1- **Software testing:** is a way to assess the quality of the software and to reduce the risk of software failure in operation.

2- **Dynamic testing:** is a testing which does involve the execution of the component or system being tested.

3- **Static testing:** is a testing does involve the execution of the component or system being tested.

4- **Objectives of Testing:**

- To prevent defects by evaluate work products such as requirements, user stories, design, and code
- To verify whether all specified requirements have been fulfilled
- To check whether the test object is complete and validate if it works as the users and other stakeholders expect
- To build confidence in the level of quality of the test object
- To find defects and failures thus reduce the level of risk of inadequate software quality
- To provide sufficient information to stakeholders to allow them to make informed decisions, especially regarding the level of quality of the test object
- To comply with contractual, legal, or regulatory requirements or standards

5- **Errors, Defects, and Failures:** A person can make an error (mistake), which can lead to the introduction of a defect (fault or bug) in the software code or related work product. If a defect in the code is executed, this may cause a failure

6- **A/B testing:** A statistical testing approach to determine which of two systems or components performs better.

7- **Acceptance criteria:** The criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity.

8- **Acceptance testing:** A test level that focuses on determining whether to accept the system.

9- **Accessibility:** The degree to which a component or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.

10- **Actual result:** The behavior produced when a component or system is tested.

11- **Debugging:** is the development activity that finds, analyzes, and fixes such defects

12- **Defect:** An imperfection or deficiency in a work product where it does not meet its requirements or specifications.

13- **Reasons that cause failures:**

By

- defects in the code
- environmental conditions. For example, radiation, electromagnetic fields, and pollution can cause defects in firmware
- influence the execution of software by changing hardware conditions

14- **Error:** A human action that produces an incorrect result

15- **Failure:** An event in which a component or system does not perform a required function within specified limits.

16- **Quality:** The degree to which a component or system satisfies the stated and implied needs of its various stakeholders.

17- **False positives:** occur due to errors in the way tests were executed, or due to defects in the test data, the test environment, or other test ware, or for other reasons. false positives are reported as defects, but aren't actually defects.

18- **False negatives:** are tests that do not detect defects that they should have detected.

19- **Quality Assurance:** Activities focused on providing confidence that quality requirements will be fulfilled.

20- **Quality control:** involves various activities, including test activi1ties, that support the achievement of appropriate levels of quality.

21- **Root Cause:** A source of a defect such that if it is removed, the occurrence of the defect type is decreased or removed.

22- **The root causes of defects:** are the earliest actions or conditions that contributed to creating the defects.

23- **Test Analysis:** The activity that identifies test conditions by analyzing the test basis.

24- **Test Basis:** The body of knowledge used as the basis for test analysis and design.

25- **Test Case:** A set of preconditions, inputs, actions (where applicable), expected results and postconditions, developed based on test conditions.

26- **Test Completion:** The activity that makes test ware available for later use, leaves test environments in a satisfactory condition and communicates the results of testing to relevant stakeholders.

27- **Test Condition:** A testable aspect of a component or system identified as a basis for testing.

28- **Test Oracle:** A source to determine an expected result to compare with the actual result of the system under test.

29- **Test Planning:** The activity of establishing or updating a test plan.

30- **Test Procedure:** A sequence of test cases in execution order, and any associated actions that may be required to set up the initial preconditions and any wrap up activities post execution.

31- **Test Process:** The set of interrelated activities comprising of test planning, test monitoring and control, test analysis, test design, test implementation, test execution, and test completion.

32- **Test Suite (Set):** A set of test scripts or test procedures to be executed in a specific test run

33- **Test ware:** Work products produced during the test process for use in planning, designing, executing, evaluating and reporting on testing.

34- **Traceability:** The degree to which a relationship can be established between two or more work products.

Q #1) What is Dynamic Testing?

Answer: Dynamic testing is done by executing the code or program with various input values and later on the output is verified.

Q #2) What is GUI Testing?

Answer: GUI or Graphical User Interface testing is the process of testing the software's user interface against the provided requirements/mockups/HTML designs etc.,

Q #3) What is Formal Testing?

Answer: Software verification, carried out by following a test plan, testing procedures and proper documentation with approval from the customer is termed as Formal Testing.

Q #4) What is Risk-Based Testing?

Answer: Identifying the critical functionality in the system and then deciding the orders in which these functionalities are to be tested and perform testing is termed as Risk-based Testing.

Q #5) What is Early Testing?

Answer: Perform testing as soon as possible in the development lifecycle to find defects at the early stages of STLC. Early testing is helpful to reduce the cost of fixing defects at the later stages of STLC.

Q #6) What is Exhaustive Testing?

Answer: Testing functionality with all valid, invalid inputs and pre-conditions is called Exhaustive testing.

Q #7) What is Defect Clustering?

Answer: Any small module or functionality may contain a number of defects and to concentrate more on testing these functionalities is known as Defect Clustering.

Q #8) What is Pesticide Paradox?

Answer: If already prepared test cases do not find defects, add/revise test cases to find more defects, this is known as Pesticide Paradox.

Q #9) What is Static Testing?

Answer: Manual verification of the code without executing the program is called Static Testing. In this process, the issues are identified in the code by verifying code, requirement and design documents.

Q #10) What is Positive Testing?

Answer: It is the form of testing which is conducted on the application to determine if the system works properly or not. Basically, it is known as the “test to pass” approach.

Q #11) What is Negative Testing?

Answer: Testing software with a negative approach to check if the system is not “showing error when not supposed to” and “not showing error when supposed to” is termed as Negative Testing.

Negative Testing is a software testing type used to check the software application for unexpected input data and conditions.

Q #12) What is an End-to-End Testing?

Answer: Testing the overall functionality of the system including the data integration among all the modules is called End-to-End Testing.

Q #13) What is Exploratory Testing?

Answer: Exploring the application, understanding its functionalities, adding (or) modifying the existing test cases for better testing is called Exploratory testing.

Q #14) What is Monkey Testing?

Answer: Testing conducted on an application without any plan and carried out randomly with the tests to find any system crash with the intention of finding tricky defects is called Monkey Testing.

Q #15) What is Non-Functional Testing?

Answer: Validating various non-functional aspects of the system such as user interfaces, user-friendliness, security, compatibility, Load, Stress, and Performance, etc., is called Non-Functional testing.

Q #16) What is Usability Testing?

Answer: Checking how easily the end-users are able to understand and operate the application is called Usability Testing.

Q #17) What is Security Testing?

Answer: Validating whether all security conditions are properly implemented in the software (or) not is called Security testing.

Q #18) What is Performance Testing?

Answer: The process of measuring various efficiency characteristics of a system such as response time, load stress transactions per minute, transaction mix, etc., is termed Performance Testing.

Q #19) What is Load Testing?

Answer: Analyzing both the functional and performance behavior of an application under various conditions is called Load Testing.

Q #20) What is Stress Testing?

Answer: Checking the application behavior under stress conditions

(or)

Reducing the system resources and keeping the load as constant and checking how the application is behaving is called Stress Testing.

Q #21) What is Process?

Answer: A process is a set of practices performed to achieve a given purpose; it may include tools, methods, materials or people.

Q #22) What is Software Configuration Management?

Answer: The process of identifying, organizing and controlling changes to Software development and maintenance.

(or)

It is a methodology to control and manage a software development project.

Q #23) What is a Testing Process / Life Cycle?

Answer: It includes the below factors:

- Writing a Test Plan
- Test Scenarios
- Test Cases
- Executing the Test Cases
- Test Results
- Defect Reporting
- Defect Tracking
- Defect Closing
- Test Release

Q #24) What is the full form of CMMI?

Answer: Capability Maturity Model Integration

Q #25) What is a Code Walk Through?

Answer: An informal analysis of the program source code to find the defects and verify the coding techniques is termed as a Code Walk Through.

Q #26) What is Unit Level Testing?

Answer: Testing of single programs, modules or unit of code is termed as Unit Level Testing.

Q #27) What is Integration Level Testing?

Answer: Testing of related programs, modules (or) unit of code.

(or)

Partitions of the system which are ready for testing with other partitions of the system are termed as Integration level testing.

Q #28) What is System Level Testing?

Answer: Testing of the entire computer system across all the modules is termed as System-level testing. This kind of testing can include Functional as well as Structural Testing.

Q #29) What is Alpha Testing?

Answer: Testing of a whole computer system before rolling out to the UAT is termed as Alpha testing.

Q #30) What is User Acceptance Testing (UAT)?

Answer: UAT is the form of testing of a computer system by the client to verify if it adhered to the provided requirements or not.

Q #31) What is a Test Plan?

Answer: It is a document describing the scope, approach, resources, and schedule of testing activities. It identifies test items, features to be tested, testing tasks, who will do each task, and any risks requiring contingency planning.

Q #32) What is a Test Scenario?

Answer: Identifying all the possible areas to be tested (or) what is to be tested is termed as Test Scenario.

Q #33) What is ECP (Equivalence Class Partition)?

Answer: It is a method for deriving test cases.

Q #34) What is a Defect?

Answer: Any flaw or imperfection in a software work product is termed as a Defect.

(or)

When the expected result does not match with the application actual result, it is termed as a Defect.

Q #35) What is Severity?

Answer: It defines the importance of the defect from the functional point of view i.e. how critical is a defect with respect to the application.

Q #36) What is Priority?

Answer: It indicates the importance or urgency of fixing a defect

Q #37) What is Re-Testing?

Answer: Re-testing the application means verifying whether the defects have been fixed or not.

Q #38) What is Regression Testing?

Answer: Verifying an existing functional and non-functional area after making changes to the part of a software or addition of new features is termed as Regression Testing.

Q #39) What is Recovery Testing?

Answer: Checking whether the system is able to handle some unexpected or unpredictable situations is called Recovery Testing.

Q #40) What is Globalization Testing?

Answer: It is the process of verifying whether the software can be run independently of its geographical and cultural environment. Verifying if the application has the feature to set and change language, date, format, and currency or if it is designed for global users.

Q #41) What is Localization Testing?

Answer: Verifying globalized application for a particular locality of users, under cultural and geographical conditions is termed as Localization Testing.

Q #42) What is Installation Testing?

Answer: Checking whether we are able to install a software successfully (or) not, as per the guidelines given in the installation document is called Installation Testing.

Q #43) What is Un-Installation Testing?

Answer: Checking whether we are able to uninstall the software from the system successfully (or) not is called Un-Installation Testing

Q #44) What is Compatibility Testing?

Answer: Checking whether the application is compatible with different software and hardware environment or not is called Compatibility Testing.

Q #45) What is a Test Strategy?

Answer: It is a part of a test plan describing how testing is carried out for the project and what testing types need to be performed on the application.

Q #46) What is a Test Case?

Answer: A Test case is a set of pre-conditional steps to be followed with input data and expected behavior to validate the functionality of a system.

Q #47) What is Business Validation Test Case?

Answer: A test case that is prepared to check the business condition or a business requirement is called the Business Validation test case.

Q #48) What is a Good Test Case?

Answer: A Test case that has a high priority of catching defects is called a Good Test Case.

Q #49) What is Use Case Testing?

Answer: Validating a software to confirm whether it is developed as per the use cases or not is called Use Case testing.

Q #50) What is a Defect Age?

Answer: The time gap between the date of detection & the date of closure of a defect is termed as Defect Age.

Q #51) What is the Showstopper Defect?

Answer: A defect that does not permit testing to continue further is called Showstopper Defect.

Q #52) What is a Test Closure?

Answer: It is the last phase of the STLC, where the management prepares various test summary reports that explain the complete statistics of the project based on the testing carried out.

Q #53) What is Bucket Testing?

Answer: Bucket testing is also known as A/B testing. It is mostly used to study the impact of various product designs on website metrics. Two simultaneous versions run on a single or a set of web pages to measure the difference in click rates, interface, and traffic.

Q #54) What is meant by Entry Criteria and Exit Criteria in Software Testing?

Answer: Entry Criteria is the process that must be present when a system begins, like,

- SRS – Software
- FRS
- Use Case
- Test Case
- Test Plan

Exit criteria ensure whether the testing is completed and the application is ready for release, like,

- Test Summary Report
- Metrics
- Defect Analysis Report

Q #55) What is Concurrency Testing?

Answer: This is a multiple user testing to access the application at the same time to verify the effect on code, module or DB and it is mainly used to identify the locking and deadlocking situations in the code.

Q #56) What is Web Application Testing?

Answer: Web application testing is done on a website to check – load, performance, security, functionality, interface, compatibility and other usability-related issues.

Q #57) What is Unit Testing?

Answer: Unit testing is done to check whether the individual modules of the source code are working properly or not.

Q #58) What is Interface Testing?

Answer: Interface testing is done to check whether the individual modules are communicating properly as per the specifications or not. Interface testing is mostly used to test the user interface of GUI applications.

Q #59) What is Gamma Testing?

Answer: Gamma testing is done when the software is ready for release with the specified requirements, this testing is done directly by skipping all the in-house testing activities.

Q #60) What is the Test Harness?

Answer: Test Harness is configuring a set of tools and test data to test an application under various conditions, which involves monitoring the output with the expected output for correctness.

The benefits of Testing Harness are: Productivity increase due to process automation and increase in the product quality

Q #61) What is Scalability Testing?

Answer: It is used to check whether the functionality and performance of a system are capable to meet the volume and size changes as per the requirements.

Scalability testing is done using the load test by changing various software, hardware configurations, and testing environment.

Q #62) What is Fuzz Testing?

Answer: Fuzz testing is a black-box testing technique that uses random bad data to attack a program to check if anything breaks in the application.

Q #63) What is the difference between QA, QC, and Testing?

Answer:

- QA: It is process-oriented and its aim is to prevent the defects in an application.
- QC: QC is product-oriented and it is a set of activities used to evaluate a developed work product.
- Testing: Executing and verifying an application with the intention of finding defects.
-

Q #64) What is Data-Driven Testing?

Answer: It is an Automation testing process in which an application is tested with multiple sets of data with different preconditions as an input to the script.

1. What is Exploratory Testing?

Exploratory testing is a hands-on approach in which testers are involved in minimum planning and maximum test execution. The planning involves the creation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort, the objectives and possible approaches to be used. The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts. This does not mean that other, more formal testing techniques will not be used. For example, the tester may decide to use boundary value analysis but will think through and test the most important boundary values without necessarily writing them down. Some notes will be written during the exploratory-testing session so that a report can be produced afterward.

2. What is "use case testing"?

In order to identify and execute the functional requirement of an application from start to finish "use case" is used and the techniques used to do this is known as "Use Case Testing."

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

SDLC deals with development/coding of the software while STLC deales with validation and verification of the software

4. What is traceability matrix?

The relationship between test cases and requirements is shown with the help of a document. This document is known as a traceability matrix.

5. What is Equivalence partitioning testing?

Equivalence partitioning testing is a software testing technique which divides the application input test data into each partition at least once of equivalent data from which test cases can be derived. By this testing method, it reduces the time required for software testing.

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

White box testing technique involves selection of test cases based on an analysis of the internal structure (Code coverage, branches coverage, paths coverage, condition coverage, etc.) of a component or system. It is also known as Code-Based testing or Structural testing. Different types of white box testing are

1. Statement Coverage
2. Decision Coverage

7. In white box testing, what do you verify?

In white box testing following steps are verified.

1. Verify the security holes in the code
2. Verify the incomplete or broken paths in the code
3. Verify the flow of structure according to the document specification
4. Verify the expected outputs
5. Verify all conditional loops in the code to check the complete functionality of the application
6. Verify the line by line coding and cover 100% testing

8. What is black box testing? What are the different black box testing techniques?

Black box testing is the software testing method which is used to test the software without knowing the internal structure of code or program. This testing is usually done to check the functionality of an application. The different black box testing techniques are

1. Equivalence Partitioning
2. Boundary value analysis
3. Cause-effect graphing

9. What is the difference between static and dynamic testing?

Static testing: During Static testing method, the code is not executed, and it is performed using the software documentation.

Dynamic testing: To perform this testing the code is required to be in an executable form.

10. What are verification and validation?

Verification is a process of evaluating software at the development phase. It helps you to decide whether the product of a given application satisfies the specified requirements. Validation is the process of evaluating software at the after the development process and to check whether it meets the customer requirements.

11. What are the different test levels?

There are four test levels

1. Unit/component/program/module testing
2. Integration testing
3. System testing
4. Acceptance testing

12. What is Integration testing?

Integration testing is a level of software testing process, where individual units of an application are combined and tested. It is usually performed after unit and functional testing.

13. What Test Plans consists of?

Test design, scope, test strategies, approach are various details that Test plan document consists of.

1. Test case identifier
2. Scope
3. Features to be tested
4. Features not to be tested
5. Test strategy & Test approach
6. Test deliverables
7. Responsibilities
8. Staffing and training
9. Risk and Contingencies

14. What is the difference between UAT (User Acceptance Testing) and System testing?

System Testing: System testing is finding defects when the system undergoes testing as a whole; it is also known as end-to-end testing. In such type of testing, the application suffers from beginning till the end.

UAT: User Acceptance Testing (UAT) involves running a product through a series of specific tests which determines whether the product will meet the needs of its users.

15. Mention the difference between Data Driven Testing and Retesting?

Retesting: It is a process of checking bugs that are actioned by the development team to verify that they are fixed.

Data Driven Testing (DDT): In data driven testing process, the application is tested with multiple test data. The application is tested with a different set of values.

16. What are the valuable steps to resolve issues while testing?

- Record: Log and handle any problems which have happened
- Report: Report the issues to higher level manager
- Control: Define the issue management process

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

Difference between test scenarios and test cases is that

Test Scenarios: A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility.

Test Cases: It is a document that contains the steps that have to be executed; it has been planned earlier.

Test Script: It is written in a programming language and it’s a short program used to test part of the functionality of the software system. In other words, a written set of steps that should be performed manually.

18. What is Latent defect?

Latent defect: This defect is an existing defect in the system which does not cause any failure as the exact set of conditions has never been met

19. What are the two parameters which can be useful to know the quality of test execution?

To know the quality of test execution, we can use two parameters

- Defect reject ratio
- Defect leakage ratio

Parameters for quality of test execution

20. What is the function of the software testing tool "phantom"?

Phantom is a freeware and is used for windows GUI automation scripting language. It allows us to take control of windows and functions automatically. It can simulate any combination of keystrokes and mouse clicks as well as menus, lists and more.

21. Explain what Test Deliverables is?

Test Deliverables are a set of documents, tools and other components that have to be developed and maintained in support of testing.

There are different test deliverables at every phase of the software development lifecycle

- Before Testing
- During Testing
- After the Testing

22. What is mutation testing?

Mutation testing is a technique to identify if a set of test data or test case is useful by intentionally introducing various code changes (bugs) and retesting with original test data/ cases to determine if the bugs are detected.

23. What all things you should consider before selecting automation tools for the AUT?

- Technical Feasibility
- Complexity level
- Application stability
- Test data
- Application size
- Re-usability of automated scripts
- Execution across environment

24. How will you conduct Risk Analysis?

For the risk analysis following steps need to be implemented

1. Finding the score of the risk
2. Making a profile for the risk
3. Changing the risk properties
4. Deploy the resources of that test risk
5. Making a database of risk

25. What are the categories of debugging?

Categories for debugging

1. Brute force debugging
2. Backtracking
3. Cause elimination
4. Program Slicing
5. Fault tree analysis

26. What is fault masking explain with example?

When the presence of one defect hides the presence of another defect in the system, it is known as fault masking.

Example: If the "Negative Value" cause a firing of unhandled system exception, the developer will prevent the negative values input. This will resolve the issue and hide the defect of unhandled exception firing.

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

A test plan can be defined as a document describing the scope, approach, resources, and schedule of testing activities and a test plan should cover the following details.

- Test Strategy
- Test Objective
- Exit/Suspension Criteria
- Resource Planning
- Test Deliverables

28. How can you eliminate the product risk in your project?

It helps you to eliminate product risk in your project, and there is a simple yet crucial step that can reduce the product risk in your project.

- Investigate the specification documents
- Have discussions about the project with all stakeholders including the developer
- As a real user walk around the website

29. What is the common risk that leads to project failure?

The common risk that leads to a project failure are

- Not having enough human resource
- Testing Environment may not be set up properly
- Limited Budget
- Time Limitations

30. On what basis you can arrive at an estimation for your project?

To estimate your project, you have to consider the following points

- Divide the whole project into the smallest tasks
- Allocate each task to team members
- Estimate the effort required to complete each task

31. Explain how you would allocate a task to team members?

Task	Member
• Analyze software requirement specification	• All the members
• Create the test specification	• Tester/Test Analyst
• Build up the test environment	• Test administrator
• Execute the test cases	• Tester, a Test administrator
• Report defects	• Tester

32. Explain what is testing type and what are the commonly used testing type?

To get an expected test outcome, a standard procedure is followed which is referred to as Testing Type.

Commonly used testing types are

- Unit Testing: Test the smallest code of an application
- API Testing: Testing API created for the application
- Integration Testing: Individual software modules are combined and tested
- System Testing: Complete testing of the system
- Install/Uninstall Testing: Testing done from the point of client/customer view
- Agile Testing: Testing through Agile technique

33. While monitoring your project what all things you have to consider?

The things that have to be taken in considerations are

- Is your project on schedule
- Are you over budget?
- Are you working towards the same career goal?
- Have you got enough resources?
- Are there any warning signs of impending problems?
- Is there any pressure from management to complete the project sooner?

34. What are the common mistakes which create issues?

- Matching resources to wrong projects
- Test manager lack of skills
- Not listening to others
- Poor Scheduling
- Underestimating
- Ignoring the small problems
- Not following the process

35. What does a typical test report contain? What are the benefits of test reports?

A test report contains the following things:

- Project Information
- Test Objective
- Test Summary
- Defect

The benefits of test reports are:

- Current status of project and quality of product are informed
- If required, stakeholder and customer can take corrective action
- A final document helps to decide whether the product is ready for release

36. What is test management review and why it is important?

Management review is also referred to as Software Quality Assurance or SQA. SQA focusses more on the software process rather than the software work products. It is a set of activities designed to make sure that the project manager follows the standard process. SQA helps test manager to benchmark the project against the set standards.

37. What are the best practices for software quality assurance?

The best practices for an effective SQA implementation is

- Continuous Improvement
- Documentation
- Tool Usage
- Metrics
- Responsibility by team members
- Experienced SQA auditors

38. When is RTM (Requirement Traceability Matrix) prepared?

RTM is prepared before test case designing. Requirements should be traceable from review activities.

39. What is the difference between Test matrix and Traceability matrix?

Test Matrix: Test matrix is used to capture actual quality, effort, the plan, resources and time required to capture all phases of software testing

Traceability Matrix: Mapping between test cases and customer requirements is known as Traceability Matrix

40. In manual testing what are stubs and drivers?

Both stubs and drivers are part of incremental testing. In incremental testing, there are two approaches namely bottom-up and top-down approach. Drivers are used in bottom-up testing and stub is used for a top-down approach. In order to test the main module, the stub is used, which is a dummy code or program.

41. What is the step you would follow once you find the defect?

Once a defect is found you would follow the step

a) Recreate the defect

b) Attach the screenshot

c) Log the defect

42. Explain what is “Test Plan Driven” or “Key Word Driven” method of testing?

This technique uses the actual test case document developed by testers using a spreadsheet containing special “key Words”. The key words control the processing.

43. What is the DFD (Data Flow Diagram)?

When a “flow of data” through an information system is graphically represented, then it is known as Data Flow Diagram. It is also used for the visualization of data processing.

44. Explain what LCSAJ is?

LCSAJ stands for 'linear code sequence and jump.' It consists of the following
three items

a) Start of the linear sequence of executable statements

b) End of the linear sequence

c) The target line to which control flow is transferred at the end of the linear sequence

45. Explain what N+1 testing is?

The variation of regression testing is represented as N+1. In this technique, the testing is performed in multiple cycles in which errors found in test cycle 'N' are resolved and re-tested in test cycle N+1. The cycle is repeated unless there are no errors found.

46. What is Fuzz testing and when it is used?

Fuzz testing is used to detect security loopholes and coding errors in software. In this technique, random data is added to the system in an attempt to crash the system. If vulnerability persists, a tool called fuzz tester is used to determine potential causes. This technique is more useful for bigger projects but only detects a major fault.

47. Mention what the main advantages of statement coverage metric of software testing are?

The benefit of statement coverage metric is that

a) It does not require processing source code and can be applied directly to object code

b) Bugs are distributed evenly through the code, due to which percentage of executable statements covered reflects the percentage of faults discovered

48. How to generate test cases for “replace a string” method?

a) If characters in new string > characters in the previous string. None of the characters should get truncated

b) If characters in new string< characters in the previous string. Junk characters should not be added

c) Spaces after and before the string should not be deleted

d) String should be replaced only for the first occurrence of the string

49. How will you handle a conflict amongst your team members?

- I will talk individually to each person and note their concerns
- I will find a solution to the common problems raised by team members
- I will hold a team meeting, reveal the solution and ask people to co- operate

50. Mention what are the categories of defects?

Mainly there are three defect categories

- **Wrong**: When a requirement is implemented incorrectly
- **Missing**: It is a variance from the specification, an indication that a specification was not implemented or a requirement of the customer is not met
- **Extra**: A requirement incorporated into the product that was not given by the end customer. It is considered as a defect because it is a variance from the existing requirements