

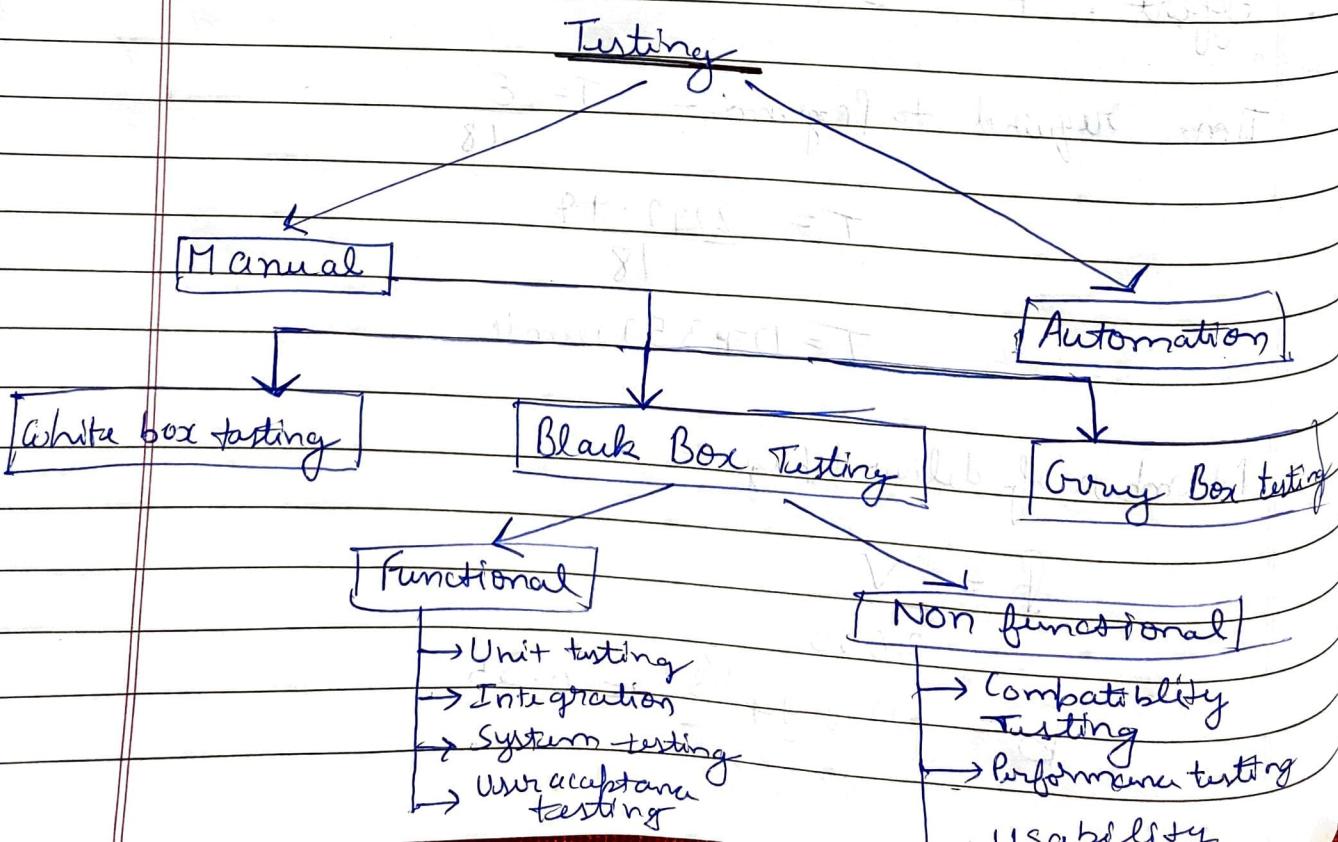
## UNIT-5

### Software Testing

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.

Testing is mandatory because it will be a dangerous situation if the software fails any of time due to lack of testing. So, without testing software cannot be deployed to end user.

### Type of Testing



Black Box Testing :- It is a technique of Software testing which examines the functionality of Software without peering into its internal Structure or Coding.

The primary source of black box testing is a Specification of Requirements that is stated by Customer.

Functional Testing :- It is a type of testing Software testing which is used to verify the functionality of Software application, whether the function is working according to the requirement specification.

In functional testing each function tested by giving the value, determining the output and verifying the actual output with the expected value.

functional testing performed as BBT which is presented to confirm ~~that~~ that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application.

Functional testing also called BBT because it focuses on application Specification rather than actual code. rather than actual code.

## Advantages of Functional Testing:-

- It focuses on the requirements of the ~~act~~ guidelines for end users to test the various models of application.
- It accepts realistic facts and figures to make testing reports.
- This type of testing is conducted when the project is very close to customer like QAs.
- It is a testing technique to reduce the gap b/w the business organization and their users.

## Disadvantages of functional testing:-

- It is a process in which various logical mistakes in software are not detected.
- The possibility of conducting redundant testing is high in functional testing.
- This testing didn't give the freedom of the Software to go live.

## White Box Testing

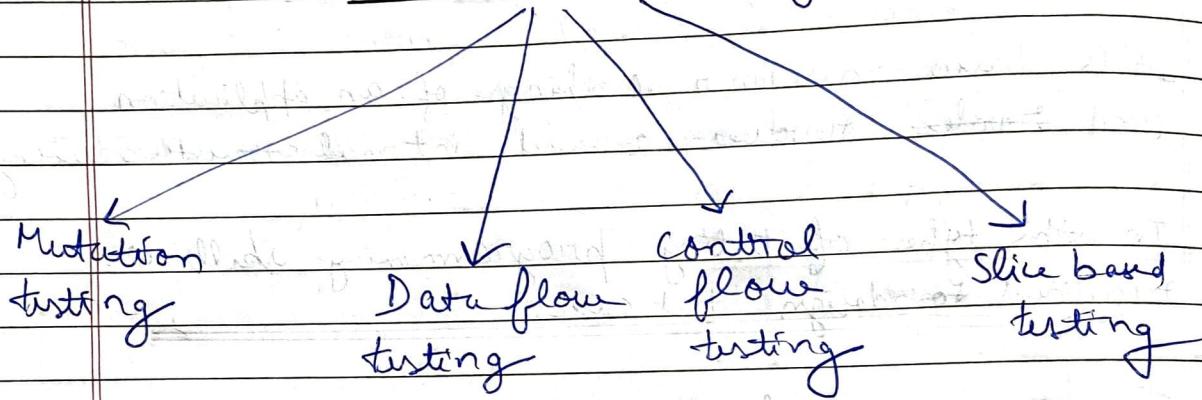
- It tests internal coding and infrastructure of a software focuses on checking of predefined inputs against expected and desired output.
- It is based on inner workings of an application and revolves around internal structure testing.
- In this type of testing programming skills are required to design test cases.
- The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of software.

## Structured testing

- It is used to test the internal design of software or structure of coding for the particular software.
- In this testing development team members are included in testing team to execute the software's internal design.
- Structural testing working is opposite to behavioral testing.
- It is also known as WBT.

- It requires in depth knowledge of programming language and is opposite to functional tests.

### Structural testing



### Advantages of Structural testing

- It is automated
- It is less time consuming
- All the early defects can easily be identified
- It removes dead code easily
- It provides easy coding and implementation
- It delivers detailed testing of Software

## Disadvantages of Structural testing

- In depth knowledge of programming language is required.
- It is expensive
- Even though structural testing is automated, it might not be very difficult because it involves training in the tool used for testing.

Q) Differentiate b/w Black Box testing and White Box testing?

Ans:- Black Box testing.

- It is a way of software testing in which the code is hidden.

- Mostly done by software tester.

- No knowledge of implementation is needed.

- It is functional test of software.

- It is behaviour testing.

- Least time consuming.

- Also called Closed testing.

White Box testing.

- It is a way of testing in which test has the knowledge about code.

- Mostly done by developers.

- Knowledge of implementation is ~~not~~ needed.

- It is structural test of software.

- It is logical testing.

- Most time consuming.

- Also called Open box testing.

## Path testing

- Path testing is a structural testing method that involves using the source code of a program in order to find every possible executable path.
- It helps to determine all the faults lying within a piece of code.

### Steps for basic Path-testing

- (1) Draw a control graph
- (2) Calculate cyclomatic complexity
- (3) Find a basic set of Paths
- (4) Generate test cases to exercise each path

## Boundary Value Analysis

Boundary Value Analysis is one of the widely used test design technique for black box testing. It is used to test boundary values because the input values near the boundary have higher chances of error.

Whenever we do the testing by boundary value analysis the tester focuses on, while entering boundary value whether the software is producing correct output or not.

Boundary values are those that contain the upper and lower limit of a variable. Assume that age is a variable of any function and its minimum value is 18 and the maximum value is 30, both 18 and 30 will be considered as boundary values.

There is 18 and 30 are the boundary values that's why tester plays more attention to these values. But this doesn't mean that the middle values like 19, 20, 21, 27, 29 are ignored. Test cases are developed for each and every value of the range.

Testing of boundary values is done by making valid and invalid partitions. Invalid partitions are tested because of output. An adverse condition is also essential.

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

12 13 14 15 16 17 18 19 20 22 24 26 28 30 31 32 34 36 38 40

Invalid Partition

Valid Partition

Invalid Partition

## Equivalence Class testing (Equivalence Class Partitioning)

It is a black box testing technique used in software testing as a major step in the SDLC. This testing technique is better than many of the testing techniques like boundary value analysis, ~~worst~~ worst case testing, robust case testing and many more in terms of time consumption and terms of precision of the test cases.

D) Given a date in the format of day-month-year, you need to find the next date for the given date. Perform boundary value analysis and equivalence class testing for this?

$$D: 1 \leq \text{Day} \leq 31$$

$$M: 1 \leq \text{Month} \leq 12$$

$$Y: 1800 \leq \text{Year} \leq 2048$$

Ay :- Boundary Value analysis :-

$$\begin{aligned} \text{no of test cases } (n = \text{no of variables}) &= 4n + 1 \\ &= 4 \times 3 + 1 \\ &= 13 \end{aligned}$$

$$\text{no of test cases} = 13$$

## Test Cases:

Test Case ID	Day	Month	Year	Expected Output
1	10	6	2000	2-6-2000
2	2	6	2000	3-6-2000
3	15	6	2000	16-6-2000
4	30	6	2000	1-7-2000
5	31	6	2000	Invalid Date
6	15	1	2000	16-1-2000
7	15	2	2000	16-2-2000
8	15	11	2000	16-11-2000
9	15	12	2000	16-12-2000
10	15	6	1800	16-6-1800
11	15	6	1801	16-6-1801
12	15	6	2047	16-6-2047
13	15	6	2048	16-6-2048

## Part Testing (Continued) .....

### Control Flow GRAPH

Every program is composed of

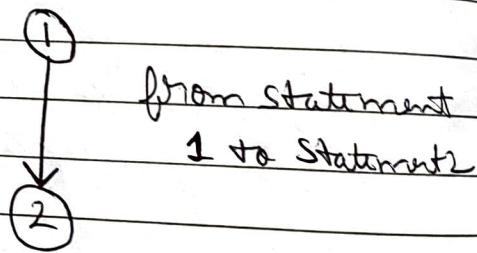
- Sequence
- Selection
- Iteration

If we see

#### 1 Sequence

1  $a = 5$

2  $b = a * b - 1;$



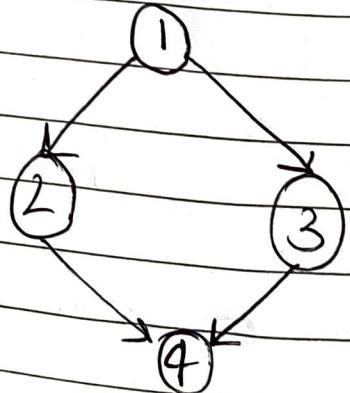
#### 2 Selection

1  $\text{if } (a > b) \text{ then}$

2  $C = 3;$

3  $\text{else } C = 5;$

4  $C = C * C;$



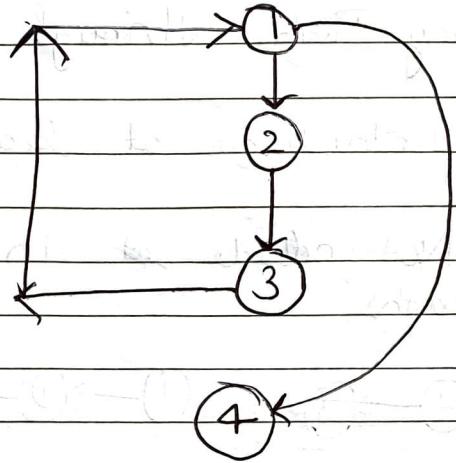
### 3 Iteration

1 While ( $a > b$ ) {

2      $b = b * a;$

3      $b = b - 1; }$

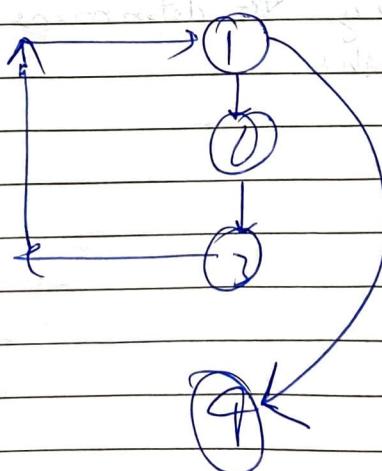
4      $c = b + d;$



Path :- A node and edge sequence from starting node to a terminal node of the control flow graph.

### All Path Criterion

In the presence of loops, the number paths can become extremely large.  
This makes all path testing impractical

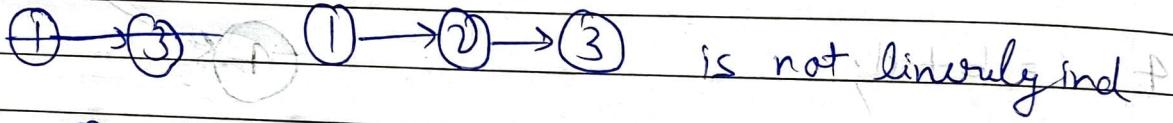


## Linearly Independent Path

Any Path through the program that:

- Introduces at least one new edge:

• Not include ~~in~~ in any other independent paths



① → ② → ③ → ④ is linearly independent

Independent Path

• It is straight forward:

- To identify linearly independent paths of simple programs.

• For complicated programs:

- It is not easy to determine the number of independent paths.

## McCabe's Cyclomatic Metric:

- An upper bound:

- for the number of linearly independent paths of program

- Provides a practical way of determining:

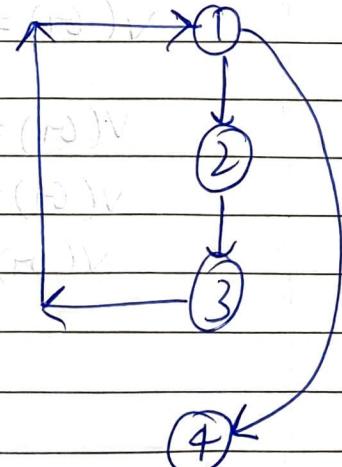
- The maximum number of tests cases required for basis path testing.

### Q How to Compute Cyclomatic Metric?

A Given a Control flow graph  $G_1$ , what is its Cyclomatic complexity  $V(G_1)$ ?

$$V(G_1) = E - N + 2$$

- $N$  is the number of nodes in  $G_1$ .
- $E$  is the no of edges in  $G_1$ .



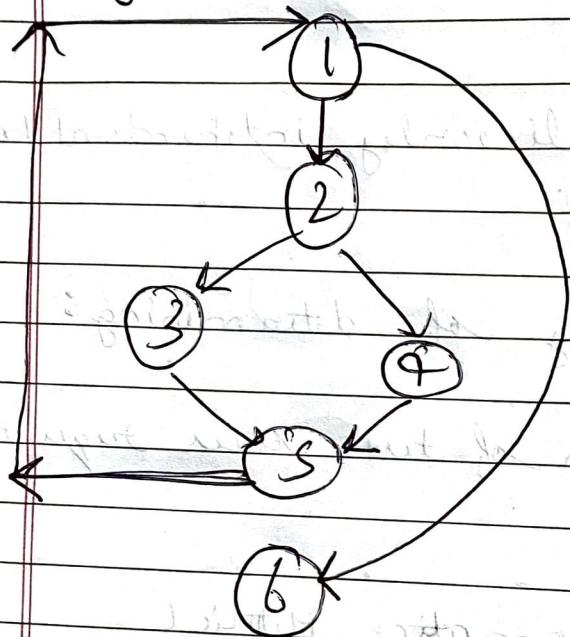
Total no of Nodes = 4

Total no of Edges = 4

$$V(G_1) = 4 - 4 + 2$$

$$V(G_1) \geq 2$$

Find Cyclomatic complexity in the given graph?



Ans:-

$$\text{Total no. of nodes } (N) = 6$$

$$\text{Total no. of Edges } (E) = 7$$

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

$$V(G) = 7 - 6 + 2$$

$$V(G) = 1 + 2$$

$$V(G) = 3$$

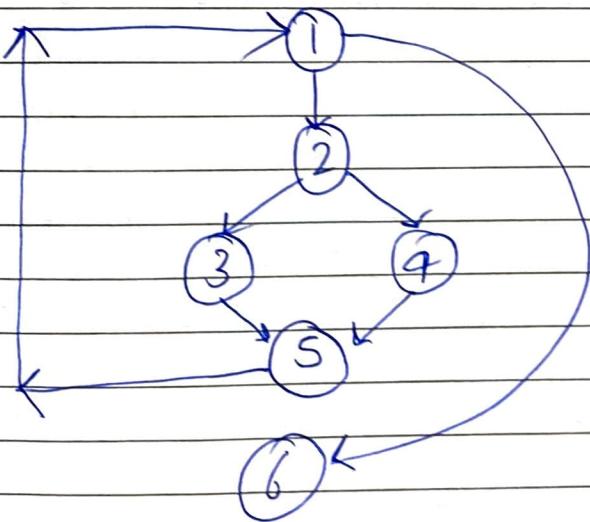
Q) Find Cyclomatic Complexity of given Code?

Int f1 (int x, int y){

```

1 while (x1 = y) {
2   if (x > 1) then
3     x = x - y;
4   else y = y - x;
5 }
6 return x; }
```

Soln First draw CFG



$$N = 6$$

$$E = 7$$

$$V(M) = E - N + 2$$

$$V(M) = 7 - 6 + 2$$

$$V(M) = 1 + 2$$

$$V(M) = 3$$