

TEK School

API Interview Questions

1. What is an API?
2. What are main differences between API and Web Service?
3. What are the Limits of API Usage?
4. What are some architectural styles for creating a Web API?
5. Who can use a Web API?
6. What is API Testing?
7. What are the advantages of API Testing?
8. Some common protocols used in API testing?
9. What is the test environment of API?
10. What are principles of an API test design?
11. What are the common API testing types?
12. What is the procedure to perform API testing?
13. What must be checked when performing API testing?
14. What is the best approach method to perform API testing?
15. What are tools could be used for API testing?
16. What are differences between API Testing and Unit Testing?
17. What are differences between API Testing and UI Testing?
18. What are major challenges faced in API testing?
19. What are the testing methods that come under API testing?
20. Why is API testing considered as the most suitable form for Automation testing?
21. What are common API errors that often founded?
22. What kinds of bugs that API testing would often find?
23. What is API documentation?
24. What are API documentation templates that are commonly used?
25. When writing API document, what must be considered?
26. How often are the APIs changed and, more importantly, deprecated?
27. What is REST?

28. What is a RESTful Web Services?
29. What is a "Resource" in REST?
30. What is the most popular way to represent a resource in REST?
31. Which protocol is used by RESTful Web services?
32. What are some key characteristics of REST?
33. What is messaging in RESTful Web services?
34. What are the core components of an HTTP request?
35. What are the most commonly used HTTP methods supported by REST?
36. Can GET request be used instead of PUT to create a resource?
37. Is there any difference between PUT and POST operations?
38. Which purpose does the OPTIONS method serve for the RESTful Web services?
39. What is URI? What is the main purpose of REST-based web services and what is its format?
40. What is payload in RESTful Web services?
41. What is the upper limit for a payload to pass in the POST method?
42. What is the caching mechanism?
43. What are SOAP Web services?
44. How does SOAP work?
45. When to use SOAP API?
46. How users utilize the facilities provided by SOAP?
47. What is the major obstacle users faced when using SOAP?
48. What are the various approaches available for developing SOAP based web services?
49. What are the elements of a SOAP message structure?
50. What are the syntax rules for a SOAP message?
51. What is the transport method in SOAP?
52. What are some important characteristics of a SOAP envelope element?
53. What are the major functionalities provided by the SOAP protocol class?
54. What are the web relation functionalities provided by SOAP protocol?
55. How does the message security model allow the creation of SOAP more secure to use?

- 56. What is the difference between top down & bottom up approach in SOAP Web services?
- 57. What are advantages of SOAP?
- 58. What are disadvantages of SOAP?
- 59. What are the differences between SOAP and REST?
- 60. SOAP or Rest APIs, which method to use?
- 61. What are the factors that help to decide which style of Web services – SOAP or REST – to use?

1. What is an API?

An API (Application Programming Interface) is a software intermediary that enables two applications to communicate with each other. It comprises a number of subroutine definitions, logs, and tools for creating application software.

In an API testing interview, you could be asked to give some API examples, here are the well-known ones: Google Maps API, Amazon Advertising API, Twitter API, YouTube API, etc.

2. What are main differences between API and Web Service?

- All Web services are APIs but not all APIs are Web services.
- Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.
- A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication whereas API may be exposed to in multiple ways.
- A Web service always needs a network to operate while APIs don't need a network for operation.

3. What are the Limits of API Usage?

Many APIs have a certain limit set up by the provider. Thus, try to estimate your usage and understand how that will impact the overall cost of the offering. Whether this will be a problem depends in large part on how data is leveraged. Getting caught by a quota and effectively cut-off because of budget limitations will render the service (and any system or process depending on it) virtually useless.

4. What are some architectural styles for creating a Web API?

This is one of the fundamental Web API interview questions. Bellows are four common Web API architectural styles:

- HTTP for client-server communication
- XML/JSON as formatting language
- Simple URI as the address for the services
- Stateless communication

5. Who can use a Web API?

Web API can be consumed by any clients which support HTTP verbs such as GET, PUT, DELETE, POST. Since Web API services do not require configuration, they can be easily used by any client. In fact, even portable devices such as mobile devices can easily use Web API, which is undoubtedly the biggest advantage of this technology.

6. What is API Testing?

API testing is a kind of software testing which determines if the developed APIs meet expectations regarding the functionality, reliability, performance, and security of the application.

7. What are the advantages of API Testing?

In an API interview, they are likely to ask about the advantages of API testing. So be prepared with the significant ones such as:

- **Test for Core Functionality:** API testing provides access to the application without a user interface. The core and code-level of functionalities of the application will be tested and evaluated early before the GUI tests. This will help detect the minor issues which can become bigger during the GUI testing.
- **Time Effective:** API testing usually is less time consuming than functional GUI testing. The web elements in GUI testing must be polled, which makes the testing process slower. Particularly, API test automation requires less code so it can provide better and faster test coverage compared to GUI test automation. These will result in the cost saving for the testing project.
- **Language-Independent:** In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent, allowing users to select any code language when adopting automation testing services for the project.
- **Easy Integration with GUI:** API tests enable highly integrable tests, which is particularly useful if you want to perform functional GUI tests after API testing. For instance, simple integration would allow new user accounts to be created within the application before a GUI test started.

8. Some common protocols used in API testing?

Many protocols are now available to be used in API testing, such as JMS, REST, HTTP, UDDI and SOAP.

9. What is the test environment of API?

Setting up the API's test environment is not an easy task, so you should have a ready answer if your API testing interview is coming. The test environment of API is a bit complete and requires the configuration of the database and server, depending on the software requirements. No GUI (Graphical User Interface) is available in this test form.

When the installation process is complete, API is verified for the proper operation. Throughout the process, the API called from the original environment is set up with different parameters to study the test results.

10. What are principles of an API test design?

The five most important principles of an API test design are:

- **Setup:** Create objects, start services, initialize data, etc
- **Execution:** Steps to apply API or the scenario, including logging

- Verification: Oracles to evaluate the result of the execution
- Reporting: Pass, failed or blocked
- Clean up: Pre-test state

11. What are the common API testing types?

While there are certainly specialty tests, and no list can be asked to be comprehensive in this realm, most tests fit broadly into these following nine categories that you should remember before attending in an API testing interview.

1. Validation Testing
2. Functional Testing
3. UI testing
4. Load testing
5. Runtime/ Error Detection
6. Security testing
7. Penetration testing
8. Fuzz testing
9. Interoperability and WS Compliance testing

12. What is the procedure to perform API testing?

1. Choose the suite to add the API test case
2. Choose the test development mode
3. Demand the development of test cases for the required API methods
4. Configure the control parameters of the application and then test conditions
5. Configure method validation
6. Execute the API test
7. Check test reports and filter API test cases
8. Arrange all API test cases

13. What must be checked when performing API testing?

During the API testing process, a request is raised to the API with the known data. This way you can analyze the validation response. While testing an API, you should consider:

- Accuracy of data

- Schema validation
- HTTP status codes
- Data type, validations, order and completeness
- Authorization checks
- Implementation of response timeout
- Error codes in case API returns, and
- Non-functional testing like performance and security testing

14. What is the best approach method to perform API testing?

The following factors should be considered when performing API testing:

- Defining the correct input parameters
- Verifying the calls of the mixture of two or more added value parameters
- Defining the basic functionality and scope of the API program
- Writing appropriate API test cases and making use of testing techniques such as equivalence class, boundary value, etc. to check the operability
- Testing case execution
- Comparing the test result with the expected result
- Verifying the API behavior under conditions such as connection to files and so on.

15. What are tools could be used for API testing?

There are myriad of different API testing tools available. A few of common tools are Katalon Studio, Postman, SoapUi Pro, Apigee, etc. While doing Unit and API testing, both targets source code. If an API method uses code based in .NET then another supporting tool must have .NET.

17. What are differences between API Testing and UI Testing?

- API enables communication between two separate software systems. A software system implementing an API contains functions or subroutines that can be executed by another software system.
- On the other hand, UI (User Interface) testing refers to testing graphical interface such as how users interact with the applications, testing application elements like fonts, images, layouts etc. UI testing basically focuses on look and feel of an application.

18. What are major challenges faced in API testing?

If you can overcome the challenges in API Testing, you can be confident in the API testing interview too. They are:

- Parameter Selection
- Parameter Combination
- Call sequencing
- Output verification and validation
- Another important challenge is providing input values, which is very difficult as GUI is not available in this case.

19. What are the testing methods that come under API testing?

One of the most common Web API testing interview questions is about the testing methods. They are:

- Unit testing and Functional testing
- Load testing to test the performance under load
- Discovery testing to list, create and delete the number of calls documented in API
- Usability and Reliability testing to get consistent results
- Security and Penetration testing to validate all types of authentication
- Automation testing to create and run scripts that require regular API calls
- End to end Integration and Web UI testing
- API documentation testing to determine its efficiency and effectiveness

20. Why is API testing considered as the most suitable form for Automation testing?

API testing is now preferred over GUI testing and is considered as most suitable because:

- It verifies all the functional paths of the system under test very effectively.
- It provides the most stable interface.
- It is easier to maintain and provides fast feedback.

21. What are common API errors that often founded?

Not only API fundamental questions, the interviewer also determine your knowledge and experience by asking about the API errors in a Web API testing interview. So the most common ones are:

- Missing module errors

- Documentation errors
- Parameter validation errors
- And some standard error expectations as if the result is not so predicted then the occurrence of errors can be seen and for the same warnings are specified in the form of a message. There can be one or more warnings within an individual module.

22. What kinds of bugs that API testing would often find?

- Missing or duplicate functionality
- Fails to handle error conditions gracefully
- Stress
- Reliability
- Security
- Unused flags
- Not implemented errors
- Inconsistent error handling
- Performance
- Multi-threading issues
- Improper errors

23. What is API documentation?

The API documentation is a complete, accurate technical writing giving instructions on how to effectively use and integrate with an API. It is a compact reference manual that has all the information needed to work with the API, and helps you answer all the API testing questions with details on functions, classes, return types, arguments, and also examples and tutorials.

24. What are API documentation templates that are commonly used?

There are several available API documentation templates help to make the entire process simple and straightforward, which could be answered in your API testing interview, such as:

- Swagger
- Miredot
- Slate

- FlatDoc
- API blueprint
- RestDoc
- Web service API specification

25. When writing API document, what must be considered?

- Source of the content
- Document plan or sketch
- Delivery layout
- Information needed for every function in the document
- Automatic document creation programs

26. How often are the APIs changed and, more importantly, deprecated?

APIs, especially modern RESTful APIs, are a nice creation that can certainly simplify and accelerate integration efforts, which makes it more likely you will benefit from them. But APIs can and do change for various reasons, sometimes abruptly, and hence REST APIs do not differ from traditional integration methods in this respect. If an API call is obsolete and disappears, your procedure will interrupt and it is important to understand how often the APIs you depend on change or are deprecated.

27. What is REST?

REST (Representational State Transfer) is an architectural style for developing web services which exploit the ubiquity of HTTP protocol and uses HTTP method to define actions. It revolves around resource where every component being a resource that can be accessed through a shared interface using standard HTTP methods.

In REST architecture, a REST Server provides access to resources and REST client accesses and makes these resources available. Here, each resource is identified by URIs or global IDs, and REST uses multiple ways to represent a resource, such as text, JSON, and XML. XML and JSON are nowadays the most popular representations of resources.

28. What is a RESTful Web Services?

Mostly, there are two kinds of Web Services which should be remembered in your next API testing interview:

1. SOAP (Simple Object Access Protocol) – an XML-based method to expose web services.
2. Web services developed in the REST style are referred to as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service

usually defines a URI, Uniform Resource Identifier a service, provides resource representation like JSON and a set of HTTP methods.

29. What is a “Resource” in REST?

REST architecture treats any content as a resource, which can be either text files, HTML pages, images, videos or dynamic business information.

REST Server gives access to resources and modifies them, where each resource is identified by URIs/global IDs.

30. What is the most popular way to represent a resource in REST?

REST uses different representations to define a resource like text, JSON, and XML.

XML and JSON are the most popular representations of resources.

31. Which protocol is used by RESTful Web services?

RESTful web services use the HTTP protocol as a medium of communication between the client and the server.

32. What are some key characteristics of REST?

Key characteristics of REST are likely asked in a Web API Testing interview. So please get the answer ready in your mind with these 2 ones:

- REST is stateless, therefore the SERVER has no status (or session data)

With a well-applied REST API, the server could be restarted between two calls, since all data is transferred to the server

- Web service uses POST method primarily to perform operations, while REST uses GET for accessing resources.

33. What is messaging in RESTful Web services?

RESTful web services use the HTTP protocol as a communication tool between the client and the server. The technique that when the client sends a message in the form of an HTTP Request, the server sends back the HTTP reply is called Messaging. These messages comprise message data and metadata, that is, information on the message itself.

34. What are the core components of an HTTP request?

An HTTP request contains five key elements:

1. An action showing HTTP methods like GET, PUT, POST, DELETE.
2. Uniform Resource Identifier (URI), which is the identifier for the resource on the server.

3. HTTP Version, which indicates HTTP version, for example-HTTP v1.1.
4. Request Header, which carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, format supported by the client, format of a message body format, cache settings, and so on.
5. Request Body, which indicates the message content or resource representation.

35. What are the most commonly used HTTP methods supported by REST?

- GET is only used to request data from a specified resource. Get requests can be cached and bookmarked. It remains in the browser history and has length restrictions. GET requests should never be used when dealing with sensitive data.
- POST is used to send data to a server to create/update a resource. POST requests are never cached and bookmarked and do not remain in the browser history.
- PUT replaces all current representations of the target resource with the request payload.
- DELETE removes the specified resource.
- OPTIONS is used to describe the communication options for the target resource.
- HEAD asks for a response identical to that of a GET request, but without the response body.

36. Can GET request be used instead of PUT to create a resource?

The PUT or POST method should be used to create a resource. GET is only used to request data from a specified resource.

37. Is there any difference between PUT and POST operations?

PUT and POST operation are quite similar, except the terms of the result generated by them.

PUT operation is idempotent, so you can cache the response while the responses to POST operation are not cacheable, and if you retry the request N times, you will end up having N resources with N different URIs created on server.

In a Web API Testing interview, you should give a specific example for PUT and POST operations to make crystal clear to the interviewer. Below is an example:

Scenario: Let's say we are designing a network application. Let's list down few URIs and their purpose to get to know when to use POST and when to use PUT operations.

GET /device-management/devices : Get all devices

POST /device-management/devices : Create a new device

GET /device-management/devices/{id} : Get the device information identified by "id"

PUT /device-management/devices/{id} : Update the device information identified by “id”

DELETE /device-management/devices/{id} : Delete device by “id”

38. Which purpose does the OPTIONS method serve for the RESTful Web services?

The OPTIONS Method lists down all the operations of a web service supports. It creates read-only requests to the server.

39. What is URI? What is the main purpose of REST-based web services and what is its format?

URI stands for Uniform Resource Identifier. It is a string of characters designed for unambiguous identification of resources and extensibility via the URI scheme.

The purpose of a URI is to locate a resource(s) on the server hosting of the web service.

A URI's format is <protocol>://<service-name>/<ResourceType>/<ResourceID>.

40. What is payload in RESTful Web services?

The “payload” is the data you are interested in transporting. This is differentiated from the things that wrap the data for transport like the HTTP/S Request/Response headers, authentication, etc.

41. What is the upper limit for a payload to pass in the POST method?

<GET> appends data to the service URL. But, its size shouldn't exceed the maximum URL length. However, <POST> doesn't have any such limit.

So, theoretically, a user can pass unlimited data as the payload to POST method. But, if we consider a real use case, then sending POST with large payload will consume more bandwidth. It'll take more time and present performance challenges to your server. Hence, a user should take action accordingly.

42. What is the caching mechanism?

Caching is just the practice of storing data in temporarily and retrieving data from a high-performance store (usually memory) either explicitly or implicitly.

When a caching mechanism is in place, it helps improve delivery speed by storing a copy of the asset you requested and later accessing the cached copy instead of the original.

43. What are SOAP Web services?

This is one of the fundamental Web services testing questions that you must know the answer. The SOAP (Simple Object Access Protocol) is defined as an XML-based protocol. It is known for designing and developing web services as well as enabling communication between applications developed on different platforms using various programming languages over the Internet. It is both platform and language independent.

44. How does SOAP work?

SOAP is used to provide a user interface that can be accessed by the client object, and the request that it sends goes to the server, which can be accessed using the server object. The user interface creates some files or methods consisting of server object and the name of the interface to the server object. It also

contains other information such as the name of the interface and methods. It uses HTTP to send the XML to the server using the POST method, which analyzes the method and sends the result to the client. The server creates more XML consisting of responses to the request of user interface using HTTP. The client can use any approach to send the XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

45. When to use SOAP API?

Use the SOAP API to create, retrieve, update or delete records, like accounts, leads, and user-defined objects. With more than 20 different calls, you can also use the SOAP API to manage passwords, perform searches, etc. by using the SOAP API in any language that supports web services.

46. How users utilize the facilities provided by SOAP?

- **PutAddress():** It is used to enter an address in the webpage and has an address instance on the SOAP call.
- **PutListing():** It is used to allow the insertion of a complete XML document into the web page. It receives the XML file as an argument and transports the XML file to XML parser liaison, which reads it and inserts it into the SOAP call as a parameter.
- **GetAddress():** It is used to get a query name and gets the result that best matches a query. The name is sent to the SOAP call in the form of text character string.
- **GetAllListing():** It is used to return the full list in an XML format.

47. What is the major obstacle users faced when using SOAP?

When using SOAP, users often see the firewall security mechanism as the biggest obstacle. This block all the ports leaving few like HTTP port 80 and the HTTP port used by SOAP that bypasses the firewall. The technical complaint against SOAP is that it mixes the specification for message transport with the specification for message structure.

48. What are the various approaches available for developing SOAP based web services?

There are two different methods available for developing SOAP-based web services, which are explained below:

- **Contract-first approach:** the contract is first defined by XML and WSDL, and then Java classes are derived from the contract.
- **Contract-last approach:** Java classes are first defined, and then the contract is generated, which is normally the WSDL file from the Java class.

“Contract-first” method is the most popular approach.

49. What are the elements of a SOAP message structure?

It is a common XML document that contains the elements as a SOAP message

Envelope: It is an obligatory root element that translates the XML document and defines the beginning and end of the message.

Header: It is an optional item which contains information about the message being sent.

Body: It contains the XML data comprising the message being sent.

Fault: It provides the information on errors that occurred while during message processing.

50. What are the syntax rules for a SOAP message?

- Must use encoded XML
- Envelope namespace must be used
- Encoding namespace must be used
- Must not consist of a DTD reference
- Must not have XML processing instruction

51. What is the transport method in SOAP?

Application layer and transport layers of a network are used by SOAP; HTTP and SMTP are the valid protocol of the application layer used as the transport for SOAP. HTTP is more preferable, since it works well with the current Internet infrastructure, in particular with firewalls.

The SOAP requests can be sent using an HTTP GET method while the specification only contains details about HTTP POST.

52. What are some important characteristics of a SOAP envelope element?

- SOAP message has a root Envelope element
- Envelope is an obligatory part of the SOAP message.
- If an envelope includes a header element, it should not contain more than one.
- Envelope version will change if the SOAP version changes.
- The SOAP envelope is indicated by the prefix ENV and the envelope element.
- The optional SOAP encoding is also specified using a namespace and the optional encoding style element.

53. What are the major functionalities provided by the SOAP protocol class?

The SOAP protocol is used to provide simple access methods for all the applications available on the Internet, providing the following functionalities:

- **Call:** A class which provides the main functionality for a remote method for which a call is needed. It is used to create the call() and to specify the encoding style of the registry that will be assigned when if necessary. This call() function is used by the RPC call, which represents the options of the call object.
- **Deployment Descriptor:** A class used to provide the information about the SOAP services. It enables easy deployment without the need for other approaches.

- DOM2 Writer: A class that serializes and uses DOM node as XML string to provide more functionalities.
- RPC Message: A class used as the base class that calls and replies to the request submitted to the server.
- Service Manager: A class that provides, lists and then outputs all SOAP services.

54. What are the web relation functionalities provided by SOAP protocol?

- HTTPUtils: This provides the functionality of the POST method to safely meet the requirements.
- Parameter: It is an argument for an RPC call used by both the client and the server.
- Response: It is an object that represents an RPC reply from both client and server, but the result will not be displayed until after the method call.
- TCPTunnel: It is an object that provides the ability to listen on a specific port and to forward all the host and port names.
- TypeConverter: It helps to convert an object of one type into another type and this is called using the class in the form object.

55. How does the message security model allow the creation of SOAP more secure to use?

The security model includes the given security tokens. These tokens comprise digital signatures for protection and authentication of SOAP messages. Security tokens can be used to provide the bond between authentication secrets or keys and security identities. Security token uses the authentication protocols and an X.509 certificate to define the relationship between the public key and identity key. The signatures are used to verify the messages and their origin, generate knowledge to confirm the security tokens to bind the identity of a person to the identity of the originator. Security model prevents different attacks and can be used to protect the SOAP architecture.

56. What is the difference between top down & bottom up approach in SOAP Web services?

- Top down SOAP Web services include creating WSDL document to create a contract between the web service and the client, with a required code as an option. This is also known as Contract-first approach. The top-down approach is difficult to implement because classes must be written to confirm the contract defined in WSDL. One of the benefits of this method is that both client and server code can be written in parallel.
- Bottom up SOAP web services require the code to be written first and then WSDL is generated. It is also known as Contract-last approach. Since WSDL is created based on the code, bottom-up approach is easy to implement and client codes must wait for WSDL from the server side to start working.

57. What are advantages of SOAP?

- SOAP is both platform and language independent.
- SOAP separates the encoding and communications protocol from the runtime environment.

- Web service can retrieve or receive a SOAP user data from a remote service, and the source's platform information is completely independent of each other.
- Everything can generate XML, from Perl scripts through C++ code to J2EE app servers.
- It uses XML to send and receive messages.
- It uses standard internet HTTP protocol.
- SOAP runs over HTTP; it eliminates firewall problems. When protocol HTTP is used as the protocol binding, an RPC call will be automatically assigned to an HTTP request, and the RPC response will be assigned to an HTTP reply.
- Compared to RMI, CORBA and DCOM, SOAP is very easy to use.
- SOAP acts as a protocol to move information in a distributed and decentralized environment.
- SOAP is independent of the transport protocol and can be used to coordinate different protocols.

58. What are disadvantages of SOAP?

SOAP is typically significantly slower than other types of middleware standards, including CORBA, because SOAP uses a detailed XML format. A complete understanding of the performance limitations before building applications around SOAP is hence required.

SOAP is usually limited to pooling and not to event notifications when HTTP is used for the transport. In addition, only one client can use the services of one server in typical situations.

If HTTP is used as the transport protocol, firewall latency usually occurs since the firewall analyzes the HTTP transport. This is because HTTP is also leveraged for Web browsing, and so many firewalls do not understand the difference between using HTTP within a web browser and using HTTP within SOAP.

SOAP has different support levels, depending on the supported programming language. For instance, SOAP supported in Python and PHP is not as powerful as it is in Java and .NET

60. SOAP or Rest APIs, which method to use?

SOAP is the heavyweight choice for Web service access. It provides the following advantages when compared to REST:

- SOAP is not very easy to implement and requires more bandwidth and resources.
- SOAP message request is processed slower as compared to REST and it does not use web caching mechanism.
- WS-Security: While SOAP supports SSL (just like REST) it also supports WS-Security which adds some enterprise security features.
- WS-AtomicTransaction: Need ACID Transactions over a service, you're going to need SOAP.

- **WS-ReliableMessaging:** If your application needs Asynchronous processing and a guaranteed level of reliability and security. Rest doesn't have a standard messaging system and expects clients to deal with communication failures by retrying.

- If the security is a major concern and the resources are not limited then we should use SOAP web services. Like if we are creating a web service for payment gateways, financial and telecommunication related work, then we should go with SOAP as here high security is needed.

REST is easier to use for the most part and is more flexible. It has the following advantages when compared to SOAP:

- Since REST uses standard HTTP, it is much simpler.
- REST is easier to implement, requires less bandwidth and resources.
- REST permits many different data formats whereas SOAP only permits XML.
- REST allows better support for browser clients due to its support for JSON.
- REST has better performance and scalability. REST reads can be cached, SOAP based reads cannot be cached.
- If security is not a major concern and we have limited resources. Or we want to create an API that will be easily used by other developers publicly then we should go with REST.
- If we need Stateless CRUD operations then go with REST.
- REST is commonly used in social media, web chat, mobile services and Public APIs like Google Maps.
- RESTful service returns various MediaTypes for the same resource, depending on the request header parameter "Accept" as application/xml or application/json for POST and /user/1234.json or GET /user/1234.xml for GET.
- REST services are meant to be called by the client-side application and not the end user directly.
- ST in REST comes from State Transfer. You transfer the state around instead of having the server store it, this makes REST services scalable.

61. What are the factors that help to decide which style of Web services – SOAP or REST – to use?

Generally, REST is preferred due to its simplicity, performance, scalability, and support for multiple data formats.

However, SOAP is favorable to use where service requires an advanced level of security and transactional reliability.

But you can read the following facts before opting for any of the styles.

- Does the service expose data or business logic? REST is commonly used for exposing data while SOAP for logic.
- The requirement from clients or providers for a formal contract. SOAP can provide contract via WSDL.
- Support multiple data formats.
- Support for AJAX calls. REST can apply the XMLHttpRequest.
- Synchronous and asynchronous calls. SOAP enables both synchronous/ asynchronous operations whereas REST has built-in support for synchronous.
- Stateless or Stateful calls. REST is suited for stateless operations.
- Security. SOAP provides a high level of security.
- Transaction support. SOAP is good at transaction management.
- Limited bandwidth. SOAP has a lot of overhead when sending/receiving packets since it's XML based, requires a SOAP header. However, REST requires less bandwidth to send requests to the server. Its messages are mostly built using JSON.
- Ease of use. REST based application is easy to implement, test, and maintain.

Cucumber Interview Questions

Q #1) Explain Cucumber shortly.

Ans: Cucumber is a tool that is based on Behavior Driven Development (BDD) methodology.

The main aim of Behavior Driven Development framework is to make various project roles such as Business Analysts, Quality Assurance, Developers etc., understand the application without diving deep into the technical aspects.

Q #2) What language is used by Cucumber?

Ans: Gherkin is the language that is used by the Cucumber tool. It is a simple English representation of the application behavior. Gherkin language uses several keywords to describe the behavior of application such as Feature, Scenario, Scenario Outline, Given, When, Then etc.

Q #3) What is meant by a feature file?

Ans: A feature file must provide a high-level description of an Application Under Test (AUT). The first line of the feature file must start with the keyword 'Feature' following the description of the application under test.

A feature file may include multiple scenarios within the same file. A feature file has the extension (.feature).

Q #4) What are the various keywords that are used in Cucumber for writing a scenario?

Ans: Mentioned below are the keywords that are used for writing a scenario:

- Given
- When
- Then
- And
-

Q #5) What is the purpose of Scenario Outline in Cucumber?

Ans: Scenario outline is a way of parameterization of scenarios. This is ideally used when the same scenario needs to be executed for multiple sets of data, however, the test steps remain the same. Scenario Outline must be followed by the keyword 'Examples', which specify the set of values for each parameter.

Q #6) What programming language is used by Cucumber?

Ans: Cucumber tool provides support for multiple programming languages such as Java, .Net, Ruby etc. It can also be integrated with multiple tools such as Selenium, Capybara etc.

Q #7) What is the purpose of Step Definition file in Cucumber?

Ans: A step definition file in Cucumber is used to segregate the feature files from the underlying code. Each step of the feature file can be mapped to a corresponding method on the Step Definition file.

While feature files are written in an easily understandable language such as Gherkin, Step Definition files are written in programming languages such as Java, .Net, Ruby etc.

Q #8) What are the major advantages of Cucumber framework?

Ans: Given below are the advantages of Cucumber Gherkin framework that make Cucumber an ideal choice for rapidly evolving Agile methodology in today's corporate world.

- Cucumber is an open source tool.
- Plain Text representation makes it easier for non-technical users to understand the scenarios.
- It bridges the communication gap between various project stakeholders such as Business Analysts, Developers, and Quality Assurance personnel.
- Automation test cases developed using the Cucumber tool are easier to maintain and understand as well.
- Easy to integrate with other tools such as Selenium

Q #9) Provide an example of a feature file using the Cucumber framework.

Ans: Following is an **Example** of a feature file for the scenario 'Login into the application':

Feature: Login to the application under test.

Scenario: Login to the application.

Given Open Chrome browser and launch the application.

When User enters the username onto the UserName field.

And User enters the password into the Password field.

When User clicks on the Login button.

Then Validate if the user login is successful.

Q #10) Provide an example of Scenario Outline using Cucumber framework.

Ans: The following is an **Example** of Scenario Outline keyword for the scenario 'Upload a file'. Number of parameter values to be included in the feature file is based on the tester's choice.

Scenario Outline: Upload a file

Given that the user is on upload file screen.

When a user clicks on the Browse button.

And user enters <filename> onto the upload textbox.

And user clicks on the enter button.

Then verify that the file upload is successful.

Examples:

|filename|

|file1|

|file2|

Q #11) What is the purpose of Behavior Driven Development (BDD) methodology in the real world?

Ans: BDD is a methodology to understand the functionality of an application in simple plain text representation.

The main aim of Behavior Driven Development framework is to make various project roles such as Business Analysts, Quality Assurance, Developers, Support Team understand the application without diving deep into the technical aspects.

Q #12) What is the limit for the maximum number of scenarios that can be included in the feature file?

Ans: A feature file can contain a maximum of 10 scenarios, but the number can vary from project to project and from one organization to another. But it is generally advisable to limit the number of scenarios included in the feature file.

Q #13) What is the use of Background keyword in Cucumber?

Ans: Background keyword is used to group multiple given statements into a single group. This is generally used when the same set of given statements are repeated in each scenario of the feature file.

Q #14) What symbol is used for parameterization in Cucumber?

Ans: Pipe symbol (|) is used to specify one or more parameter values in a feature file.

Q #15) What is the purpose of Examples keyword in Cucumber?

Ans: Examples keyword is used to specify values for each parameter used in the scenario. Scenario Outline keyword must always be followed by the keyword Examples.

Q #16) What is the file extension for a feature file?

Ans: File Extension for a feature file is .feature. A feature file is ideally written in a notepad file and is saved with the extension feature.

Q #17) Provide an example of step definition file in Cucumber.

Ans: Step definition corresponding to the step “Open Chrome browser and launch the application” may look like the code mentioned below:

```
@Given("^Open Chrome browser and launch the application$")
```

```
public void openBrowser()
```

```
{
```

```
driver = new ChromeDriver();
```

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

```
driver.get("www.facebook.com");
```

```
}
```

Q #18) What is the purpose of Cucumber Options tag?

Ans: Cucumber Options tag is used to provide a link between the feature files and step definition files. Each step of the feature file is mapped to a corresponding method on the step definition file.

Below is the syntax of Cucumber Options tag:

```
@CucumberOptions(features="Features",glue={"StepDefinition"})
```

Q #19) How can Cucumber be integrated with Selenium WebDriver?

Ans: Cucumber can be integrated with Selenium webdriver by downloading the necessary JAR files.

Given below are the list of JAR files that are to be downloaded for using Cucumber with Selenium web driver:

- cucumber-core-1.2.2.jar
- cucumber-java-1.2.2.jar
- cucumber-junit-1.2.2.jar
- cucumber-jvm-deps-1.0.3.jar
- cucumber-reporting-0.1.0.jar
- gherkin-2.12.2.jar

Q #20) When is Cucumber used in real time?

Ans: Cucumber tool is generally used in real time to write acceptance tests for an application.

Q #21) Provide an example of Background keyword in Cucumber.

Ans:

Background: Given user is on the application login page.

Q #22) What is the use of Behavior Driven Development in Agile methodology?

Ans: The advantages of Behavior Driven Development are best realized when non-technical users such as Business Analysts use BDD to draft requirements and provide the same to the developers for implementation.

In Agile methodology, user stories can be written in the format of feature file and the same can be taken up for implementation by the developers.

Q #23) Explain the purpose of keywords that are used for writing a scenario in Cucumber.

Ans: “Given” keyword is used to specify a precondition for the scenario. When a keyword is used to specify an operation to be performed. Then keyword is used to specify the expected result of a performed action. “And” keyword is used to join one or more statements together into a single statement.

Q #24) What is the name of the plugin that is used to integrate Eclipse with Cucumber?

Ans: Cucumber Natural Plugin is the plugin that is used to integrate Eclipse with Cucumber.

Q #25) What is the meaning of TestRunner class in Cucumber?

Ans: TestRunner class is used to provide the link between feature file and step definition file. Below is the sample representation of how TestRunner class will look like. A TestRunner class is generally an empty class with no class definition.

Q #26) Provide an example of TestRunner class in Cucumber.

Ans:

```
Package com.sample.TestRunner
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;

@RunWith(Cucumber.class)
@CucumberOptions(features="Features", glue={"StepDefinition"})
public class Runner
{
}
```

Q #27) What is the starting point of execution for feature files?

Ans: When integrated with Selenium, the starting point of execution must be from TestRunner class.

Q #28) Should any code be written within TestRunner class?

Ans: No code should be written under the TestRunner class. It should include the tags @RunWith and @CucumberOptions.

Q #29) What is the use of features property under Cucumber Options tag?

Ans: Features property is used to let Cucumber framework identify the location of the feature files.

Q #30) What is the use of glue property under Cucumber Options tag?

Ans: Glue property is used to let Cucumber framework identify the location of step definition files.

Q #31) What is the maximum number of steps that are to be written within a scenario?

Ans: The maximum number of steps to be written in a scenario is 3-4 steps. But depends on scenario and it could be more than that

Additional Resources:

<https://career.guru99.com/top-15-cucumber-interview-questions/>

<https://www.educba.com/cucumber-interview-questions/>

<https://www.zeolearn.com/interview-questions/cucumber>

<https://www.enozom.com/blog/top-10-cucumber-interview-questions>

Tek School

Java Interview Questions

Part 1

1. What is Java?

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

2. Mention some features of Java?

Some of the features which play important role in the popularity of java are as follows:

- **Object-Oriented:** Java is an object-oriented programming language. Everything in Java is an Object.

Portable: Java run time environment uses a bytecode verification process to make sure that code loaded over the network doesn't violate Java security constraints.

- **Platform independent:** Java is platform independent. Java is a write once, run anywhere language. Without any modifications, we can use a program in different platforms.
- **Secured:** Java is well known for its security. It delivers virus free systems.
- **High Performance:** Java enables high performance with the use of JIT (Just-In-Time) compilers
- **Multithreaded:** Java Multithreaded features allows us to write programs that can perform many tasks simultaneously. Multithreading concept of Java shares a common memory area. It doesn't occupy memory for each thread.

3. What is the difference between Declaration and Definition in Java?

Declaration: If you just declare a class or method/function or variable without mentioning anything about what that class or method/function or variable looks like is called as declaration in Java.

Definition: If you define how a class or method/function or variable is implemented then it is called definition in Java.

When we create an interface or abstract class, we simply declare a method/function but not define it.

4. What is an Object in Java?

An object is an instance of a class. Objects have state (variables) and behavior (methods).

Example: A dog is an object of Animal class. The dog has its states such as color, name, breed, and behaviors such as barking, eating, wagging her tail.

5. What is a Class in Java?

A class can be defined as a collection of objects. It is the blueprint or template that describes the state and behavior of an object.

6. What is Constructor in Java?

Constructor in Java is used in the creation of an Object that is an instance of a Class. Constructor name should be same as class name. It looks like a method but it's not a method. It won't return any value. We have seen that methods may return a value. If there is no constructor in a class, then compiler automatically creates a default constructor.

7. What is Local Variable and Instance Variable?

Local Variable:

Local variable is a variable which we declare inside a Method. A method will often store its temporary state in local variables.

Instance Variable (Non-static):

Instance variable is a variable which is declared inside a Class but outside a Method. We don't declare this variable as Static because these variables are non-static variables.

8. What are the OOPs concepts?

OOPS Stands for Object Oriented Programming System. It includes Abstraction, Encapsulation, Inheritance, Polymorphism, Interface etc.,

9. What is Inheritance in Java?

Inheritance is a process where one class inherits the properties of another class.

10. What is Polymorphism?

Polymorphism allows us to perform a task in multiple ways. Let's break the word Polymorphism and see it, 'Poly' means 'Many' and 'Morphos' means 'Shapes'.

Assume we have four students and we asked them to draw a shape. All the four may draw different shapes like Circle, Triangle, and Rectangle.

11. What are the types of Polymorphism?

There are two types of Polymorphism in Java

1. Compile time polymorphism (Static binding) – Method overloading
2. Runtime polymorphism (Dynamic binding) – Method overriding

We can perform polymorphism by 'Method Overloading' and 'Method Overriding'

12.What is Method Overloading?

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

There are three ways to overload a method.

- Parameters with different data types
- Parameters with different sequence of a data types
- Different number of parameters

13.What is Method Overriding?

Declaring a method in child class which is already present in the parent class is called Method Overriding.

In simple words, overriding means to override the functionality of an existing method.

In this case, if we call the method with child class object, then the child class method is called. To call the parent class method we have to use **super** keyword.

14.What is Abstraction in Java?

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

Example: Mobile Phone.

A layman who is using mobile phone doesn't know how it works internally but he can make phone calls.

15.What is Abstract Class in Java?

We can easily identify whether a class is an abstract class or not. A class which contains abstract keyword in its declaration then it is an Abstract Class.

Points to remember:

- Abstract classes may or may not include abstract methods
- If a class is declared abstract then it cannot be instantiated.
- If a class has abstract method then we have to declare the class as abstract class
- When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class. However, if it does not, then the subclass must also be declared abstract.

16.What is Abstract Method?

An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon), like this:

In order to use an abstract method, you need to override that method in sub class.

17.What is Interface in Java?

An interface in Java looks similar to 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. Read more on

18.What is Encapsulation in Java?

Encapsulation is a mechanism of binding code and data together in a single unit. Let's take an example of Capsule. Different powdered or liquid medicines are

encapsulated inside a capsule. Likewise in encapsulation, all the methods and variables are wrapped together in a single class. Read more on

19. Difference between Array and ArrayList?

ARRAY	ARRAYLIST
Array is static	ArrayList is dynamic
Size of the array should be given at the time of array declaration. We cannot change the size of array after creating it	Size of the array may not be required. It changes the size dynamically. Capacity of ArrayList increases automatically whenever we add elements to an ArrayList
Array can contain both primitive data types as well as objects	ArrayList cannot contain primitive data types. It contains only objects

ARRAY	ARRAYLIST
Arrays are multidimensional	ArrayList is always single dimension

20. Difference between ArrayList and HashSet in Java?

ARRAYLIST	HASHSET
ArrayList implements List interface	HashSet implements Set interface
ArrayList allows duplicates	HashSet doesn't allow duplicates
ArrayList is an ordered collection and maintains insertion order of elements	HashSet is an unordered collection and doesn't maintain insertion order
ArrayList is backed by an Array	HashSet is backed by an HashMap instance

ARRAYLIST	HASHSET
ArrayList is an index based	HashSet is object based
In ArrayList, we can retrieve object by calling get() method or remove object by calling remove() method	In HashSet, we can't achieve get() method

21. What are the different access modifiers available in Java?

Access modifiers are subdivided into four types such as Default, Public, Private, Protected

default: The scope of default access modifier is limited to the package only. If we do not mention any access modifier, then it acts like a default access modifier.

private: The scope of private access modifier is only within the classes.

Note: Class or Interface cannot be declared as private

protected: The scope of protected access modifier is within a package and also outside the package through inheritance only.

Note: Class cannot be declared as protected

public: The scope of public access modifier is everywhere. It has no restrictions. Data members, methods and classes that declared public can be accessed from anywhere.

22. Difference between static binding and dynamic binding?

1. Static binding is also known as early binding whereas dynamic binding is also known as late binding.
2. Determining the type of an object at compile time is Static binding whereas determining the type of an object at run time is dynamic binding
3. Java uses static binding for overloaded methods and dynamic binding for overridden methods.

23. Difference between Abstract Class and Interface?

ABSTRACT CLASS	INTERFACE
To declare Abstract class we have to use abstract keyword	To declare Interface we have to use interface keyword
In an Abstract class keyword abstract is mandatory to declare a method as an abstract	In an Interface keyword abstract is optional to declare a method as an abstract. Compiler treats all the methods as abstract by default

ABSTRACT CLASS	INTERFACE
An abstract class contains both abstract methods and concrete methods(method with body)	An interface can have only abstract methods
An abstract class provides partial abstraction	An interface provides fully abstraction
An abstract class can have public and protected abstract methods	An interface can have only public abstract methods
An abstract class can have static, final or static final variables with any access modifiers	An interface can have only public static final variables
An abstract class can extend one class or one abstract class	An interface can extend any number of interfaces

ABSTRACT CLASS	INTERFACE
Abstract class doesn't support multiple inheritance	Interface supports multiple inheritance

24. What is Multiple Inheritance?

If a class implements multiple interfaces, or an interface extends multiple interfaces then it is known as multiple inheritance.

We will update this post “Java Interview Questions For Selenium Testers” ASAP. Keep visiting.

If you like this post, share it with your friends.

25. What are the Java IDE's?

Ans: Eclipse and NetBeans are the IDE's of JAVA.

26. What is mean by Exception?

Ans: An Exception is a problem that can occur during the normal flow of an execution. A method can throw an exception when something fails at runtime. If that exception couldn't be handled, then the execution gets terminated before it completes the task.

If we handled the exception, then the normal flow gets continued. Exceptions are a subclass of java.lang.Exception.

27. What are the types of Exceptions?

Ans: Two types of Exceptions are explained below in detail.

Checked Exception:

These exceptions are checked by the compiler at the time of compilation. Classes that extend Throwable class except Runtime exception and Error are called checked Exception.

Checked Exceptions must either declare the exception using throws keyword (or) surrounded by appropriate try/catch.

E.g. ClassNotFoundException

Unchecked Exception:

These exceptions are not checked during the compile time by the compiler. The compiler doesn't force to handle these exceptions.

It includes:

- Arithmetic Exception
- ArrayIndexOutOfBoundsException
-

28.What are the different ways to handle exceptions?

Ans: Two different ways to handle exception are explained below:

#1) Using try/catch:

A risky code is surrounded by try block. If an exception occurs, then it is caught by the catch block which is followed by the try block.

#2) By declaring throws keyword:

At the end of the method, we can declare the exception using throws keyword.

29.What are the Advantages of Exception handling?

Ans: Given below are the advantages:

- The normal flow of the execution won't be terminated if exception got handled
- We can identify the problem by using catch declaration

30.What are Exception handling keywords in Java?

Ans: Given below are the two Exception Handling Keywords:

try:

When a risky code is surrounded by a try block. An exception occurring in the try block is caught by a catch block. Try can be followed either by catch (or) finally (or) both. But any one of the blocks is mandatory.

catch:

This is followed by try block. Exceptions are caught here.

finally:

This is followed either by try block (or) catch block. This block gets executed regardless of an exception. So generally clean up codes are provided here.

31.What is the final keyword in Java?

Ans:

Final variable:

Once a variable is declared as final, then the value of the variable could not be changed. It is like a constant.

Example:

```
final int = 12;
```

Final method:

A final keyword in a method that couldn't be overridden. If a method is marked as a final, then it can't be overridden by the subclass.

Final class:

If a class is declared as final, then the class couldn't be subclassed. No class can extend the final class.

32.What is a Thread?

Ans: In Java, the flow of a execution is called Thread. Every java program has at least one thread called main thread, the Main thread is created by JVM. The user can define their own threads by extending Thread class (or) by implementing Runnable interface. Threads are executed concurrently.

33.How do you make a thread in Java?

Ans: There are two ways available in order to make a thread.

#1) Extend Thread class:

Extending a Thread class and override the run method. The thread is available in `java.lang.thread`.

The disadvantage of using a thread class is that we cannot extend any other classes because we have already extend the thread class. We can overload the `run ()` method in our class.

#2) Implement Runnable interface:

Another way is implementing the runnable interface. For that we should provide the implementation for `run ()` method which is defined in the interface.

34.What does yield method of the Thread class do?

Ans: A `yield ()` method moves the currently running thread to a runnable state and allows the other threads for execution. So that equal priority threads have a chance to run. It is a static method. It doesn't release any lock.

`Yield ()` method moves the thread back to the Runnable state only, and not the thread to `sleep ()`, `wait ()` (or) `block`.

35.Explain about wait () method.

Ans: wait () method is used to make the thread to wait in the waiting pool. When a `wait ()` method is executed during a thread execution then immediately the thread gives up the lock on the object and goes to the waiting pool. `Wait ()` method tells the thread to wait for a given amount of time.

Then the thread will wake up after `notify ()` (or) `notify all ()` method is called.

`Wait()` and the other above-mentioned methods do not give the lock on the object immediately until the currently executing thread completes the synchronized code. It is mostly used in synchronization.

36.Difference between notify() method and notifyAll() method in Java.

Ans: Given below are few differences between `notify()` method and `notifyAll()` method

notify()	notifyAll()
This method is used to send a signal to wake up a single thread in the waiting pool.	This method sends the signal to wake up all the threads in a waiting spool.

37.How to stop a thread in java? Explain about sleep () method in a thread?

Ans: We can stop a thread by using the following thread methods.

- Sleeping
- Waiting
- Blocked

Sleep:

Sleep () method is used to sleep the currently executing thread for the given amount of time. Once the thread is wake up it can move to the runnable state. So sleep () method is used to delay the execution for some period.

It is a static method.

Example:

Thread. Sleep (2000)

So it delays the thread to sleep 2 milliseconds. Sleep () method throws an `InterruptedException`

38.When to use Runnable interface Vs Thread class in Java?

Ans: If we need our class to extend some other classes other than the thread then we can go with the runnable interface because in java we can extend only one class.

If we are not going to extend any class then we can extend the thread class.

39.Difference between start() and run() method of thread class.

Ans: Start() method creates new thread and the code inside the run () method is executed in the new thread. If we directly called the run() method then a new thread is not created and the currently executing thread will continue to execute the run() method.

40.What is Multi-threading?

Ans: Multiple threads are executed simultaneously. Each thread starts their own stack based on the flow (or) priority of the threads.

On the 1st line execution, JVM calls the main method and the main thread stack looks as shown below.

41.What is Synchronization?

Ans: Synchronization makes only one thread to access a block of code at a time. If multiple thread accesses the block of code, then there is a chance for inaccurate results at the end. To avoid this issue, we can provide synchronization for the sensitive block of codes.

The synchronized keyword means that a thread needs a key in order to access the synchronized code.

Locks are per objects. Every Java object has a lock. A lock has only one key. A thread can access a synchronized method only if the thread can get the key to the objects lock.

For this, we use “Synchronized” keyword.

42.What is the disadvantage of Synchronization?

Ans: Synchronization is not recommended to implement all the methods. Because if one thread accesses the synchronized code then the next thread should have to wait. So it makes slow performance on the other end.

43.What is meant by Serialization?

Ans: Converting a file into a byte stream is known as Serialization. The objects in the file is converted to the bytes for security purposes. For this, we need to implement java.io.Serializable interface. It has no method to define.

Variables that are marked as transient will not be a part of the serialization. So we can skip the serialization for the variables in the file by using a transient keyword.

44.What is the purpose of a transient variable?

Ans: Transient variables are not part of the serialization process. During deserialization, the transient variables values are set to default value. It is not used with static variables.

Example:

transient int numbers;

45.Which methods are used during Serialization and Deserialization process?

Ans: ObjectOutputStream and ObjectInputStream classes are higher level java.io. package. We will use them with lower level classes FileOutputStream and FileInputStream.

ObjectOutputStream.writeObject —->Serialize the object and write the serialized object to a file.

ObjectInputStream.readObject —> Reads the file and deserializes the object.

To be serialized, an object must implement the serializable interface. If superclass implements Serializable, then the subclass will automatically be serializable.

46.What is the purpose of a Volatile Variable?

Ans: Volatile variable values are always read from the main memory and not from thread's cache memory. This is used mainly during synchronization. It is applicable only for variables.

Example:

volatile int number;

To read more about all the questions above, please refer to the following links:

<https://www.softwaretestingmaterial.com>

<https://www.softwaretestinghelp.com>

Tek School

Java Interview Questions

Part 2

1. Difference between Serialization and Deserialization in Java.

Ans: These are the difference between serialization and deserialization in java:

Serialization	Deserialization
Serialization is the process which is used to convert the objects into byte stream	Deserialization is the opposite process of serialization where we can get the objects back from the byte stream.
An object is serialized by writing it an ObjectOutputStream.	An object is deserialized by reading it from an ObjectInputStream.

<https://www.quora.com/What-is-the-difference-between-serialization-and-deserialization-in-core-Java>

2. What is serialVersionUID?

Ans: Whenever an object is Serialized, the object is stamped with a version ID number for the object class. This ID is called the serialVersionUID. This is used during deserialization to verify that the sender and receiver that are compatible with the Serialization.

<https://www.geeksforgeeks.org/serialversionuid-in-java/>

3. Write a program to swap 2 numbers without a temporary variable?

```
int a=10;
int b=20;

a=a+b;// first this should be there a=10+20=30
b=a-b; // b= 30-20=10
a=a-b; //a=30-10=20

System.out.println(a);
System.out.println(b);

//swap strings
String x="Hello";
String y="TekSchool";
```

```
x=x+y; //HelloTekSchool
y=x.substring(0,(x).length()-y.length());
x=x.substring(y.length());
```

```
System.out.println(x);
System.out.println(y);
```

<https://www.geeksforgeeks.org/swap-two-numbers-without-using-temporary-variable/>

4. Find out how many alpha characters present in a string?

```
String given="wefeqf878979797fewfewrf879797efds&^&^*^^^";
String replaced=given.replaceAll("[a-zA-Z]", "");
int alphaChar=given.length()-replaced.length();
System.out.println(alphaChar);
```

<https://stackoverflow.com/questions/7252142/how-to-check-how-many-letters-are-in-a-string-in-java/7252166>

5. How to find whether given number is odd number?

```
int a=22;
```

```
if (a%2!=0) {
    System.out.println(a+" is an odd number");
}else {
    System.out.println(a+" is not an odd number");
}
```

<https://stackoverflow.com/questions/7342237/check-whether-number-is-even-or-odd/7342253>

6. How to find out the part of the string from a string? What is substring? Find number of words in string?

```
String a="You Made it to Tek School Interview Session";
```

```
System.out.println(a.substring(35));
String [] words=a.split(" ");
```

```
System.out.println(words.length);
```

```
for (String string : words) {  
    System.out.println(string);  
}
```

<https://www.geeksforgeeks.org/searching-for-character-and-substring-in-a-string/>

7. Write java program to reverse String? Reverse integer?

Reverse String:

```
String a= "Hello Amerca";  
// 1 method  
StringBuffer sb=new StringBuffer(a);  
System.out.println(sb.reverse());  
  
//2 method  
String reverse="";  
for (int i=a.length()-1; i>=0; i--) {  
    reverse=reverse+a.charAt(i);  
}  
System.out.println(reverse);  
  
// 3 method  
String reverse1="";  
for (int i=a.length(); i>=1; i--) {  
    reverse1=reverse1+a.substring(i-1, i);  
}  
System.out.println(reverse1);
```

Reverse Integer

```
int a=123;  
// 1 method - cheapest way  
String numbers=String.valueOf(a);  
StringBuffer sb=new StringBuffer(numbers);  
System.out.println(sb.reverse());
```

Look for other solution if interested

```
int currentNum=12345;  
int reversedNum=0;
```



```
while(currentNum!=0) {  
    reversedNum=reversedNum*10+currentNum%10;  
    currentNum=currentNum/10;  
}  
System.out.println(reversedNum);
```

<https://www.programmingsimplified.com/java/source-code/java-program-reverse-number>

8. Write a program to sort array in ascending order?

//1 method

```
int [] a= {12,24,2,4,1};  
Arrays.sort(a);  
System.out.println(Arrays.toString(a));
```

<https://www.sanfoundry.com/java-program-sort-array-ascending-order/>

9. How can you convert a String in to an array? Array to String?

Converting String to an Array

```
String a="TekSchoolAmerica";  
char[] array=a.toCharArray();  
System.out.println(array.length);  
for (int i=0; i<array.length; i++) {  
    System.out.println(array[i]);  
}
```

Converting Array to String

```
System.out.println(Arrays.toString(array));  
https://www.quora.com/How-do-I-convert-string-array-to-string-in-Java
```

10. Verify whether given two strings are equal?

```
String a="Hello";  
String b= "Hello";
```

1 method

```
if (a.equals(b)) {
```

```
System.out.println("String "+a+" and String "+b+" are equal");
}else {
System.out.println("String "+a+" and String "+b+" are not equal");
}
```

2 method

```
if (a.compareTo(b)==0) {
System.out.println("String "+a+" and String "+b+" are equal");
}else {
System.out.println("String "+a+" and String "+b+" are not equal");
}
```

<https://www.geeksforgeeks.org/compare-two-strings-in-java/>

11.What is the difference between String and StringBuffer? What is mutable and immutable?

The most important difference between String and StringBuffer in java is that String object is

immutable whereas StringBuffer object is mutable.

The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.

By immutable, we mean that the value stored in the String object cannot be changed.

For example we cannot reverse string directly, only through using StringBuffer class.

<https://stackoverflow.com/questions/2971315/string-stringbuffer-and-stringbuilder>

immutability vs. mutability

String is immutability class it means once we are creating String objects it is not possible to

perform modifications on existing object. (String object is fixed object)

StringBuffer is a mutability class it means once we are creating StringBuffer objects on that

existing object it is possible to perform modification.

```
class Test {

public static void main(String[] args) {
```

```
String a="Hello";
String b="Hello";

StringBuffer sb=new StringBuffer("Hello TekSchool");

a=a.concat(" TekSchool");
System.out.println(a);
sb=sb.append(" America");
System.out.println(sb);
    }
}
```

<https://www.interviewcake.com/concept/java/mutable>

12.What are the collections you have used?

Mostly in my current project we use ArrayList (There are other collections Set and Maps - just for your awareness and you can mention that you don't use them in your project).

If you know it please explain to the interviewer.

<http://www.java67.com/2013/01/difference-between-set-list-and-map-in-java.html>

13.What are arrays and list? Difference between them?

1. Arrays are fixed in size but ArrayLists are dynamic in size.
2. Arrays can store homogeneous elements whereas collections can store both.
Example: in
Array we can store either int or String or boolean whereas in Array list we can store all of them together
3. To find the size on an Array we use `ArrayName.length` and for arrayList we use `ArrayListName.size()`

ArrayList:

```
ArrayList al=new ArrayList();
al.add("JackMa");
al.add("JackMa");
al.add(10);
for (int i=0; i<al.size(); i++) {
    System.out.println(al.get(i));
}
```

Array:

```
[] array=new String[3];
array[0]="John";
array[1]="Jordon";
array[2]="Jordon";
for(int i=0; i<array.length; i++) {
    System.out.println(array[i]);
}
```

<https://www.quora.com/What-is-the-difference-between-an-array-and-an-array-list>

14.What is constructor? Use of constructor in class?

Constructor gets invoked when a new object is created.

Every class has a constructor. If we do not explicitly write a constructor for a class the java compiler

builds a default constructor for that class.

Constructors must have the same name as the class and can not return a value.

They are only called once while regular methods could be called many times.

```
public BaseProject() {
    try {
        prop = new Properties();
        FileInputStream fis = new FileInputStream("filePath.properties");
        prop.load(fis);
    } catch (IOException e) {
        e.getMessage();
    }
}
```

OR

```
public LoginPage() {
    //initialize element using driver and this means current class objects
    PageFactory.initElements(driver, this);
}
```

<https://www.javatpoint.com/java-constructor>

15.How can we access variable without creating an object instance of it?

What is Instance

variables and how you use it?

By declaring variable as a static we can access it from different class - those variables called class variables and also known as static variables are declared with the **static** keyword in a class, but outside a method, constructor or a block. Whereas, **Instance** variables are declared in a class, but outside a method, constructor or any block.

```
//class variables
public class TestBase {
    public static WebDriver driver;
    public static Properties prop;
```

```
//instance variable
public class LoginStepDefinitions {
    WebDriver driver;
```

<https://www.dummies.com/programming/java/what-is-the-static-keyword-in-java/>

16.How to print without main method "hello"?

You can say that personally you don't find a practical use of such way, but you know it can be done using static block.

```
    static
    {
        System.out.println("hello");
    }
```

NOTE: Static block has the highest priority in java.

So, any thing that is written in static block is executed first.

<https://stackoverflow.com/questions/8605137/printing-message-on-console-without-using-main-method>

17.How can you handle exceptions? How many catch blocks can we have?

By using "Try Catch Block".

There can be any number of catch block for a single try block and It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block or a finally block.

However only the catch block encountered first on the call stack that satisfies the condition for the exception will be executed for that particular exception, rest will be ignored.

```
public TestBase() {  
    try {  
        prop=new Properties();  
        FileInputStream fis=new  
        FileInputStream("fileNamePath.properties");  
        prop.load(fis);  
    } catch (FileNotFoundException e) {  
        e.getMessage();  
    } catch (IOException e) {  
        e.getMessage();  
    } catch (Exception e) {  
        e.getMessage();  
    }  
    System.out.println("Running script after exception");  
}
```

<https://stackabuse.com/exception-handling-in-java-a-complete-guide-with-best-and-worst-practices/>

18.What is the difference between throw and throws?

Throws :

- is used to declare an exception, which means it works similar to the try-catch block.
- is used in method declaration.
- is followed by exception class names.
- you can declare multiple exception with throws
- throws declare at method it might throws Exception
- used to handover the responsibility of handling the exception occurred in the method to the caller method.

1 Example:

```
public void readPropFile() throws FileNotFoundException, IOException{
    Properties prop=new Properties();
    FileInputStream fis=new FileInputStream("fileNamePath.properties");
    prop.load(fis);
}
```

2 Example:

```
public class Test {
    public static void main(String[] args) throws InterruptedException {
        Test test = new Test();
        test.company();
    }
    void studentDetails() throws InterruptedException {
        System.out.println("School is over");
        Thread.sleep(3000);
        System.out.println("Please do not disturb.....");
    }
    void tsa() throws InterruptedException {
        studentDetails();
    }
    void company() throws InterruptedException {
        tsa();
    }
}
```

Throw :

- is used in the method body to throw an exception
- throw is followed by an instance variable
- you cannot declare multiple exceptions with throw
- throw use for throwing actual Exception
- The throw keyword is used to handover the instance of the exception created by the programmer to the JVM manually.
- throw keyword uses the exception object

Example

```
class Test {

    public static void main(String args[]) {
```

```
classAge(15);  
}  
public static void classAge(int age) {  
    try {  
        if (age > 18) {  
            System.out.println("You are eligible to enroll in class");  
        }else {  
            System.out.println("You are not eligible to enroll in class");  
        }throw new ArithmeticException("student is less than 18 years  
old");  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
}  
}
```

<https://beginnersbook.com/2013/04/difference-between-throw-and-throws-in-java/>

19.What is the difference between interface and a class? Example from your framework?

Class :

- Class will contain concrete methods
- Class is extended
- We can create an Object of the class
- Class can inherit only one Class and can implement many interfaces

Interface :

- Interface will have Interface keyword.
- Interface will contain only abstract methods
- We cannot create object of interface
- Interface needs to be implemented
- Class can extends many interfaces
- We need to provide implementation to all methods when we implement interface to the class

```
public interface Insurances {  
    public void auto();  
    public void rental();
```



```
    public void travel();  
}
```

Interface is like a contract with the class, which says I only accept stuff, which looks like the signature that I provide. And class that implements the interface says, sure I will make sure that I look like that.

Practical Example:

Basic statement we all know in Selenium is

```
WebDriver driver = new FirefoxDriver();
```

WebDriver itself is an Interface.

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 and FirefoxDriver is a class.

<https://www.geeksforgeeks.org/difference-between-abstract-class-and-interface-in-java/>

20.Difference between abstract class and interface?

Abstract class:

- Abstract class will have abstract keyword.
- Abstract class can have both abstract as well as concrete methods
- Abstract methods in the sense no implementation only declaration.
- We cannot create an Object of Abstract Class

```
public abstract class Insurance {  
    public void auto() {  
        System.out.println("Insurance auto");  
    }  
    public void rental() {  
        System.out.println("Insurance rental");  
    }  
    //abstract method  
    public abstract void pet()  
}
```

<https://javapapers.com/core-java/abstract-and-interface-core-java-2/difference-between-a-java-interface-and-a-java-abstract-class/>

21. When to use abstract class and interface in Java?

- An abstract class is good if you think you will plan on using inheritance since it provides a common base class implementation to derived classes.
- An abstract class is also good if you want to be able to declare non-public members. In an interface, all methods must be public.
- If you think you will need to add methods in the future, then an abstract class is a better choice. Because if you add new method headings to an interface, then all of the classes that already implement that interface will have to be changed to implement the new methods. That can be quite a hassle.

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

22. Explain OOPS concepts?

OOP concepts in Java are the main ideas behind Java's Object Oriented Programming. They are an abstraction, encapsulation, inheritance, and polymorphism.

<https://stackify.com/oops-concepts-in-java/>

23. What is polymorphism? Types of polymorphism? How do you use it in your code?

Polymorphism is the ability of an object to take on many forms.

Combination of overloading and overriding is known as Polymorphism.

Method Overloading is called as Static Polymorphism/Compile time polymorphisms.

Method Overriding is also called Dynamic Polymorphism/Run time polymorphism.

<https://www.sitepoint.com/quick-guide-to-polymorphism-in-java/>

24. Can we override/overload main method? Explain the reason? Can you override static method?

We cannot override static method, so we cannot override main method.

However, you can overload main method in Java. But the program doesn't execute the overloaded main method when you run your program, you have to call the overloaded main method from the actual main method.

Practically I do not see any use of it and we don't use it in my framework.

```
public class MainMethodOverload {  
    public static void main(String[] args) {  
        main(5); //if comment this line nothing will get executed  
    }  
    public static void main(int r) {  
        System.out.println("Hello");  
    }  
}
```

<http://www.java67.com/2015/06/can-you-overload-or-override-main-in-java.html>

25. How does method override differ from abstraction and inheritance?

Override

When methods with same name and arguments present in child and parents class and class inheriting the method from its **superclass** (parent class) has the option to override it. Benefit of overriding is the ability to define behaviour specific to particular class. Method Overriding is possible only by inheritance.

Inheritance is a process where one class inherits the properties of another class or simply we can say that extending one class into other class is known as **Inheritance**.

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

<https://stackoverflow.com/questions/6308178/what-is-the-main-difference-between-inheritance-and-polymorphism>

26.Method overloading & overriding? How do you use it in your framework?

Method overloading in Java occurs when two or more methods in the same class have the exact same name but different parameters (remember that method parameters accept values passed into the method).

Overloading: Same method name with different arguments **in same class**

Example

```
public static void dropdown(WebElement Object,int value){
    Select index= new Select(Object);
    index.selectByIndex(value);
}
public static void dropdown(WebElement Object,String value){
    Select visibletext= new Select(Object);
    visibletext.selectByValue(value);
}
public static void dropdown(WebElement Object,Object value){
    Select visibletext= new Select(Object);
    visibletext.selectByVisibleText((String) value);
}
```

An overridden method would have the exact same method name, return type, number of parameters, and types of parameters as the method in the parent class, and the only difference would be the body of the method.

Overriding: Changing the implementation of the inherited methods **in the subclass(child class)**

With method overriding a child class can give its own specific implementation to an inherited method without modifying the parent class method.

Combination of overloading and overriding is known as Polymorphism.

<https://softwaretestingboard.com/q2a/2915/provide-example-overloading-overriding-selenium-framework#axzz5orLva55a>

27.What is singleton and have used singleton concept in your project ?

I know what is singleton class, however in my current project I don not use this concept.

(A singleton class is a class that can have only one object (an instance of the class) at a time.)

<https://stackoverflow.com/questions/3192095/where-exactly-the-singleton-pattern-is-used-in-real-application>

28.Can we achieve 100% abstraction in JAVA? Can we achieve 100% abstraction in JAVA

with use of the interfaces?

We cannot achieve 100% abstraction in JAVA unless we use Interfaces

<https://stackoverflow.com/questions/34954592/how-can-interface-achieve-100-abstraction-in-java>

29.What is encapsulation?

It is the technique of making the fields in a class private and providing access to the fields via public

methods. If a field is declared private, it cannot be accessed by anyone outside the class, thereby

hiding the fields within the class. Therefore encapsulation is also referred to as data hiding.

The main benefit of encapsulation is the ability to modify our implemented code without breaking the

code of others who use our code. With this Encapsulation gives maintainability, flexibility and

extensibility to our code.

https://www.tutorialspoint.com/java/java_encapsulation.htm

30.What is the Difference between final, finally?

Final keyword:

- Used to declare constant values. The variable declared as final should be initialized only once and cannot be changed.
- Used to prevent inheritance. Java classes declared as final cannot be extended.
- Used to prevent method overriding. Methods declared as final cannot be overridden.

Example 1:

```
final int b=30;  
b=37; //cannot change value of final variable
```

Example 2:

```
public final class Test {  
    public static void main(String args[]) {  
        System.out.println("I am parent");  
    }  
}
```

//you will get an error "Cannot subclass final class"

```
public class Child extends Test{  
    public static void main(String[] args) {  
        System.out.println("I am a child");  
    }  
}
```

Example 3:

```
public class Test {  
  
    public final void testFinalKey() {  
  
        System.out.println("Parent final method");  
    }  
}
```

//you will get an error "Cannot override the final method"

```
public class Child extends Test{  
  
    public void testFinalKey() {  
  
        System.out.println("Child final method");  
    }  
}
```

Finally block :

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

```
try {
    Properties prop = new Properties();
    FileInputStream fis = new FileInputStream("FilePath");
    prop.load(fis);
} catch (Exception e) {
    System.out.println("I am an exception block");
} finally {
    System.out.println("I am final block");
}
System.out.println("Running script after exception");
```

<https://www.javatpoint.com/difference-between-final-finally-and-finalize>

31.What is Access Modifiers (Private, public, protected)? How did you use them?

Java provides access modifiers to set access levels for classes, variables, methods and constructors.

public: A class or interface may be accessed from outside the package.

Constructors, inner classes, methods and field variables may be accessed wherever their class is accessed.

protected: Accessed by other classes in the same package or any subclasses of same package or different package.

private: Accessed only within the class in which they are declared.

default: When no access modifier is specified for a class , method or data member – It is said to be

having the default access modifier by default.

https://www.tutorialspoint.com/java/java_access_modifiers.htm

32.What is the output for this program?

```
for (int i = 0; i < 3; i++) {
    for (int j = 3; j >= 0; j--) {
        if (i == j)
            continue;
        System.out.println(i + " " + j);
    }
}
```

The continue keyword can be used in any of the loop control structures. It causes the loop to immediately jump to the next iteration of the loop. So the output of the program will be:

```
0 3
0 2
0 1
1 3
1 2
1 0
2 3
2 1
2 0
```

33. Here is the arrayList, how can I remove all duplicates from it?

```
List<String> al = new ArrayList<String>();
al.add("Ajay");
al.add("Becky");
al.add("Chaitanya");
al.add("Ajay");
al.add("Rock");
al.add("Becky");
HashSet hs=new HashSet();
for (int i=0; i<al.size(); i++) {
    hs.add(al.get(i));
}
System.out.println(hs);
```

<https://www.geeksforgeeks.org/how-to-remove-duplicates-from-arraylist-in-java/>

34. What is the output of the following program?

```
class Parent{
    m1(){
        System.out.println("In parent class m1");
    }
}
class Subclass extends parent{
```



```
m1(){
    System.out.println("In child class m1");
}
m2(){
    System.out.println("In m2");
}
public static void main(String args[]){
    Parent obj= new Subclass();
    obj.m1();
    obj.m2();
}
}
```

Program won't run !

```
public static void main(String args[])
Parent obj= new Subclass();
obj.m1(); - will give child output ("In child class m1")
obj.m2(); - this method won't be accessible (child class object is referred by
parent
class reference variable)
```

Tek School of America
Maven, GIT and Jenkins
Interview Questions

What is Maven Tool?

A) Maven is a build automation tool used primarily for Java projects. Maven is a build automation tool used primarily for Java projects.

Why should we use Maven?

A) Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. Maven repository is a directory of packaged JAR file with pom.xml file. Maven searches for dependencies in the repositories.

What is a Maven POM file?

A) A Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects.

What are the repositories in Maven?

A) In Maven terminology, a repository is a directory where all the project jars, library jar, plugins or any other project specific artifacts are stored and can be used by Maven easily.

What does Maven dependency plugin do?

A) Apache Maven Dependency Plugin. The dependency plugin provides the capability to manipulate artifacts. It can copy and/or unpack artifacts from local or remote repositories to a specified location.

What is meant by a goal in Maven?

A) Executing a phase means executes all previous phases. Plugin is a collection of goals. Plugin is a class and goals are methods within the class. Maven is based around the central concept of a build lifecycle.

Where are Maven dependencies downloaded to?

A) The jars , dependency files and other files which are downloaded by Maven reside in the Maven local repository. By default the Maven local repository is the .m2 folder. You can copy the jar directly into where it is meant to go. Maven will find this file next time it is runs.

What is the use of Maven clean?

- A) The Maven Clean Plugin, as the name implies, attempts to clean the files and directories generated by Maven during its build. While there are plugins that generate additional files, the Clean Plugin assumes that these files are generated inside the target directory.

Where are Maven dependencies stored?

A) The maven local repository is a local folder that is used to store all your project's dependencies (plugin jars and other files which are downloaded by Maven). In simple, when you build a Maven project, all dependency files will be stored in your Maven local repository.

How do you tell Maven to use the latest version of a dependency?

In Maven, dependencies are usually set up like this:

```
<dependency>
<groupId>wonderful-inc</groupId>
<artifactId>dream-library</artifactId>
<version>1.2.3</version>
</dependency>
```

How do you find Oracle JDBC driver in Maven repository?

I want to add the oracle jdbc driver to my project as dependency (runtime scope) – ojdbc14. In MVNrepository site the dependency to put in the POM is:

```
<dependency>
<groupId>com.oracle</groupId>
<artifactId>ojdbc14</artifactId>
<version>10.2.0.3.0</version>
</dependency>
```

What is force maven update?

A) mvn clean install -U

-U means force update of snapshot dependencies. Release dependencies can't not be updated this way.

What is GIT?

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.

What is a repository in GIT?

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

What is the command you can use to write a commit message?

The command that is used to write a commit message is "git commit -a". The -a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use "git add<file>" before git commit -a if new files need to be committed for the first time.

What is the difference between GIT and SVN?

The difference between GIT and SVN is

- a) Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.
- b) GIT does not support 'commits' across multiple branches or tags. Subversion allows the creation of folders at any location in the repository layout.
- c) Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

What are the advantages of using GIT?

- a) Data redundancy and replication
- b) High availability
- c) Only one .git directory per repository
- d) Superior disk utilization and network performance
- e) Collaboration friendly
- f) Any sort of projects can use GIT

What is the function of 'GIT PUSH' in GIT?

'GIT PUSH' updates remote refs along with associated objects.

What is the function of git clone?

The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, 'cloning' is the most common way used by programmers.

What is the function of 'git config'?

The 'git config' command is a convenient way to set configuration options for your Git installation. Behaviour of a repository, user info, preferences etc. can be defined through this command.

What does commit object contain?

- a) A set of files, representing the state of a project at a given point of time
- b) Reference to parent commit objects
- c) An SHA1 name, a 40 character string that uniquely identifies the commit object.

How can you create a repository in Git?

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

What is ‘head’ in git and how many heads can be created in a repository?

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”. A repository can contain any number of heads.

What is the purpose of branching in GIT?

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

What is the common branching pattern in GIT?

The common way of creating branch in GIT is to maintain one as “Main“

branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

How can you bring a new feature in the main branch?

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

What is a ‘conflict’ in git?

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

How can conflict in git resolved?

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge, run “git commit”. Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

To delete a branch what is the command that is used?

Once your development branch is merged into the main branch, you don’t need development branch. To delete a branch use, the command “git branch -d [head]”.

What is the difference between 'git remote' and 'git clone'?

'git remote add' just creates an entry in your git config that specifies a name for a particular URL. While, 'git clone' creates a new git repository by copying and existing one located at the URI.

What is 'git status' is used for?

As 'Git Status' shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively

What is Jenkins?

Answer: Jenkins is a free open source Continuous Integration tool and automation server to monitor continuous integration and delivery. It is written in Java.

It is known as an automated Continuous Delivery tool that helps to build and test the software system with easy integration of changes to the system. Jenkins follows Groovy Scripting.

Also, it enables developers to continuously check in their code and also analyze the post-build actions. The automation testers can use to run their tests as soon as the new code is added or code is modified.

What are the features of Jenkins?

Answer: Jenkins comes with the following features:

1. Free open source.
2. Easy installation on various operating systems.
3. Build Pipeline Support.
4. Workflow Plugin.
5. Test harness built around JUnit.
6. Easy upgrades.
7. Rapid release cycle.
8. Easy configuration setup.
9. Extensible with the use of third-party plugins.

What are the advantages of Jenkins? Why we use Jenkins?

Answer: Jenkins is used to continuously monitor the large code base in real time. It enables developers to find bugs in their code and fix. Email notifications are made to the developers regarding their check-ins as a post-build action.

Advantages of Jenkins are as follows:

- Build failures are cached during the integration stage.
- Notifies the developers about build report status using LDAP (Lightweight Directory Access Protocol) mail server.
- Maven release project is automated with simple steps.
- Easy bug tracking.
- Automatic changes get updated in the build report with notification.
- Supports Continuous Integration in agile development and test-driven development.

Mention some of the important plugins in Jenkins?

Answer: Plugins in Jenkins includes:

- Gits
- Maven 2 Project
- HTML Publisher
- Copy Artcraft
- Join
- Green Balls
- Amazon EC2

What is Continuous Integration in Jenkins?

Answer: Continuous integration is the process of continuously checking-in the developer's code into a version control system and triggering the build to check and identify bugs in the written code.

This is a very quick process and also gives them a chance to fix the bugs. Jenkins is one such continuous integration tool.

In software development, multiple developers work on different software modules.

While performing integration testing all the modules are being integrated together. It is considered as the development practice to integrate the code into the source repository

Whenever the programmer/developer makes any change to the current code, then it automatically gets integrated with the system running on the tester's machine and makes the testing task easy and speedy for the system testers.

Continuous Integration comprises of:

- Development and Compilation
- Database Integration
- Unit Testing
- Production Deployment
- Code Labeling
- Functional Testing

- Generating and Analyzing Reports

What is Jenkins pipeline? What is a CI CD pipeline?

Answer: The pipeline can be defined as the suite of plugins supporting implementation and integration of continuous delivery pipelines in Jenkins.

Continuous integration or continuous delivery pipeline consists of build, deploy, test, release pipeline. The pipeline feature saves a lot of time and error in maintaining the builds. Basically, a pipeline is a group of build jobs that are chained and integrated in sequence.

What are Scripted Pipelines in Jenkins?

Answer: Scripted Pipeline follows Groovy

What are Triggers?


Answer: Trigger in Jenkins defines the way in which the pipeline should be executed frequently. PollSCM, Cron, etc are the currently available Triggers.

How to make sure that your project build does not break in Jenkins?

Answer: You need to follow the below-mentioned steps to make sure that the Project build does not break:

1. Clean and successful installation of Jenkins on your local machine with all unit tests.
2. All code changes are reflected successfully.
3. Checking for repository synchronization to make sure that all the differences and changes related to config and other settings are saved in the repository.
4. **How you can set up a Jenkins job?**
5. **Answer:**
6. **Setting up a new job in Jenkins is elaborated below with snapshots:**
7. **Step 1:** Go to the Jenkins Dashboard and Login with your registered login credentials.

← → ↻ localhost:8084/jenkins/login?from=%2Fjenkins%2F

 **Jenkins**

Jenkins ▶

User:

Password:

☐ Remember me on this computer

log in







9. **Step 2:** Click on the **New Item** that is shown in the left panel of the page.

← → ↻ localhost:8080

Jenkins

search

Jenkins ▶

-  **New Item**
-  People
-  Build History
-  Manage Jenkins
-  My Views
-  Credentials

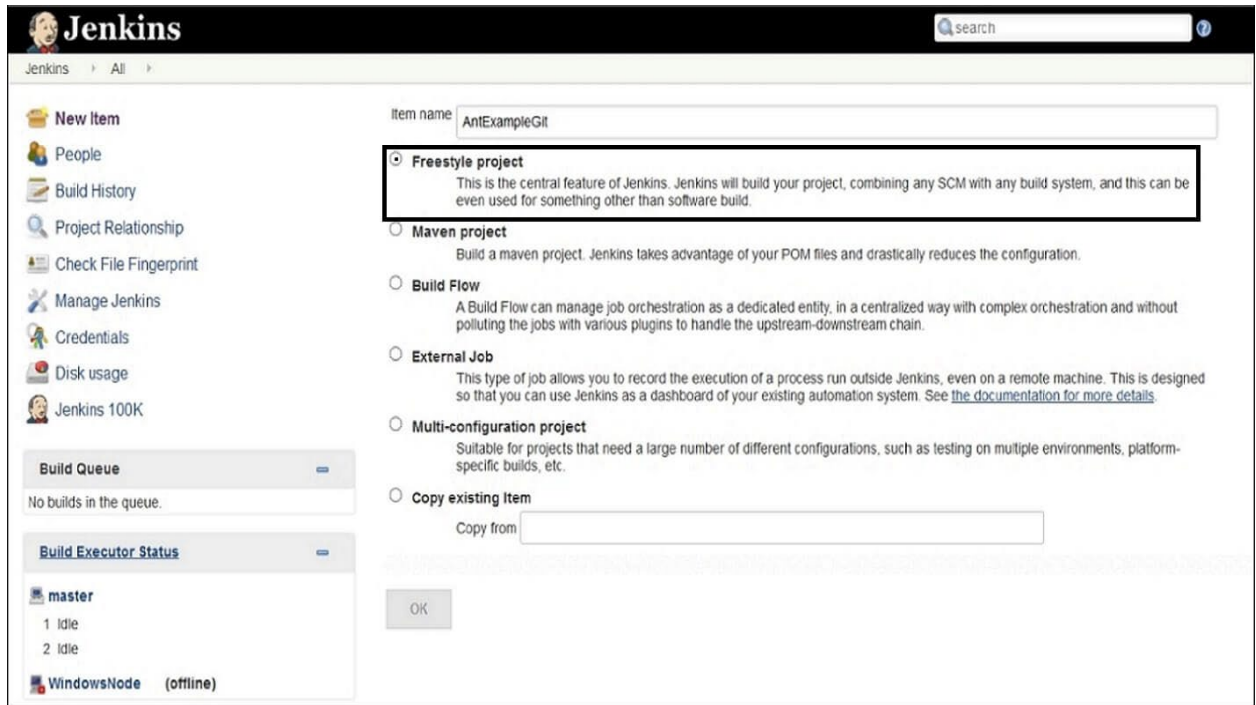
Build Queue

No builds in the queue.

Build Executor Status

1	Idle
2	Idle

11. **Step 3:** Click on the **Freestyle Project** from the given list on the upcoming page and specify the item name in the text box.



The image shows the Jenkins 'New Item' configuration page. The 'Item name' field is set to 'AntExampleGit'. The 'Freestyle project' option is selected and highlighted with a red box. Other options include 'Maven project', 'Build Flow', 'External Job', 'Multi-configuration project', and 'Copy existing Item'. The left sidebar shows navigation links like 'New Item', 'People', 'Build History', etc. The bottom left shows the 'Build Queue' and 'Build Executor Status' sections.

Jenkins

Item name:

☒ **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

☐ **Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

☐ **Build Flow**
A Build Flow can manage job orchestration as a dedicated entity, in a centralized way with complex orchestration and without polluting the jobs with various plugins to handle the upstream-downstream chain.

☐ **External Job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

☐ **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

☐ **Copy existing Item**
Copy from:

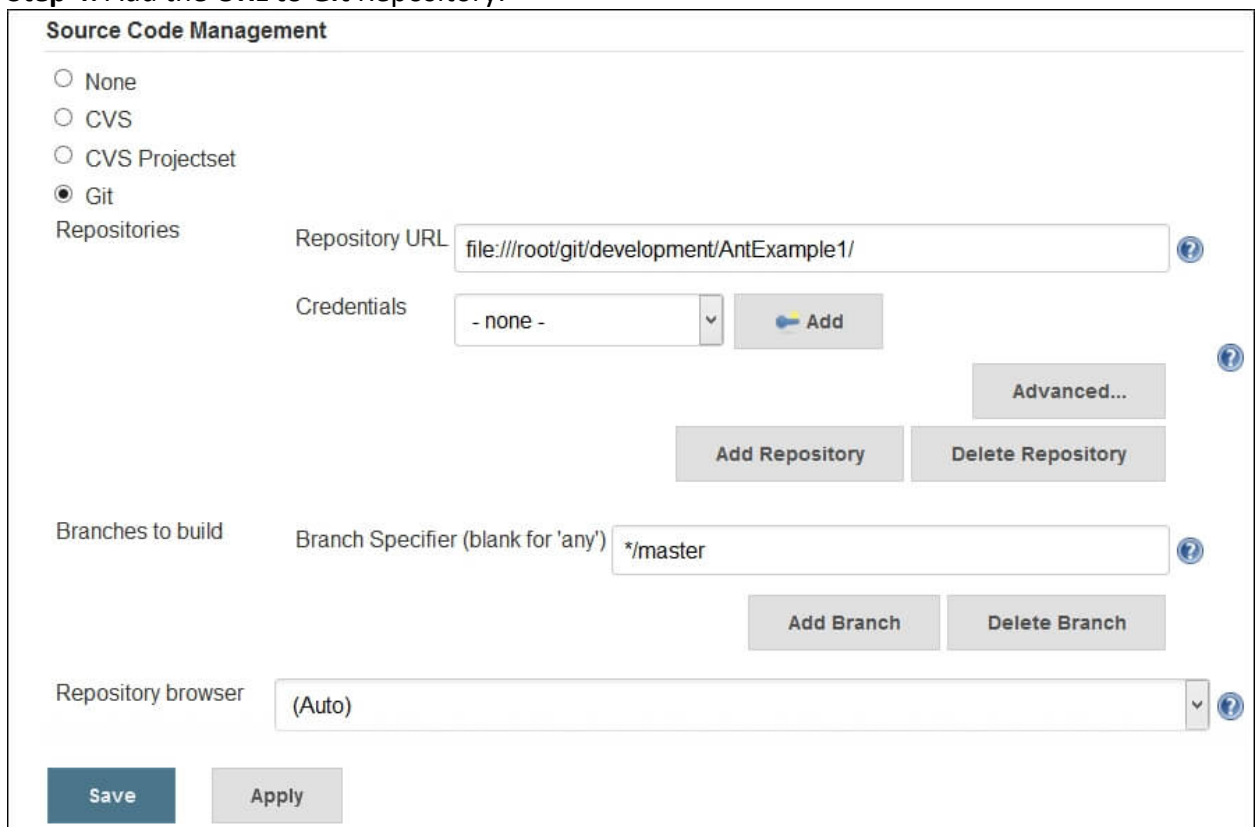
Build Queue
No builds in the queue.

Build Executor Status

master
1 Idle
2 Idle

WindowsNode (offline)

13. **Step 4: Add the URL to Git Repository.**



The image shows the 'Source Code Management' configuration page in Jenkins. The 'Git' option is selected under 'Repositories'. The 'Repository URL' is set to 'file:///root/git/development/AntExample1/'. The 'Credentials' dropdown is set to '- none -'. The 'Branches to build' section has a 'Branch Specifier (blank for 'any')' set to '*/master'. The 'Repository browser' is set to '(Auto)'. There are buttons for 'Add Repository', 'Delete Repository', 'Add Branch', 'Delete Branch', 'Advanced...', 'Save', and 'Apply'.

Source Code Management

☐ None
☐ CVS
☐ CVS Projectset
☒ **Git**

Repositories

Repository URL:

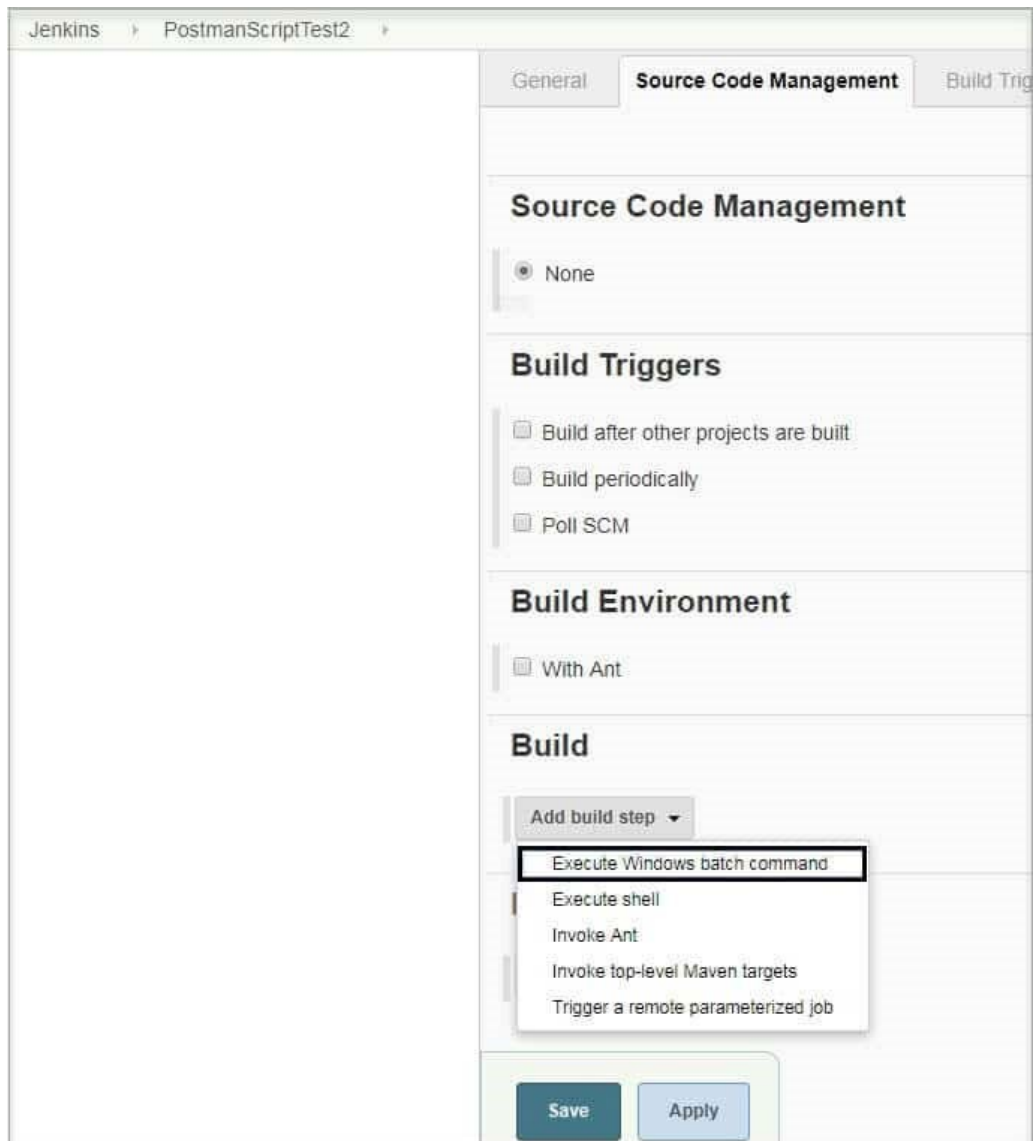
Credentials:

Branches to build

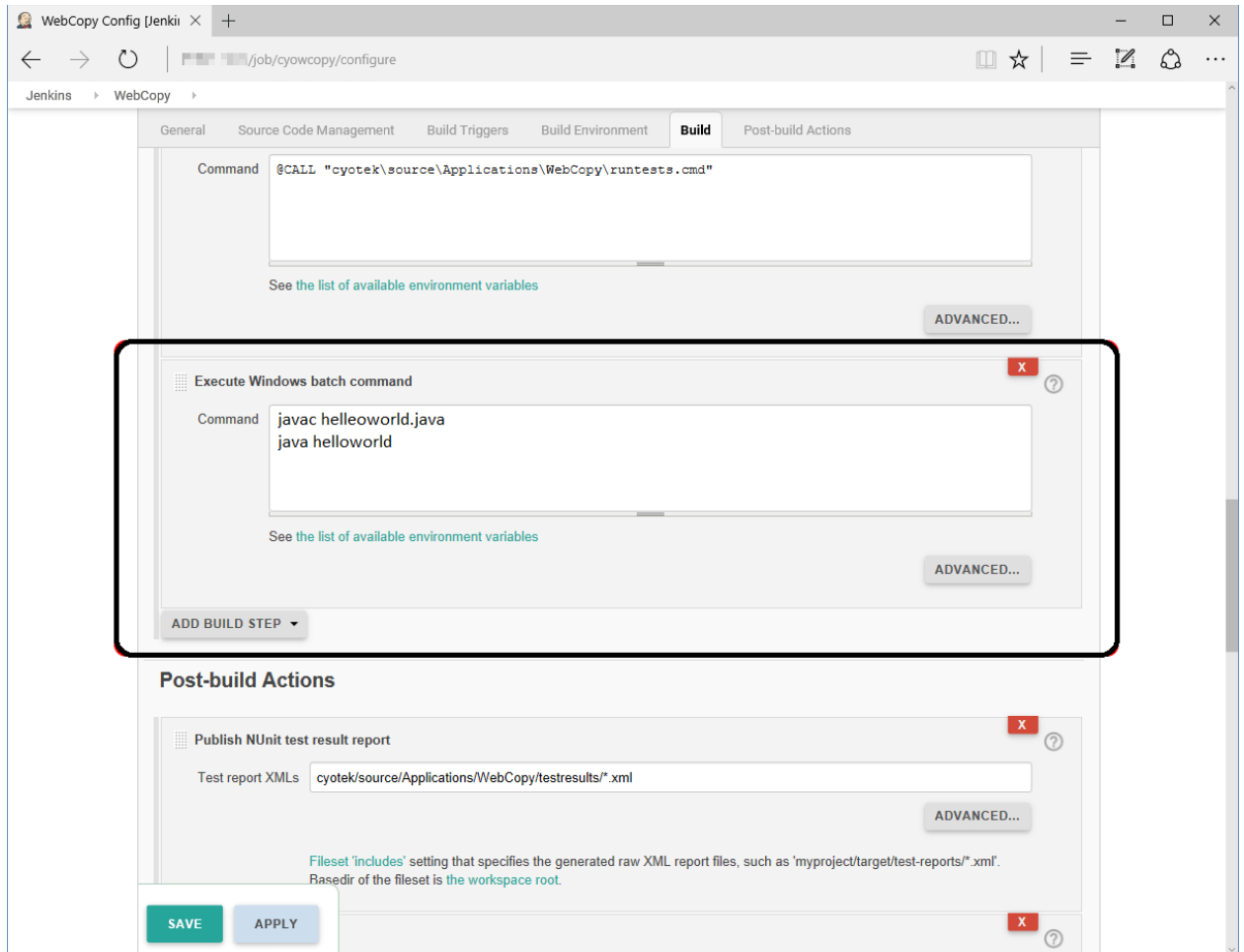
Branch Specifier (blank for 'any'):

Repository browser:

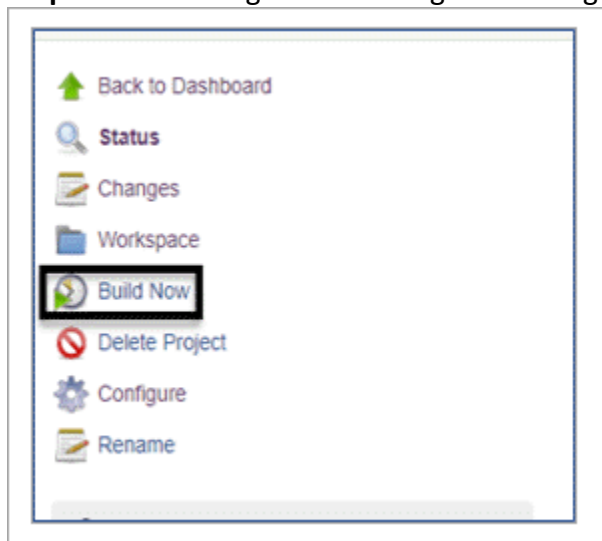
15. **Step 5: Go to the Build section and click on the Add build step => Execute Windows batch command.**



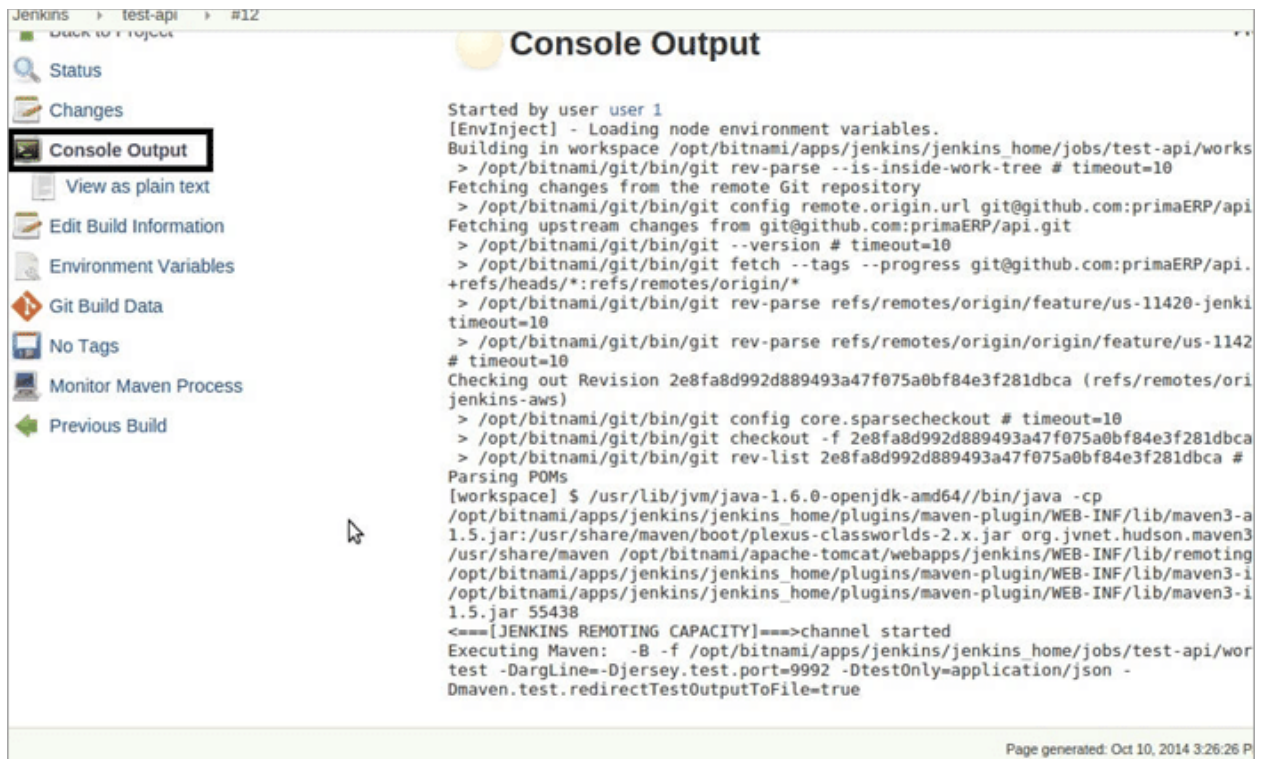
17. **Step 6:** Enter the command in the **command window** as shown below.



19. **Step 7:** After saving all the settings and changes click on **Build Now**.



21. **Step 8:** To see the status of the build click on **Console Output**.

The screenshot shows the Jenkins web interface for a build named 'test-api' (build #12). The 'Console Output' tab is selected in the left sidebar. The main area displays the build's log, which includes commands for cloning a Git repository, fetching updates, and running Maven. The log shows the build is using a workspace at /opt/bitnami/apps/jenkins/jenkins_home/jobs/test-api/works and is running on a system with Java 1.6.0 and Maven 3.0. The build is currently in a 'channel started' state.

```
Started by user user 1
[EnvInject] - Loading node environment variables.
Building in workspace /opt/bitnami/apps/jenkins/jenkins_home/jobs/test-api/works
> /opt/bitnami/git/bin/git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> /opt/bitnami/git/bin/git config remote.origin.url git@github.com:primaERP/api.git
Fetching upstream changes from git@github.com:primaERP/api.git
> /opt/bitnami/git/bin/git --version # timeout=10
> /opt/bitnami/git/bin/git fetch --tags --progress git@github.com:primaERP/api.
+refs/heads/*:refs/remotes/origin/*
> /opt/bitnami/git/bin/git rev-parse refs/remotes/origin/feature/us-11420-jenki
timeout=10
> /opt/bitnami/git/bin/git rev-parse refs/remotes/origin/origin/feature/us-1142
# timeout=10
Checking out Revision 2e8fa8d992d889493a47f075a0bf84e3f281dbca (refs/remotes/ori
jenkins-aws)
> /opt/bitnami/git/bin/git config core.sparsecheckout # timeout=10
> /opt/bitnami/git/bin/git checkout -f 2e8fa8d992d889493a47f075a0bf84e3f281dbca
> /opt/bitnami/git/bin/git rev-list 2e8fa8d992d889493a47f075a0bf84e3f281dbca #
Parsing POMs
[workspace] $ /usr/lib/jvm/java-1.6.0-openjdk-amd64/bin/java -cp
/opt/bitnami/apps/jenkins/jenkins_home/plugins/maven-plugin/WEB-INF/lib/maven3-a
1.5.jar:/usr/share/maven/boot/plexus-classworlds-2.x.jar org.jvnet.hudson.maven3
/usr/share/maven /opt/bitnami/apache-tomcat/webapps/jenkins/WEB-INF/lib/remoting
/opt/bitnami/apps/jenkins/jenkins_home/plugins/maven-plugin/WEB-INF/lib/maven3-i
/opt/bitnami/apps/jenkins/jenkins_home/plugins/maven-plugin/WEB-INF/lib/maven3-i
1.5.jar 55438
<==[JENKINS REMOTING CAPACITY]==>channel started
Executing Maven: -B -f /opt/bitnami/apps/jenkins/jenkins_home/jobs/test-api/wor
test -DargLine=-Djersey.test.port=9992 -DtestOnly=application/json -
Dmaven.test.redirectTestOutputToFile=true
```

What are the two components (pre-requisites) that Jenkins is mainly integrated with?

Answer: Jenkins integrates with:

1. Build tools/ Build working script like Maven script.
2. Version control system/Accessible source code repository like Git repository.

How can You Clone a Git Repository via Jenkins?

Answer: To create a clone repository via Jenkins you need to use your login credentials in the Jenkins System.

To achieve the same you need to enter the Jenkins job directory and execute the **git config** command.

What is the solution if you find a broken build for your project?

Answer: To resolve the broken build follow the below-mentioned steps:

(i) Open console output for the build and check if any file change has missed.

OR

(ii) Clean and update your local workspace to replicate the problem on the local system and try to resolve it (In case you couldn't find out the issue in the console output).

What are the various ways in which the build can be scheduled in Jenkins?

Answer: The build can be triggered in the following ways:

1. After the completion of other builds.
2. By source code management (modifications) commit.
3. At a specific time.
4. By requesting manual builds.

Why is Jenkins called a Continuous Delivery Tool?

Answer: We have seen the Continuous Delivery workflow in the previous question, now let's see the step by step process of why Jenkins is being called as a Continuous Delivery Tool:

1. Developers work on their local environment for making changes in the source code and push it into the code repository.
2. When a change is detected, Jenkins performs several tests and code standards to check whether the changes are good to deploy or not.
3. Upon a successful build, it is being viewed by the developers.
4. Then the change is deployed **manually** on a staging environment where the client can have a look at it.
5. When all the changes get approved by the developers, testers, and clients, the final outcome is saved **manually** on the production server to be used by the end users of the product.

In this way, Jenkins follows a **Continuous Delivery approach** and is called the **Continuous Delivery Tool**.

Tek School of America

Selenium Interview Questions

What is Automation Testing?

Automation testing is the process of testing a software or application using an automation testing tool to find the defects. In this process, executing the test scripts and generating the results are performed automatically by automation tools. It is required when we have huge amount of regression test cases. Some most popular tools to do automation testing are HP QTP/UFT, Selenium WebDriver, etc.,

1. What are the benefits of Automation Testing?

This is one of the common interview questions in any Automation testing job.

1. Saves time and money. Automation testing is faster in execution.
2. Reusability of code. Create one time and execute multiple times with less or no maintenance.
3. Easy reporting. It generates automatic reports after test execution.
4. Easy for compatibility testing. It enables parallel execution in the combination of different OS and browser environments.
5. Low-cost maintenance. It is cheaper compared to manual testing in a long run.
6. Automated testing is more reliable.
7. Automated testing is more powerful and versatile. Automation tools allow us to integrate with Cross Browser Testing Tools, Jenkins, Github etc.,
8. It is mostly used for regression testing. Supports execution of repeated test cases.
9. Minimal manual intervention. Test scripts can be run unattended.
10. Maximum coverage. It helps to increase the test coverage.

2. What are the challenges and limitations of Selenium WebDriver?

As we all know Selenium WebDriver is a tool which automates the browser to mimic real user actions on the web. Selenium is a free open source testing tool. Some of the challenges with selenium webdriver are as follows

1. We cannot test windows application
2. We cannot test mobile apps – Appium
3. Limited reporting-
4. Handling dynamic Elements -
5. Handling page load -
6. Handling pop up windows – Robot class or autoIT

7. Handling captcha-

3. What type of tests have you automated?

Our main focus is to automate test cases to do *Regression testing, Smoke testing, and Sanity testing*. Sometimes based on the project and the test time estimation, we do focus on End to End testing.

4. How many test cases you have automated per day?

It depends on Test case scenario complexity and length. I did automate 1-2 test scenarios per day when the complexity is limited. Sometimes just 1 or fewer test scenarios in a day when the complexity is high.

5. What is a Framework?

A framework defines a set of rules or best practices which we can follow in a systematic way to achieve the desired results. There are different types of automation frameworks and the most common ones are:

- Data Driven Testing Framework
- Keyword Driven Testing Framework
- Hybrid Testing Framework
- BDD Framework – Cucumber

6. Have you created any Framework?

If you are a beginner: You can say “No, I didn’t get a chance to create framework from the scratch. I have used the framework which is already available. My contribution is mostly in creating test cases by using the existing framework.”

If you are a beginner but have good knowledge on creating framework: You can say “Yes, I have involved in developing framework along with other automation tester in my company.”

If you are an experienced tester: You can say “I have contributed in developing framework.” or You can say “Yes, I have created framework from the scratch. There was no automation process in my previous company. I designed the framework from the scratch.”

7. Why do you prefer Selenium Automation Tool?

1. Free and open source
2. Have large user base and helping communities
3. Cross-browser compatibility
4. Platform compatibility
5. Multiple programming languages support such as Java, Perl, Python, PHP, C# etc.,

8. What is Selenium?

Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers. It has gained a lot of popularity in terms of web-based automated testing and giving a great competition to the famous commercial tool HP QTP (Quick Test Professional) AKA HP UFT (Unified Functional Testing).

Selenium is a set of different software tools. Each tool has a different approach in supporting web based automation testing.

It has four components namely,

- Selenium IDE (Selenium Integrated Development Environment)- outdated
- Selenium RC (Selenium Remote Control)
- Selenium WebDriver
- Selenium Grid

What is Selenium IDE?

Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. It allows us to record and playback the scripts. Even though we can create scripts using Selenium IDE, we need to use Selenium RC or Selenium WebDriver to write more advanced and robust test cases.

8. What is Selenese?

Selenese is the language which is used to write test scripts in Selenium IDE.

9. Which is the only browser that supports Selenium IDE to be used?

Firefox

10. What is Selenium RC?

Selenium RC AKA Selenium Remote control / Selenium 1. Selenium Remote Control was the main Selenium project for a long time before the WebDriver merge brought up Selenium 2. Selenium 1 is still actively supported (in maintenance mode). It relies on JavaScript for automation. It supports Java, Javascript, Ruby, PHP, Python, Perl and C#. It supports almost every browser out there.

11. What is Selenium WebDriver?

Selenium WebDriver AKA Selenium 2 is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.

12. What is Selenium Grid?

Selenium Grid is a tool used together with Selenium RC to run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems.

In simple words, it is used to distribute your test execution on multiple platforms and environments concurrently.

13. When do you use Selenium Grid?

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution

14. What are the advantages of Selenium Grid?

It allows running test cases in parallel thereby saving test execution time.

It allows multi-browser testing

It allows us to execute test cases on multi-platform

15. What is a hub in Selenium Grid?

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

16. What is a node in Selenium Grid?

Node is the machine which is attached to the hub. There can be multiple nodes in Selenium Grid.

17. What are the types of WebDriver APIs available in Selenium?

- Firefox Driver
- Gecko Driver
- InternetExplorer Driver
- Chrome Driver
- HTMLUnit Driver
- Opera Driver
- Safari Driver
- Android Driver
- iPhone Driver
- EventFiringWebDriver

18. Which WebDriver implementation claims to be the fastest?

The fastest implementation of WebDriver is the HTMLUnitDriver or headless browsers . It is because the HTMLUnitDriver does not execute tests in the browser.

19. What are the Programming Languages supported by Selenium WebDriver?

- Java
- C#
- Python
- Ruby
- Perl
- PHP

20. What are the Operating Systems supported by Selenium WebDriver?

- Windows
- Linux
- Apple

21. What are the Open-source Frameworks supported by Selenium WebDriver?

- JUnit
- TestNG

22. What are the Locators available in Selenium?

In Selenium WebDriver, there are 8 different types of locators:

1. ID –
2. ClassName

3. Name
4. TagName
5. LinkText
6. PartialLinkText
7. XPath
8. CSS Selector

23. What is an XPath?

XPath is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements such as textbox, button, checkbox, Image etc., in a web page.

24. What is the difference between “/” and “//”

Single Slash “/” – Single slash is used to create XPath with absolute path i.e. the XPath would be created to start selection from the document node/start node.

Double Slash “//” – Double slash is used to create XPath with relative path i.e. the XPath would be created to start selection from anywhere within the document.

25. What is the difference between Absolute Path and Relative Path?

Absolute XPath starts from the root node and ends with desired descendant element's node. It starts with top HTML node and ends with input node. It starts with a single forward slash(/) as shown below

```
/html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input
```

Relative XPath starts from any node in between the HTML page to the current element's node(last node of the element). It starts with a double forward slash(//) as shown below.

```
//input[@id='email']
```

26. What is the difference between Assert and Verify in Selenium?

Assert: In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.

Verify: In simple words, there won't be any halt in the test execution even though the verify condition is true or false.

27. What are Soft Assert and Hard Assert in Selenium?

Soft Assert: Soft Assert collects errors during `@Test`. Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

Hard Assert: Hard Assert throws an `AssertionException` immediately when an assert statement fails and test suite continues with next `@Test`.

28. What are the verification points available in Selenium?

In Selenium IDE, we use Selenese Verify and Assert Commands as Verification points. In Selenium WebDriver, there are no built-in features for verification points. It totally depends on our coding style. Some of the Verification points are

- To check for page title
- To check for certain text
- To check for certain element (text box, button, drop down, etc.)

29. How to launch a browser using Selenium WebDriver?

WebDriver is an Interface. We create an Object of a required driver class such as `FirefoxDriver`, `ChromeDriver`, `InternetExplorerDriver` etc.,

To launch Firefox Driver:

```
WebDriver driver = new FirefoxDriver();
```

Note: If you use geckodriver with Selenium, you must upgrade to Selenium 3.3. Here we have to set the property as follows

```
System.setProperty("webdriver.gecko.driver", "D:\\Selenium  
Environment\\Drivers\\geckodriver.exe");
```

To launch Chrome Driver:

```
WebDriver driver = new ChromeDriver();
```

To launch Internet Explorer Driver:

```
WebDriver driver = new InternetExplorerDriver();
```

To launch Safari Driver:

```
WebDriver driver = new SafariDriver();
```

30. Is the FirefoxDriver a Class or an Interface?

FirefoxDriver is a Java class, and it implements the *WebDriver* interface.

31. What is the super interface of WebDriver?

SearchContext.

32. Explain the line of code *Webdriver driver = new FirefoxDriver();* ?

```
Webdriver driver = new FirefoxDriver();
```

‘*WebDriver*’ is an interface and we are creating an object of type *WebDriver* instantiating an object of *FirefoxDriver* class.

33. Why we do create a reference variable ‘driver’ of type WebDriver

```
WebDriver driver = new FirefoxDriver();
```

instead of creating

```
FirefoxDriver driver = new FirefoxDriver();
```

What is the purpose of doing this way?

If we create a reference variable driver of type WebDriver then we could use the same driver variable to work with any browser of our choice such as IEDriver, SafariDriver etc.,

34. What are the different exceptions you have faced in Selenium WebDriver?

Some of the exceptions I have faced in my current project are

1. ElementNotVisibleException
2. StaleElementReferenceException

Element Not visible Exception:

This exception will be thrown when you are trying to locate a particular element on webpage that is not currently visible even though it is present in the DOM. Also sometimes, if you are trying to locate an element with the xpath which associates with two or more element.

Stale Element Reference Exception:

A stale element reference exception is thrown in one of two cases, the first being more common than the second.

The two reasons for Stale element reference are

1. The element has been deleted entirely.
2. The element is no longer attached to the DOM.

We face this stale element reference exception when the element we are interacting is destroyed and then recreated again. When this happens the reference of the element in the DOM becomes stale. Hence we are not able to get the reference to the element.

Some other exceptions we usually face are as follows:

- WebDriverException
- IllegalStateException
- TimeoutException
- NoAlertPresentException
- NoSuchWindowException
- NoSuchElementException

35. How to Login into any site if it is showing an Authentication Pop-Up for Username and Password?

To do this we pass username and password with the URL

http://username:password@url

e.g. <http://myUserName:myPassword@softwaretestingmaterial.com>

36. What are the types of waits available in Selenium WebDriver?

In Selenium we could see three types of waits such as Implicit Waits, Explicit Waits and Fluent Waits.

- Implicit Waits – Click to view detailed post
- Explicit Waits – Click to view detailed post
- Fluent Waits – Click to view detailed post

37. What is Implicit Wait In Selenium WebDriver?

Implicit waits tell to the WebDriver to wait for a certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception. The default setting is 0 (zero). We need to set some wait time to make WebDriver to wait for the required time.

38. What is WebDriver Wait In Selenium WebDriver?

WebDriverWait is applied on a certain element with defined *expected condition* and *time*. This wait is only applied to the specified element. This wait can also throw an exception when an element is not found.

39. What is Fluent Wait In Selenium WebDriver?

FluentWait can define the maximum amount of time to wait for a specific condition and frequency with which to check the condition before throwing an *"ElementNotVisibleException"* exception.

40. How to input text in the text box using Selenium WebDriver?

By using sendKeys() method

```
WebDriver driver = new FirefoxDriver();
driver.get("https://www.gmail.com");
driver.findElement(By.xpath("xpath")).sendKeys("Software Testing Material
Website");
```

41. How to input text in the text box without calling the sendKeys()?

```
// To initialize js object
JavascriptExecutor JS = (JavascriptExecutor)webdriver;
// To enter username
JS.executeScript("document.getElementById('User').value='SoftwareTestingMaterial.com'");
;
// To enter password
JS.executeScript("document.getElementById('Pass').value='tester'");
```

42. How to clear the text in the text box using Selenium WebDriver?

By using clear() method

```
WebDriver driver = new FirefoxDriver();
driver.get("https://www.gmail.com");
driver.findElement(By.xpath("xpath_of_element1")).sendKeys("Software Testing Material
Website");
driver.findElement(By.xpath("xpath_of_element1")).clear();
```

43. How to get a text of a web element?

By using getText() method

```
String availableText = driver.findElement(By.xpath("//*[@id='gbw']/div/div/div[1]/div[1]/a")).getText();
```

44. How to get an attribute value using Selenium WebDriver?

By using `getAttribute(value)`;

It returns the value of the attribute passed as a parameter.

HTML:

```
<input name="nameSelenium" value="valueSelenium">SoftwareTestingMaterial</input>
```

Selenium Code:

```
String attributeValue = driver.findElement(By.name("nameSelenium")).getAttribute("value");
```

```
System.out.println("Available attribute value is :"+attributeValue);
```

Output: valueSelenium

45. How to click on a hyperlink using Selenium WebDriver?

We use `click()` method in Selenium to click on the hyperlink

```
driver.findElement(By.linkText("Software Testing Material Website")).click();
```

46. How to submit a form using Selenium WebDriver?

We use “submit” method on element to submit a form

```
driver.findElement(By.id("form_1")).submit();
```

Alternatively, you can use `click` method on the element which does form submission

47. How to press ENTER key on text box In Selenium WebDriver?

To press ENTER key using Selenium WebDriver, We need to use Selenium Enum Keys with its constant ENTER.

```
driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER);
```

48. How to pause a test execution for 5 seconds at a specific point?

By using `java.lang.Thread.sleep(long milliseconds)` method we could pause the execution for a specific time. To pause 5 seconds, we need to pass parameter as 5000 (5 seconds)

```
Thread.sleep(5000)
```

49. Is Selenium Server needed to run Selenium WebDriver Scripts?

When we are distributing our Selenium WebDriver scripts to execute using Selenium Grid, we need to use Selenium Server.

50. What happens if I run this command. `driver.get("www.softwaretestingmaterial.com")` ;

An exception is thrown. We need to pass HTTP protocol within `driver.get()` method.

```
driver.get("https://www.softwaretestingmaterial.com");
```

51. What is the alternative to `driver.get()` method to open an URL using Selenium WebDriver?

Alternative method to `driver.get("url")` method is `driver.navigate.to("url")`

52. What is the difference between `driver.get()` and `driver.navigate.to("url")`?

`driver.get()`: To open an URL and it will wait till the whole page gets loaded

`driver.navigate.to()`: To navigate to an URL and It will not wait till the whole page gets loaded

53. Can I navigate back and forth in a browser in Selenium WebDriver?

We use Navigate interface to do navigate back and forth in a browser. It has methods to move back, forward as well as to refresh a page.

`driver.navigate().forward();` – to navigate to the next web page with reference to the browser's history

`driver.navigate().back();` – takes back to the previous webpage with reference to the browser's history

`driver.navigate().refresh();` – to refresh the current web page thereby reloading all the web elements

`driver.navigate().to("url");` – to launch a new web browser window and navigate to the specified URL

54. What are the different types of navigation commands?

Refer above question (Can I navigate back and forth in a browser)

55. How to fetch the current page URL in Selenium?

To fetch the current page URL, we use *getCurrentURL()*

```
driver.getCurrentUrl();
```

56. How can we maximize browser window in Selenium?

To maximize browser window in selenium we use *maximize()* method. This method maximizes the current window if it is not already maximized

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

57. How to delete cookies in Selenium?

To delete cookies we use *deleteAllCookies()* method

```
driver.manage().deleteAllCookies();
```


58. What are the ways to refresh a browser using Selenium WebDriver?

There are multiple ways to refresh a page in selenium

- Using *driver.navigate().refresh()* command as mentioned in the question 45
- 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());*
- Using *sendKeys(Keys.F5)* on any textbox on the webpage

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

driver.getWindowHandle() – It returns a handle of the current page (a unique identifier)

driver.getWindowHandles() – It returns a set of handles of the all the pages available.

60. What is the difference between *driver.close()* and *driver.quit()* methods?

Purpose of these two methods (*driver.close* and *driver.quit*) is almost same. Both allow us to close a browser but still, there is a difference.

driver.close(): To close current WebDriver instance

driver.quit(): To close all the opened WebDriver instances

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

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

- *findElement()* returns a single *WebElement* (found first) based on the locator passed as parameter. Whereas *findElements()* returns a list of *WebElements*, all satisfying the locator value passed.
- Syntax of *findElement()*-
`WebElement textbox = driver.findElement(By.id("textBoxLocator"));`
Syntax of *findElements()*-
`List <WebElement> elements = element.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.

62. How to find whether an element is displayed on the web page?

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

`isDisplayed()`

```
boolean elePresent = driver.findElement(By.xpath("xpath")).isDisplayed();
```

`isSelected()`

```
boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected();
```

`isEnabled()`

```
boolean eleEnabled= driver.findElement(By.xpath("xpath")).isEnabled()
```

63. How to select a value in a dropdown?

By using *Select* class

```
WebElement mySelectElement = driver.findElement(By.name("dropdown"));
Select dropdown = new Select(mySelectElement);
dropdown.selectByVisibleText(Text);
dropdown.selectByIndex(Index);
dropdown.selectByValue(Value);
```

64. How to capture Screenshot in Selenium WebDriver?

Test cases may fail while executing the test scripts. While we are executing the test cases manually we just take a screenshot and place in a result repository. The same can be done by using Selenium WebDriver.

Some of the scenarios we may need to capture a screenshot using Selenium WebDriver are

- i. Application issues
- ii. Assertion Failure
- iii. Difficulty to find Webelements on the web page
- iv. Timeout to find Webelements on the web page

Selenium provides an interface called *TakesScreenshot* which has a method *getScreenShotAs* which can be used to take a screenshot of the application under test.

In Selenium 3, we may face few issues while capturing Screenshots. To overcome we use aShot utility. Click on below links to see posts related to the normal way of capturing a screenshot and capturing a screenshot using aShot utility.

Capture screenshot using Selenium WebDriver

Full Page Screenshot using aShot utility

Failed Test Cases Screenshot

65. How to mouse hover on a web element using WebDriver?

By using Actions class

```
WebElement ele = driver.findElement(By.xpath("xpath"));  
//Create object 'action' of an Actions class  
Actions action = new Actions(driver);  
//Mouseover on an element  
action.moveToElement(ele).perform();
```

66. How can we handle web based pop-up?

To handle alerts popups we need to do switch to the alert window and call Selenium WebDriver Alert API methods.

67. How can we handle windows based pop up?

Selenium doesn't support windows based applications. It is an automation testing tool which supports only web application testing. We could handle windows based popups in Selenium using some third party tools such as AutoIT, Robot class etc.

68. How to handle hidden elements in Selenium WebDriver?

We can handle hidden elements by using javaScript executor

```
(JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");
```

69. How to find more than one web element in the list?

```
// To store the list
List <WebElement> eleList = driver.findElements(By.xpath("xpath"));
// To fetch the size of the list
int listSize = eleList.size();
//for loop
for (int i=0; i<listSize; i++)
{
    // Clicking on each link
    links.get(i).click();
    // Navigating back to the previous page that stores the links
    driver.navigate().back();
}
```

70. How to read a JavaScript variable in Selenium WebDriver?

By using JavascriptExecutor

```
// To initialize the JS object.
JavascriptExecutor JS = (JavascriptExecutor) webdriver;
// To get the site title.
```

```
String title = (String)JS.executeScript("return document.title");
System.out.println("Title of the webpage : " + title);
```

71. What is JavaScriptExecutor and in which cases JavaScriptExecutor will help in Selenium automation?

In general, we click on an element using click() method in Selenium.

For example:

```
driver.findElement(By.id("Id Value")).click();
```

Sometimes web controls don't react well against selenium commands and we may face issues with the above statement (click()). To overcome such kind of situation, we use JavaScriptExecutor interface.

It provides a mechanism to execute Javascript through Selenium driver. It provides "executescript" & "executeAsyncScript" methods, to run JavaScript in the context of the currently selected frame or window.

There is no need to write a separate script to execute JavaScript within the browser using Selenium WebDriver script. Just we use predefined interface named 'Java Script Executor'. We need to import the JavascriptExecutor package in the script.

```
import org.openqa.selenium.JavascriptExecutor;
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript(Script,Arguments);
```

Script – The JavaScript to execute

Arguments – The arguments to the script(Optional). May be empty.

Returns – One of Boolean, Long, String, List, WebElement, or null.

Let's see some scenarios we could handle using this Interface:

1. To type Text in Selenium WebDriver without using sendKeys() method
2. To click a Button in Selenium WebDriver using JavaScript

3. To handle Checkbox
4. To generate Alert Pop window in selenium
5. To refresh browser window using Javascript
6. To get innertext of the entire webpage in Selenium
7. To get the Title of our webpage
8. To get the domain
9. To get the URL of a webpage
10. To perform Scroll on an application using Selenium
11. To click on a SubMenu which is only visible on mouse hover on Menu
12. To navigate to different page using Javascript

72. How do you read test data from excels?

Test data can efficiently be read from excel using JXL or POI API. POI API has many advantages than JXL.

[Click here](#) to see a practical example of using Apache POI.

73. Is it possible to automate the captcha using Selenium?

No, It's not possible to automate captcha and bar code reader.

74. How to handle Ajax calls in Selenium WebDriver?

Handling AJAX calls is one of the common issues when using Selenium WebDriver. We wouldn't know when the AJAX call would get completed and the page has been updated. In this post, we see how to handle AJAX calls using Selenium.

AJAX stands for Asynchronous JavaScript and XML. AJAX allows the web page to retrieve small amounts of data from the server without reloading the entire page. AJAX sends HTTP requests from the client to server and then process the server's response without reloading the entire page. To handle AJAX controls, wait commands may not work. It's just because the actual page is not going to refresh.

When you click on a submit button, the required information may appear on the web page without refreshing the browser. Sometimes it may load in a second and sometimes it may take longer. We have no control over loading time. The best approach to handle this kind of

situations in selenium is to use dynamic waits (i.e. WebDriverWait in combination with ExpectedCondition)

Some of the methods which are available are as follows:

titleIs() – The expected condition waits for a page with a specific title.

```
wait.until(ExpectedConditions.titleIs("Deal of the Day"));
```

elementToBeClickable() – The expected condition waits for an element to be clickable i.e. it should be present/displayed/visible on the screen as well as enabled.

```
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("xpath")));
```

3. alertIsPresent() – The expected condition waits for an alert box to appear.

```
wait.until(ExpectedConditions.alertIsPresent()) != null;
```

4. textToBePresentInElement() – The expected condition waits for an element having a certain string pattern.

```
wait.until(ExpectedConditions.textToBePresentInElement(By.id("title"), "text to be found"));
```

75. List some scenarios which we cannot automate using Selenium WebDriver?

1. Bitmap comparison is not possible using Selenium WebDriver
2. Automating Captcha is not possible using Selenium WebDriver
3. We can not read bar code using Selenium WebDriver

76. What is Object Repository in Selenium WebDriver?

Object Repository is used to store element locator values in a centralized location instead of hard coding them within the scripts. We do create a property file (*.properties*) to store all the element locators and these property files act as an object repository in Selenium WebDriver.

77. How you build Object Repository in your project?

In QTP, there is an Object Repository concept. When a user records a test, the objects and its properties are captured by default in an Object Repository. QTP uses this Object Repository to play back the scripts. Coming to Selenium, there is no default Object Repository concept. It doesn't mean that there is no Object Repository in Selenium. Even though there is no default one still we could create our own. In Selenium, we call objects as locators (such as ID, Name, Class Name, Tag Name, Link Text, Partial Link Text, XPath, and CSS). Object repository is a collection of objects. One of the ways to create Object Repository is to place all the locators in a separate file (i.e., properties file). But the best way is to use Page Object Model. In the Page Object Model Design Pattern, each web page is represented as a class. All the objects related to a particular page of a web application are stored in a class.

78. What is Page Object Model in Selenium?

Page Object Model is a Design Pattern which has become popular in Selenium Test Automation. It is widely used design pattern in Selenium for enhancing test maintenance and reducing code duplication. Page object model (POM) can be used in any kind of framework such as modular, data-driven, keyword driven, hybrid framework etc. A page object is an object-oriented class that serves as an interface to a page of your Application Under Test(AUT). The tests then use the methods of this page object class whenever they need to interact with the User Interface (UI) of that page. The benefit is that if the UI changes for the page, the tests themselves don't need to change, only the code within the page object needs to change. Subsequently, all changes to support that new UI is located in one place.

79. What is Page Factory?

We have seen that 'Page Object Model' is a way of representing an application in a test framework. For every 'page' in the application, we create a Page Object to reference the 'page' whereas a 'Page Factory' is one way of implementing the 'Page Object Model'.

80. What is the difference between Page Object Model (POM) and Page Factory?

Page Object is a class that represents a web page and hold the functionality and members. Page Factory is a way to initialize the web elements you want to interact with within the page object when you create an instance of it.

81. What are the advantages of Page Object Model Framework?

Code reusability – We could achieve code reusability by writing the code once and use it in different tests.

Code maintainability – There is a clean separation between test code and page specific code such as locators and layout which becomes very easy to maintain code. Code changes only on Page Object Classes when a UI change occurs. It enhances test maintenance and reduces code duplication.

Object Repository – Each page will be defined as a java class. All the fields in the page will be defined in an interface as members. The class will then implement the interface.

Readability – Improves readability due to clean separation between test code and page specific code

82. How can you use the Recovery Scenario in Selenium WebDriver?

By using “*Try Catch Block*” within Selenium WebDriver Java tests.

```
try {  
    driver.get("www.SoftwareTestingMaterial.com");  
}catch(Exception e){  
    System.out.println(e.getMessage());  
}
```

83. How to Upload a file in Selenium WebDriver?

There are two cases which are majorly used to upload a file in Selenium WebDriver such as using *SendKeys* Method and using *AutoIT* Script.

84. How to Download a file in Selenium WebDriver?

By using *AutoIT* script, we could download a file in Selenium WebDriver.

85. How to switch between frames in Selenium?

By using the following code, we could switch between frames.

```
driver.switchTo().frame();
```

once actions done inside the frame we need to switch to default content

```
driver.switchTo().defaultContent()
```

86. How to connect a Database in selenium?

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

We use JDBC Driver to connect the Database in Selenium (While using Java Programming Language).

87. How To Resize Browser Window Using Selenium WebDriver?

To resize the browser window to particular dimensions, we use 'Dimension' class to resize the browser window.

88. How To Scroll Web Page Down Or UP Using Selenium WebDriver?

JavaScript scrollBy() method scrolls the document by the specified number of pixels.

89. How To Perform Right Click Action (Context Click) In Selenium WebDriver?

We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

90. How To Perform Double Click Action In Selenium WebDriver?

We use Actions class to do Double click action in selenium.

91. How To Perform Drag And Drop Action in Selenium WebDriver?

In some applications, we may face a situation to automate drag and drop an item from one location to another location. We could not achieve these using basic elements. Selenium has provided an “Actions” class to handle this kind of scenarios. We overcome this kind of scenarios such as drag and drop using Actions Class.

To achieve this, we use Actions class in Selenium WebDriver.

92. How To Highlight Element Using Selenium WebDriver?

By using JavascriptExecutor interface, we could highlight the specified element

93. Have you used any crossbrowsertesting tool to run selenium scripts on cloud?

I have used BrowserStack to run selenium tests on multiple browsers & Multiple operating systems in parallel. Earlier we have made a video on how to use BrowserStack to run selenium scripts on the cloud. Find the link in the description below.

94. What is desired capabilities?

In Selenium we use desired capabilities to handle SSL certificates in chrome browser

We need to create an instance of DesiredCapabilities

```
DesiredCapabilities desiredCapability = DesiredCapabilities.chrome();
```

95. What is Continuous Integration?

Continuous Integration is abbreviated as CI. Continuous Integration is a development practice which aims to make sure the correctness of a software. After each commit, a suite of tests run automatically and test the software to ensure whether the software is running without any breaks. If any test fails, we will get immediate feedback say “build is broken”.

In simple words, continuous integration is a process of verifying the correctness of a software.

Some of the continuous integration tools are Jenkins, TeamCity, Bamboo, Travis, Circle Ci, Bitbucket.

We can schedule the test suite execution using these CI Tools.

96. How to achieve Database testing in Selenium?

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

Sometimes, we may face a situation to get the data from the Database or to modify (update/delete) the data from the Database. If we plan to automate anything outside the vicinity of a browser, then we need to use other tools to achieve our task. To achieve the Database connection and work on it, we need to use JDBC API Driver.

The Java Database Connectivity (JDBC) API provides universal data access from the Java programming language. Using the JDBC API, you can access virtually any data source, from relational databases to spreadsheets and flat files. It lets the user connect and interact with the Database and fetch the data based on the queries we use in the automation script. JDBC is a SQL level API that allows us to execute SQL statements. It creates a connectivity between Java Programming Language and the database.

Using JDBC Driver we could do the following

- i. Establish a Database connection
- ii. Send SQL Queries to the Database
- iii. Process the results

97. How to delete Browser Cookies with Selenium Web Driver?

```
driver.Manage().Cookies.DeleteAllCookies();
```

The logo features a large yellow square with a dark gray corner cutout at the top right. To its right is a solid dark gray square. Below the yellow square is a dark gray square with a white corner cutout at the bottom left. The text 'TEK School' is centered in the yellow square.

TEK School

Updated 3/27/2020

SQA 110-Agile & Manual Testing

Contents

What is Agile Testing?.....	3
What is Agile Manifesto?	3
What are the principles of Agile Software Development?	3
What are the main roles in Scrum?	3
Product Owner:	3
Scrum Master:	3
Scrum Development Team:.....	3
Scrum Team:.....	4
What approach do you follow when requirements change continuously?.....	4
How is Agile Testing different from other traditional Software Development Models?	4
In what way does agile development methodology differ from other development methodologies?	4
When do we use Agile Scrum Methodology?	4
What is a Sprint?	4
What are Product Backlog and Sprint Backlog?	5
Product Backlog	5
Sprint Backlog:.....	5
What is the difference between Burn-up and Burn-down chart?.....	5
Burn Down Charts.....	5
Burn-up charts	5
What are the types of burn-down charts?.....	5
What is Product Burndown Chart?	5
What is Sprint Burndown Chart?	5
What is Release Burndown Chart?	5
What is Defect Burndown Chart?	5
What is a Daily Stand-up Meeting?	6
What is a Sprint Planning Meeting?	6
What is a Sprint Review Meeting?.....	6
What is a Sprint Retrospective Meeting?	6
What is a Task Board?.....	6
What is DevOps?	7
What is the differenceS between Agile and Waterfall model?	7
How long were your sprints?	7

What are the disadvantages of the agile model?.....	7
What is an impediment in Scrum?	8
What are the examples of impediments?.....	8
What kind of impediments should a scrum master remove?	8
What is Velocity?	8
What is traceability matrix?	8
What is Verification in software testing?.....	8
What is Validation in software testing?.....	9
What is White Box Testing?	9
What is Black Box Testing?.....	9
What is Grey Box Testing?	9
What is Positive and Negative Testing?.....	9
What is Test Suite?	9
What is Test Scenario?.....	9
What is Test Case?.....	9
What is Test Environment?	9
What is Test Data?.....	9
What is Test Harness?.....	10
What is Unit Testing?.....	10
What is Integration Testing?	10
What is Functional Testing?	10
What is Non-Functional Testing?	10
Homework.....	11

WHAT IS AGILE TESTING?

Agile testing is a software testing practice that follows the principles of agile software development. It is an iterative software development methodology where requirements keep changing as per the customer needs. Testing is done in parallel to the development of an iterative model. Test team receives frequent code changes from the development team for testing an application.

WHAT IS AGILE MANIFESTO?

Agile manifesto defines 4 key points:

- i. Individuals and interactions over process and tools
- ii. Working software over comprehensive documentation
- iii. Customer collaboration over contract negotiation
- iv. Responding to change over following a plan

WHAT ARE THE PRINCIPLES OF AGILE SOFTWARE DEVELOPMENT?

1. Highest priority is to satisfy the customer through early and continuous delivery of business valuable software
2. Welcome changing requirements, even late in development
3. Deliver working software frequently
4. Business people and developers must work together with transparency on daily basis throughout the project
5. Build projects around motivated individuals
6. The best form of communication is to do face-to-face conversation
7. Working software is the primary measure of progress
8. Able to maintain a constant pace
9. Continuous attention to technical excellence
10. Simplicity – the art of maximizing the amount of work not done – is essential
11. Self-organizing teams
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

WHAT ARE THE MAIN ROLES IN SCRUM?

Scrum consists of three main roles:

PRODUCT OWNER: Product Owner usually represents the Client and acts as a point of contact from the Client side. The one who prioritizes the list of Product Backlogs which Scrum Team should finish and release.

SCRUM MASTER: Scrum Master acts as a facilitator to the Scrum Development Team. Clarifies the queries and organizes the team from distractions and teach the team how to use scrum and also concentrates on Return on Investment (ROI). Responsible for managing the sprint.

SCRUM DEVELOPMENT TEAM: Developer's, QA's. Who develops the product. Scrum development team decides the effort estimation to complete a Product Backlog Item.

SCRUM TEAM: A cross-functional, self-organizing group of dedicated people (Group of Product Owner, Business Analyst, Developer's and QA's). Recommended size of a scrum team is 7 plus or minus 2 (i.e, between 5 to 9 members in a team).

WHAT APPROACH DO YOU FOLLOW WHEN REQUIREMENTS CHANGE CONTINUOUSLY?

In Agile methodology, change in requirement is possible. It's not like other traditional methodologies where the requirements are locked down at the requirement phase. Every team member should be ready to handle the changes in the project.

The team should work closely with the Product Owner to understand the scope of requirement change and to negotiate to keep the requirement changes to a minimum or to adopt those changes in the next sprint. Based on the requirement changes Test Team could update the Test Plan and Test Cases to achieve the deadlines. The team should understand the risk in the requirement change and prepare a contingency plan. It is a best practice not to go for the automation process until requirements are finalized.

HOW IS AGILE TESTING DIFFERENT FROM OTHER TRADITIONAL SOFTWARE DEVELOPMENT MODELS?

It is one of the common Agile Testing Interview Questions.

In Agile Methodology, testing is not a phase like other traditional models. It is an activity parallel to development in the Agile. The time slot for the testing is less in the Agile compared to the traditional models. The testing team works on small features in Agile whereas the test team works on a complete application after development in the traditional models.

IN WHAT WAY DOES AGILE DEVELOPMENT METHODOLOGY DIFFER FROM OTHER DEVELOPMENT METHODOLOGIES?

In Agile methodology, the code is broken down into small branches and only one branch is developed and tested at a time. At one particular time, only one particular branch is developed and tested. Agile teams follow several processes in the agile methodology like continuous communication with the team, frequent changes to get the optimal results etc. This makes the agile process more flexible and focused. This is not the case with other development methodologies.

WHEN DO WE USE AGILE SCRUM METHODOLOGY?

- i. When the client is not so clear on requirements
- ii. When the client expects quick releases
- iii. When the client doesn't give all the requirements at a time

WHAT IS A SPRINT?

In Scrum, the project is divided into Sprints. Each Sprint has a specified timeline (2 weeks to 1 month). This timeline will be agreed by a Scrum Team during the Sprint Planning Meeting. Here, User Stories are split into different modules. The end result of every Sprint should be a potentially shippable product.

WHAT ARE PRODUCT BACKLOG AND SPRINT BACKLOG?

PRODUCT BACKLOG: Product Backlog is a repository where the list of Product Backlog Items stored and maintained by the Product Owner. The list of Product Backlog Items are prioritized by the Product Owner as high and low and also could re-prioritize the product backlog constantly.

SPRINT BACKLOG: Group of user stories which scrum development team agreed to do during the current sprint (Committed Product Backlog items). It is a subset of the product backlog.

WHAT IS THE DIFFERENCE BETWEEN BURN-UP AND BURN-DOWN CHART?

BURN DOWN CHARTS provide proof that the project is on track or not. Both the burn-up and burn-down charts are graphs used to track the progress of a project.

BURN-UP CHARTS represent how much work has been completed in a project whereas Burn-down chart represents the remaining work left in a project.

WHAT ARE THE TYPES OF BURN-DOWN CHARTS?

There are four popularly used burn down charts in Agile.

- i. Product burndown chart
- ii. Sprint burndown chart
- iii. Release burndown chart
- iv. Defect burndown chart

WHAT IS PRODUCT BURNDOWN CHART?

A graph which shows how many Product Backlog Items (User Stories) implemented/not implemented.

WHAT IS SPRINT BURNDOWN CHART?

A graph which shows how many Sprints implemented/not implemented by the Scrum Team.

WHAT IS RELEASE BURNDOWN CHART?

A graph which shows List of releases still pending, which Scrum Team have planned.

WHAT IS DEFECT BURNDOWN CHART?

A graph which shows how many defects identified and fixed.

WHAT IS A DAILY STAND-UP MEETING?

Daily Stand-up Meeting is a daily routine meeting. It brings everyone up to date on the information and helps the team to stay organized.

Each team member reports to the peers the following:

1. What did you complete yesterday?
2. What do you do today?
3. Any impediments in your way?

In general, it's not a recorded meeting. Reporting will be between peers not to Scrum Master or Product Owner. It is normally timeboxed to a maximum of 15 minutes. It is aka 15 Minute Stand-up Meeting.

Here is a screenshot from Slack Application on daily standup meeting.

WHAT IS A SPRINT PLANNING MEETING?

The first step of Scrum is the Sprint Planning Meeting where the entire Scrum Team attends. Here the Product Owner selects the Product Backlog Items (User Stories) from the Product Backlog.

Most important User Stories at the top of the list and least important User Stories at the bottom. Scrum Development Team decides and provides effort estimation.

WHAT IS A SPRINT REVIEW MEETING?

In the Sprint Review Meeting, Scrum Development Team presents a demonstration of a potentially shippable product. Product Owner declares which items are completed and not completed. Product Owner adds the additional items to the product backlog based on the stakeholder's feedback.

WHAT IS A SPRINT RETROSPECTIVE MEETING?

Scrum Team meets again after the Sprint Review Meeting and documents the lessons learned in the earlier sprint such as "What went well", "What could be improved". It helps the Scrum Team to avoid the mistakes in the next Sprints.

WHAT IS A TASK BOARD?

A task board is a dashboard which illustrates the progress that an agile team is making in achieving their sprint goals.

In general, the columns used in a task board are as follows

- i. User Story: Actual Business Requirement (Description)
- ii. To Do: All the tasks of current sprint
- iii. In Progress: Any task being worked on

- iv. To Verify: Tasks pending for verification
- v. Done: Tasks which are completed

WHAT IS DEVOPS?

The term DevOps was formed by combining “Development” and “Operations”. DevOps is an operational philosophy that promotes collaboration between development and operation teams. DevOps focuses on bringing development and operations team together in order to build, test, and release software faster and more reliably.

WHAT IS THE DIFFERENCES BETWEEN AGILE AND WATERFALL MODEL?

AGILE	WATERFALL
Testing is done in parallel with the development activity which means that as the development progresses so does the testing	Testing is generally done at the end of the development
Agile is an incremental approach	It is a sequential design process where design, development, testing and other phases happens one after another in a sequential way
Testing is performed concurrently with software development	"Testing" phase comes after the "Build" phase
Agile methodology is known for its flexibility	Waterfall is a structured software development methodology, and often times can be quite rigid
It believes in constant feedback and accepts changes to requirements	Customer feedback is usually not collected until the very end of the project, and changes are discouraged
Agile focuses on collaboration, small releases and customer feedback	Team coordination is very limited
Self-motivated and self-organizing teams drive the project	Project manager drives the project as a central controlling authority

HOW LONG WERE YOUR SPRINTS?

An ideal sprint length is anywhere between 1 week to 4 weeks. 2 week-long Sprints are most common for IT and software product development.

WHAT ARE THE DISADVANTAGES OF THE AGILE MODEL?

Some of the disadvantages of using the agile model are as follows:

- It is not easy to predict. When you encounter a large project, it becomes more problematic to estimate the amount of effort needed in the project
- In case the guidelines given by the customers is not understood properly, then the final outcome of the project will not meet the customer requirements. It leads to the customer dissatisfaction
- It is not possible to properly focus on the design and documentation of a project sometimes

- Basically, High-level decision making is in the hands of the higher authorities. The team members with little or no experience are not involved in decision-making, thus they don't get a chance to advance their knowledge

WHAT IS AN IMPEDIMENT IN SCRUM?

Impediments are the obstacles faced by the scrum team. Any obstacle that keeps the team from getting work done and that slows velocity is known as Impediment. Scrum Master is responsible for removing impediments.

In Agile Scrum interviews, you may be asked to give some examples of impediments.

WHAT ARE THE EXAMPLES OF IMPEDIMENTS?

Impediments come in many forms. Some of the impediments are mentioned below

- Resource unavailability (Sick team member)
- Lack of management support
- Business issues
- Lack of skill
- Technical, operational issues
- Even external issues such as weather

WHAT KIND OF IMPEDIMENTS SHOULD A SCRUM MASTER REMOVE?

Scrum Master's main responsibility is to identify, track and help remove impediments. Scrum Master shouldn't remove impediments initially even though Scrum Master can remove impediments on behalf of the Scrum team. The Scrum Master should not pamper nor overrule the Scrum Team. Scrum Master should motivate the Scrum team to become independent enough to face problems and take a decision and perform every task by themselves. The Scrum team should be able to make their own decisions. The Scrum master supports and guides the Scrum Team to operate as efficiently as possible. Sometimes, impediments are beyond the ability of the Scrum Team to remove. In such cases, the Scrum Master may get support from outside of the Scrum Team.

WHAT IS VELOCITY?

Velocity is a key metric that is calculated at the end of each sprint by addition of all effort estimates associated with user stories completed in a sprint. It predicts how much work an agile software development team can successfully complete within a sprint and how much time will it need to finish a project. Points from partially-completed or incomplete user stories should not be counted in calculating velocity.

WHAT IS TRACEABILITY MATRIX?

The relationship between test cases and requirements is shown with the help of a document. This document is known as traceability matrix.

WHAT IS VERIFICATION IN SOFTWARE TESTING?

Verification is the process, to ensure that whether we are building the product right

WHAT IS VALIDATION IN SOFTWARE TESTING?

Validation is the process, whether we are building the right product i.e., to validate the product which we have developed is right or not.

WHAT IS WHITE BOX TESTING?

It is based on applications internal code structure (you work with the code too). In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases.

WHAT IS BLACK BOX TESTING?

Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.

WHAT IS GREY BOX TESTING?

Grey box is the combination of both White Box and Black Box Testing.

WHAT IS POSITIVE AND NEGATIVE TESTING?

POSITIVE TESTING: It is to determine what system supposed to do. It helps to check whether the application is justifying the requirements or not.

NEGATIVE TESTING: It is to determine what system not supposed to do. It helps to find the defects from the software.

WHAT IS TEST SUITE?

Test Suite is a collection of test cases.

WHAT IS TEST SCENARIO?

Test Scenario gives the idea of what we have to test.

WHAT IS TEST CASE?

Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions and actual results.

WHAT IS TEST ENVIRONMENT?

Test Environment is the combination of hardware and software on which Test Team performs testing.

WHAT IS TEST DATA?

Test data is the data that is used by the testers to run the test cases.

WHAT IS TEST HARNESS?

A test harness is the collection of software and test data configured to test a program unit by running it under varying conditions which involves monitoring the output with expected output.

WHAT IS UNIT TESTING?

Unit Testing is also called as Module Testing or Component Testing. It is done to check whether the individual unit or module of the source code is working properly. It is done by the developers in the developer's environment.

WHAT IS INTEGRATION TESTING?

Integration Testing is the process of testing the interface between the two software units. Integration testing is done by three ways. Big Bang Approach, Top-Down Approach, Bottom-Up Approach

WHAT IS FUNCTIONAL TESTING?

To verify that each function of the software application behaves as specified in the requirement document.

WHAT IS NON-FUNCTIONAL TESTING?

Non-functional testing refers to various aspects of the software such as performance, load, stress, scalability, security, compatibility etc., Main focus is to improve the user experience on how fast the system responds to a request.

HOMework

In short responses reply to the following questions:

Question	Answers
<i>Scrum</i>	Scrum is an agile process for managing SDLC
<i>Scrum Team</i>	A Scrum Team is usually 8 or nine members working together to deliver the required products
<i>Product Owner (PO)</i>	is the key stakeholder in the project and is responsible for maintaining the product backlog.
<i>Scrum Master (SM)</i>	is the facilitator for an agile development team.
<i>Tester</i>	software tester is responsible to ensure the quality of application
<i>Developer</i>	is responsible for writing code to create the software
<i>Business Analyst</i>	
<i>Project Manager</i>	
<i>Task Board</i>	
<i>Product Backlog - (PB)</i>	
<i>Sprint Backlog – (SB)</i>	
<i>Burndown Chart – (BC)</i>	
<i>Release Backlog – (RB)</i>	
<i>Sprint Planning</i>	
<i>Daily Scrum/Meeting</i>	
<i>Three Questions</i>	
<i>Sprint Review</i>	
<i>Sprint Retrospective</i>	
<i>Sprint Demo</i>	
<i>Time Box</i>	
<i>Epic</i>	
<i>User Stories</i>	
<i>Story Points</i>	
<i>Team Capacity</i>	
<i>Team Velocity</i>	
<i>Scrum of Scrums</i>	
<i>Backlog Grooming</i>	
<i>Planning Poker</i>	
<i>Relative Estimation</i>	
<i>Iteration</i>	
<i>Definition of Done (DoD)</i>	
<i>Kanban Method</i>	
<i>Kanban Board</i>	
<i>API testing</i>	
<i>alpha testing</i>	
<i>Agile testing</i>	
<i>acceptance testing</i>	

<i>accessibility testing</i>	
<i>accuracy testing</i>	
<i>ad hoc testing</i>	
<i>analytical testing</i>	
<i>big-bang testing</i>	
<i>bottom-up testing</i>	
<i>Exploratory testing</i>	
<i>branch testing</i>	
<i>component integration testing</i>	
<i>component testing</i>	
<i>code coverage</i>	
<i>commercial off-the-shelf (COTS)</i>	
<i>Usability Testing</i>	
<i>Continuous deployment</i>	
<i>Continuous Integration</i>	
<i>Refactoring</i>	
<i>Extreme Programming (XP)</i>	
<i>Test-driven development</i>	
<i>Unit Test</i>	
<i>actual result</i>	
<i>Agile Manifesto</i>	
<i>black-box testing</i>	
<i>white-box testing</i>	
<i>gray-box testing</i>	

Tek School

SQL Interview Questions

1. What is the order of SQL SELECT?

Please give an example

SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY.

```
SELECT AVG(SALARY), J.JOB_ID, JOB_TITLE
FROM EMPLOYEES E JOIN JOBS J
ON E.JOB_ID=J.JOB_ID
WHERE JOB_TITLE LIKE '%Manager%'
GROUP BY J.JOB_ID, JOB_TITLE
HAVING AVG(SALARY)>11000
ORDER BY JOB_TITLE;
```

2. How to rename a column in the output of SQL query?

Please give an example

To rename columns in the result output we would use Aliases.

That would change the column in the Results but not in the Database itself.

There is 2 ways how it can be done:

1) We can use AS Keyword

```
SELECT FIRST_NAME, LAST_NAME, SALARY AS MONTHLY_SALARY,
SALARY*12 AS YEARLY_SALARY
FROM EMPLOYEES;
```

2) We can just leave space

```
SELECT FIRST_NAME, LAST_NAME, SALARY MONTHLY_SALARY, SALARY*12
YEARLY_SALARY
FROM EMPLOYEES;
```

3. How to add new Employee details in an Employee table?

To add new record to the Table we would use INSERT DML Command

```
INSERT INTO  
EMPLOYEES(EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,HIRE_DATE,JO  
B_ID, SALARY)  
VALUES (220,'JOHN', 'SNOW', 'JSNOW', '01-JUL-08', 'IT_PROG' ,5500);
```

4. What is a constraint in SQL?
Please give an example

Constraint, as in limitation or check, can be used to specify the limit on the data type of any table column. It can be specified while creating or altering the table. Common constraints – NOT NULL, DEFAULT, UNIQUE, PRIMARY KEY, FOREIGN KEY.

5. What is a Primary key?
Please give an example

PK is a column in the Table that uniquely identifies each record and the value is never duplicated in the same table
PK cannot contain NULL Values

6. What is a Foreign key?
Please give an example

Is existence of PK in another table
Is a key used to link two tables together
It can accept Null Values
We can have more than one Foreign Key in a table.

7. How to avoid duplicate records in a SELECT query?
Please give an example

We can use DISTINCT Keyword.

```
SELECT DISTINCT DEPARTMENT_ID  
FROM EMPLOYEES  
ORDER BY DEPARTMENT_ID;
```

8. What is difference between DELETE and TRUNCATE?

Please give an example

DELETE: delete selected rows from the table, WHERE clause can be used for conditional parameters. Deleted records can be rolled back or committed.

TRUNCATE: delete ALL rows from the table >> auto-commits, i.e. cannot be rolled back.

```
DELETE FROM INSURANCE  
WHERE POLICY_NUMBER=0002;
```

```
TRUNCATE TABLE JOB_HISTORY;
```

9. What is the difference between UNION and UNION ALL?

Please give an example

UNION merges the contents of two structurally-compatible tables into a single combined table.

UNION will omit duplicate records whereas and it will also sort the results where is UNION ALL will include duplicate records.

The performance of UNION ALL will typically be better than UNION, since UNION requires the server to do the additional work of removing any duplicates.

```
SELECT FIRST_NAME, LAST_NAME, EMAIL FROM EMPLOYEES  
UNION ALL  
SELECT FIRST_NAME, LAST_NAME, EMAIL FROM MY_TABLE;
```

```
SELECT FIRST_NAME FROM EMPLOYEES  
UNION  
SELECT LAST_NAME FROM EMPLOYEES;
```

10. What is the difference between BETWEEN and IN condition operators?

Please give an example

BETWEEN: used to display rows based on a range of values

IN: used to check for values contained in a specific set of values.

```
SELECT * FROM EMPLOYEES  
WHERE SALARY IN (6000, 10000);
```

```
SELECT * FROM EMPLOYEES  
WHERE SALARY BETWEEN 6000 AND 10000;
```

11. What is the difference between the WHERE and HAVING clauses?
Please give an example

The WHERE clause is used to filter records from a result. The filtering occurs before any groupings are made.

The HAVING clause is used to filter values from a group (i.e., to check conditions after aggregation into groups has been performed).

```
SELECT COUNT(*), DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID IN(10,50,80,100,110,120,150)  
GROUP BY DEPARTMENT_ID  
HAVING COUNT(*)>5;
```

12. What do you mean by DML?
Please give an example

Data Manipulation Language (DML) - deals with data manipulation (used to retrieve, store, modify, delete and update data in database).

DML Commands: SELECT, UPDATE, DELETE, INSERT (SUDI)

```
DELETE FROM INSURANCE  
WHERE POLICY_NUMBER=0002;
```

```
INSERT INTO  
EMPLOYEES(EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,HIRE_DATE,JO  
B_ID, SALARY)  
VALUES (220,'JOHN', 'SNOW', 'JSNOW', '01-JUL-08', 'IT_PROG',5500);
```

13. Which Operators available in SQL?

Please give an example

SQL Operator is a reserved word used primarily in an SQL statement's WHERE clause to perform operations, such as arithmetic operations and comparisons. These are used to specify conditions in an SQL statement.

There are three types of Operators.

1. Arithmetic Operators

```
SELECT FIRST_NAME, LAST_NAME, SALARY*12  
FROM EMPLOYEES;
```

2. Comparison Operators

```
SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90;
```

3. Logical Operators

```
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90 AND LAST_NAME='King';
```

```
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90 OR LAST_NAME='King';
```

```
SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID IN (90,100,110,10);
```

14. What is the difference between an inner and outer join?

Please give an example

An inner join returns rows when there is at least some matching data between two (or more) tables that are being compared.

An outer join returns rows from both tables that include the records that are unmatched from one or both the tables.

```
SELECT COUNT(EMPLOYEE_ID), E.DEPARTMENT_ID, DEPARTMENT_NAME  
FROM EMPLOYEES E INNER JOIN DEPARTMENTS D  
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID  
GROUP BY E.DEPARTMENT_ID, DEPARTMENT_NAME;
```

```
SELECT COUNT(EMPLOYEE_ID), E.DEPARTMENT_ID, DEPARTMENT_NAME  
FROM EMPLOYEES E FULL OUTER JOIN DEPARTMENTS D  
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID  
GROUP BY E.DEPARTMENT_ID, DEPARTMENT_NAME;
```

15. What is a Subquery ?

Please give an example

A Subquery is a SQL query within another query. It is a subset of a Select statement whose return values are used in filtering the conditions of the main query.

```
SELECT * FROM EMPLOYEES  
WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES);
```

```
SELECT * FROM EMPLOYEES  
WHERE DEPARTMENT_ID IN  
(SELECT DEPARTMENT_ID FROM DEPARTMENTS WHERE  
DEPARTMENT_NAME LIKE 'A%');
```


16. What is Database Testing?

Please give an example

It is AKA back-end testing or data testing.

Database testing involves in verifying the integrity of data in the front end with the data present in the back end. It involves in updating records in a database and verifying the same on the front end.

17. What are aggregate functions in SQL?

Please give an example

SQL aggregate functions return a single value, calculated from values in a column. Some of the aggregate functions in SQL are as follows

- AVG() – This function returns the average value
- COUNT() – This function returns the number of rows
- MAX() – This function returns the largest value
- MIN() – This function returns the smallest value
- ROUND() – This function rounds a numeric field to the number of decimals specified
- SUM() – This function returns the sum

```
SELECT ROUND(AVG(SALARY),2) FROM EMPLOYEES;
```

18. Explain SQL Data Types?

Please give an example

In SQL Server, each column in a database table has a name and a data type. We need to decide what type of data to store inside each and every column of a table while creating a SQL table.

There are many different Data Types, the common are:

Numbers – 1234

VarChar2 - String, letters, characters (Variable Length)

Char - String, letters, characters (Fixed Length)

Date - 02/13/201

Difference between Char() and varChar2()in Memory Allocation

19.How to Test for *NULL* Values?

Please give an example

A field with a NULL value is a field with no value. NULL value cannot be compared with other NULL values. Hence, It is not possible to test for NULL values with comparison operators, such as =, <, or <>. For this, we have to use the IS NULL and IS NOT NULL operators.

```
SELECT * FROM EMPLOYEES  
WHERE COMMISSION_PCT IS NULL;
```

```
SELECT * FROM EMPLOYEES  
WHERE COMMISSION_PCT IS NOT NULL;
```

20.What is a GROUP BY Clause? Have you use it?

Please give an example

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

```
SELECT COUNT(*), DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID IN(10,50,80,100,110,120,150)  
GROUP BY DEPARTMENT_ID  
HAVING COUNT(*)>5;
```

21.When and why do we use ORDER BY Clause

Please give an example

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

```
SELECT EMPLOYEE_ID, FIRST_NAME  
FROM EMPLOYEES  
WHERE ROWNUM<=5  
ORDER BY 1 DESC;
```

22. What is the difference between AND and OR operators
Please give an example

The AND and OR operators are used to filter records based on more than one condition:

The AND operator displays a record if all the conditions separated by AND is TRUE.

The OR operator displays a record if any of the conditions separated by OR is TRUE.

```
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90 AND LAST_NAME='King';
```

```
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90 OR LAST_NAME='King';
```

23. How can you get first 5 records from the Table? Last 5 records?
Please give an example

For that we can use ROWNUM Keyword:

```
SELECT EMPLOYEE_ID, FIRST_NAME  
FROM EMPLOYEES  
WHERE ROWNUM<=5  
ORDER BY EMPLOYEE_ID;
```

```
SELECT EMPLOYEE_ID, FIRST_NAME
```

```
FROM EMPLOYEES
WHERE ROWNUM<=5
ORDER BY 1 DESC;
```

24. Have you done functions in SQL?
Please give an example

Yes I have

```
SELECT * FROM EMPLOYEES
WHERE HIRE_DATE=(SELECT MIN(HIRE_DATE) FROM EMPLOYEES)
UNION
SELECT * FROM EMPLOYEES
WHERE HIRE_DATE=(SELECT MAX(HIRE_DATE) FROM EMPLOYEES);
```

```
SELECT UPPER(COUNTRY_NAME)
from COUNTRIES
WHERE LENGTH(COUNTRY_NAME)>6;
```

25. Explain how would you query two or more tables when validating data in SQL Database?
Please give an example

Using Subquery:

```
SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEES
WHERE DEPARTMENT_ID =( SELECT DEPARTMENT_ID FROM
DEPARTMENTS WHERE DEPARTMENT_NAME ='Shipping');
```

Using JOINS

```
SELECT EMPLOYEE_ID, FIRST_NAME, D.DEPARTMENT_ID,
DEPARTMENT_NAME
FROM EMPLOYEES E JOIN DEPARTMENTS D
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;
```

26. How to write a query to show the details of employees from Employees table whose last names start with K? End with k?
Please write an example

To achieve that we can use LIKE Operator

There are two wildcards used in conjunction with the LIKE operator:

% - The percent sign represents zero, one, or multiple characters

_ - The underscore represents a single character

```
SELECT * FROM EMPLOYEES  
WHERE LAST_NAME LIKE 'A%';
```

```
SELECT * FROM EMPLOYEES  
WHERE LAST_NAME LIKE '%a';
```

27. What is the difference between TRUNCATE and DROP?
Please write an example

TRUNCATE: delete ALL rows from the table >> auto-commits, i.e. cannot be rolled back. Database triggers do not fire on TRUNCATE

DROP: delete a table from the database and operation cannot be rolled back.

```
TRUNCATE TABLE INSURANCE;
```

```
DROP TABLE INSURANCE;
```

28. What types of joins have you used in the past?
Please write an example

In the Past I have used Inner joins, Left Joins, Right Joins and Full Outer Joins

```
SELECT EMPLOYEE_ID, E.DEPARTMENT_ID, D.DEPARTMENT_ID,  
DEPARTMENT_NAME  
FROM EMPLOYEES E RIGHT OUTER JOIN DEPARTMENTS D  
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;
```

```
SELECT EMPLOYEE_ID, E.DEPARTMENT_ID, D.DEPARTMENT_ID,  
DEPARTMENT_NAME  
FROM DEPARTMENTS D LEFT OUTER JOIN EMPLOYEES E  
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;
```

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY,  
D.DEPARTMENT_NAME, CITY, COUNTRY_NAME  
FROM EMPLOYEES E FULL JOIN DEPARTMENTS D  
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID  
FULL JOIN LOCATIONS L  
ON L.LOCATION_ID=D.LOCATION_ID  
FULL JOIN COUNTRIES C ON C.COUNTRY_ID=L.COUNTRY_ID;
```

```
SELECT AVG(SALARY), J.JOB_ID, JOB_TITLE  
FROM EMPLOYEES E JOIN JOBS J  
ON E.JOB_ID=J.JOB_ID  
WHERE JOB_TITLE LIKE '%Manager%'  
GROUP BY J.JOB_ID, JOB_TITLE  
HAVING AVG(SALARY)>11000  
ORDER BY JOB_TITLE;
```

29.What is the difference between DDL and DML Commands?
Please write an example

BASIS FOR COMPARISON	DDL	DML
Basic	DDL is used to Specify the database schema database structure.	DML is used to access, modify or retrieve the data from the database.
Full Form	Data Definition Language	Data Manipulation Language
Commands	CREATE, ALTER, DROP, TRUNCATE AND COMMENT and RENAME, etc.	SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, etc.

30.What is the difference between Commit and Rollback?
Please write an example

s both ROLLBACK and COMMIT Commands are TCC (Transaction Control Commands), they have differences:

COMMIT is used to save the changes permanently on the server

ROLLBACK is used to undo the changes and restore previous state.

31.Let's say you working on a Employees Database. How would you write a query to find out those employees whose first name starts with character 'N', 'A', 'K' ?
Please write an example

```
SELECT * FROM EMPLOYEES WHERE LAST_NAME LIKE 'A%'
UNION
SELECT * FROM EMPLOYEES WHERE LAST_NAME LIKE 'N%'
UNION
SELECT * FROM EMPLOYEES WHERE LAST_NAME LIKE 'K%';
```

Or

```
SELECT * FROM EMPLOYEES
WHERE LAST_NAME LIKE 'A%' OR LAST_NAME LIKE 'N%' OR LAST_NAME
LIKE 'K%';
```

32. What is the difference between right join and left join ?

Please write an example

Left join and Right join are both Outer Joins and they both return matching data from both Tables **PLUS**

- **LEFT** outer join or **LEFT** join Return **non matching** data from all rows from the **LEFT TABLE** written in **YOUR QUERY**, even if there is **NO MATCH** in the **right table**
- **RIGHT** outer join or **RIGHT** join Return **non matching** data from all rows from the **RIGHT TABLE** written in **YOUR QUERY**, even if there is **NO MATCH** in the **left table**

```
SELECT EMPLOYEE_ID, E.DEPARTMENT_ID, D.DEPARTMENT_ID,
DEPARTMENT_NAME
FROM EMPLOYEES E RIGHT OUTER JOIN DEPARTMENTS D
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;
```

```
SELECT EMPLOYEE_ID, E.DEPARTMENT_ID, D.DEPARTMENT_ID,
DEPARTMENT_NAME
FROM DEPARTMENTS D LEFT OUTER JOIN EMPLOYEES E
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;
```


Tek School of America

TESTNG Interview Questions

1. What is TestNG?

TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing to integration testing

2. What are the advantages of TestNG?

1. TestNG provides parallel execution of test methods
2. It allows to define dependency of one test method over other method
3. It allows to assign priority to test methods
4. It allows grouping of test methods into test groups
5. It has support for parameterizing test cases using @Parameters annotation
6. It allows data driven testing using @DataProvider annotation
7. It has different assertions that helps in checking the expected and actual results
8. Detailed (HTML) reports

3. What are the annotations available in TestNG?

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

4. Can you arrange the below *testng.xml* tags from parent to child?

```
<test>
<suite>
<class>
<methods>
<classes>
```

The correct order of the TestNG tags are as follows

```
<suite>
<test>
<classes>
<class>
<methods>
```

5. How to create and run *testng.xml* ?

In TestNG framework, we need to create ***testng.xml*** file to create and handle multiple test classes. We do configure our test run, set test dependency, include or exclude any test, method, class or package and set priority etc in the xml file.

6. What is the importance of *testng.xml* file?

In a Selenium TestNG project, we use *testng.xml* file to configure the complete test suite in a single file. Some of the features are as follows.

- testng.xml file allows to include or exclude the execution of test methods and test groups
- It allows to pass parameters to the test cases
- Allows to add group dependencies
- Allows to add priorities to the test cases
- Allows to configure parallel execution of test cases
- Allows to parameterize the test cases

7. How to pass parameter through *testng.xml* file to a test case?

We could define the parameters in the *testng.xml* file and then reference those parameters in the source files.

Create a java test class, say, *ParameterizedTest.java* and add a test method say *parameterizedTest()* to the test class. This method takes a string as input parameter. Add the annotation `@Parameters("browser")` to this method.

```
public class ParameterizedTest {  
    @Test  
    @Parameters("browser")  
    public void parameterizedTest(String browser){  
        if(browser.equals("firefox")){  
            System.out.println("Open Firefox Driver");  
        }else if(browser.equals("chrome")){  
            System.out.println("Open Chrome Driver");  
        }  
    }  
}
```

The parameter would be passed a value from *testng.xml*, which we will see in the next step.

We could set the parameter using the below syntax in the *testng.xml* file.

```
<parameter name="browser" value="firefox"/>
```

Here, name attribute represents the parameter name and value represents the value of that parameter.

8. What is TestNG Assert and list out common TestNG Assertions?

TestNG Asserts help us to verify the condition of the test in the middle of the test run. Based on the TestNG Assertions, we will consider a successful test only if it is completed the test run without throwing any exception.

Some of the common assertions supported by TestNG are

- assertEquals(String actual,String expected)
- assertEquals(String actual,String expected, String message)
- assertEquals(boolean actual,boolean expected)
- assertTrue(condition)
- assertTrue(condition, message)
- assertFalse(condition)
- assertFalse(condition, message)

9. What is Soft Assert in TestNG?

Soft Assert collects errors during `@Test`. Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

If there is any exception and you want to throw it then you need to use `assertAll()` method as a last statement in the `@Test` and test suite again continue with next `@Test` as it is.

10. What is Hard Assert in TestNG?

Hard Assert throws an *AssertionException* immediately when an assert statement fails and test suite continues with next `@Test`

11. What is exception test in TestNG?

TestNG gives an option for tracing the Exception handling of code. You can verify whether a code throws the expected exception or not. The expected exception to validate while running the test case is mentioned using the ***expectedExceptions*** attribute value along with @Test annotation.

12. How to set test case priority in TestNG?

We use *priority* attribute to the @Test annotations. In case priority is not set then the test scripts execute in alphabetical order.

```
package TestNG;
import org.testng.annotations.*;
public class PriorityTestCase{
    @Test(priority=0)
    public void testCase1() {
        system.out.println("Test Case 1");
    }
    @Test(priority=1)
    public void testCase2() {
        system.out.println("Test Case 2");
    }
}
```

Output:

Test Case 1

Test Case 2

13. What is Parameterized testing in TestNG?

Parameterized tests allow developers to run the same test over and over again using different values.

There are two ways to set these parameters:

- using *testng.xml*
- using *Data Providers*

14. How can we create data driven framework using TestNG?

By using *@DataProvider* annotation, we can create a Data Driven Framework.

```
@DataProvider(name="getData")
public Object[][] getData(){
    //Object [][] data = new Object [rowCount][colCount];
    Object [][] data = new Object [2][2];

    data [0][0] = "FirstUid";
    data [0][1] = "FirstPWD";

    data[1][0] = "SecondUid";
    data[1][1] = "SecondPWD";

    return data;

}
```

15. How to run a group of test cases using TestNG?

TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods belong to groups, but you can also specify groups that contain other groups. Then TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set. This gives you maximum flexibility in how you partition your tests and doesn't require you to recompile anything if you want to run two different sets of tests back to back.

Groups are specified in your `testng.xml` file and can be found either under the `<test>` or `<suite>` tag. Groups specified in the `<suite>` tag apply to all the `<test>` tags underneath.

```
@Test (groups = { "smokeTest", "functionalTest" })
public void loginTest(){
    System.out.println("Logged in successfully");
}
```

16. How to create Group of Groups in TestNG?

Groups can also include other groups. These groups are called *MetaGroups*. For example, you might want to define a group *all* that includes *smokeTest* and *functionalTest*. Let's modify our `testng.xml` file as follows:

```
<groups>
  <define name="all">
    <include name="smokeTest"/>
    <include name="functionalTest"/>
  </define>
  <run>
    <include name="all" />
  </run>
</groups>
```


17. How to run test cases in parallel using TestNG?

we can use "parallel" attribute in testng.xml to accomplish parallel test execution in TestNG

The parallel attribute of suite tag can accept four values:

tests – All the test cases inside <test> tag of testng.xml file will run parallel

classes – All the test cases inside a java class will run parallel

methods – All the methods with @Test annotation will execute parallel

instances – Test cases in same instance will execute parallel but two methods of two different instances will run in different thread.

```
<suite name="softwaretestingmaterial" parallel="methods">
```

18. How to exclude a particular test method from a test case execution?

By adding the exclude tag in the *testng.xml*

```
<classes>
  <class name="TestCaseName">
    <methods>
      <exclude name="TestMethodNameToExclude"/>
    </methods>
  </class>
</classes>
```

19. How to exclude a particular test group from a test case execution?

By adding the exclude tag in the *testng.xml*

```
<groups>
  <run>
    <exclude name="TestGroupNameToExclude"/>
  </run>
</groups>
```

20. How to disable a test case in TestNG ?

To disable the test case we use the parameter `enabled = false` to the `@Test` annotation.

```
@Test(enabled = false)
```

21. How to skip a @Test method from execution in TestNG?

By using *throw new SkipException()*

Once *SkipException()* thrown, remaining part of that test method will not be executed and control will go directly to next test method execution.

```
throw new SkipException("Skipping - This is not ready for testing ");
```

22. How to Ignore a test case in TestNG?

To ignore the test case we use the parameter `enabled = false` to the `@Test` annotation.

```
@Test(enabled = false)
```

23. How TestNG allows to state dependencies?

TestNG allows two ways to declare the dependencies.

Using attributes `dependsOnMethods` in `@Test` annotations –

Using attributes `dependsOnGroups` in `@Test` annotations –

24. What are the different ways to produce reports for TestNG results?

TestNG offers two ways to produce a report.

Listeners implement the interface `org.testng.ITestListener` and are notified in real time of when a test starts, passes, fails, etc...

Reporters implement the interface `org.testng.IReporter` and are notified when all the suites have been run by TestNG. The `IReporter` instance receives a list of objects that describe the entire test run.

25. What is the use of @Listener annotation in TestNG?

TestNG listeners are used to configure reports and logging. One of the most widely used listeners in testNG is `ITestListener` interface. It has methods like `onTestStart`, `onTestSuccess`, `onTestFailure`, `onTestSkipped` etc. We should implement this interface creating a listener class of our own. Next we should add the listeners annotation (`@Listeners`) in the Class which was created.

26. How to write regular expression In testng.xml file to search @Test methods containing “smoke” keyword.

Regular expression to find `@Test` methods containing keyword “smoke” is as mentioned below.

```
<methods>
  <include name=".*smoke.*"/>
</methods>
```

27. What is the time unit we specify in test suites and test cases?

We specify the time unit in test suites and test cases is in *milliseconds*.

28. List out various ways in which TestNG can be invoked?

TestNG can be invoked in the following ways

- Using Eclipse IDE
- Using ant build tool
- From the command line
- Using IntelliJ's IDEA

29. What is the use of @Test(invocationCount=x)?

The *invocationcount* attribute tells how many times TestNG should run a test method

```
@Test(invocationCount = 10)
public void testCase1(){
```

In this example, the method *testCase1* will be invoked ten times

30. What is the use of @Test(threadPoolSize=x)?

The *threadPoolSize* attribute tells to form a thread pool to run the test method through multiple threads.

Note: This attribute is ignored if invocationCount is not specified

```
@Test(threadPoolSize = 3, <code class="plain">invocationCount =
</code><code class="value">10</code>) public void testCase1(){
```

In this example, the method *testCase1* will be invoked from three different threads

31. What does the test timeout mean in TestNG?

The maximum number of milliseconds a test case should take.

```
@Test(threadPoolSize = 3, invocationCount = 10, timeout = 10000)
public void testCase1(){
```

In this example, the function `testCase1` will be invoked ten times from three different threads. Additionally, a time-out of ten seconds guarantees that none of the threads will block on this thread forever.

32. What is @DataProvider annotation?

@DataProvider: A test method that uses DataProvider will be executed the specific methods multiple number of times based on the data provided by the DataProvider. The test method will be executed using the same instance of the test class to which the test method belongs.