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 the code 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 and logic flows to ensure they meet the specified requirements.

Process of White Box Testing:

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

Testing Techniques:

- Statement Coverage
- Branch Coverage
- Condition Coverage
- Multiple Condition Coverage
- Basis Path Testing
- Loop Testing

Features of White box Testing:

- **Code coverage analysis:** White box testing helps to analyze the code coverage of an application, which helps to identify the areas of the code that are not being tested.
- **Access to the source code:** White box testing requires access to the application's source code, which makes it possible to test individual functions, methods, and modules.
- **Knowledge of programming languages:** Testers performing white box testing must have knowledge of programming languages like Java, C++, Python, and PHP to understand the code structure and write tests.
- **Identifying logical errors:** White box testing helps to identify logical errors in the code, such as infinite loops or incorrect conditional statements.

- **Integration testing:** White box testing is useful for integration testing, as it allows testers to verify that the different components of an application are working together as expected.
- **Unit testing:** White box testing is also used for unit testing, which involves testing individual units of code to ensure that they are working correctly.
- **Optimization of code:** White box testing can help to optimize the code by identifying any performance issues, redundant code, or other areas that can be improved.
- **Security testing:** White box testing can also be used for security testing, as it allows testers to identify any vulnerabilities in the application's code.
- **Verification of Design:** It verifies that the software's internal design is implemented in accordance with the designated design documents.
- **Check for Accurate Code:** It verifies that the code operates in accordance with the guidelines and specifications.
- **Identifying Coding Mistakes:** It finds and fixes programming flaws in your code, including syntactic and logical errors.
- **Path Examination:** It ensures that each possible path of code execution is explored and tests various iterations of the code.
- **Determining the Dead Code:** It finds and removes any code that isn't used when the programme is running normally (dead code).

Advantages of Whitebox Testing

- **Thorough Testing:** White box testing is thorough as the entire code and structures are tested.
- **Code Optimization:** It results in the optimization of code removing errors and helps in removing extra lines of code.
- **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.
- **Integration with SDLC:** White box testing can be easily started in the Software Development Life Cycle.
- **Detection of Complex Defects:** Testers can identify defects that cannot be detected through other testing techniques.
- **Comprehensive Test Cases:** Testers can create more comprehensive and effective test cases that cover all code paths.
- **Testers** can ensure that the code meets coding standards and is optimized for performance.

Disadvantages of White box Testing:

- **Programming Knowledge and Source Code Access:** Testers need to have programming knowledge and access to the source code to perform tests.
- **Overemphasis on Internal Workings:** Testers may focus too much on the internal workings of the software and may miss external issues.
- **Bias in Testing:** Testers may have a biased view of the software since they are familiar with its internal workings.

- **Test Case Overhead:** Redesigning code and rewriting code needs test cases to be written again.
- **Dependency on Tester Expertise:** Testers are required to have in-depth knowledge of the code and programming language as opposed to black-box testing.
- **Inability to Detect Missing Functionalities:** Missing functionalities cannot be detected as the code that exists is tested.
- **Increased Production Errors:** High chances of errors in production.