver = new FirefoxDriver(profile);

**Ques.65. How to do drag and drop in Selenium?**

Ans. Using Action class, [drag and drop can be performed in Selenium](https://artoftesting.com/drag-and-drop-in-selenium-webdriver). Sample code-

Actions builder = new Actions(driver);

Action dragAndDrop = builder.clickAndHold(SourceElement)

.moveToElement(TargetElement)

.release(TargetElement)

.build();

dragAndDrop.perform();

**Ques.66. How to execute JavaScript code in Selenium?**

Ans. JavaScript code can be executed in Selenium using JavaScriptExecuter. Sample code for javascript execution-

WebDriver driver = new FireFoxDriver();

if (driver instanceof JavascriptExecutor) {

((JavascriptExecutor)driver).executeScript("{JavaScriptCode}");

}

**Ques.67. How to handle alerts in Selenium?**

Ans. In order to accept or dismiss an alert box, the alert class is used. This requires first switching to the alert box and then using accept() or dismiss() command as the case may be.

Alert alert = driver.switchTo().alert();

//To accept the alert

alert.accept();

Alert alert = driver.switchTo().alert();

//To cancel the alert box

alert.dismiss();

**Ques.68. What is HtmlUnitDriver?**

Ans. HtmlUnitDriver is the fastest WebDriver. Unlike other drivers (FireFoxDriver, ChromeDriver, etc), the HtmlUnitDriver is non-GUI. On executing test scripts, no browser gets launched.

**Ques.69. How to handle hidden elements in Selenium WebDriver?**

Ans. Using javaScript executor we can handle hidden elements-

(JavascriptExecutor(driver))

.executeScript("document.getElementsByClassName(locator).click();");

**Selenium Framework Interview Questions**

**Ques.70. What is Page Object Model or POM?**

Ans. Page Object Model(POM) is a design pattern in Selenium. A design pattern is a solution or a set of standards that are used for solving commonly occurring software problems.  
  
Now coming to POM – POM helps to create a framework for maintaining selenium scripts. In POM for each page of the application, a class is created having the web elements belonging to the page and methods handling the events on that page. The test scripts are maintained in separate files and the methods of the page object files are called from the test scripts file.  
In this way, we can create a robust automation framework using POM.

**Ques.71. What are the advantages of POM?**

Ans. The advantages are POM are-

1. Using POM, we can create an Object Repository i.e. a set of web elements in separate files along with their associated functions. In this way, keeping the code clean.
2. For any change in UI(or web elements) only page object files are required to be updated leaving test files unchanged.
3. It makes code reusable as well as maintainable.

**Ques.72. What is Page Factory?**

Ans. Page factory is an implementation of the Page Object Model in Selenium. It provides @FindBy annotation to find web elements. In addition, there is a PageFactory.initElements() method to initialize all web elements defined with @FindBy annotation.

public class SamplePage {

WebDriver driver;

@FindBy(id="search") WebElement searchTextBox;

@FindBy(name="searchBtn") WebElement searchButton;

//Constructor

public samplePage(WebDriver driver){

this.driver = driver;

//initElements method to initialize all elements

PageFactory.initElements(driver, this);

}

//Sample method

public void search(String searchTerm){

searchTextBox.sendKeys(searchTerm);

searchButton.click();

}

}

**Ques.73. What is an Object repository?**

Ans. An object repository is the centralized location of all the objects or WebElements of the test scripts. In Selenium, we can implement an object repository using the Page Object Model as well as Page Factory design patterns.

**Ques.74. What is a data-driven framework?**

Ans. A data-driven framework is one in which the test data is put in external files like CSV, Excel, etc. Basically, the test data is separated from the test logic that is written in test script files. The test data drives the test cases, i.e. the test methods run for each set of test data values.  
TestNG provides inherent support for data-driven testing using @dataProvider annotation.

**Ques.75. What is a keyword-driven framework?**

Ans. A keyword-driven framework is one in which the normal set of actions are associated with keywords and are kept in external files usually in tabular form.  
For example, an action of launching a browser will be associated with keyword – launchBrowser(), action to write in a textbox with keyword – writeInTextBox(webElement, textToWrite), etc.  
The code to perform the action based on a keyword specified in the external file is implemented in the framework itself.  
In this way, the test steps can be written in a file by a person of a non-programming background also (provided all the used keywords are implemented in the framework).

**Ques.76. What is a hybrid framework?**

Ans. A hybrid framework is a combination of two or more frameworks. For example, a combination of data-driven and keyword-driven frameworks can be considered as a hybrid framework.

**Ques.77. What is Selenium Grid?**

Ans. Selenium Grid is a tool that helps in the distributed testing. Using Grid, we can run test scripts in different machines having different browsers, browser versions, platforms, etc in parallel. In the Selenium grid, there is a hub that is a central server managing all the distributed machines known as nodes.

**Ques.78. What are some advantages of the Selenium grid?**

Ans. The advantages of the Selenium grid are-

1. It allows running test cases in parallel thereby saving test execution time.
2. Multi-browser testing is possible using the Selenium grid by running the test on machines having different browsers.
3. Also, we can do multi-platform testing by configuring nodes having different operating systems.

**Ques.79. What is a hub in the Selenium Grid?**

Ans. A hub is a server or a central point in the Selenium grid that controls the test executions on the different machines.

**Ques.80. What is a node in the Selenium Grid?**

Ans. Nodes are the machines that are attached to the selenium grid hub and have selenium instances running the test scripts. Unlike a hub, there can be multiple nodes in the selenium grid.

**Ques.81. Explain the line of code Webdriver driver = new FirefoxDriver(); .**

Ans. In the line of code **Webdriver driver = new FirefoxDriver();**  ‘WebDriver’ is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.

**Ques.82 What is the purpose of creating a reference variable- ‘driver’ of type WebDriver instead of directly creating a FireFoxDriver object or any other driver’s reference in the statement Webdriver driver = new FirefoxDriver();?**

Ans. By creating a reference variable of type WebDriver, we can use the same variable to work with multiple browsers like ChromeDriver, IEDriver, etc.

**Ques.83. Name an API used for reading and writing data to excel files.**

Ans. Apache POI API and JXL(Java Excel API) can be used for reading, writing, and updating excel files.

**Ques.84. Name an API used for logging in Java.**

Ans. Log4j is an open-source API widely used for logging in Java. It supports multiple levels of logging like – ALL, DEBUG, INFO, WARN, ERROR, TRACE, and FATAL.

**Ques.85. What is the use of logging in automation?**

Ans. Logging helps in debugging the tests when required and also provides storage of the test’s runtime behavior.

**Ques.86. What is TestNG?**

Ans. TestNG(NG for Next Generation) is a testing framework that can be integrated with Selenium or any other automation tool. Moreover, it provides multiple capabilities like assertions, reporting, parallel test execution, etc.

**Ques.87. What are some advantages of TestNG?**

Ans. Following are the advantages of TestNG-

1. TestNG provides different assertions that help in checking the expected and actual results.
2. It provides parallel execution of test methods.
3. We can define the dependency of one test method over others in TestNG.
4. Also, we can assign priority to test methods in selenium.
5. It allows the grouping of test methods into test groups.
6. It allows data-driven testing using @DataProvider annotation.
7. TestNG has inherent support for reporting.
8. It has support for parameterizing test cases using @Parameters annotation.

**Ques.88. What are commonly used TestNG annotations?**

Ans. The commonly used TestNG annotations are-

* @Test – The @Test annotation **marks a method as a test method**.
* @BeforeSuite – The annotated method will run only **once before all tests in this suite have run**.
* @AfterSuite -The annotated method will run only once **after all tests in this suite have run**.
* @BeforeClass – The annotated method will run **only once before the first test method in the current class is invoked**.
* @AfterClass – The annotated method will run **only once after all the test methods in the current class have been run**.
* @BeforeTest – The annotated method will run **before any test method belonging to the classes inside the <test> tag is run**.
* @AfterTest – The annotated method will run **after all the test methods** **belonging to the classes** **inside the <test> tag have run**.
* **@**BeforeMethod – The annotated method will run **before each test method** marked by @Test annotation.
* **@**AfterMethod – The annotated method will run **after each test method** marked by @Test annotation.
* @DataProvider-The @DataProvider annotation is used to **pass test data to the test method**. The test method will run as per the number of rows of data passed via the data provider method.

**Ques.89. What are some common assertions provided by TestNG?**

Ans. Some of the common assertions provided by testNG are-

1. assertEquals(String actual, String expected, String message) – (and other overloaded data type in parameters)
2. assertNotEquals(double data1, double data2, String message) – (and other overloaded data type in parameters)
3. assertFalse(boolean condition, String message)
4. assertTrue(boolean condition, String message)
5. assertNotNull(Object object)
6. fail(boolean condition, String message)
7. true(String message)

**Ques.90. What is the use of the testng.xml file?**

Ans. A testng.xml file is used for configuring the whole test suite. In this file, we can create a test suite, create test groups, mark tests for parallel execution, add listeners, and pass parameters to test scripts. Later, this testng.xml file can be used for triggering the test suite.

**Ques.91. How can we pass the parameter to test script using TestNG?**

Ans. Using @Parameter annotation and ‘parameter’ tag in testng.xml we can pass parameters to the test script.  
Sample testng.xml –

<suite name="sampleTestSuite">

<test name="sampleTest">

<parameter name="sampleParamName" value="sampleValue"/>

<classes>

<class name="TestFile" />

</classes>

</test>

</suite>

//Sample test script:

public class TestFile {

@Test

@Parameters("sampleParamName")

public void parameterTest(String paramValue) {

System.out.println("Value of sampleParamName is - " + sampleParamName);

}

}

Copy

**Ques.92. How can we create a data-driven framework using TestNG?**

Ans. Using @DataProvider we can create a data-driven framework. Basically, we can pass test data to the associated test method and then multiple iterations of the test run for the different test data values passed from the @DataProvider method. The method annotated with @DataProvider annotation return a 2D array of object.

//Data provider returning 2D array of 3\*2 matrix

@DataProvider(name = "dataProvider1")

public Object[][] dataProviderMethod1() {

return new Object[][] {{"kuldeep","rana"}, {"k1","r1"},{"k2","r2"}};

}

//This method is bound to the above data provider returning 2D array of 3\*2 matrix

//The test case will run 3 times with different set of values

@Test(dataProvider = "dataProvider1")

public void sampleTest(String s1, String s2) {

System.out.println(s1 + " " + s2);

}

Copy

**Ques.93. What is the use of @Listener annotation in TestNG?**

Ans. Listeners are used for performing some action in case an event gets triggered. Usually, testNG listeners are used for configuring reports and logging. One of the most widely used listeners in testNG is ITestListener interface.  
  
It has methods like onTestSuccess, onTestFailure, onTestSkipped, etc. We need to implement this interface creating a listener class of our own. After that using the @Listener annotation, we can specify that for a particular test class, a customized listener class should be used.

@Listeners(PackageName.CustomizedListenerClassName.class)

public class TestClass {

WebDriver driver= new FirefoxDriver();

@Test

public void testMethod(){

//test logic

}

}

Copy

**Ques.94. How can we make one test method dependent on others using TestNG?**

Ans. Using the dependsOnMethods parameter inside @Test annotation in TestNG, we can make one test method run only after the successful execution of the dependent test method.

@Test(dependsOnMethods = { "preTests" })

**Ques.95. How can we set the priority of test cases in TestNG?**

Ans. Using the priority parameter in @Test annotation in TestNG we can define the priority of test cases. The default priority of the test when not specified is integer value 0. Example-

@Test(priority=1)

**Ques.96. What is the default priority of a test method in TestNG?**

Ans. The default priority of a test when not specified is integer value 0. So, if we have one test case with priority 1 and one without any priority then the test without any priority value will get executed first.

**Ques.97. How to prevent a test case from running using TestNG?**

Ans. A Test method can be disabled from getting executed by setting the “enabled” attribute as false.

//In case of a test method

@Test(enabled = false)

public void testMethod1() {

//Test logic

}

//In case of test method belonging to a group

@Test(groups = {"NegativeTests"}, enabled = false)

public void testMethod2() {

//Test logic

}

Copy

**Ques.98. How can we run test cases in parallel using TestNG?**

Ans. In order to run the tests in parallel just add these two key-value pairs in the suite-

* parallel=”{methods/tests/classes}”
* thread-count=”{number of thread you want to run simultaneously}”.

<suite name="ArtOfTestingSuite" parallel="methods" thread-count="5">

**Ques.99. What is the use of @Factory annotation in TestNG?**

Ans. @Factory annotation helps in the dynamic execution of test cases. Using @Factory annotation, we can pass parameters to the whole test class at run time. The parameters passed can then be used by one or more test methods of that class.  
  
For example – there are two classes TestClass and the TestFactory class. Because of the @Factory annotation, the test methods in class TestClass will run twice with the data “k1” and “k2”.

public class TestClass{

private String str;

//Constructor

public TestClass(String str) {

this.str = str;

}

@Test

public void TestMethod() {

System.out.println(str);

}

}

public class TestFactory{

//The test methods in class TestClass will run twice with data "k1" and "k2"

@Factory

public Object[] factoryMethod() {

return new Object[] { new TestClass("K1"), new TestClass("k2") };

}

}

**Ques.100. What is the difference between @Factory and @DataProvider annotation?**

Ans. @Factory method creates instances of test class and runs all the test methods in that class with a different set of data.  
Whereas, @DataProvider is bound to individual test methods and run the specific methods multiple times.

*Trên đây là danh sách chọn lọc về 100 câu hỏi phỏng vấn Selenium. Thời gian tới Anh Tester đang soạn các câu hỏi và trả lời về các phần khác như TestNG, NUnit, SQL, Cucumber,...*  
  
*Nếu bạn có thắc mắc về bất kỳ câu hỏi nào được liệt kê ở trên hoặc bạn muốn giúp Anh Tester thêm một số câu hỏi phỏng vấn khác về Selenium hay các phần khác, vui lòng cho mình biết trong phần comment bên dưới hoặc vào group chuyên Automation Testing để trao đổi thêm với mình và các bạn khác.  
CẢM ƠN RẤT NHIỀU !!!*

### 1. Tại sao cần Automation Test? (Why need automation test?)

Câu hỏi này để đánh giá kiến thức sơ bộ của bạn về automation test và việc bạn có biết mục đích sử dụng automation test để sử dụng nó có hiệu quả.

* Giúp tiết kiệm tiền bạc và thời gian: nhất là trong giai đoạn bảo trì của các dự án lớn. Mỗi tuần chúng ta phải thực hiện regression test từ 1 đến 2 lần với số lượng test case rất lớn trong 1 đến 2 ngày. ĐIều này gần như không thể thực hiện bằng cách thủ công, trong khi với kiểm thử tự động chúng ta hoàn toàn có thể với nguồn nhân lực vô cùng khiêm tốn.
* Chính xác hơn: Nhờ độ ổn định cao, kiểm thử tự động có thể thực thi các test case với độ chính xác cao hơn.
* Độ bao phủ cao: Như đã nói ở trên, khi sử dụng kiểm thử tự động, chúng ta có thể thực thi số lượng lớn test case trong một thời gian ngắn. Nên độ bao phủ của nó rất cao. Điều này giúp chúng ta tăng độ bao phủ trong giai đoạn regression test .
* Hoàn thành các công việc mà con người không thể làm được: Nếu chúng ta muốn thực thi load test, performance test, thì kiểm thử tự động là cách duy nhất.

**Các trường hợp cần sử dụng automation test:**  
 a) Kiểm thử hồi quy (Regression testing): Trong trường hợp sửa lỗi hoặc triển khai module mới, tester phải đảm bảo rằng chức năng đã được triển khai hoặc không thay đổi không bị ảnh hưởng. Trong trường hợp này, tester kết thúc chạy test case hồi quy nhiều lần.  
 Ví dụ: Sau mỗi yêu cầu thay đổi hoặc sửa lỗi, sau mỗi lần lặp trong trường hợp tiếp cận phát triển gia tăng, v.v.  
 b) Kiểm thử phi chức năng: Kiểm thử các khía cạnh phi chức năng của một ứng dụng.  
 Ví dụ: kiểm thử tải (load testing) hoặc kiểm thử hiệu suất (performance testing), vv rất khó cho con người theo dõi và phân tích.  
 c) Kiểm thử tính toán phức tạp: các test scenario dễ bị lỗi khi kiểm thử thủ công.  
 d) Thực hiện lặp lại các kiểm thử giống nhau: Đôi khi, tester phải chạy cùng một bộ test case cho một bộ dữ liệu khác nhau hoặc sau mỗi lần phát hành bản dựng hoặc trên nhiều phần cứng, phần mềm hoặc kết hợp cả hai.  
 Kiểm thử tự động các test case trong các tình huống trên giúp đạt được tốc độ kiểm thử và giảm thiểu lỗi của con người.

### 2. Framework là gì? (What is the framework?)

Câu hỏi để đánh giá sơ bộ cách build framework của bạn và liệu frame work đó có hiệu quả không?  
Framework là một tập hợp cấu trúc của toàn bộ bộ kiểm thử tự động. Nó cũng là một hướng dẫn, mà nếu tuân theo có thể dẫn đến một cấu trúc dễ bảo trì và nâng cao.  
 Những hướng dẫn này bao gồm:  
 Tiêu chuẩn mã hóa  
 Xử lý dữ liệu kiểm thử  
 Duy trì và xử lý các phần tử (kho đối tượng trong QTP)  
 Xử lý tệp môi trường và tệp thuộc tính  
 Báo cáo dữ liệu  
 Xử lý nhật ký

### 3. Automation test framework là gì?

Có thể hiểu đơn giản đó là một application project được dựng lên để tự động hóa việc kiểm thử một ứng dụng nào đó. Như vậy, bản thân framework cũng chính là một ứng dụng. Nó cũng phải được thiết kế hoàn chỉnh, được apply những design pattern, và cũng phải dựa trên những định nghĩa, quy tắc cơ bản nhất của ngôn ngữ lập trình được sử dụng để phát triển nên framework đó. Framework có thể được deploy như một ứng dụng hoàn chỉnh, hoặc cũng có thể được đóng gói thành các thư viện để được tiếp tục được phát triển.

### 4. Trách nhiệm của một Automation Engineer?

Automation Engineer không chỉ làm công việc viết automation script. Họ trước hết vẫn phải là một QA Tester đúng nghĩa. Đó là phải có sự am hiểu về mặt nghiệp vụ (business) của hệ thống. Có thể hiểu ít nhất mức độ quan trọng của việc kiểm thử, biết cách viết test case, log defect. Thực tế công việc thì người Automation Engineer sẽ kiêm luôn công việc của một Manual QA, và khi đã feature nào đã được hoàn tất, họ sẽ bắt tay vào việc implement các test case liên quan tới feature đó thành automation.

Trên thực tế, từ một QA thuần manual để chuyển sang Automation QA thực sự không phải là việc dễ dàng vì có dính tới code, và cũng đòi hỏi nhiều mindset, kỹ năng của một developer. Vì vậy, bạn cũng đừng ngạc nhiên khi thấy có nhiều Developer chuyển sang làm Automation QA nhưng từ Manual QA mà chuyển sang Automation thành công lại khá hiếm. Đó là bởi vì developer đã có sẵn dev skills và coding mindset, là những thứ cần rất nhiều thời gian + năng khiếu mới có được. Khi đó, chỉ cần học hỏi thêm mindset và kỹ năng cơ bản của một Manual QA là đã có thể bắt đầu con đường của một Automation QA được rồi.

Tuy nhiên, một full-stack QA không chỉ cần có Manual và Automation skills mà còn cần phải có ít nhiều kỹ năng của một DevOps để có thể tự deploy và maintain những gì mình đã xây dựng. Và cuối cùng là khả năng ngoại ngữ + giao tiếp để có thể deliver những gì mình đã và đang làm tới khách hàng.

### 5. Nêu 4 tính chất cơ bản của Lập trình hướng đối tượng OOP (Object-Oriented Programming)?

Phần lớn các automation framework hiện nay được xây dựng dựa trên Selenium kết hợp với một ngôn ngữ lập trình hướng đối tượng (phổ biến nhất có lẽ là Java và C#). Vậy nên dĩ nhiên các câu hỏi phỏng vấn sẽ ít nhiều liên quan tới OOP.

4 tính chất cơ bản của OOP thì có lẽ ai cũng biết, đó là:

- Encapsulation (tính đóng gói).

- Abstraction (tính trừu tượng).

- Inheritance (tính kế thừa).

- Polymorphism (tính đa hình).

Nhưng để hiểu và giải thích được cặn kẽ cả 4 tính chất này thì bạn cần ít nhất 1-2 tiếng đồng hồ để thử practice và nghiền ngẫm qua các ví dụ đầy rẫy trên mạng.

### 6. Sự khác biệt giữa Interface và Abstract class?

[Text

Description automatically generated](https://lh3.googleusercontent.com/-qUlOl5b_mns/X-5tmqSuq2I/AAAAAAACDmQ/cIlyFQiBjec2lRZs6z78b8yJ3H71gg_IQCNcBGAsYHQ/s16000/image.png)  
Trong Java, chúng ta chỉ có thể extend (kế thừa) một class cha duy nhất nhưng lại có thể implements nhiều interface khác nhau.

### 7. Có cần phải xây dựng framework khi build automation? (Why do you need build the framework in automation testing?)

Tham khảo câu trả lời phỏng vấn mẫu bằng Tiếng Việt:  
"Framework là hướng dẫn và không phải là quy tắc bắt buộc, vì vậy tester hoàn toàn có thể làm mà không cần chúng, nhưng nếu bạn tạo và tuân theo nó, việc tăng cường và duy trì sẽ dễ thực hiện."  
   
"Tham khảo câu trả lời phỏng vấn mẫu bằng Tiếng Anh:  
Frameworks are instructions and not mandatory rules, so testers can do without them, but if you create and follow it, enhancement and maintenance will be easy to implement."

### 8. Liệt kê một số ưu điểm và nhược điểm của automation test – kiểm thử tự động? (List some good and bad in automation testing?)

Đối với các bạn manual chuyển sang automation đã có kinh nghiệm, trải qua thời gian thực thi auto, các bạn sẽ đúc kết được ưu điểm và nhược điểm của automation test  
   
Tham khảo câu trả lời phỏng vấn mẫu bằng Tiếng Việt:  
 - Ưu điểm:  
 Nhân lực ít hơn  
 Có thể tái sử dụng  
 Thực hiện nhiều test hơn trong thời gian ngắn hơn  
 Độ tin cậy cao  
 Thi công song song các test case  
 Nhanh  
 - Nhược điểm:  
 Thời gian phát triển và bảo trì nhiều hơn  
 Chi phí công cụ cao  
 Tài nguyên có kỹ năng được yêu cầu  
 Thiết lập môi trường  
 Test Script gỡ lỗi là một vấn đề  
   
Tham khảo câu trả lời phỏng vấn mẫu bằng Tiếng Anh:  
-Advantages:  
Less manpower  
Reusable  
Do more tests in less time  
High reliability  
Executing parallel test cases  
Fast  
   
- Defect:  
More development and maintenance time  
High tool costs  
Skilled resources are required  
Environment setting  
Test Script debugging is a problem

### 9. Bạn có thể tự động hoá bao nhiêu test case trong 1 ngày? (How many test case can you build in a day?)

Tuỳ vào mỗi level và kinh nghiệm sẽ có số test case được auto  
   
 Tham khảo câu trả lời phỏng vấn mẫu bằng Tiếng Việt:  
"Con số phụ thuộc vào độ phức tạp của các test case. Khi độ phức tạp bị hạn chế, tôi có thể tự động hóa 5 đến 6 test case mỗi ngày. Đôi khi, tôi chỉ có thể tự động hóa một test case cho các tình huống phức tạp.  
Tôi cũng đã chia các test case của mình thành các thành phần khác nhau như: lấy đầu vào, thực hiện tính toán, xác minh đầu ra.. trong trường hợp các kịch bản rất phức tạp và đã mất từ ​​2 ngày trở lên."  
   
Tham khảo câu trả lời phỏng vấn mẫu bằng Tiếng Anh:  
"The number depends on the complexity of the test case. When complexity is limited, I can automate 5 to 6 test cases per day. Sometimes, I can only automate a test case for complex situations.  
 I have also divided my test cases into different components such as taking input, performing calculations, verifying the output .. in case of very complicated scenarios and it took 2 days or more up."

### 10. Selenium là gì?

Selenium là một thư viện được tạo ra để phục vụ cho việc automation test các ứng dụng web.  
<https://anhtester.com/blog/gioi-thieu-ve-selenium-b254.html>

### 11. Appium là gì?

Appium is an open source automation tool for running scripts and testing native applications, mobile-web applications and hybrid applications on Android or iOS using a webdriver.

Nói một cách đơn giản, chúng ta sẽ cần đến Appium khi perform automation testing trên các mobile platforms như Android hay iOS.  
<https://viblo.asia/p/gioi-thieu-ve-appium-va-huong-dan-cai-dat-appium-phan-1-1Je5EYN15nL>

### 12. Cucumber là gì?

Cucumber is a software tool that supports behavior-driven development. Central to the Cucumber BDD approach is its ordinary language parser called Gherkin. It allows expected software behaviors to be specified in a logical language that customers can understand.

Cucumber là một công cụ phần mềm hỗ trợ BDD. Ngôn ngữ được sử dụng trong Cucumber là Gherkin.

### 13. Sự khác biệt giữa Git và GitHub?

Simply put, Git is a version control system that lets you manage and keep track of your source code history. GitHub is a cloud-based hosting service that lets you manage Git repositories.

Yeah, một câu hỏi tưởng chừng đơn giản nhưng cũng có thể khiến nhiều người sa lưới. Nois một cách đơn giản, Git là một hệ thống quản lý version cho phép bạn quản lý và theo dõi lịch sử của source code. Trong khi đó, GitHub là một dịch vụ cloud hosting cho phép bạn quản lý các Git repo khác nhau.

### 14. Page Object Model là gì?

<https://www.toolsqa.com/selenium-webdriver/page-object-model/>Page Object Model or POM  is a design pattern or a framework that we use in Selenium using which one can create an object repository of the different web elements across the application. To simplify, in the Page Object Model framework, we create a class file for each web page. This class file consists of different web elements present on the web page. Moreover, the test scripts then use these elements to perform different actions.

Page Object Model là một design pattern hoặc một framework mà chúng ta sử dụng trong Selenium. Nói một cách đơn giản, với mỗi trang ứng dụng cần test (ví dụ như trang Login) thì chúng ta sẽ tạo một class file tương ứng. Class này sẽ chứa các element locators tương ứng cho trang cần test (nút Login, Username field, Password field,...) và cả các keyword/action tương ứng (clickLoginButton, inputUsername,...).  
15. Browser Factory là gì?

<https://www.toolsqa.com/selenium-webdriver/c-sharp/browser-factory-or-webdriver-factory/>

From our experience with frameworks we know that maintaining and passing around a WebDriver object across different tests is a delicate process. Also, the complexity increases when we have to maintain only one instance of a WebDriver through out the test run. To overcome the problem on instantiation of WebDriver and maintaining the instance of browser we can use create a small class called Browser Factory or WebDriver Factory.

Có thể hiểu Browser Factory như một design pattern giúp khởi tạo và duy trì browser instance một cách đơn giản và dễ dàng hơn trong suốt quá trình chạy test.  
16. Data Driven là gì?

Về mặt định nghĩa, các bạn có thể tự tìm hiểu. Nhưng về mặt technical, một biểu hiện của Data Driven đó là với cùng một test method, bạn có thể run nó nhiều lần, với mỗi lần là 1 bộ data khác nhau, tương ứng với các scenario khác nhau. Data Driven hầu như đều được hỗ trợ bởi các testing framework cơ bản như TestNG, JUnit, NUnit, Xunit, v.v...

<https://www.toolsqa.com/testng/testng-dataproviders/>

### 17. Sự khác biệt giữa BDD và TDD?

<https://www.glowtouch.com/test-driven-development-vs-behavior-driven-development/>

[Text

Description automatically generated](https://lh3.googleusercontent.com/-DrVqtCnExxg/X-5stUt9CVI/AAAAAAACDmE/xiK10LOk7SkigeXQ8CMINY2h2vqGkoLwQCNcBGAsYHQ/s16000/image.png)  
18. Các kiểu chờ đợi (wait) trong Selenium?

<https://www.guru99.com/implicit-explicit-waits-selenium.html>

#### **Selenium waits bao gồm một số loại như:**

1. Implicit wait
2. Explicit wait
3. Fluent wait

## Các câu hỏi về tính cách và định hướng trong tương lai

Trong phần cuối cùng của buổi tuyển dụng Automation Tester, bạn có thể gặp các câu hỏi về tính cách và định hướng tương lai. Bạn hãy cho nhà tuyển dụng thấy mình là người cầu tiến, có khả năng làm việc trong môi trường cộng tác cũng như có ý định làm việc lâu dài với công ty.

Automation Tester là nghề nghiệp triển vọng cho các bạn sinh viên IT tại Việt Nam. Các bạn hãy chuẩn bị thật kỹ những tình huống có thể xảy ra để vượt qua các vòng tuyển dụng Automation Testermột cách tốt nhất.