1. What is software testing?
- Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.
- Software testing is an activity to check whether the actual results match the expected result.
- Ensure that the software system is defect free.

2. What is Exploratory Testing?
- It is also directed from the requirement point of view, but it is also explored during testing.
- It can also determine the number of test cases during the testing.
- High-level investigation of system or project or application.

3. What is traceability matrix?
Traceability matrix is a Document showing the relationship between test requirements and test cases.

4. What is Boundary value testing?
- Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.
- these extreme ends like Start- End, Lower- Upper, Maximum-Minimum, Just Inside- Just Outside values are called boundary values and the testing is called "boundary testing".

5. What is Equivalence partitioning testing?
- Equivalence Partitioning or Equivalence Class Partitioning is type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc.
- In this technique, input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of small number of test cases.

6. What is Integration testing?
Integration testing is a level of the software testing process where individual units are combined and tested as a group.

7. What determines the level of risk?
The likelihood of an adverse event and the impact of the event.

8. What is Alpha testing?
Alpha Testing is a type of Acceptance Testing performed by the testers who are part of the organization, in other words: internal employees. It is the final stage of testing and it is usually done to verify that an application is free of errors / bugs before being launched on the market.

9. What is beta testing?
In software development, a beta test is the second phase of software testing in which a sampling of the intended audience tries the product out.
Beta is the second letter of the Greek alphabet. Originally, the term *alpha test* meant the first phase of testing in a software development process.

10. What is component testing?
Component testing is a sub-category of software testing in which each individual component of the software is separately tested without integrating with other components. It is also known as module testing.

11. What is functional system testing?
Functional Testing is a testing technique that is used to test the features/functionality of the system or Software, should cover all the scenarios including failure paths and boundary cases.

12. What is Non-Functional Testing?
Non-Functional testing is a software testing technique that verifies the attributes of the system such as memory leaks, performance or robustness of the system. Non-Functional testing is performed at all test levels.

13. What is GUI Testing?
Graphical user interface testing (GUI testing) refers to the testing of a graphical user interface to ensure trouble-free use and implementation.
A graphical user interface is a software tool that helps end users to utilize a software environment. These tools need to work well in order to support a user base.

14. What is Ad-hoc testing?
Ad hoc Testing is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

15. What is white box testing and list the types of white box testing?

White Box Testing is a testing technique in which software's internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

Following are important WhiteBox Testing Techniques:

- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Multiple Condition Coverage
- Finite State Machine Coverage
- Path Coverage
- Control flow testing
- Data flow testing

The different types of white box testing are:

1. Unit Testing
2. Static Analysis
3. Dynamic Analysis
4. Statement Coverage
5. Branch Testing Coverage
6. Security Testing
7. Mutation Testin

16. What is black box testing?

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings.

17. What are the different black box testing techniques?

1. Equivalence Partitioning
2. Boundary Value Testing
3. Decision Table Testing
4. State Transition Testing
5. Error Guessing

18. Mention what are the categories of defects?

Defects are usually categorized into three divisions.
☞ Wrong: When the requirements are not implemented properly.
☞ Missing: This category implements that a specification has not been apparently implemented, maybe a required facility of the customer was not appropriately applied.
☞ Extra: It indicates a required facility into the product, which was not noted by the end customer. But maybe it is an attribute which the user of the product actually wished for.

19. Mention what big-bang testing is?

Big Bang Testing is an Integration testing approach in which all the components or modules are integrated together at once and then tested as a unit. This combined set of components is considered as an entity while testing. If all of the components in the unit are not completed, the integration process will not execute.

20. What is the purpose of exit criteria?

Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.

21. When should "Regression Testing" be performed?

Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged.

22. What are 7 key principles? Explain in detail?
1. Testing shows presence of Defects
2. Exhaustive Testing is Impossible!
3. Early Testing
4. Defect Clustering
5. The Pesticide Paradox
6. Testing is Context Dependent
7. Absence of Errors Fallacy

☞ **Testing shows the presence of defects:**

- The goal of software testing is to make the software fail. Software testing reduces the presence of defects.
- Software testing talks about the presence of defects and doesn't talk about the absence of defects.
- Software testing can ensure that defects are present but it can not prove that software is defect-free. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects.

☞ **Exhaustive testing is not possible:**

- It is the process of testing the functionality of the software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. Exhaustive testing is impossible means the software can never test at every test case.
- It can test only some test cases and assume that the software is correct and it will produce the correct output in every test case. If the software

will test every test case then it will take more cost, effort, etc., which is impractical.

☞ **Early Testing:**

- To find the defect in the software, early test activity shall be started. The defect detected in the early phases of SDLC will be very less expensive. For better performance of software, software testing will start at the initial phase i.e. testing will perform at the requirement analysis phase.

☞ **Defect clustering:**

- In a project, a small number of modules can contain most of the defects. Pareto Principle to software testing state that 80% of software defect comes from 20% of modules.

☞ **Pesticide paradox:**

- Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.

☞ **Testing is context-dependent:**

- The testing approach depends on the context of the software developed. Different types of software need to perform different types of testing.
- For example, The testing of the e-commerce site is different from the testing of the Android application.

☞ **Absence of errors fallacy:**

- If a built software is 99% bug-free but it does not follow the user requirement then it is unusable.
- It is not only necessary that software is 99% bug-free but it is also mandatory to fulfill all the customer requirements.

23. Difference between QA v/s QC v/s Tester

Quality Assurance (QA)	Quality Control (QC)
It is a procedure that focuses on providing assurance that quality requested will be achieved	It is a procedure that focuses on fulfilling the quality requested.
QA aims to prevent the defect	QC aims to identify and fix defects
It is a method to manage the quality-Verification	It is a method to verify the quality-Validation

It does not involve executing the program	It always involves executing a program
It's a Preventive technique	It's a Corrective technique
It's a Proactive measure	It's a Reactive measure
It is the procedure to create the deliverables	It is the procedure to verify that deliverables
QA involves in full software development life cycle	QC involves in full software testing life cycle
In order to meet the customer requirements, QA defines standards and methodologies	QC confirms that the standards are followed while working on the product
It is performed before Quality Control	It is performed only after QA activity is done
It is a Low-Level Activity, it can identify an error and mistakes which QC cannot	It is a High-Level Activity, it can identify an error that QA cannot
Its main motive is to prevent defects in the system. It is a less time-consuming activity	Its main motive is to identify defects or bugs in the system. It is a more time-consuming activity
QA ensures that everything is executed in the right way, and that is why it falls under verification activity	QC ensures that whatever we have done is as per the requirement, and that is why it falls under validation activity
It requires the involvement of the whole team	It requires the involvement of the Testing team
The statistical technique applied on QA is known as SPC or Statistical Process Control (SPC)	The statistical technique applied to QC is known as SQC or Statistical Quality Control

24. Difference between Smoke and Sanity?

SMOKE TESTING	SANITY TESTING
Smoke Testing is performed to ascertain that the critical functionalities of the program are working fine.	Sanity testing is done at random to verify that each functionality is working as expected.
Smoke testing exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system.
The main objective of the testing is to verify the stability of the system.	The main objective of the testing is to verify the rationality of the system.

Smoke testing is usually documented and scripted.	Sanity testing is not documented and is unscripted.
This testing is performed by the developers or testers.	Sanity testing in software testing is usually performed by testers.
It is a well elaborate and planned testing.	This is not a planned test and is done only when there is a shortage of time.
This is a wide and deep testing.	This is a wide and shallow testing.
Smoke testing is a subset of <u>Acceptance testing.</u>	Sanity testing is a subset of Regression Testing

25. Difference between verification and Validation

Verification	Validation
It includes checking documents, design, codes and programs.	It includes testing and validating the actual product.
Verification is the static testing.	Validation is the dynamic testing.
It does *not* include the execution of the code.	It includes the execution of the code.
Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing.
It checks whether the software conforms to specifications or not.	It checks whether the software meets the requirements and expectations of a customer or not.
It can find the bugs in the early stage of the development.	It can only find the bugs that could not be found by the verification process.
The goal of verification is application and software architecture and specification.	The goal of validation is an actual product.
Quality assurance team does verification.	Validation is executed on software code

	with the help of testing team.
It comes before validation.	It comes after verification.
It consists of checking of documents/files and is performed by human.	It consists of execution of program and is performed by computer.

26. Explain types of Performance testing.

    a. Capacity Testing.
    b. Load Testing.
    c. Volume Testing.
    d. Stress Testing.
    e. Soak Testing.

27. What is Error, Defect, Bug and failure?

Error: A mistake in coding is called error.
Defect: error found by tester is called defect.
Bug: defect accepted by development team then it is called bug.
Failure: build does not meet the requirements then it is failure.

28. Difference between Priority and Severity

Parameters	Severity in Testing	Priority in Testing
Definition	Severity is a term that denotes how severely a defect can affect the functionality of the software.	Priority is a term that defines how fast we need to fix a defect.
Parameter	Severity is basically a parameter that denotes the total impact of a given defect on any software.	Priority is basically a parameter that decides the order in which we should fix the defects.
Relation	Severity relates to the standards of quality.	Priority relates to the scheduling of defects to resolve them in software.
Value	The value of severity is objective.	The value of priority is subjective.
Change of Value	The value of Severity changes continually from time to time.	The value of Priority changes from time to time.
Who Decides the Defect	The testing engineer basically decides a defect's severity level.	The product manager basically decides a defect's priority level.
Types	There are 5 types of Severities:	There are 3 types of

	Cosmetic, Minor, Moderate, Major, and Critical.	Priorities: High, Medium, and Low.

29. What is Bug Life Cycle OR Defect Life Cycle?

☞ When a defect is found, it passes different states throughout its lifetime. The defect lifecycle helps to coordinate the defect status changes to the team members. This process is called the Bug Life Cycle or Defect life cycle.

30. Explain the difference between Functional testing and Non-Functional testing.

Functional Testing	Non-functional Testing
It verifies the operations and actions of an application.	It verifies the behavior of an application.
It is based on requirements of customer.	It is based on expectations of customer.
It helps to enhance the behavior of the application.	It helps to improve the performance of the application.
Functional testing is easy to execute manually.	It is hard to execute non-functional testing manually.
It tests what the product does.	It describes how the product does.
Functional testing is based on the business requirement.	Non-functional testing is based on the performance requirement.
Examples: **1.** Unit Testing **2.** Smoke Testing **3.** Integration Testing **4.** Regression Testing	**Examples:** **1.** Performance Testing **2.** Load Testing **3.** Stress Testing **4.** Scalability Testing

31. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

Parameter	SDLC	STLC
Origin	Development Life Cycle	Testing Life Cycle
Objective	The main object of SDLC life cycle is to complete successful development of the software including testing and other phases.	The only objective of the STLC phase is testing.
Requirement Gathering	In SDLC the business analyst gathers the requirements and create Development Plan	In STLC, the QA team analyze requirement documents like functional and non-functional documents and create System Test Plan
High & Low-Level Design	In SDLC, the development team creates the high and low-level design plans	In STLC, the test analyst creates the Integration Test Plan
Coding	The real code is developed, and actual work takes place as per the design documents.	The testing team prepares the test environment and executes them
Maintenance	SDLC phase also includes post-deployment supports and updates.	Testers, execute regression suits, usually automation scripts to check maintenance code deployed.

32. What is the difference between test scenarios, test cases, and test script?

Test Scenario	Test Case	Test Script
Is any functionality that can be tested.	Is a set of actions executed to verify particular features or functionality.	Is a set of instructions to test an app automatically.
Is derived from test artifacts like Business Requirement Specification (BRS) and Software Requirement Specification (SRS).	Is mostly derived from test scenarios.	Is mostly derived from test cases.
Helps test the end-to-end functionality in an Agile way.	Helps in exhaustive testing of an app.	Helps to test specific things repeatedly.
Is more focused on what to test.	Is focused on what to test and how to test.	Is focused on the expected result.
Takes less time and fewer resources to create.	Requires more resources and time.	Requires less time for testing but more resources for scripts creating and updating.
Includes an end-to-end functionality to be tested.	Includes test steps, data, expected results for testing.	Includes different commands to develop a script.
The main task is to check the full functionality of a software application.	The main task is to verify compliance with the applicable standards, guidelines, and customer requirements.	The main task is to verify that nothing is skipped, and the results are true as the desired testing plan.
Allows quickly assessing the testing scope.	Allows detecting errors and defects.	Allows carrying out an automatic execution of test cases.

33. Explain what Test Plan is? What is the information that should be covered.

Test Plan is A document describing the scope, approach, resources, and schedule of intended test activities.

34. What are the different Methodologies in Agile Development Model?

☞ Scrum
☞ Extreme Programming
☞ Dynamic system Development method(DSDM)
☞ Test Driven Development(TDD)
☞ Feature Driven Development(FDD)
☞ XBreed
☞ Crystal
☞ Kanban

35. Explain the difference between Authorization and Authentication in Web testing.

Authentication	Authorization
In the authentication process, the identity of users are checked for providing the access to the system.	While in authorization process, a the person's or user's authorities are checked for accessing the resources.
In the authentication process, users or persons are verified.	While in this process, users or persons are validated.
It is done before the authorization process.	While this process is done after the authentication process.
It needs usually the user's login details.	While it needs the user's privilege or security levels.
Authentication determines whether the person is user or not.	While it determines **What permission does the user have?**
Generally, transmit information through an ID Token.	Generally, transmit information through an Access Token.
The OpenID Connect (OIDC) protocol is an authentication protocol that is generally in charge of user authentication process.	The OAuth 2.0 protocol governs the overall system of user authorization process.
Popular Authentication Techniques- • Password-Based Authentication	Popular Authorization Techniques- • Role-Based Access Controls (RBAC)

Authentication	Authorization
• Passwordless Authentication • 2FA/MFA (Two-Factor Authentication / Multi-Factor Authentication) • Single sign-on (SSO) • Social authentication	• SON web token (JWT) Authorization • SAML Authorization • OpenID Authorization • OAuth 2.0 Authorization
The authentication credentials can be changed in part as and when required by the user.	The authorization permissions cannot be changed by user as these are granted by the owner of the system and only he/she has the access to change it.
The user authentication is visible at user end.	The user authorization is not visible at the user end.
The user authentication is identified with username, password, face recognition, retina scan, fingerprints, etc.	The user authorization is carried out through the access rights to resources by using roles that have been pre-defined.

36. What are the common problems faced in Web testing?

Integration testing exposes problems with interfaces among different program components before deployment

- ☞ Interoperability. ...
- ☞ Security. ...
- ☞ Performance. ...
- ☞ Usability. ...
- ☞ Quality Testing, Exceptional Services.