UNIT-4

Testing-Unit testing-Black box testing-White box testing-Integration and System testing-Regression testing-Debugging-Program Analysis-Symbolic Execution-Model Checking-Case Study

1. TESTING:

Software Testing is a method to assess the functionality of the software program. The process checks whether the actual software matches the expected requirements and ensures the software is bug-free.

The purpose of software testing is to identify the errors, faults, or missing requirements in contrast to actual requirements. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

1. UNIT TESTING

Unit testing is a type of software testing that focuses on individual units or components of a software system.

The purpose of unit testing is to validate that each unit of the software works as intendedand meets the requirements.

Unit testing is typically performed by developers, and it is performed early in the development process before the code is integrated and tested as a whole system.

Unit tests are automated and are run each time the code is changed to ensure that new code does not break existing functionality. Unit testsare designed to validate the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system. This allows developers to quickly identify and fix any issues

Early in the development process, improving the overall quality of the software and reducing the time required for later testing.

Objective of Unit Testing:

The objective of Unit Testing is:

1. To isolate a section of code.
2. To verify the correctness of the code.
3. To test every function and procedure.
4. To fix bugs early in the development cycle and to save costs.
5. To help the developers understand the code base and enablethem to make changes quickly.
6. To help with code reuse.
7. To isolate a section of code.
8. To verify the correctness of the code.
9. To test every function and procedure.
10. To fix bugs early in the development cycle and to save costs.
11. To help the developers understand the code base and enablethem to make changes quickly.
12. To help with code reuse.

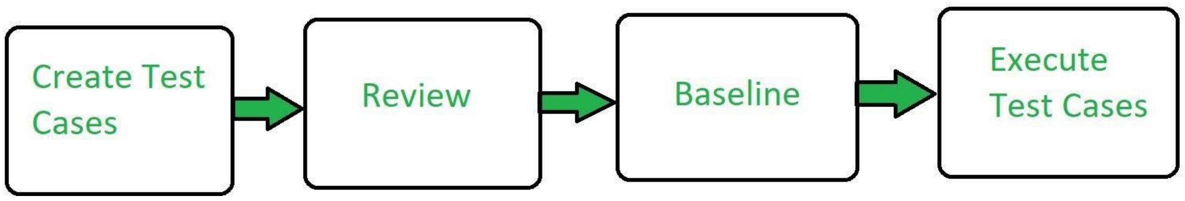
TYPES OF UNIT TESTING:

There are 2 types of Unit Testing:

1. **Manual**

2. Automated.

Workflow of Unit Testing:



Unit Testing Techniques:

There are 3 types of Unit Testing Techniques. They are

1. Black Box Testing: This testing technique is used in covering the unit tests for input, user interface, and outputparts.

2. White Box Testing: This technique is used in testing the functional behavior of the system by giving the input and checking the functionality output including the internal design structure and code of the modules.

3. Gray Box Testing: This technique is used in executing the relevant test cases, test methods, and test functions, and analyzing the code performance for the modules.

Advantages of Unit Testing:

1. Unit Testing allows developers to learn what functionality is provided by a unit and how to use it to gain a basic understanding of the unit API.
2. Unit testing allows the programmer to refine code and make sure the module works properly.
3. Unit testing enables testing parts of the project withoutwaiting for others to be completed.

Disadvantages of Unit Testing:

1. The process is time-consuming for writing the unit testcases.
2. Unit Testing will not cover all the errors in the module because there is a chance of having errors in the modules while doing integration testing.
3. Unit Testing is not efficient for checking the errors in the UI (User Interface) part of the module.
4. It requires more time for maintenance when the source codeis changed frequently.

BLACK BOX TESTING

Black-box testing is a type of software testing in which the tester is not concerned with the internal knowledge or implementation detailsof the software but rather focuses on validating the functionality based on the provided specifications or requirements.

Black box testing can be done in the following ways:

1. Syntax-Driven Testing – This type of testing is applied to systems that can be syntactically represented by some language. For example, language can be represented by context-free grammar. In this, the test cases are generated so that each grammar rule is used atleast once.

2. Equivalence partitioning – It is often seen that many types of inputs work similarly so instead of giving all of them separately we can group them and test only one input of each group. The idea is to partition the input domain of the system into several equivalence classes such that each member of the class works similarly, i.e., if a test case in one class results in some error, other members of the class would also result in the same error.

Types of Black Box Testing

There are many types of Black Box Testing but the following are the prominent ones –

- **Functional testing** – This black box testing type is related to the functional requirements of a system; it is done by software testers.
- **Non-functional testing** – This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.
- **Regression testing** – Regression Testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

Tools used for Black Box Testing:

Tools used for Black box testing largely depend on the type of black box testing you are doing.

- For Functional/ Regression Tests you can use – QTP, Selenium
- For Non-Functional Tests, you can use – LoadRunner, Jmeter

WHITE BOX TESTING

White box testing techniques analyze the internal structures the used data structures, internal design, code structure, and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

White Box Testing is also known as transparent testing or open box testing.

White box testing is a software testing technique that involves testing the internal structure and workings of a software application. The tester has access to the source code and uses this knowledge to design test cases that can verify the correctness of the software at thecode level.

White box testing is also known as structural testing or code-based testing, and it is used to test the software's internal logic, flow, and structure. The tester creates test cases to examine the code paths andlogic flows to ensure they meet the specified requirements.

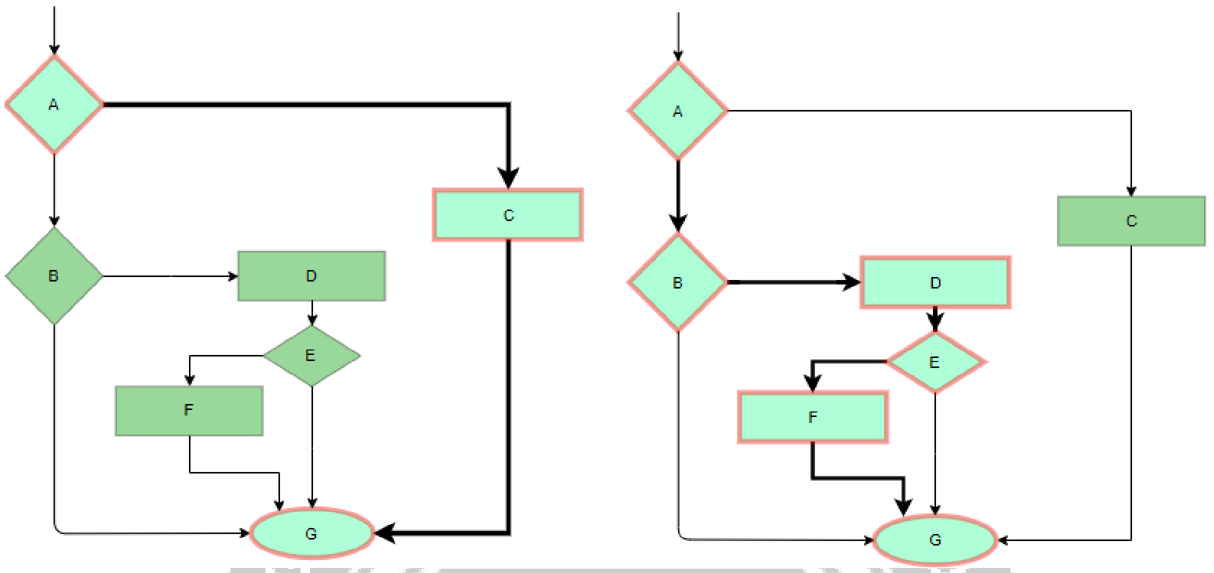
Process of White Box Testing

1. **Input:** Requirements, Functional specifications, design documents, source code.
2. **Processing:** Performing risk analysis to guide through theentire process.
3. **Proper test planning:** Designing test cases to cover the entire code. Execute rinse-repeat until error-free software isreached. Also, the results are communicated.
4. **Output:** Preparing final report of the entire testing process.

Testing Techniques

1. Statement Coverage

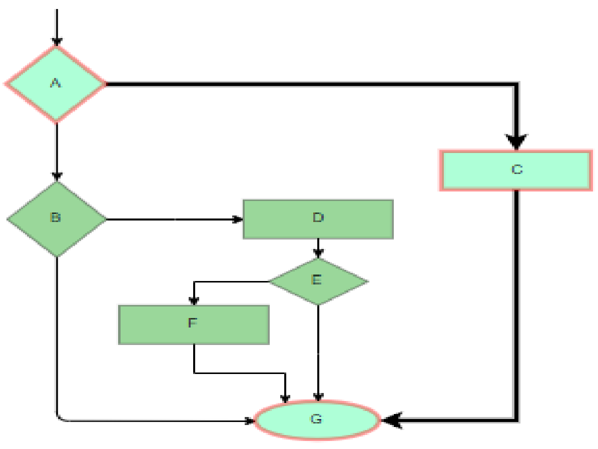
In this technique, the aim is to traverse all statements at least once. Hence, each line of code is tested. In the case of a flowchart, everynode must be traversed at least once. Since all lines of code are covered, it helps in pointing out faulty code.

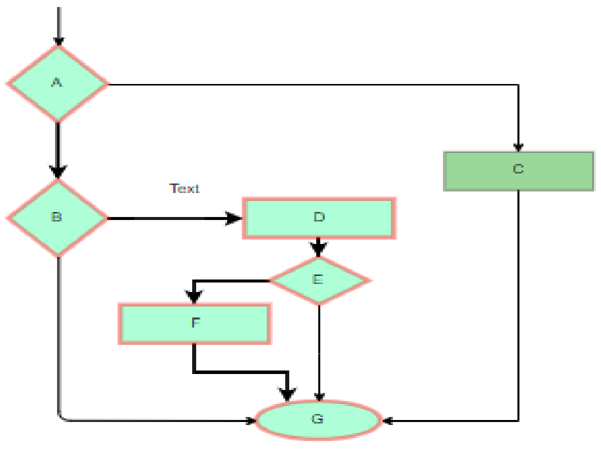


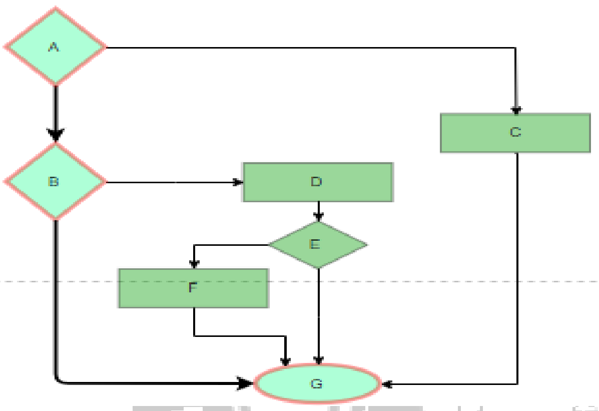
Statement Coverage Example

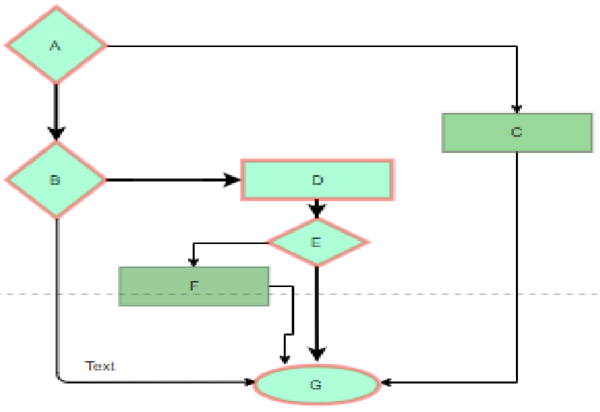
2. Branch Coverage:

In this technique, test cases are designed so that each branch from all decision points is traversed at least once. In a flowchart, all edges must be traversed at least once.


A
C
B
D
E
F
G


A
C
Text
B
D
E
F
G


A
C
B
D
E
F
G


A
C
B
D
E
F
Text
G

3.test cases are required such that all branches of all decisions are covered, i.e, all edges of the flowchart are covered

4. Condition Coverage

In this technique, all individual conditions must be covered as shown in the following example:

- READ X, Y
- IF(X == 0 || Y == 0)
- PRINT '0'
- #TC1 – X = 0, Y = 55
- #TC2 – X = 5, Y = 0

5. Multiple Condition Coverage

In this technique, all the possible combinations of the possible outcomes of conditions are tested at least once. Let's consider the following example:

- READ X, Y
- IF(X == 0 || Y == 0)
- PRINT '0'
- #TC1: X = 0, Y = 0
- #TC2: X = 0, Y = 5
- #TC3: X = 55, Y = 0
- #TC4: X = 55, Y = 5

6.Basis Path Testing

In this technique, control flow graphs are made from code or flowchart and then Cyclomatic complexity is calculated which defines the number of independent paths so that the minimal number of test cases can be designed for each independent path. **Steps:**

- Make the corresponding control flow graph
- Calculate the cyclomatic complexity
- Find the independent paths
- Design test cases corresponding to each independent path
- V(G) = P + 1, where P is the number of predicate nodes in the flow graph
- V(G) = E – N + 2, where E is the number of edges and N is the total number of nodes

- V(G) = Number of non-overlapping regions in the graph
- #P1: 1 – 2 – 4 – 7 – 8
- #P2: 1 – 2 – 3 – 5 – 7 – 8
- #P3: 1 – 2 – 3 – 6 – 7 – 8
- #P4: 1 – 2 – 4 – 7 – 1 – . . . – 7 – 8

3. Loop Testing

Loops are widely used and these are fundamental to many algorithms hence, their testing is very important. Errors often occur at the beginnings and ends of loops.

- **Simple loops:** For simple loops of size n, test cases are designed that:

1. Skip the loop entirely
2. Only one pass through the loop
3. 2 passes

4. m passes, where m < n
5. n-1 ans n+1 passes

- **Nested loops:** For nested loops, all the loops are set to their minimum count, and we start from the innermost loop. Simple loop tests are conducted for the innermost loop and this is worked outwards till all the loops have been tested.
- **Concatenated loops:** Independent loops, one after another. Simple loop tests are applied for each. If they're not independent, treat them like nesting.

White Testing is performed in 2 Steps

1. Tester should understand the code well
2. Tester should write some code for test cases and executethem

Tools required for White box testing:

- PyUnit
- Sqlmap
- Nmap
- Parasoft Jtest
- Nunit
- VeraUnit
- CppUnit
- Bugzilla
- Fiddler
- JSUnit.net
- OpenGrok
- Wireshark
- HP Fortify
- CSUnit

Advantages of Whitebox Testing

1. **Thorough Testing**: White box testing is thorough as the entire code and structures are tested.
2. **Code Optimization:** It results in the optimization of code

removing errors and helps in removing extra lines of code.

3. **Early Detection of Defects:** It can start at an earlier stage as it doesn't require any interface as in the case of black box testing.
4. **Integration with SDLC:** White box testing can be easily started in Software Development Life Cycle.
5. **Detection of Complex Defects:** Testers can identify defects that cannot be detected through other testing techniques.
6. **Comprehensive Test Cases:** Testers can create more comprehensive and effective test cases that cover all codepaths.
7. Testers can ensure that the code meets coding standards andis optimized for performance.

Disadvantages of White box Testing

1. **Programming Knowledge and Source Code Access:** Testers need to have programming knowledge and access to the source code to perform tests.
2. **Overemphasis on Internal Workings:** Testers may focus too much on the internal workings of the software and maymiss external issues.
3. **Bias in Testing:** Testers may have a biased view of the software since they are familiar with its internal workings.
4. **Test Case Overhead:** Redesigning code and rewriting code needs test cases to be written again.
5. **Dependency on Tester Expertise:** Testers are required to have in-depth knowledge of the code and programming language as opposed to black-box testing.
6. **Inability to Detect Missing Functionalities:** Missing functionalities cannot be detected as the code that exists istested.
7. **Increased Production Errors:** High chances of errors in production.

Differences between Black Box Testing vs White BoxTesting:

Black Box Testing	White Box Testing
It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it.	It is a way of testing the software in which the tester hasknowledge about the internal structure or the code or the program of the software.
Implementation of code is not needed for black box testing.	Code implementation is necessary for white box testing.
It is mostly done by software testers.	It is mostly done by software developers.
No knowledge of implementation is needed.	Knowledge of implementationis required.
It can be referred to as outer or external software testing.	It is the inner or the internal software testing.
It is a functional test of the software.	It is a structural test of the software.
This testing can be initiatedbased on the requirement specifications document.	This type of testing of softwareis started after a detail design document.
No knowledge of programmingis required.	It is mandatory to have knowledge of programming.
It is the behavior testing of the software.	It is the logic testing of the software.

Black Box Testing	White Box Testing
It is applicable to the higher levels of testing of software.	It is generally applicable to the lower levels of software testing.
It is also called closed testing.	It is also called as clear box testing.
It is least time consuming.	It is most time consuming.
It is not suitable or preferred for algorithm testing.	It is suitable for algorithm testing.
Can be done by trial and error ways and methods.	Data domains along with inner or internal boundaries can be better tested.
Example: Search something on google by using keywords	Example: By input to check and verify loops
Black-box test design techniques- • Decision table testing • All-pairs testing • Equivalence partitioning • Error guessing	White-box test design techniques- • Control flow testing • Data flow testing • Branch testing
Types of Black Box Testing: • Functional Testing • Non-functional testing • Regression Testing	Types of White Box Testing: • Path Testing • Loop Testing • Condition testing

Black Box Testing	White Box Testing
It is less exhaustive as compared to white box testing.	It is comparatively more exhaustive than black box testing.