**Black-Box Testing**

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while 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.

Advantages:

• Well suited and efficient for large code segments.

• Code access is not required.

• Clearly separates the 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.

**White-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, a tester needs to know the internal workings 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.

Advantages:

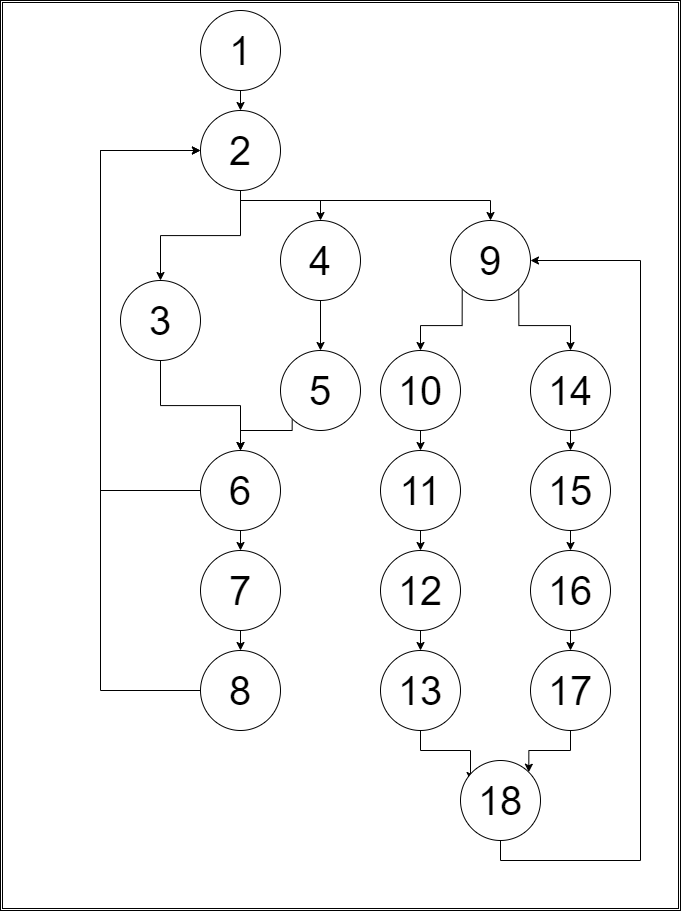
• As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.

• 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.

**Test Case**



**Control Flow Diagram**

**Cyclomatic Complexity**

Cyclomatic Complexity = E – N + 2

where E = Number of edges in the program flow graph and,

N = Number of nodes in the program flow graph.

Here, E = 22, N = 18

Cyclomatic Complexity = 22 – 18 + 2 = 6

Since the cyclomatic complexity is 6, therefore the number of independent paths will be 6.

**Independent Paths**

**Path A:** 1 > 2 > 3 > 6 > 2

**Path B:** 1 > 2 > 4 > 5 > 6 > 2

**Path C:** 1 > 2 > 3 > 6 > 7 > 8 > 2

**Path D:** 1 > 2 > 4 > 5 > 6 > 7 > 8 > 2

**Path E:** 1 > 2 > 9 > 10 > 11 > 12 > 13 > 18 > 9

**Path F:** 1 > 2 > 9 > 14 > 15 > 16 > 17 > 18 > 9