Module 4

Whitebox Testing

Index

- **Whitebox testing**
- **Control flow testing**
- **Data flow testing**

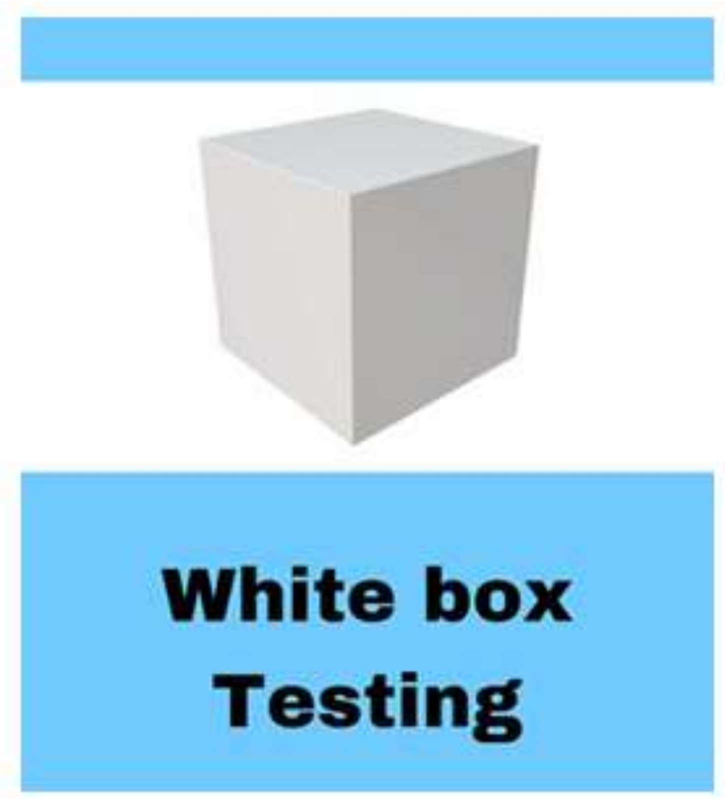
❖ Whitebox testing

- **White Box Testing** is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security
- In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing

➢ What do you verify in WhiteBox Testing?

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis



➢ How do you perform White Box Testing?

- Step 1) Understand the source code
- Step 2) Create test cases and Execute

Types of WhiteBox Testing

- **Unit Testing**: It is often the first type of testing done on an application. Unit Testing is performed on each unit or block of code as it is developed. Unit Testing is essentially done by the programmer.
- **Testing for Memory Leaks**: Memory leaks are leading causes of slower running applications. A QA specialist who is experienced at detecting memory leaks is essential in cases where you have a slow running software application.

- **White Box Penetration Testing**: In this testing, the tester/developer has full information of the application's source code, detailed network information, IP addresses involved and all server information the application runs on. The aim is to attack the code from several angles to expose security threats.
- **White Box Mutation Testing**: The idea behind mutation testing is to make a few arbitrary changes to a program at a time. Each time the program is changed, it is called a mutated program and the change effected is called a mutant.
- Mutation-based testing approach is that it is computationally very expensive, since a large number of possible mutants can be generated.
- Several test tools are available that automatically generate mutants for a given program

Advantages of WhiteBox Testing

- Code optimization by finding hidden errors
- White box tests cases can be easily automated
- Testing is more thorough as all code paths are usually covered

Disadvantages of WhiteBox Testing

- White box testing can be quite complex and expensive
- White box testing requires professional resources, with a detailed understanding of programming and implementation.
- White-box testing is time-consuming, bigger programming applications take the time to test fully

➢ WhiteBox Testing Techniques

- **Control flow testing**
- **Data flow testing**

1) Control flow testing

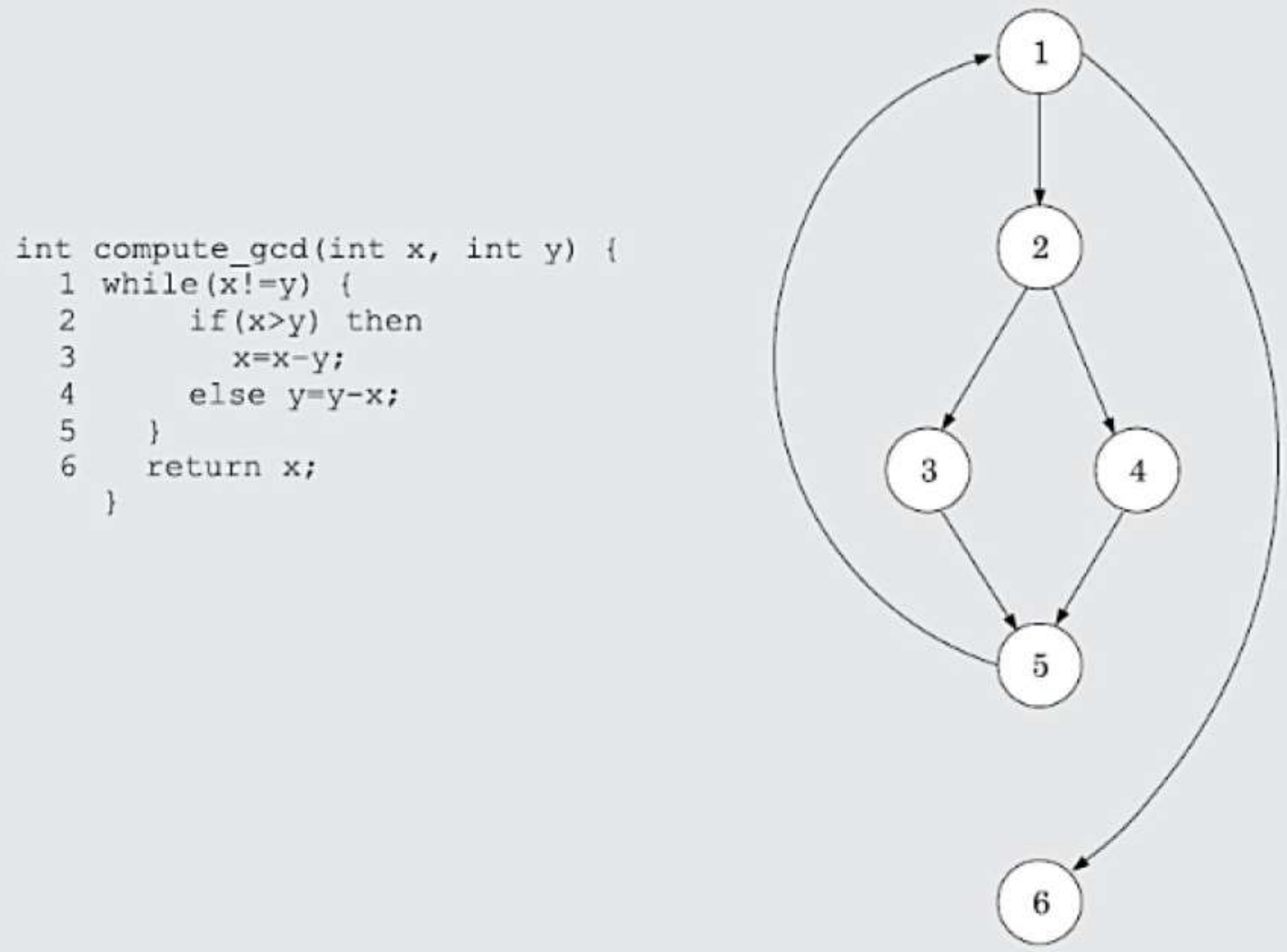
- Control flow testing is a testing technique that comes under white box testing
- The aim of this technique is to determine the execution order of statements or instructions of the program through a **control structure**
- The control structure of a program is used to develop a test case for the program
- In this technique, a particular part of a large program is selected by the tester to set the testing path

- It is mostly used in unit testing
- Test cases represented by the control graph of the program.
- **Control Flow Graph** is formed from the node, edge, decision node, junction node to specify all possible execution path

Notations used for Control Flow Graph

1) **Node** : Used to create a path of procedures. Basically, it represents the sequence of procedures
2) **Edge** : Used to link the direction of nodes
3) **Decision Node** : Used to decide next node of procedure as per the value
4) **Junction node** : Point where at least three links meet

E



```
int compute_gcd(int x, int y) {
  1 while(x!=y) {
  2       if(x>y) then
  3         x=x-y;
  4       else y=y-x;
  5     }
  6    return x;
    }
```

(a) An example program

(b) Control flow graph

- path coverage testing does not try to cover all paths, but only a subset of paths called linearly independent paths(basis paths).
- if each path in the set introduces at least one new edge that is not included in any other path in the set.
- **McCabe’s Cyclomatic Complexity Metric**
- McCabe’s cyclomatic complexity defines an upper bound on the number of independent paths in a program.
- Given a control flow graph G of a program, the cyclomatic complexity V(G) can be computed as: $V(G) = E – N + 2$
- where, N is the number of nodes of the control flow graph and E is the number of edges in the control flow graph.

2) Data flow testing

- Data Flow Testing is a specific strategy of software testing that focuses on data variables and their values
- It makes use of the control flow graph
- It can be considered as a type of white box testing and structural types of testing
- It keeps a check at the data receiving points by the variables and its usage points
- The process is conducted to detect the bugs because of the incorrect usage of data variables or data values

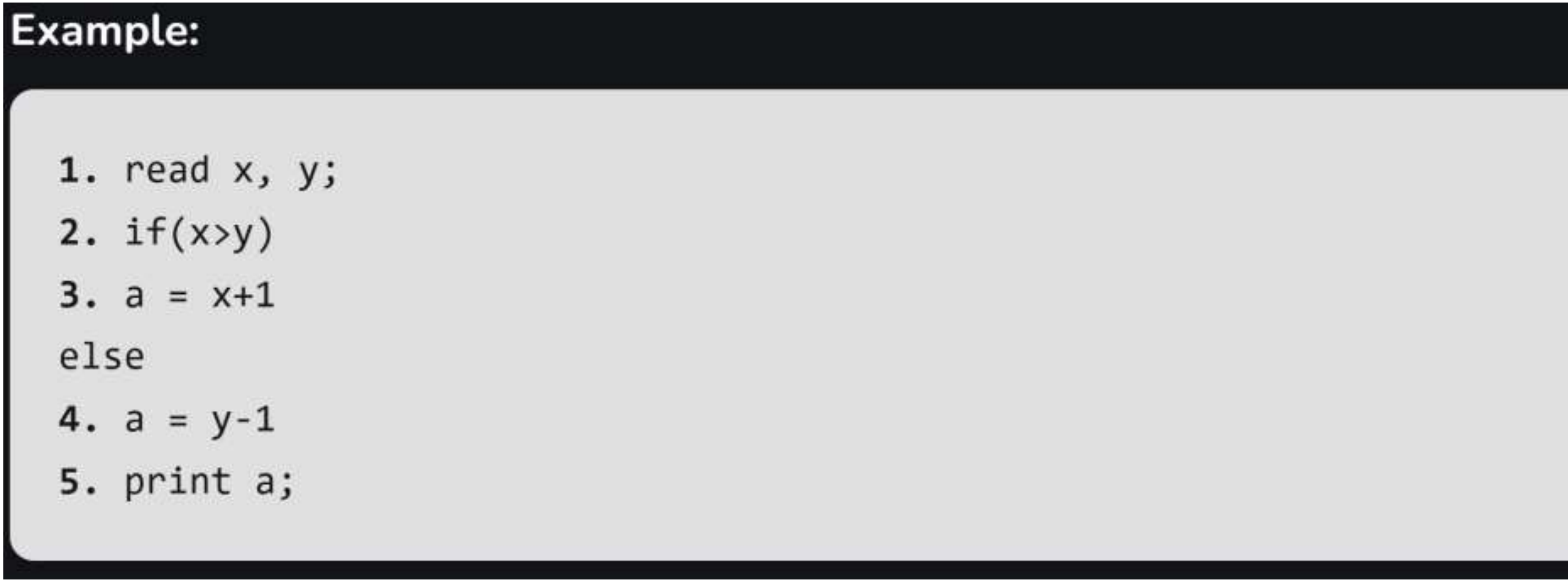
What is Data flow Testing?

- The programmer can perform numerous tests on data values and variables. This type of testing is referred to as data flow testing.
- It is performed at two abstract levels: static data flow testing and dynamic data flow testing.
- The static data flow testing process involves analyzing the source code without executing it.
- Static data flow testing exposes possible defects known as data flow anomaly.
- Dynamic data flow identifies program paths from source code.

2) Data flow testing

- Furthermore, it is concerned with:
  - Statements where variables receive values,
  - Statements where these values are used or referenced.
- To illustrate the approach of data flow testing, assume that each statement in the program is assigned a unique statement number. For a statement number S-
  - DEF(S) = {X | statement S contains the definition of X}
  - USE(S) = {X | statement S contains the use of X}

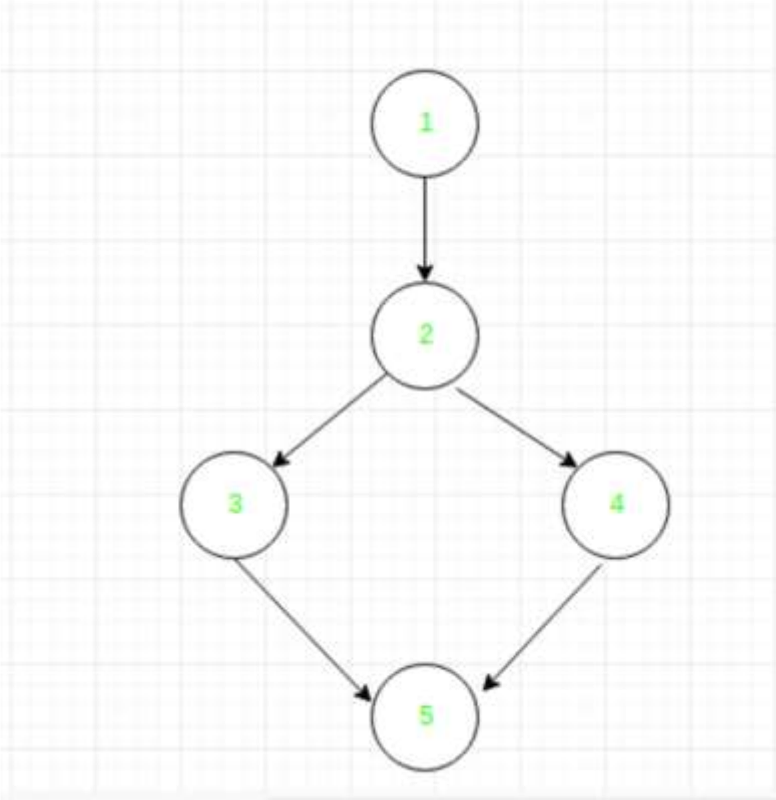
2) Data flow testing



Example:

```
1. read x, y;
2. if(x>y)
3. a = x+1
else
4. a = y-1
5. print a;
```

2) Data flow testing



Variable	Defined at node	Used at node
x	1	2, 3
y	1	2, 4
a	3, 4	5

Advantages of Data Flow Testing:

Data Flow Testing is used to find the following issues-

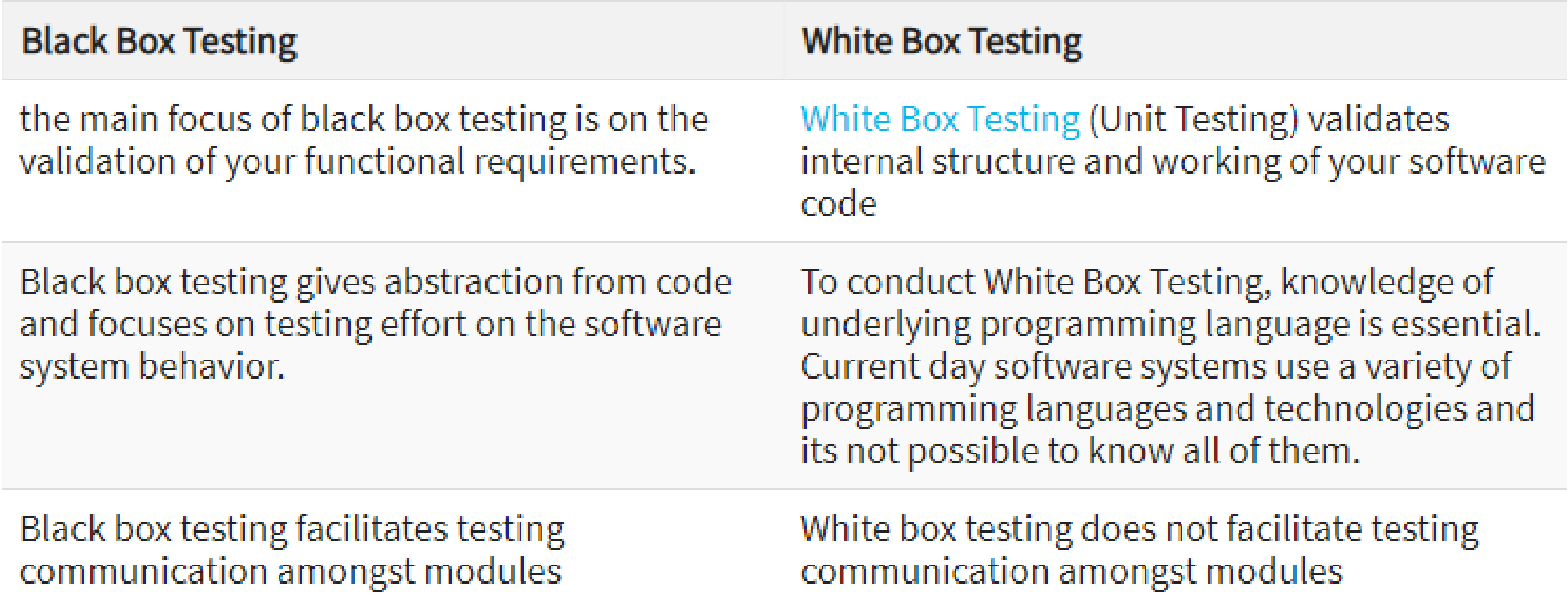
- To find a variable that is used but never defined,
- To find a variable that is defined but never used,
- To find a variable that is defined multiple times before it is use,
- Deallocating a variable before it is used.

Disadvantages of Data Flow Testing

- Time-consuming and costly process
- Requires knowledge of programming languages

❖ Comparison of Black Box and White Box Testing:

Which to choose ???



Black Box Testing	White Box Testing
the main focus of black box testing is on the validation of your functional requirements.	White Box Testing (Unit Testing) validates internal structure and working of your software code
Black box testing gives abstraction from code and focuses on testing effort on the software system behavior.	To conduct White Box Testing, knowledge of underlying programming language is essential. Current day software systems use a variety of programming languages and technologies and its not possible to know all of them.
Black box testing facilitates testing communication amongst modules	White box testing does not facilitate testing communication amongst modules