Automation Testing Interview Guide by Subburaj S

(subburajs.automation@gmail.com)

General:

- Tell me about yourself?
- What are your Role and Responsibilities?
- Architecture of your current application?
- Which Automation framework you are using in your organization can you tell me the folder structure and flow of your framework?

Java:

- Why String is **immutable** in Java?
  https://www.youtube.com/watch?v=1s4wciDYxqs&list=PLFGoYjJG_fqr84PKCp88iPkQDWJSfzaqz&index=16

- What is **static** in java?
  https://www.youtube.com/watch?v=qRCEdWQ0f4Q&t=209s

- What is **final** in java?
  https://www.youtube.com/watch?v=7m1Zmf_Z3B0

- What is **this** word in java?

  https://www.youtube.com/watch?v=HD5IyaOgdPQ

- What is **finally** and where do we use it?

  https://www.youtube.com/watch?v=oR2CVVjCsCA

- What is **Autoboxing** and **unboxing**?

  https://www.youtube.com/watch?v=thqLx4zUeY0

- What is **serialization** and **deserialization**?

  https://www.youtube.com/watch?v=qo9S2CeoqQE

- What is an **abstract** modifier?

  https://www.youtube.com/watch?v=yY6XMBUCNYE

- What is **call by reference** and **call by value**?

- **Primitives** and **Non-Primitives** datatypes in Java? String is primitive or non-primitive?

  https://www.youtube.com/watch?v=QfKiQX_PX1g

- What is the method of overloading?

  https://www.youtube.com/watch?v=T9fRL52TP_M

- Why is it important to override **HashCode**() when you override equals()?

  https://www.youtube.com/watch?v=Nr56SlbMed4

- What is the difference between a **checked** and **unchecked** exceptions?

  https://www.youtube.com/watch?v=alGOQMLg3UY

- Difference between **final**, **finally**, **finalize**?

  https://www.youtube.com/watch?v=MZa8llTCQJ8

- Difference between **abstract** and **interface**?

  https://www.javatpoint.com/difference-between-abstract-class-and-interface

  https://www.journaldev.com/1607/difference-between-abstract-class-and-interface-in-java

  https://www.youtube.com/watch?v=4oEV1aYcOwI&t=198s

- Difference between **String Buffer** and **String Builder**?

  https://www.youtube.com/watch?v=oYcb0N1YfVw

Parameter	String	StringBuffer	StringBuilder
Storage	String Pool	Heap	Heap
Mutability	Immutable	Mutable	Mutable
Thread Safe	Not used in a threaded environment	Used in a multi-threaded environment	Used in a single-threaded environment
Performance	Slow	Slower than StringBuilder but faster than String	Faster than StringBuffer
Syntax	`String var =“Edureka”;` `String var=new String(“Edureka”);`	`StringBuffer var = new StringBuffer("Edureka");`	`StringBuilder var = new StringBuilder("Edureka");`

- Difference between **Array** and **ArrayList**?

  https://www.youtube.com/watch?v=cLAQdFC3nkE

- Difference between **ArrayList** and **LinkedList**?

  https://www.youtube.com/watch?v=QWMyhFUtFHo

- Difference between **HashMap** and **HashTable**?
  https://www.youtube.com/watch?v=z5tZ0Zb5rJQ

- How to define dynamic array?

  https://www.tutorialspoint.com/How-to-declare-Java-array-with-array-size-dynamically

- Can we create the object for the abstract classes?

  https://www.youtube.com/watch?v=w2bp0Vdq6Fw

- Can we create the object for an interface?

  https://www.youtube.com/watch?v=uwTBa_hzDjI

- Can we create constructor of abstract class?

  https://www.youtube.com/watch?v=dn0n78xF5Zs

- Can constructor be overloaded. Explain why?

  Yes! Java supports **constructor overloading**.
  In constructor loading, we create **multiple constructors** with the same name but with different parameters types or with different no of parameters.

- Can main method be **overloaded**?

  Yes, we can overload the main method in java.

https://www.youtube.com/watch?v=YWL-FyLihE8

- Can main method be **overridden**?

No, we cannot override main method of java because **a static method cannot be overridden.**
The **static method in java is associated with class** whereas the **non-static method is associated with an object**. Static belongs to the class area, static methods don't need an object to be called. Static methods can be called directly by using the classname (classname.static_method_name()).
So, whenever we try to execute the derived class static method, it will automatically execute the base class static method.
Therefore, **it is not possible to override the main method in java**

- Can we **override** static method?

  https://www.youtube.com/watch?v=MuEenxrkDwU

- Can we **overload** static method?

  https://www.youtube.com/watch?v=7mu8L6AzsSQ

- Can we write non-abstract methods in **Interface**?

  No

- Can we execute a java program **without main method**?

  Yes, we can execute a java program without a main method by using a **static block**.

  Static block in Java is a group of statements that gets executed only once when the class is loaded into the memory by

Java ClassLoader, It is also known as a static initialization block. Static initialization block is going directly into the stack memory.

- Can we call a **non-static variable** in static method?

  https://www.youtube.com/watch?v=msXJVIecGp0

- Can I execute multiple **catch blocks without try** will it give me compile time error?

  No

- How to achieve **serialization** and **deserialization**?

  Serialization is a mechanism of converting the state of an object into a byte stream. **Deserialization** is the reverse process where the byte stream is used to recreate the actual **Java** object in memory. This mechanism is used to persist the object. The byte stream created is platform independent.

  Serializable interface for serializing the object.

- If we declare the main method as private what will happen?

  Yes, we can declare the main method as **private** in Java.
  It compiles successfully without any errors but at the runtime, it says that the main method is not public.

- How to check whether the array is empty and null?

  To check if an array is null, use **equal to** operator and check if array is equal to the value **null**.

  To check if an array has no elements, **get length property** of the array and check if the length is **zero**.

- What are the classes available in a **list interface**?

  AbstractList, AbstractSequentialList, ArrayList, AttributeList, CopyOnWriteArrayList, LinkedList, RoleList, RoleUnresolvedList, Stack, Vector.

- What is the use of **constructor** in java?

  Constructor is a special method that is called at runtime during the object creation by using the **new** operator.

- What is **Hashmap**? Can we store objects in **Hashmap** and how to retrieve them?

  HashMap in Java is a collection based on Map and consists of **key-value pairs**. A HashMap is denoted by < Key, Value > or < K, V >. A HashMap element can be accessed using a Key i.e. we must know the key to access the HashMap element.

  A **HashMap** uses a technique called "**Hashing**". In hashing, a longer string is converted into a shorter string by applying some algorithm or 'hash function'. A string is converted to a shorter string as it helps in searching that is faster. It is also used for efficient indexing.

  https://www.softwaretestinghelp.com/hashmap-in-java/

  Custom class should follow the contract between **hashCode**() and **equals**().

  Make custom class **immutable**.

- Difference between **Hash Map** and **Hash Set**?

https://www.javatpoint.com/difference-between-hashset-and-hashmap

- Where did you use **HashMap** in your project and also oops concepts in your Automation Framework?

  https://automationtalks.com/2017/11/03/map-hashmap-testng-dataprovider/

- Access modifiers in java and its scope?

  Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

  Default: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

  Protected: The access level of a protected modifier is within the package and **outside the package through child class**. If you do not make the child class, it cannot be accessed from outside the package.

  Public: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

- What is **singleton** class in java?

  A class should ensure that only a **single instance must be created** and single object can be used by all other classes.

  Advantage

  Singleton Pattern can saves memory because object is not created at each request.

Provide a global point of access to the object.

Allow multiple instances in the future without affecting a singleton class's clients.

How to create Singleton design pattern?

Static member, Private constructor and Static factory method.

- What is the difference between **static binding** and **dynamic binding**?

  https://www.tutorialspoint.com/difference-between-static-binding-and-dynamic-binding-in-java

- Is **Hashmap** thread safe?

  Non-synchronized and it's **not** thread safe.

- What is **static**, How to set value of **static variable?**

  A **static** method can be accessed without creating an object of the class.

  https://www.softwaretestinghelp.com/java/static-in-java/

- Can we **overload** private methods?

  Yes, we can overload private methods in Java but, you can access these from the same class.

- Is it possible to **extend Final Class**?

  No, it cannot be extended.

- Is it possible to **override Static method?**

  No

- Is it possible to **overload main method?**

  Yes

- Is it possible to **initialize a variable present in an Interface?**

  No

- What would happen, if **multiple inheritance** is possible, in Java?

  Multiple inheritance is not possible in Java.

- Explain Exceptions hierarchy in java?

  https://rollbar.com/blog/blog/java-exceptions-hierarchy-explained

- Explain **Set** and **Map** in Java?

  Set and Map both are **interfaces**, it is a part of the collection framework.

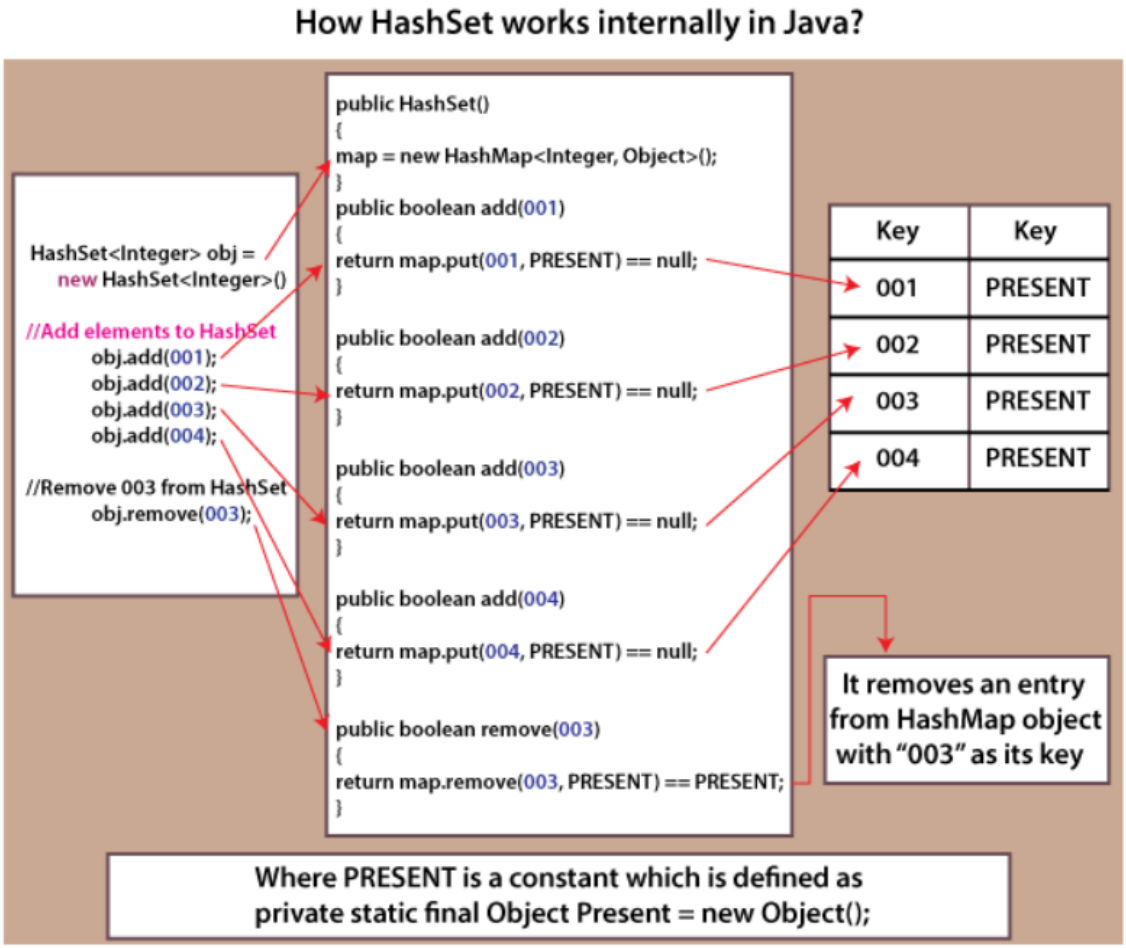
  1. **Duplicate**:- Set doesn't allow duplicates. Set and all of the classes which implements Set interface should have unique elements.
  Map stored the elements as key & value pair. Map doesn't allow duplicate keys while it allows duplicate values.

  2. List is an **ordered collection** it maintains the insertion order, which means upon displaying the list content it will display the elements in the same order in which they got inserted into the list.

  3. Null values:.
  Set allows single null value at most.

Map can have single null key at most and any number of null values.

4.Method: - If you want to add the object in the set interface add will be used, on the other side in the map interface objects will be stored into the key and values format, here the key must be unique.



- Explain about **Inheritance**.

  https://www.javatpoint.com/inheritance-in-java

  Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object.

- Difference between **overloading** and **overriding**?

  https://www.journaldev.com/32182/overriding-vs-overloading-in-java

- Difference **Encapsulation** and **Abstraction** ?

  https://www.javatpoint.com/abstraction-vs-encapsulation-in-java

Abstraction	Encapsulation
Abstraction is a feature of OOPs that hides the **unnecessary** detail but shows the essential information.	Encapsulation is also a feature of OOPs. It hides the code and data into a **single** entity or unit so that the data can be protected from the outside world.
It solves an issue at the **design** level.	Encapsulation solves an issue at **implementation** level.
It focuses on the **external** lookout.	It focuses on **internal** working.
It can be implemented using **abstract classes** and **interfaces**.	It can be implemented by using the **access modifiers** (private, public, protected).
It is the process of **gaining** information.	It is the process of **containing** the information.
In abstraction, we use **abstract classes** and **interfaces** to hide the code complexities.	We use the **getters** and **setters** methods to hide the data.
The objects are **encapsulated** that helps to perform abstraction.	The object need not to **abstract** that result in encapsulation.

- Difference between **throw** and **throws**?

  https://www.javatpoint.com/difference-between-throw-and-throws-in-java

No.	throw	throws
1)	Java throw keyword is used to explicitly throw an exception.	Java throws keyword is used to declare an exception.
2)	Checked exception cannot be propagated using throw only.	Checked exception can be propagated with throws.
3)	Throw is followed by an instance.	Throws is followed by class.
4)	Throw is used within the method.	Throws is used with the method signature.
5)	You cannot throw multiple exceptions.	You can declare multiple exceptions e.g. public void method()throws IOException,SQLException.

- What is **polymorphism?**
- How and when to use **interface**?
- Can we instantiate an interface?
- Can we over load main method in Java?
- Can we **override** constructor?
- Where do you use polymorphism in java?
- What is the system.out.println? and use of it?
- Why do we use **finally** and how it differs from the **final** keyword?
- Can we use multiple catches? When can we use multiple catches?
- Different between POI and JXL?
- How to prevent the **override** method in Java?
- Why is the main method static?
- What is the use of **static variables**?
- What is the difference between **list** and **set**?
- How will you access default and protected class?
- Why Object creation not possible in **Abstract classes**?

- Design pattern in JAVA.
- What All of the classes in the Java Collection Framework have?
- Situations when we use **abstraction** and **Interface** - explanation about **loosely coupled** and **tightly coupled**
- Will Java provide default constructor by own ? How
- Difference between **Arraylist** and **Linked List**,In which situation they are used ?
  Difference between
  List<String> list = new ArrayList<String>() and
  ArrayList<String> list = new ArrayList<String>();
- Difference between **HashMap** and **MultiMap**?
- In which situation the method should be **static** and when **non static**?
- How does **HashMap** is implemented using key value pair?
- Suppose you have class and abstract class in class there is a user defined constructor and main method which one will get executed first?
- What do you mean by **POJO** why we use POJO?
- class A have 3 method, class B have 2 method, class B inherited class A, how do you call method of class A by creating object of class B?

Selenium:

- What are the challenges you have faced during testing?
- What strategies you followed while building a selenium framework from scratch?
- Where do you perform the **singleton design pattern**? If you don't use it, then do you have an idea about this?
- Difference between **Implicit**, **Explicit** and **Fluent** waits in Selenium?
- Pros and cons of Implicit wait and Explicit wait.
- Why we prefer **explicit wait instead of fluent wait**? What are the disadvantages of fluent wait?
- Without **implicit wait** selenium script will work or not?
- What is **default polling time** in explicit wait and in implicit wait?

- Explain about **synchronization** in selenium?
- Which concept they have implemented in **explicit** and **fluent wait**?
- Explain **abstraction** and **interface** respect of selenium with some example
- Difference between **Factory design** and **Singleton framework**?
- What is **page object** and **page factory model**?
- Have you used **interface** in your framework other than selenium interfaces?
- How do you achieve **inheritance** in your framework?
- What is **Webdriver**, Name methods which do not have the implementation?
- What are the methods present in the webdriver interface?
- What's the fastest locator in Selenium?
- What does **:: (doubles colon)** in sibling xpaths represent?
- Explain. “Driver.manage.window.maximize” (talk about option interface here)
- What is difference between **get()** and **navigate().to()** in Selenium?
- How would you check the broken links, in the webpage?
- Difference between **submit()** and **click()** in Selenium?
- Difference between **absolute XPath (/)** and **relative XPath (//)**
- Difference between **findelement** and **findelements?**
- Difference between **frames** and **iframes?**
- Return type of **findelement** and **findelements**?
- What error will be thrown when no element found for **findelement** and **findelements**?
- State some exception which you have faced in your framework? (Don’t mention only selenium explain. Explain java exception also)
- Types of Exceptions and how to handle **stale element exception**?
- What are the **interface** used in selenium?
- Where do you used **inheritance** in selenium?

- How do you initialize web elements in POM? What error or exception will come if not initiated?
- If both wait method that is implicit and explicit is mentioned in the script, then which one is work? is it good practice to mention both in good?
- What is the difference between **close** and **quit** in selenium?
- How do you handle **Alert** in Selenium?
- In a web page, there are several Pop-up, but we don't when the pop-up will appear, in this case how we will handle the Pop-up using Selenium WebDriver (Java)
- How to handle file upload when type attribute does not file for upload web element.
- How to cover character keyboard operation from the context menu utilizing user-defined keyword?
- Consider this snippet
  Web driver driver=new chromedriver();
  what does the above code snippet mean?
- Where can "Dynamic Polymorphism" in Selenium WebDriver be observed?
- What is the difference between **"/"** and **"//"** in XPath?
- If proper Xpath, CssSelector and ID are not available, how do you identify an object?
- Attributes of CSS Selector?
- Which is most faster xpath or css?
- How to **get n-th element** using XPath and CSS?
- Consider you are only allowed to use css locator, how will you find the parent/grandparent of a web element?
- Will **driver.findelements()** throw an exception ?
- What is returned by driver().manage() ?
- In selenium, if you want to access the element that has the text "This element has an ID that changes every time the page is loaded" in it, then which of the following will you use?
- On page Object Model Framework (POM), how do you initialize the elements of a page to be used in the runner class?

(name of the PageObjects class is “”SignupPage.java”” and the dirver object name is “”driver””).

- Get the values from the dropdown and print them in **Ascending** order
- Using **TreeSet** to find elements command
- Does takes screenshot is interface or class
- Selenium uses lots of third parties jars for scripting. Then why do we still go for selenium?
- Why do we have to use **build()** and **perform()** with the action object
- Can we use **perform()** along in scripting without build()
- What is difference between **build()** and **perform()** in selenium?
- Return type of **getwindowhandle()** and **getwindowhandles()?**
- Window Handling in Selenium -Switching from another window to Parent window
- If the button is **disabled**? how to check -using **getattribute()**
- Explain method overloading with selenium and some example
- How do you read excel in the script? (very careful while answering. the counter-question will come as per your answer)
- Do you use the **property file** in your framework? If yes, then which java concept gets utilize here? (Java Collection)
- On a web page, there are several Pop-up, but **we don’t when the pop-up will appear**, in this case how we will handle the Pop-up using Selenium WebDriver (Java)
- Started automation test suite execution and few test cases are failed in a test run. How can you execute only failed test cases at once (with one click) what design pattern do we use when we trigger different browsers?
- What are approached to **handle dynamic WebElement**?
- Click last option in the dropdown (Last drop-down changes dynamically)
- How to calculate links on a page? (Answer with HTML tag)
- Write the code to read the value from the excel sheet.
- What is **Page Factory** in POM Design pattern?

- Suppose you have 10 pages in your application then how to achieve POM. What you will do?
- Annotation used in Page Object Model?
- Ways to find broken links in Selenium?
- How to handle **frame** in Selenium?
- How to handle **Alerts** in Selenium?
- Different types of Navigation Commands?
- Difference between **assert** and **verify**?
- How to download a file using Selenium?
- How do you manage a set of **Data Tables** in Selenium?
- How do you automate **localization testing** -diff language in UI?
- How to avoid **NoSuchElementException** without using try/catch block and with try/catch block?
- How to handle web tables whose values change **dynamically**?
- How to check whether web element is enabled or **Disabled** without using **isEnabled** method?
- Why is CSS locator faster than Xpath?
- Even though CSS is faster than Xpath ,why do 95% of the companies use XPath ?
- Error is throwing as Element not found but when I go and check that element is available in the web page? The element is not hidden so no need to use Java script executor? How do you solve this?
- How do you execute using **headless mode**?
- In Selenium, how to get text value from text-box if **gettext()** is not working?
- If we are using correct locator but still getting element not found error then how you will resolve this error?
- In **Page object** model once you create loginpage.java class what is the first thing you start with writing initially. How are you initiating writing something into a page class?
- What if Windows popup occurs during test execution and due to that can't execute automated tests, how u will resolve this error?
- Different ways to **handle hidden elements**?

- What is the difference between **click() function** in **webelement** interface and **click() function** in **Actions class**?
- Is it possible to change the behavior of a test at runtime?
- Describe how to handle the below items using selenium

  -iframe -windows -table -Alerts

- How to click right click of mouse?
- How to scroll down a page?
- What will **driver.getWindowHandels()** return?
- How will you automate Windows based application?

Java Program:

- Swap string **without 3rd variable**?
- Duplicates in a String?
- How to find the length of the string **without using length**?
- Largest Number in an Array?
- Reverse string **without using reverse function**
- Write code to print the Fibonacci series?
- Write code to print only the even numbers from an array.
- Write code to find special character, number, capital and small letter in a given string.
- Write code to check if a number is **palindrome**?
- Write a code to reverse the code, without using the built-in method.
- Write a Java code to identify, if the pair of strings are an **Anagram** or not?
- Write a code to get Highest number using array.

TestNG:

- What is the importance of the testng framework?
- Why we use TestNG in your framework
- What is the purpose of testing XML
- Explain the purpose of listeners? is it the selenium concept of TestNG?
- Case Scenario: How to run the same method 100 times in TestNG with the same data?
- What is the reporting tool in your framework? and why?
- Some questions in TestNG XML?
- What are different **testng annotations**?
- How can you configure tests in testng?
- What is **@dataprovider**?
- Difference between **@Factory** and **@DataProvider**?
- **@Factory** explain with real time example?
- **Test Order** in TestNG?
- How to add/remove test cases in **Testng.xml**?
- Explain the difference between **beforemethod**, **beforetest**, and **beforeclass**
- List out the testng annotation hierarchy order?
- How you **achieve parallel execution using testng**?
- Out of 50 testcases, how you will run only the failed testcases?
- How can you take the screenshot for the failed testcases?
- How can you run the same tests for 10 times?
- Types of Listeners?
- TestNG: Parallel executions, Grouping?
- Difference between **after suite** and **before suite**?
- What is the use of **testng.xml**?
- How many suits can be there in **testNG** , what if I run all the suits?
- Syntax to perform parallel testing in TestNG and what do you write in **<suite tag>** also what do you mention in double quotes like **parallel = " "**

- In **testNG**, do we have multiple suite in **one XML** file and what If I want to run all suits?
- What is **invocationcount** in **testng**?
- Cucumber tags and annotations?
- What is **background** in Cucumber?
- Difference between **Scenario** and **Scenario Outline**?
- Write skeleton of test runner?
- Explain retry analyzer?
- Cucumber tags? And how to run different combinations of tags when multiple tags are present
- Difference between **hooks** and **tags**?

Maven:

- Lifecycle of Maven
- Use of Maven **surfire plugin.** If yes, where and why?
- What is the use of **pom.xml**?
- CI / CD tools
- What is Jenkins?
- How will you handle dependencies in Maven at run time?
- Today we have executed some tests using **maven**, but tomorrow when you see that someone **deleted all** dependencies from **.pom file** then in that case will you be able to execute tests or not.
- Consider you have to write test/suite for different environments(qa, preproduction, production) and pass different set of data for each environment. How will you do it using maven file(Pom.xml)?
- Can you give some basic commands used in maven project?
- How will you configure Jenkins job?
- What are two components Jenkins is integrated with?
- How you will schedule the deployments?
- What is the purpose of version control tool?
- What are the git commands you have used?
- What is difference between **group id** and **artifact id**?
- How and when is Jenkins is used in your Automation?

WebService:

- Difference between REST and SOAPUI.
- Method in REST.
- Difference between **PUT** and **PATCH** call
- How to integrate postman to project?
- How will you handle dynamic **payloads** in API?
- How do you capture specific responses value and pass to other request?
- What challenges you faced in API testing?
- What is difference between **Authorization** and **Authentication**?
- What are the **API status codes**, you have come across?
- What is difference between **OAuth1.0** and **OAuth2.O** ,When and where do you use and how. Can you write a sample code?
- How you get the response from one api and send to another api?

Functional Testing:

- What is **Test Plan**?
- Explain the bug life cycle?
- Difference between **smoke** and **sanity tests**?
- Difference between **regression** and **retesting**?
- Difference between **functional** and **regression testing**?

- Difference between **severity** and **priority**?
- Difference between **Test Plan** and **Test Strategy**?
- Difference between boundary value analysis and equivalence partitioning?
- Difference between **white box** and **black box testing**?
- If we are having 1000 test cases, what type of testing carried for automation testing? Can we write a method that returns two or more values? If then, how?
- **Bug Life Cycle**?
- **Bug Triage**?
- What is **exploratory testing**?
- What is **adhoc testing**?
- What is build acceptance testing?
- What is difference between **Validation** and **Verification**?
- Explain severity and priority and High severity with low priority, low severity and high priority?
- Given the test cases having priority of -1,0,1,2 tell me the sequence of execution.
- Explain inner join and outer join in SQL?
- Difference between **DELETE**,**DROP** & **TRUNCATE**?
- What are test design techniques?
- What is **deferred bug**?
- How you will decide **what tests to automate**?
- When you decide to stop the testing?
- If your **testsuite** takes 1 and half to run, what you will do to reduce the time?
- How many test cases in your regression **testsuite**? How much time it will take to execute?
- How to cover character keyboard operation from the context menu utilizing user-defined keyword?
- What is most important things to define in a bug?
- Can tell about any achievements you have done in automation?
- Can you tell me the difference between bdd and tdd?

- What is a test plan and what are the steps to create a test plan?
- What are the inbound and outbound for testing?
- What is smoke, regression and sanity testing?
- What is the next step to be taken, if developer rejects the Open defect?
- Explain the test metrices?
- How would you priorities Tests?
- How do you perform **Automation Code Review/ Walk Through** in your project?
- There are 250 manual test cases , how will you segregate (on what basis) the regression , sanity , smoke suite?
- When Regression ,Sanity , smoke test scripts are executed ?
- How will you decide that this test case is feasible or good candidate for automation ?

Agile:

- What is an agile methodology?
- What is the scrum and who is your scrum master?
- Ceremonies followed in Agile methodology?
- Retrospective meeting?
- Describe Scrum ceremony?
- When do you automate in current sprint or next sprint?
- Explain velocity in sprint.
- In tight sprint schedule, if a new requirement is added, how will you handle this situation?
- What is **backlog** in Scrum methodology?
- You have 30 + sprints in your release how will you design your test scripts and run them?

Managerial round:

- How you will be an asset to the team?
- Why you are looking for a change?
- How soon you can join?

- Suppose if we give manual testing for six months or one year what will you do?
- How interested are you in learning new technologies?
- Failures in your work life.
- As Lead, how do you define quality of product before releases?
- Suppose you are the leam QA and 1 new member join your team and at the same time you have a deadline to meet in next 2 or 3 days so how will you involve that new member in team so that you can utilise him/her to meet deadlines?
- As a QA, where do you see yourself after 3 years?
- What are your strengths and Weakness?
- What are some best practices you learnt and how much difference it made in testing career? Explain before and after situations
- After you have run a full regression test, and find new regression bugs, which bugs would you prioritize. Bugs that suggest that functionality has regressed, or bugs that appear in new features?

References;
https://www.youtube.com/watch?v=1s4wciDYxqs&list=PLFGoYjJG_fqr84PKCp88iPkQDWJSfzaqz&index=15

Telusko