

- Experiment 12 :-

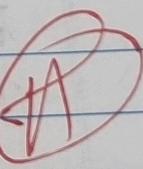
What is Testing.

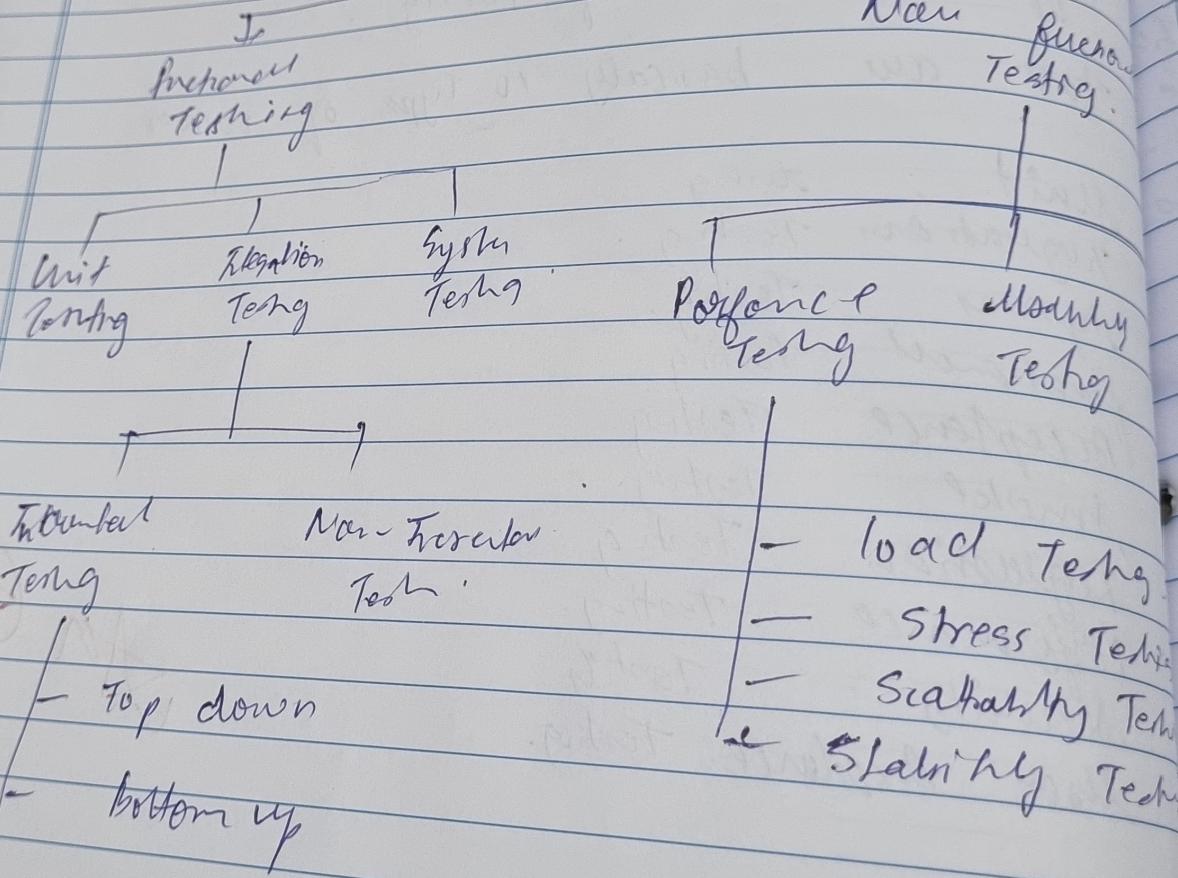
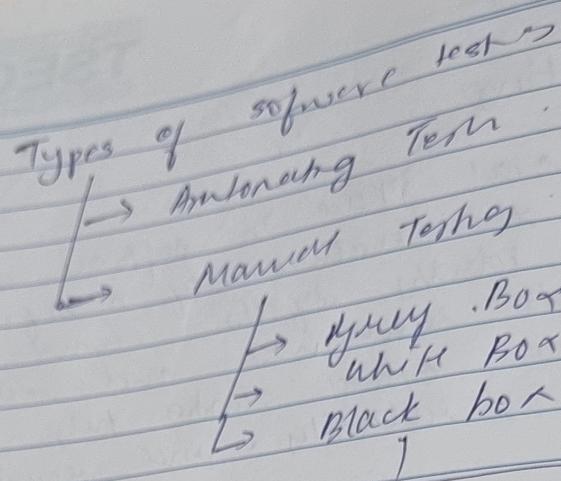
Testing is a crucial phase in software development that involves the evaluation of a software application to identify defects and ensure its functionality meets requirements and verify that it performs as expected. Testing helps improve software quality, reliability and security.

Different Types of Testing.

There are basically 10 types of tests:

- Unit testing.
- Integration Testing.
- System Testing.
- functional testing.
- Acceptance Testing.
- Smoke Testing.
- Regression Testing.
- Performance Testing.
- Security Testing.
- User Acceptance Testing.

✓ An 



Unit Testing: It involves testing individual code components or function in isolation, ensuring they produce the expected output. It is typically an automated and helps identify errors early in the development process.

Integration Testing: It checks how different components or modules interact when combined. It ensures the data flows smoothly between them and that they work together as a cohesive system.

System Testing: It evaluates the entire software as a whole. It assesses whether the individual components function correctly and meet specific requirements, focusing on the system behavior and functionality.

Functional Testing: Functional testing verifies the software's functions and features to ensure they operate according to predetermined requirements. Test cases are designed to validate that the software performs its intended tasks accurately.

Acceptance Testing: Acceptance testing is the final phase before deployment and involves end-user or stakeholder testing the software to determine if it meets their acceptance criteria. It verifies that the software is ready for production use.

Smoke Testing: it is a preliminary test to check software work without major issues. It quickly identifies showstopper defects.

Regression testing: it ensures that new code changes do not introduce defects or break existing functionality by re-running previously executed test cases often automated to save time.

Performance testing: evaluates the SW's speed, scalability, and stability under various conditions. This includes load testing, stress testing, and capacity testing to measure performance metrics.

Security Testing: - it assesses the SW's resistance to security threats and vulnerabilities. It includes penetration testing, vulnerability scanning, and authentication checks to protect data and functionality.

User Acceptance Testing: - UAT involves end-users or stakeholders validating that the SW aligns with their business needs and operates as expected in a real-world environment, ensuring it meets user expectations before deployment.

Explaining White Box Testing.

white Box testing, also known as structure or glass box testing, examines the internal structure code logic and paths within a software application. Testers use knowledge of the code to design test cases that exercise specific code paths. This testing method aims to ensure that all code branches, statements and conditions have been tested.

Key steps in White Box Tests:

Identify code paths: Analyze the source code to identify possible execution paths.

Design test cases: Create test cases that target specific code paths.

Execute test cases: Run the test cases and record the results.

Analyze results: Check if the code executes as expected and identify any discrepancies.

Repeat: Repeat the process until all code paths are tested and issue all unresolved.

That is -

Junit is widely used testing framework for java that provides a framework for writing & running test cases.

Cyclomatic complexity:

It is a new metric that quantifies the no. independent paths or decision points or decision points in a program controls flow or it helps assess code complexity. It uses test cases to cover all possible paths thus aiding in code quality analysis. One of the three ways to compute cyclomatic complexity.

$v(G)$ for a flowgraph, G , is defined as

$$v(G) = E - N + 2$$

where $E \rightarrow$ no. of edges in flowgraph
 $N \rightarrow$ no. of flowgraph nodes

$$\text{or } v(G) = p\beta$$

where β is no. of preimage nodes