Shree H.N. Shukla College of I.T. &Mgmt, Rajkot

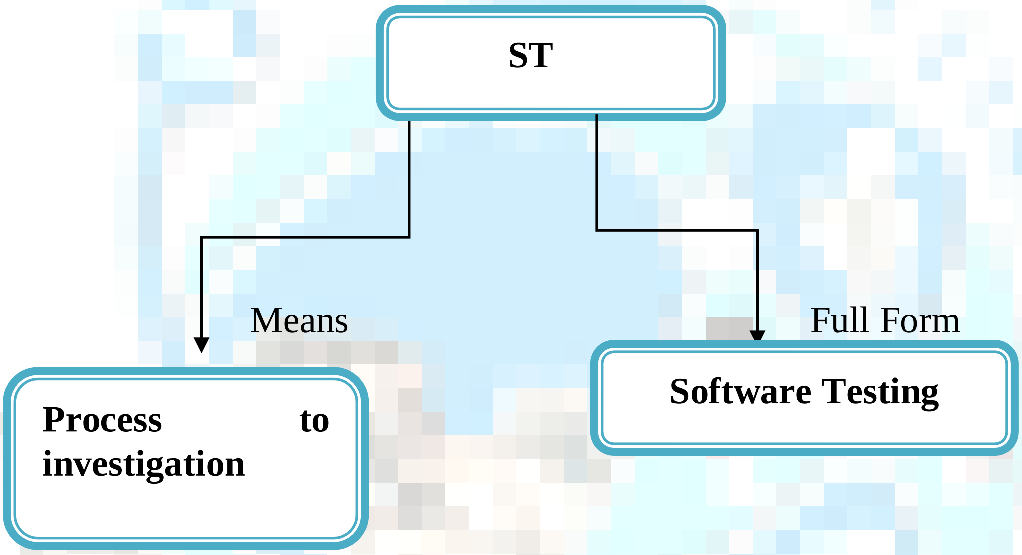
2-Vaishalinagar,
Raiya Road,
Nr. Amrapali Under Bridge,
Rajkot 360001,

3-vaishlinagar,
Raiya Road,
Nr. Amrapali Under Bridge,
Rajkot 360001.

UNIT -2(PART – 1)

BASIC OF SOFTWARE TESTING

- **TOPIC:-** Introduction to Software Testing (2 MARKS)



- Testing is one type of investigation process for any product whether it is as per specification or not.
- Software Testing is a **process to investigation** or insurance to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate.
- Software Testing also provides an objective, independent view of the software to allow the **business to appreciate and understand the risks at implementation of the software**.
- Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

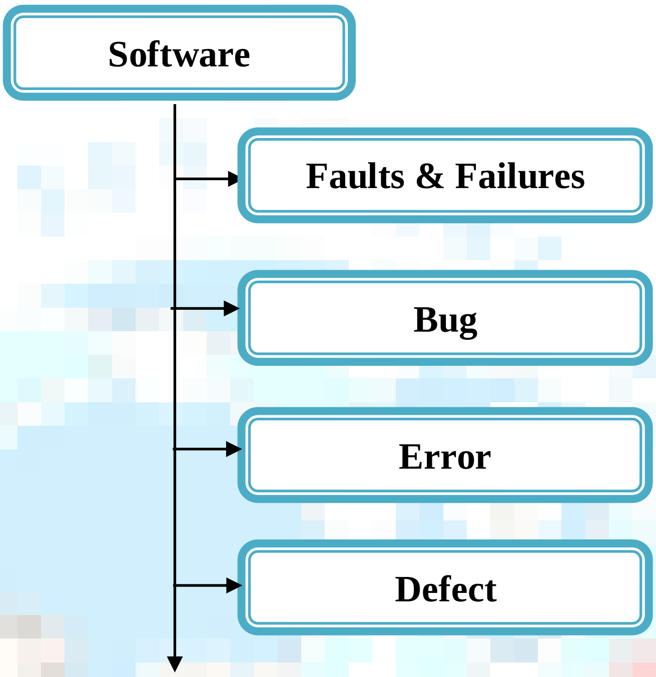
- It can also be stated as **the process of validating and verifying that** a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics.
- Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

➢ **SUMMARY:-**

- Testing is one type of investigation process for any product whether it is as per specification or not.
- Software Testing is a **process to investigation** or insurance to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate.
- It can also be stated as **the process of validating and verifying that** a software program/application/product meets the business and technical requirements

NO.	**QUESTION**	**ANSWER**
1.	ST	SOFTWARE TESTING
2.	What is Software Testing?	process to investigation
3.	Testing?	Testing is one type of investigation process for any product whether it is as per specification or not.

- **TOPIC:-** Software Faults, Failures, Bug, Error and Defect



- **<u>Software Fault & Failure:</u>**

- **<u>A system failure</u>** occurs when the delivered service no longer complies with the specifications, the latter being an agreed description of the system's expected function and/or service.
- This definition applies to both hardware and software system failures.

- **<u>Faults or bugs in hardware or a software component cause errors.</u>**

- An error is defined as that part of the system which is liable to lead to subsequent failure, and an error affecting the service is an indication that a failure occurs or has occurred.

- If the system comprises of multiple components, errors can lead to a component failure. As various components in the system interact, failure of one component might introduce one or more faults in another.

Bug: (1 MARK)

- **A software bug**is an **error, flaw, mistake, failure, or fault in a computer program** that prevents it from behaving as intended (e.g., producing an incorrect or unexpected result).

- Most bugs arise from mistakes and errors made by people in either a program's source code or its design, and a few are caused by compilers producing incorrect code.

- A program that contains a large number of bugs, and/or bugs that seriously interfere with its functionality, is said to be buggy.

- Reports detailing bugs in a program are commonly known as **bug reports, fault reports, problem reports,** trouble reports, change requests, and so forth.

Error: (1 MARK)

- **The word error** has different meanings and usages relative to how it is conceptually applied.

- **The concrete meaning of the Latin word error is "wandering" or "straying".** To the contrary of **an illusion, an error or a mistake** can sometimes be dispelled through knowledge (knowing that one is looking at a mirage and not at real water doesn't make the mirage disappear).

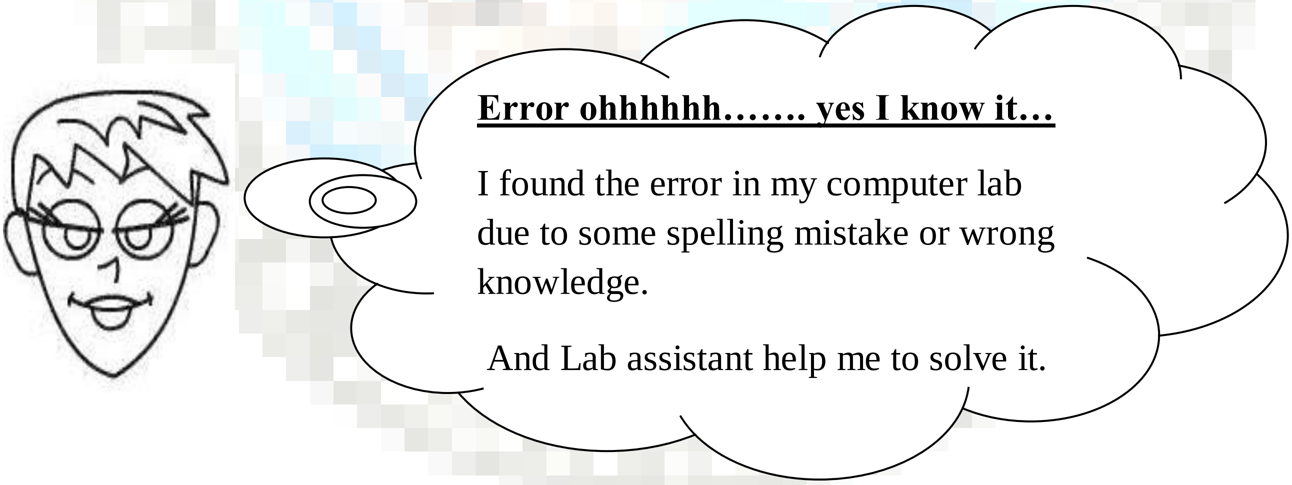
- However, some errors can occur even when individuals have the required knowledge to perform a task correctly.

- Examples include forgetting to collect your change after buying chocolate from a vending machine, forgetting the original document after making photocopies, and forgetting to turn the gas off after cooking a meal. These slip errors can occur when an individual is distracted by something else.

Defect: (1 MARK)

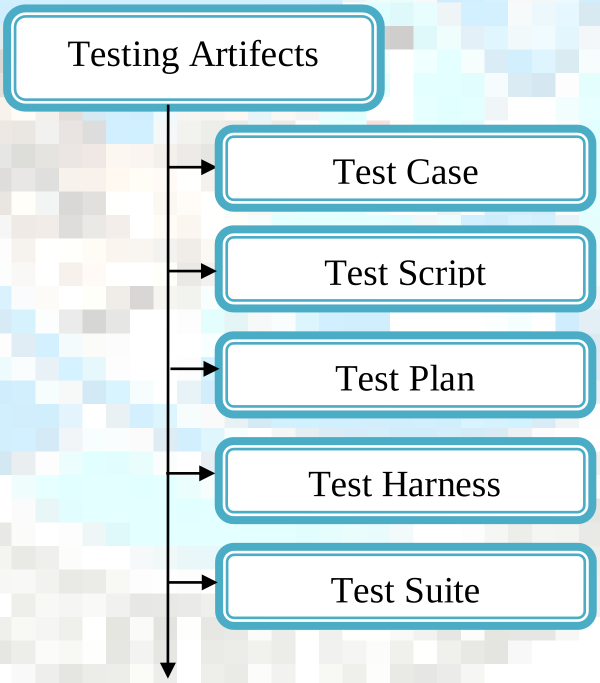
- **The defect** means the **failure of some angles** to add up to the expected amount of 360° or 180°, when such angles in the plane would.

- The opposite notion is the excess. The software product is in working but fail to perform its task properly.


Error ohhhhhh……. yes I know it…
I found the error in my computer lab due to some spelling mistake or wrong knowledge.
And Lab assistant help me to solve it.

NO.	QUESTION	ANSWER
1.	Bug	error, flaw, mistake, failure, or fault in a computer program
2.	The Latin word error is____.	"Wandering" or "straying".
3.	Defect	the failure of some angles

- **TOPIC:-** Different Testing places or Testing Artifacts.



Test case: (2 OR 3 MARKS)

- A test case is usually **a single step,** or sequence of steps, and its expected result, along with various additional **pieces of information.**
- It can occasionally be a series of steps but with one expected result or expected outcome.
- **A test case is a set of conditions by which a tester can determine whether an application or software system is working correctly.**
- The optional fields are a test case ID, Description, test step / order of execution number, related requirement(s), depth, test category, author, and check boxes for whether the test is automatable and has been automated.
- Larger test cases may also contain prerequisite states or steps.
- A test case should also contain a place for the actual result. These steps can be stored in a word processor document, spreadsheet, database or other common repository.
- **Test cases include (1) Formal Test Cases and (2) Informal Test Cases.**
- A formal written test-case includes some input and for find expected output, which is worked out before the test is executed.
- The known input should test a precondition and the expected output should test a post condition.
- Informal Test Cases works For applications or systems without formal requirements, test cases can be written based on the accepted normal operation of programs of a similar class.
- In some schools of testing, test cases are not written at all but the activities and results are reported after the tests have been run.

SUMMARY:-

- A test case is usually **a single step,** or sequence of steps, and its expected result, along with various additional **pieces of information.**

- **A test case is a set of conditions by which a tester can determine whether an application or software system is working correctly.**

- **Test cases include (1) Formal Test Cases and (2) Informal Test Cases.**

Test Script: (2 OR 3 MARKS)

- A test script in software testing is a **set of instructions that** will be performed on the system under test to test that the system functions as expected.

- These steps can be executed manually or automatically.

- There are two ways to execute test scripts.

  →Manual testing
  →Automated testing

- **Test cases are called manual testing.**

- **Test scripts are short programs that written in a programming language.** It is used to testing the part of the functionality of a software system.

- Test scripts written as short programs so it can either be written using a special automated functional **GUI test tool(Quick Test Professional) orin a well-known programming language(C,C++,PHP,JAVA)**

- **Advantages** of automated testing are:-

✓ Tests may be executed continuously.
✓ No continuous need of human interaction.
✓ It is easily repeatable.
✓ It is fast.

- **Disadvantages :-**

✓ The major disadvantage is that if the automated tests may be poorly written and can break during playback.

✓ Due to this disadvantage most systems are designed with human interaction in mind, it is good practice that a human tests the system at some point.

✓ **Automated tests can only examine what they have been programmed to examine.**

✓ A trained manual tester can notice that the system under test is misbehaving without being prompted or directed.

✓ Therefore, manual testers can find new bugs while ensuring that old bugs do not reappear while an automated test can only ensure the latter.

➢ **SUMMARY:-**

- A test script in software testing is a **set of instructions that** will be performed on the system under test to test that the system functions as expected. These steps can be executed manually or automatically.

- There are two ways to execute test scripts.

→**Manual testing**
→**Automated testing**

- **Test cases are called manual testing.**

Test Suite: (2 OR 3 MARKS)

- The most common term for a **collection of test cases is a testsuite.** The test suite often also contains more detailed instructions / goals for each collection of test cases.
- It definitely contains a section where the tester identifies the system configuration used during testing.
- **A group of test cases** may also contain prerequisite states or steps, and descriptions of the following tests. Collections of test cases are sometimes incorrectly termed a test plan.
- They may also be called a test script, or even a test scenario.
- In software development, a test suite, less **commonly known as a validation suite**, is a collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviors.
- **A test suite often contains detailed instructions** or goals for each collection of test cases and information on the system configuration to be used during testing. A group of test cases also contain required states or steps, and descriptions of the following tests.

■ Different types of test suites:

- Test suites are used to **group similar test cases together.**

- An executable test suite is executed by a program. This usually means that a test harness, which is integrated with the suite.
- The test suite and the test harness together can work on a sufficiently detailed level to correctly communicate with **the system under test (SUT**).

- A test suite for primarily testing subroutine might consist of a list number and testing subroutine. The testing subroutine would supply each number in the list to the primarily tester, and verify that the result of each test is correct.

➢ SUMMARY:-

- The most common term for a **collection of test cases is a test suite.** The test suite often also contains more detailed instructions / goals for each collection of test cases.

- **A group of test cases** may also contain prerequisite states or steps, and descriptions of the following tests.

- They may also be called a test script, or even a test scenario.

Test plan: (2 OR 3 MARKS):

- It is the approach that will be used to test the system, not the individualtests.

- Most companies that use automated testing will call the code that is used their test scripts.

- Test plan is one type of documents that insure that the software or product is as per specification and satisfied the customer.

- **A test plan is a systematic approach to testing a system such as a machine or software.** The plan typically contains a detailed understanding of what the eventual workflow will be.

- A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with proper inputs from Test Engineers.

- Depending on the product and the responsibility of the organization to which the test plan applies, **a test plan may include one or more of the following:**

  - ✓ **Design Verification test** – this testing is performed during the development or approval stages of the product or software.

  - ✓ **Manufacturing or Production test** – this testing is performed during preparation or assembly of the product in an ongoing manner for purposes of performance verification and quality control.

  - ✓ **Acceptance test –** this testing is performed at the time of delivery or installation of the product.

  - ✓ **Service and Repair test** – this is performed as per requirement during the life of the product.

  - ✓ **Regression test** – Here the platform is upgraded and the existing application will be run. The testing is performed on an existing product, to verify that existing functionality didn't get broken when other environment is changed.

- ✓ **<u>Note: -</u>** (Here remember the steps of SDLC –preliminary investigation,-system analysis,-system design,-coding,-testing,-installation. ---Now compare with Test plan testing points)

Just think that if you have any complex work then what you do???

→Yes your answer is right you are doing ***<u>planning.</u>***

Test Harness (harness means tools or equipment): <u>(2 OR 3 MARKS):</u>

- In software testing, a test harness or **automated test framework** is a **collection of software and test data.**
- Here **different equipments and tools are used to configured test programs and its units by running them under different conditions and monitoring its behavior and outputs.**
- It has two main parts: the Test execution engine and the Test script.
- **Test harnesses allow for the automation of tests.** They can call functions with supplied parameters and print out and compare the results to the desired value.
- The test harness is a hook to the developed code, which can be tested using an automation framework.
- A test harness should allow specific tests to run (this helps in optimizing), orchestrate a runtime environment, and provide a capability to analyze results.

- **Objectives of test harness:**
    - **Automation in the testing process.**
    - **Execute test suites of test cases.**
    - **Generate associated test reports.**

- **Benefits of test harness:**

- Increased productivity due to automation of the testing process.

- Increased probability by regression testing.
  (Regression testing is one type of software testing that is used to find uncover Errors and new software bug.).

- Increased quality of software product.

- Ensure that all the software parts will be checked.

- No human interaction is needed continuously.

- Testing done with simulate, which is not possible in real time environment.

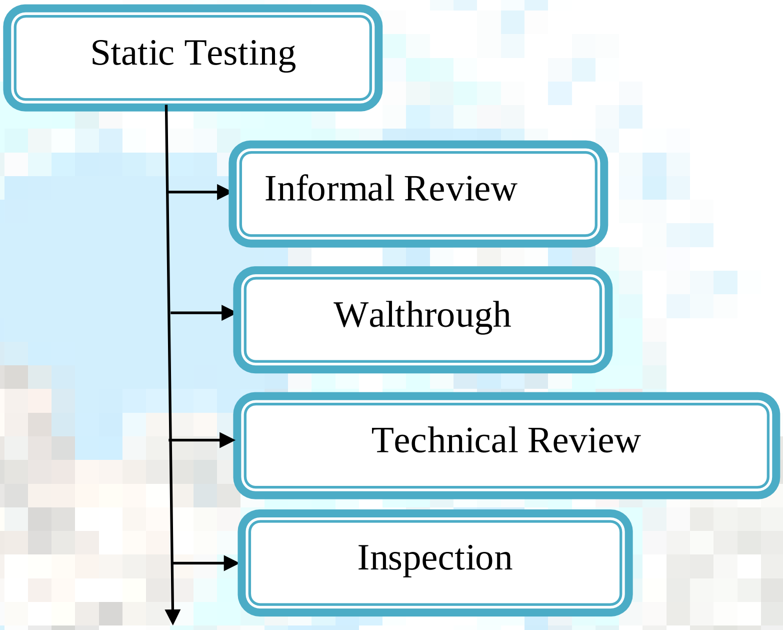
NO.	QUESTION	ANSWER
1.	Testing Artifects	Test Case, Test script, Test Suite, Test plan, Test Harness
2.	How Many Types of Test Cases?	Formal Test Cases, Informal Test Cases.
3.	Test Script Means?	set of instructions
4.	How Many ways Execute Test Scripts?	Manual testing, Automated testing

5.	Test Suites Means?	Collection of test cases is a test suite.
6.	Test Harness Means?	tools or equipment
7.	Objectives of test harness.	**Automation in the testing process, Execute test suites of test cases, Generate associated test reports.**

UNIT – 2(PART – 2)

TYPES OF SOFTWARE TESTING, VERIFICATION AND VALIDATION

➢ **TOPIC:-** Static testing: (3 OR 5 MARKS)



- Static testing is a **testing where the actual software is not used or tested.** It means that here the **software parts like coding, algorithms or documents are tested to find the errors.**

- It is done by software developers who actually wrote the code. The primary syntax checking and manually reviewing the coding testing is done by developers at the time of software development.

- **It is not detail testing because here the developers are involves in the testing and during the software development the testing is performed with primary basis.**

- It just checks basic coding, algorithm and document of software.

- **The main aim of this testing is to find errors at startup.** It is primarily syntax checking of the code and/or manually reviewing the code or document **to find errors.**

- This type of testing can be used by the developer who wrote the code.
  Static testing is a process of evaluating the software documents. This document describes static test techniques such as reviews, code analysis and static analysis

- It includes:-

  1. Code reviews (Technical review and informal review)
  2. Inspections
  3. Walkthroughs

- A **software review** is "A process or meeting during which a software product is [examined by] project personnel, managers, users, customers, user representatives, or other interested parties for comment or approval"

- **Formal reviews** greatly outperform **informal reviews** in cost-effectiveness. Informal reviews may often be unnecessarily expensive (because of time-wasting through lack of focus), and frequently provide a sense of security which is quite unjustified by the relatively small number of real defects found and repaired.

- **Technical review** is a form of peer review in which "a team of qualified personnel examines the suitability of the software product for its intended use and identifies discrepancies from specifications and standards.

- Technical reviews may also provide recommendations of alternatives and examination of various alternatives"

- In software engineering, a **walkthrough** or **walk-through** is a form of software peer review "in which **a designer or programmer leads members of the development team a**nd other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems“

- In general, a walkthrough has one or two broad objectives: to gain feedback about the technical quality or content of the document; and/or to familiarize the audience with the content.

- **A walkthrough** is normally organized and directed by the author of the technical document. Any combination of interested or technically qualified personnel (from within or outside the project) may be included as seems appropriate.

- **Inspection** in software engineering refers to peer review of any work product by trained individuals who look for defects using a well defined process.

- (Hey just remember that in your exam hall some experts appointed for inspection to stop any copy case).

- **An inspection is one of the most common sorts of review practices found in software projects. The goal of the inspection is for all of the inspectors** to reach consensus on a work product and approve it for use in the project.

- Commonly inspected work products include software requirements specifications and test plans. In an inspection, a work product is selected for review and a team is gathered for an inspection meeting to review the work product.

- **The goal of the inspection is to identify defects**. In an inspection, a defect is any part of the work product that will keep an inspector from approving it.

- The stages in **the inspections process are: Planning, Overview meeting, Preparation, Inspection meeting, Rework and Follow-up. The Preparation, Inspection meeting and Rework stages might be iterated.**

- **SUMMARY: -**

  - Static testing is a **testing where the actual software is not used or tested.** It means that here the **software parts like coding, algorithms or documents are tested to find the errors.**

  - **It is not detail testing.** It just checks basic coding, algorithm and document of software.

    - It includes:- **Code reviews (Technical review and informal review)**
    - **Inspections**
    - **Walkthroughs**

  - **Reviews : -**

    - **Software reviews**
    - **Formal reviews**
    - **Technical reviews**
    - **Walkthrough**
    - **Inspection**

NO.	QUESTION	ANSWER
1.	Static testing	Testing where the actual software is not used
2.	How many Static testing?	**1.Code reviews** **2.Inspections** **3.Walkthroughs**

- **TOPIC: - Dynamic** testing (3 OR 5 MARKS).

  - **Dynamic testingis also known as dynamic analysis.** It is a term used in software engineering to describe the testing **of the dynamic behavior of code.**

  - That is, **dynamic analysis refers to the examination of the physical response from the system to variables that are not constant and change with time.**

  - In dynamic testing the software must be compiled and run. Dynamic Testing involves working with the software, giving input values, dummy data and checking if the output is as per expected or not.

  - Dynamic testing means testing based on specific test cases by execution of the test object or running programs.

  - Dynamic testing is involves during Unit Tests, Integration Tests, System Tests and Acceptance Tests.

- The difference between Static testing and dynamic testing is that in dynamic testing tester is used to test software through executing it. This is in contrast to Static testing.

Difference between Static testing and Dynamic testing: (3 OR 5 MARKS)

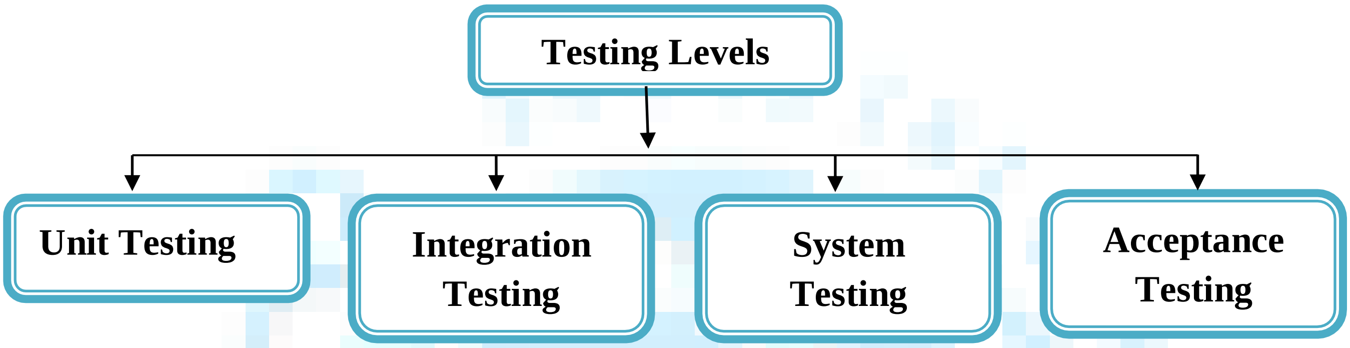
No	**Static testing**	**Dynamic testing**
1	Static testing done without executing the program.	Dynamic testing done by executing the program.
2	Static testing does verification process.	Dynamic testing does validation process.
3	Static testing is about prevention of defects.	Dynamic testing is about finding and fixing the defects.
4	Static testing gives assessment of code and documentation.	Dynamic testing involves test cases for execution.
5	Static testing involves checklist and process to be followed.	Dynamic testing involves test cases for execution.
6	This testing can be performed before compilation.	This testing is performed after compilation.
7	This testing covers the structural and statement coverage testing.	This testing covers the executable file of the code.
8	Cost of finding defects and fixing is less.	Cost of finding and fixing defects is high.
9	Return on investment will be high as	Return on investment will be low as this

	this process involved at early stage.	process involves after the development phase.
10	More reviews comments are highly recommended for good quality.	More defects are highly recommended for good quality.
11	Requires loads of meetings.	Comparatively requires lesser meetings.

- **SUMMARY : -**
  - **Dynamic testingis also known as dynamic analysis.**
  - Testing **of the dynamic behavior of code.**
  - Dynamic Testing involves working with the software, giving input values,
  - Dummy data and checking if the output is as per expected or not.

NO.	QUESTION	ANSWER
1.	Dynamic testing	Dynamic testingis also known as dynamic analysis. It is a term used in software engineering to describe the testing of the dynamic behavior of code.

- **TOPIC:-** Testing Levels: (2 OR 3 MARKS)



- **Unit testing - Testing of individual software components or modules.** Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code.

- May require developing test driver modules or test harnesses.

- This type of testing is **performed by the developers before the setup is handed over to the testing team to formally execute the test cases.**

- Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team.

- The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

- **Limitations of Unit Testing**
  - Testing cannot catch each and every bug in an application. It is impossible to evaluate every execution path in every software application. The same is the case with unit testing.
  - There is a limit to the number of scenarios and test data that the developer can use to verify the source code. So after he has exhausted all options there is no choice but to stop unit testing and merge the code segment with other units.
- **Integration testing** - Testing of integrated modules to verify combined functionality after integration.

- **Modules are typically code modules, individual applications, client and server applications on a network; etc.This type of testing is especially relevant to client/server and distributed systems.**

- The testing of combined parts of an application to determine if they function correctly together is Integration testing. There are two methods of doing Integration Testing Bottom-up Integration testing and Top down Integration testing.

Integration Testing Method

➢ **Bottom-up integration**

- This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

➢ Top-Down integration:

- This testing, **the highest-level modules are tested first and progressively lower-level modules are tested after that**.

- In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

- The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic those it will encounter in customers' computers, systems and network

- **System testing tests** a completely integrated system to verify that it meets its requirements.

- This is the next level in the testing and tests the system as a whole. **Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.**

- This type of testing is performed by a specialized testing team.

- System testing is so important because of the following reasons:

  - System Testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

- The application is tested thoroughly to verify that it meets the functional and technical specifications.

- The application is tested in an environment which is very close to the production environment where the application will be deployed.
- System Testing enables us to test, verify and validate both the business requirements as well as the Applications Architecture.

- **System integration testing** verifies that a system is integrated to any external or third party systems defined in the system requirements.

- **Regression testing -** Testing the application as a whole for the modification in any module or functionality.

- Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

- Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change.

- To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing.

- The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application.

- **Regression testing is so important because of the following reasons:**

- Minimize the gaps in testing when an application with changes made has to be tested.

- Testing the new changes to verify that the change made did not affect any other area of the application.

- Mitigates Risks when regression testing is performed on the application.

- Test coverage is increased without compromising timelines.

- Increase speed to market the product.

Acceptance testing

- Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer does this testing to determine whether to accept application.

- This is arguably the most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirements.

- The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.

- More ideas will be shared about the application and more tests can be performed on it to gauge its accuracy and the reasons why the project was initiated.

- Acceptance tests are not only intended to point out simple spelling mistakes, cosmetic errors or Interface gaps, but also to point out any bugs in the application that will result in system crashers or major errors in the application.
- By performing acceptance tests on an application the testing team will deduce how the application will perform in production. There are also legal and contractual requirements for acceptance of the system.

- **Alpha testing** is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site.

- This test is the first stage of testing and will be performed amongst the teams (developer and QA teams). Unit testing, integration testing and system testing when combined are known as alpha testing**.**

- **During this phase, the following will be tested in the application:**
  - **Spelling Mistakes**
  - **Broken Links**
  - **Cloudy Directions**

- The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

- **Beta testing comes after alpha testing.** Versions of the software, known as beta versions, are released to a limited audience outside of the programming team.

- The software is released to groups of people so that further testing can ensure the product has few faults or bugs.

- This test is **performed after Alpha testing has been successfully performed.** In beta testing a sample of the intended audience tests the application.

- Beta testing is also known as pre-release testing. Beta test versions of software are ideally distributed to a wide audience on the Web, partly to give the program **a "real-world" test** and partly to provide a preview of the next release. In this phase the audience will be testing the following:
    - Users will install, run the application and send their feedback to the project team.
    - Typographical errors, confusing application flow, and even crashes.
    - Getting the feedback, the project team can fix the problems before releasing the software to the actual users.
    - The more issues you fix that solve real user problems, the higher the quality of your application will be.
    - Having a higher-quality application when you release to the general public will increase customer satisfaction.
    -

- **SUMMARY (TESTING LEVEL): -**

- UNIT TESTING:-

- **Testing of individual software components or modules.**
- **Performed by the developers before the setup is handed over to the testing team to formally execute the test cases.**
- The goal of unit testing is to isolate each part of the program.

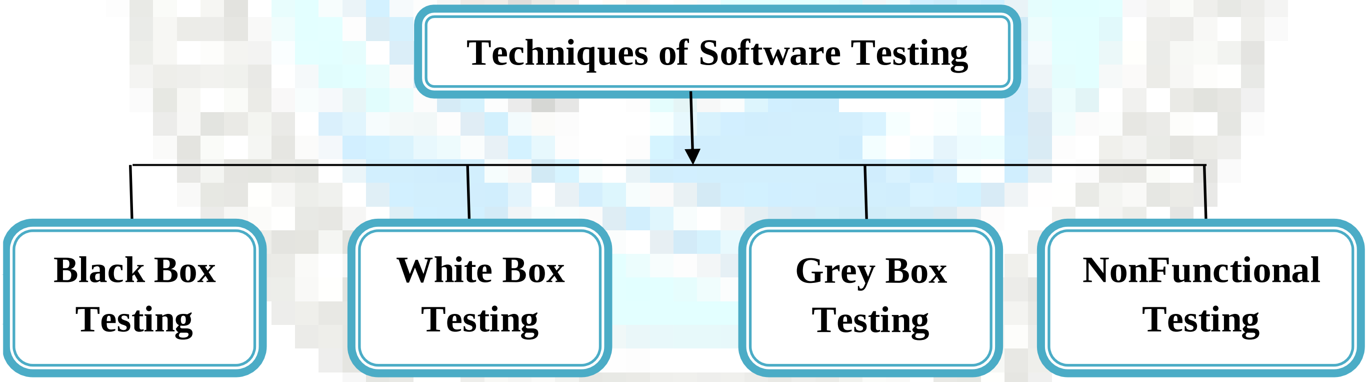
➢ INTEGRATION TESTING:-

- Testing of integrated modules to verify combined functionality after integration.
- Modules are:- Code module,individual applications, client and server applications on a network
- Bottom-up integration:-
  This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.
- Top-down integration:-
  - This testing, the highest-level modules are tested first and progressively lower-level modules are tested after that.
- **Regression testing:-**
- Testing the application as a whole for the modification in any module or functionality.
- **Acceptance testing:-**

- This type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application.

NO.	QUESTION	ANSWER
1.	Unit Testing	Testing of individual software components or modules
2.	**Integration testing**	**Testing of integrated modules to verify combined functionality after integration**

- **TOPIC:-** Techniques of Software Testing.



- Black Box Testing

(Equivalence Partitioning, Boundary Data Analysis, Decision Table Testing, State Transition Testing)

- White Box Testing
  (Statement Testing and Coverage, Decision Testing and Coverage)

- Grey Box Testing

- NonFunctional Testing
  (Performance Testing, Stress Testing, Load Testing, Usability Testing, Security Testing)

<u>Black box testing: (blind testing,Behavioral Testing)</u>

- **The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.**

- The tester is oblivious to the system architecture and does not have access to the source code.

- Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

- **Black box testing tests all possible combinations of end-user actions.** Black box testing **assumes no knowledge of code and is intended to simulate the end-user experience.**

- You can use sample applications to integrate and test the application block for black box testing.

- You can begin planning for black box testing immediately after the requirements and the functional specifications are available.
- **Black box testing treats the software as a "black box"—without any knowledge of internal implementation.**
- Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

Specification-based testing:

- Specification-based testing aims to test the functionality of software according to the applicable requirements.
- Thus, the tester **inputs data into, and only sees the output from**, the test object.
- This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.
- Specification-based testing is necessary, but it is insufficient to guard against certain risks.
- **Advantages and disadvantages**:
- The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs.

- Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not.

- But, on the other hand, black box testing has been said to be **"like a walk in a darkwithout a flashlight**," because the tester doesn't know how the software being tested was actually constructed.

- As a result, there are situations when

(1) A tester writes many test cases to check something that could have been tested by only one test case, and/or

(2) Some parts of the back-end are not tested at all.Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

➢ **Black Box Testing Steps:**

- Black box testing involves testing external interfaces to ensure that the code meets functional and nonfunctional requirements. The various steps involved in black box testing are the following:

1. **Create test plans.** Create prioritized test plans for black box testing.

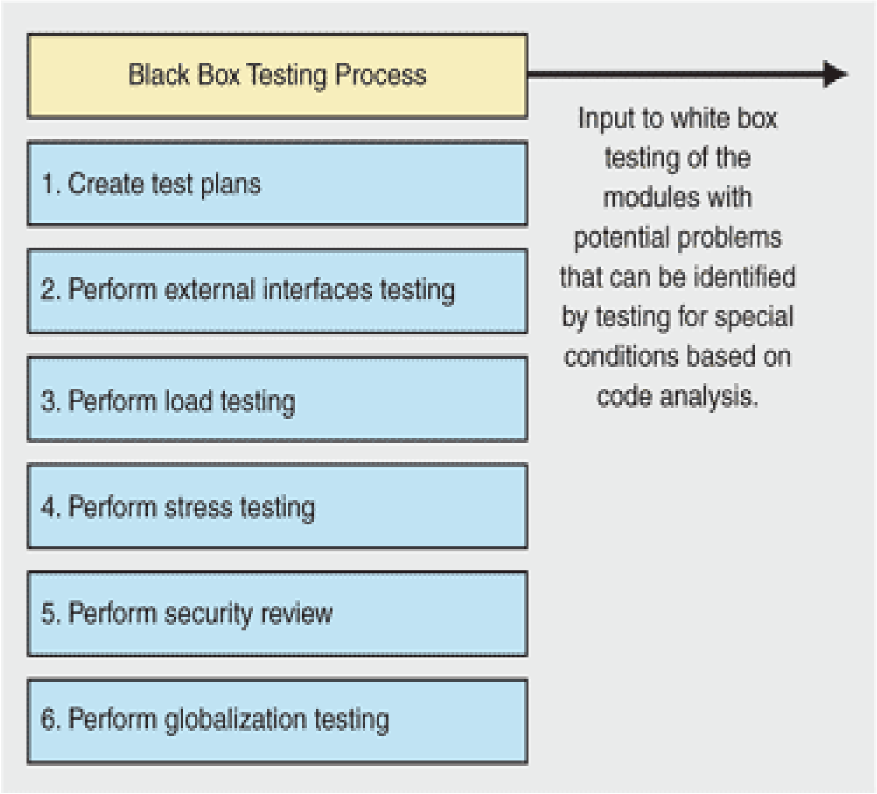
2. **Test the external interfaces.** Test the external interfaces for various types of inputs using automated test suites, such as NUnit suites and custom prototype applications.

3. **Perform load testing.** Load test the application block to analyze the behavior at various load levels. This ensures that it meets all performance objectives that are stated as requirements.

4. **Perform stress testing.** Stress test the application block to analyze various bottlenecks and to identify any issues visible only under extreme load conditions, such as race conditions and contentions.

5. **Perform security testing.** Test for possible threats in deployment scenarios. Deploy the application block in a simulated target environment and try to hack the application by exploiting any possible weakness of the application block.

6. **Perform globalization testing.** Execute test cases to ensure that the application block can be integrated with applications targeted toward locales other than the default locale used for development.


Black Box Testing Process
1. Create test plans
2. Perform external interfaces testing
3. Perform load testing
4. Perform stress testing
5. Perform security review
6. Perform globalization testing
Input to white box testing of the modules with potential problems that can be identified by testing for special conditions based on code analysis.

Advantage and Disadvantage of Black box testing

Advantages	Disadvantages
-Well suited and efficient for large code segments. -Code Access not required. -Clearly separates user's perspective from the developer's perspective through visibly defined roles. -Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language or operating systems.	-Limited Coverage since only a selected number of test scenarios are actually performed. -Inefficient testing, due to the fact that the tester only has limited knowledge about an application. -Blind Coverage, since the tester cannot target specific code segments or error prone areas. -The test cases are difficult to design.

White box testing:

- **White box testing is also known as Structural Testing, clear box testing, open box testing or Logic-driven Testing or Glass Box Testing.**

- White box testing is the detailed investigation of internal logic and structure of the code. **White box testing is also called glass testing or open box testing.**

- In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.

- **The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.**

- In white box testing, you create test cases by looking at the code to detect any potential failure scenarios.

- You determine the suitable input data for testing various APIs and the special code paths that need to be tested by analyzing the source code for the application block.

- Therefore, the test plans need to be updated before starting white box testing and only after a stable build of the code is available.

- A failure of a white box test may result in a change that requires all black boxes testing to be repeated and white box testing paths to be reviewed and possibly changed.

- **White box testing** is **when the tester has access to the internal data structures and algorithms including the code that implement these**.

✓ **<u>Types of white box testing</u>**

■ The following types of white box testing exist:

- **API** testing (application programming interface) - testing of the application using public and private APIs.

- **Code coverage** - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

- **Fault injection** methods - improving the coverage of a test by introducing faults to test code paths

- **Static testing** - White box testing includes all static testing.

- **Test coverage**White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods.

- This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

- Two common forms of code coverage are:

- **Function coverage**, which reports on functions executed.

- **Statement coverage**, which reports on the number of lines executed to complete the test.

- They both return code coverage metric, measured as a percentage.

- White box testing involves the following steps:

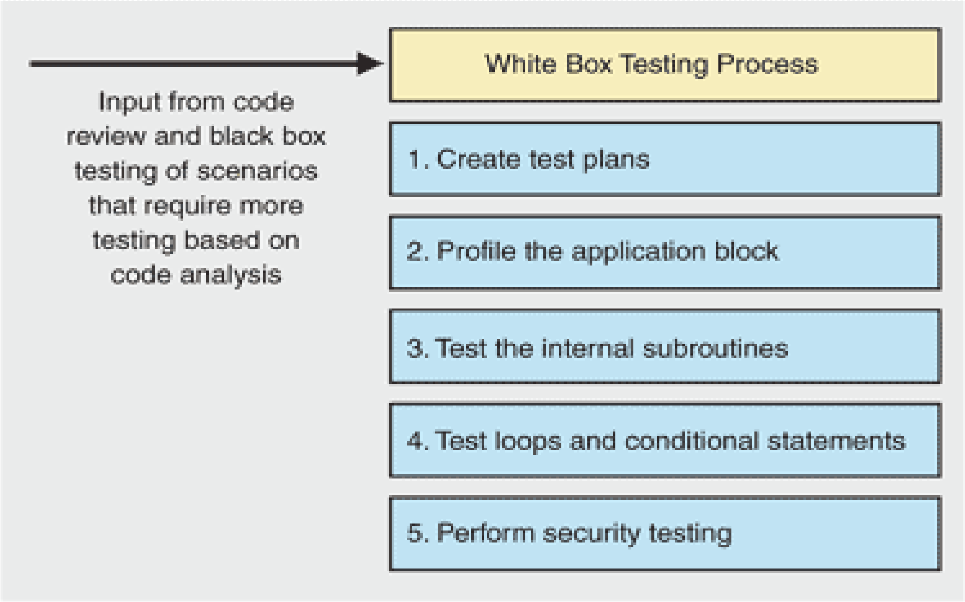
1. **Create test plans.** Identify all white box test scenarios and prioritize them.

2. **Profile the application block.** This step involves studying the code at run time to understand the resource utilization, time spent by various methods and operations, areas in code that are not accessed, and so on.

3. **Test the internal subroutines.** This step ensures that the subroutines or the nonpublic interfaces can handle all types of data appropriately.

4. **Test loops and conditional statements.** This step focuses on testing the loops and conditional statements for accuracy and efficiency for different data inputs.

5. **Perform security testing.** White box security testing helps you understand possible security loopholes by looking at the way the code handles security.


White Box Testing Process
Input from code review and black box testing of scenarios that require more testing based on code analysis
1. Create test plans
2. Profile the application block
3. Test the internal subroutines
4. Test loops and conditional statements
5. Perform security testing

- **Source Code is available for Testing**

  1) Structural Testing process
  2) Program Logic-driven Testing
  3) Design-based Testing
  4) Examines the internal structure of program

- **White box testing techniques**

  1) Basis path testing
  2) Complexity testing

- **White box testing is**

  1) Derive test cases:
  2) Based on program structure.
  3) To ensure that all independent paths within a module of the program have been tested.

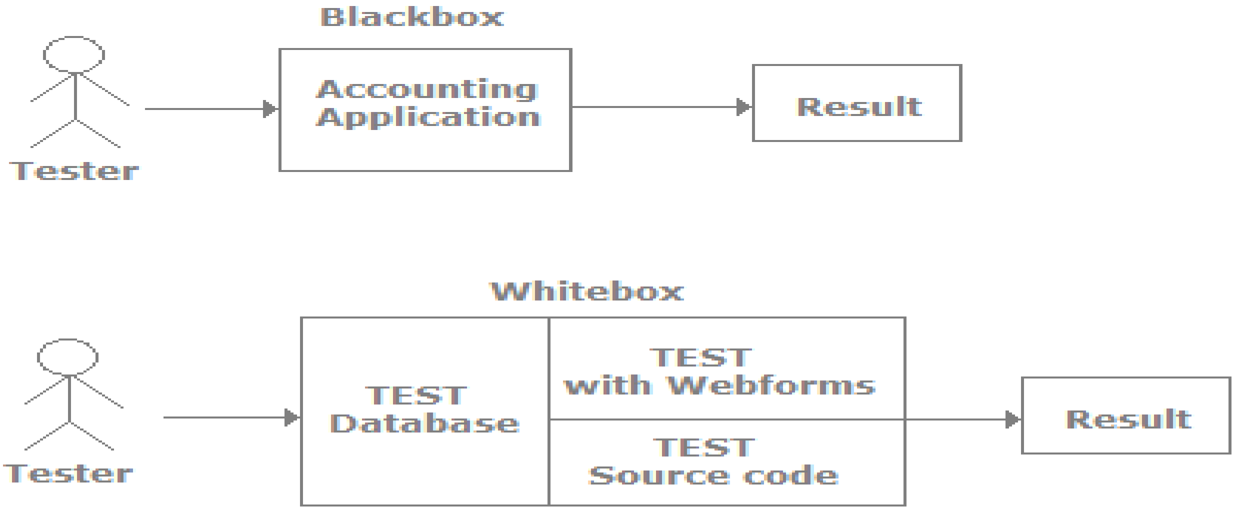
- **Maximum criteria for White Box Testing**

1) Code
2) Document High Level Design
3) Low level Design Document
4) Application Requirements Specification

Advantage and Disadvantage of White box testing:

Advantages	**Disadvantages**
- As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively. - It helps in optimizing the code. -Extra lines of code can be removed which can bring in hidden defects. - Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.	- Due to the fact that a skilled tester is needed to perform white box testing, the costs are increased. - Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems as many paths will go untested. - It is difficult to maintain white box testing as the use of specialized tools like code analyzers and debugging tools are required.

A simple diagram to understand Black box testing and White box testing.



- **Grey box testing:**

  - **Grey box testinginvolves having knowledge of internal data structures and algorithms for purposes of designing the test cases, but testing** at the user, or black-box level.

  - **Grey Box testing is a technique to test the application with limited knowledge of the internal workings of an application.**

  - In software testing, the term the more you know the better carries a lot of weight when testing an application.

  - Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge.

- Unlike black box testing, where the tester only tests the application's user interface, in grey box testing, the tester has access to design documents and the database.

- Having this knowledge, the tester is able to better prepare test data and test scenarios when making the test plan.

- Manipulating input data and formatting output do not qualify as grey box, because the input and output are clearly outside of the "black-box" that we are calling the system under test.

- This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test.

- However, modifying a data repository does qualify as grey box, as the user would not normally be able to change the **data outside of the system under test**.

- Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

➢ Advantage and Disadvantage of Grey box testing:

Advantages	**Disadvantages**
- Offers combined benefits of black box and white box testing wherever possible.	- Since the access to source code is not available, the ability to go over the code and test coverage is limited.

- Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications. - Based on the limited information available, a grey box tester can design excellent test scenarios especially around communication protocols and data type handling. - The test is done from the point of view of the user and not the designer.	- The tests can be redundant if the software designer has already run a test case. - Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.

- **Finally general idea about these three testing fundas.**

- **Black Box v/s Grey Box v/s White Box (2 OR 3 MARKS)**

S.N.	Black Box Testing	Grey Box Testing	White Box Testing
1	The Internal Workings of an application are not required to be known	Somewhat knowledge of the internal workings are known	Tester has full knowledge of the Internal workings of the application
2	Also known as closed box testing, data driven testing, behavior testing and functional testing	Another term for grey box testing is translucent testing as the tester has limited knowledge of the insides of the application	Also known as clear box testing, structural testing or code based testing

3	Performed by end users and also by testers and developers	Performed by end users and also by testers and developers	Normally done by testers and developers
4	Testing is based on external expectations - Internal behavior of the application is unknown	Testing is done on the basis of high level database diagrams and data flow diagrams	Internal workings are fully known and the tester can design test data accordingly
5	This is the least time consuming and exhaustive	Partly time consuming and exhaustive	The most exhaustive and time consuming type of testing
6	Not suited to algorithm testing	Not suited to algorithm testing	Suited for algorithm testing
7	This can only be done by trial and error method	Data domains and Internal boundaries can be tested, if known	Data domains and Internal boundaries can be better tested

- **Nonfunctional Testing: (3 MARKS)**

  - This section is based upon the **testing of the application from its non-functional attributes.** Non-functional testing of Software **involves testing the Software from the requirements which are non functional in nature related but important a well such as performance, security, user interface etc.**

  - Some of the important and commonly used non-functional testing types are mentioned as follows:

  - **Performance testing** - Term often used interchangeably with ’stress’ and ‘load’ testing.

- To check whether system meets performance requirements. Used different performance and load tools to do this.

- It is mostly used to identify any bottlenecks or performance issues rather than finding the bugs in software. There are different causes which contribute in lowering the performance of software:

  - **Network delay.**
  - **Client side processing.**
  - **Database transaction processing.**
  - **Load balancing between servers.**
  - **Data rendering.**

- Performance testing is considered as one of the important and mandatory testing type in terms of following aspects:

  - Speed (i.e. Response Time, data rendering and accessing)
  - Capacity
  - Stability
  - Scalability

- It can be either qualitative or quantitative testing activity and can be divided into different sub types such as Load testing and Stress testing.

- **Stress testing** - System is stressed beyond its specifications to check how and when it fails.

- **Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.**

- This testing type includes the testing of Software behavior under abnormal conditions. Taking away the resources, applying load beyond the actual load limit is Stress testing.

- The main intent is to test the Software by applying the load to the system and taking over the resources used by the Software to identify the breaking point. This testing can be performed by testing different scenarios such as:

  - Shutdown or restart of Network ports randomly.
  - Turning the database on or off.
  - Running different processes that consume resources such as CPU, Memory, server etc.

Load testing - It's a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.

- A process of testing the behavior of the Software by applying maximum load in terms of Software accessing and manipulating large input data.

- It can be done at both normal and peak load conditions. This type of testing identifies the maximum capacity of Software and its behavior at peak time.

- Most of the time, Load testing is performed with the help of automated tools such as Load Runner, AppLoader, IBM Rational Performance Tester, Apache JMeter, Silk Performer, Visual Studio Load Test etc.

- Virtual users (VUsers) are defined in the automated testing tool and the script is executed to verify the Load testing for the Software.

- The quantity of users can be increased or decreased concurrently or incrementally based upon the requirements.

- **Security testing** :-

- Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks.

- Security testing involves the testing of Software in order to identify any flaws ad gaps from security and vulnerability point of view. Following are the main aspects which Security testing should ensure:

  - Confidentiality.
  - Integrity.
  - Authentication.
  - Availability.

- Authorization.
- Non-repudiation.
- Software is secure against known and unknown vulnerabilities.
- Software data is secure.
- Software is according to all security regulations.
- Input checking and validation.
- SQL insertion attacks.
- Injection flaws.
- Session management issues.
- Cross-site scripting attacks.
- Buffer overflows vulnerabilities.
- Directory traversal attacks.

- **Usability testing** is needed to check if the user interface is easy to use and understand.

  - This section includes different concepts and definitions of Usability testing from Software point of view. It is a black box technique and is used to identify any error(s) and improvements in the Software by observing the users through their usage and operation.

SUMMARY (BLACKBOX TESTING): -

- Testing without having any knowledge of the interior workings of the application is Black Box testing.

- Tests all possible combinations of end-user actions.
- Assumes no knowledge of code and is intended to simulate the end-user experience.
- Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

- Black Box Testing Steps:-

1. Create test plans.
2. Test the external interfaces.
3. Perform load testing.
4. Perform stress testing.
5. Perform security testing.
6. Perform globalization testing.

- **SUMMARY (WHITEBOX TESTING): -**
  - Also known as Structural Testing, clear box testing, open box testing or Logic-driven Testing or Glass Box Testing.
  - White box testing is the detailed investigation of internal logic and structure of the code.
  - The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.
  - When the tester has access to the internal data structures and algorithms including the code that implement these.

  - White box testing steps:

1. Create test plans.
2. Profile the application block.
3. Test the internal subroutines.
4. Test loops and conditional statements.
5. Perform security testing.

- **<u>SUMMARY (GRAY TESTING): -</u>**

  - Having knowledge of internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level.

  - Technique to test the application with limited knowledge of the internal workings of an application.

  - This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test.

- **<u>SUMMARY (Nonfunctional TESTING): -</u>**

  - **Testing of the application from its non-functional attributes.**

  - **Involves testing the Software from the requirements which are non functional in nature related but important a well such as performance, security, user interface etc.**

  - Non-functional testing types are mentioned as follows:

    - **Performance testing**
      - **Stress testing**
      - **Load testing**
      - **Security testing**

- **Usability testing**

NO.	QUESTION	ANSWER
1.	Black box testing also Known as	blind testing, Behavioral Testing
2.	White box testing is also known as	Structural Testing, clear box testing, open box testing or Logic-driven Testing or Glass Box Testing
3.	White box testing techniques	Basis path testing,Complexity testing