



Sri  
**SAI RAM**  
ENGINEERING COLLEGE  
INSTITUTE OF TECHNOLOGY  
West Tambaram, Chennai - 44

# **SAIRAM** DIGITAL RESOURCES



**24ITPC302**

**SOFTWARE ENGINEERING**

## **UNIT NO 4**

### **4.SOFTWARE TESTING**

#### **4.1 White Box Testing**

**INFORMATION TECHNOLOGY**



## 4.2 White box testing- basis path testing

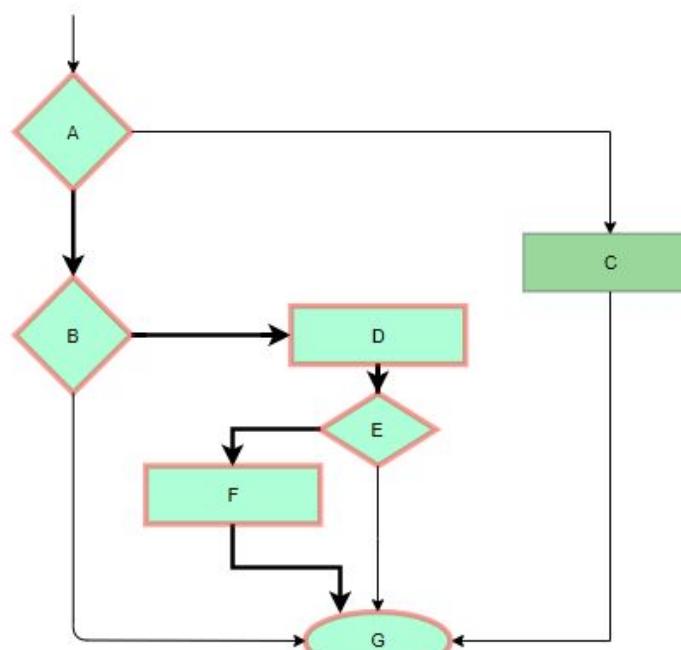
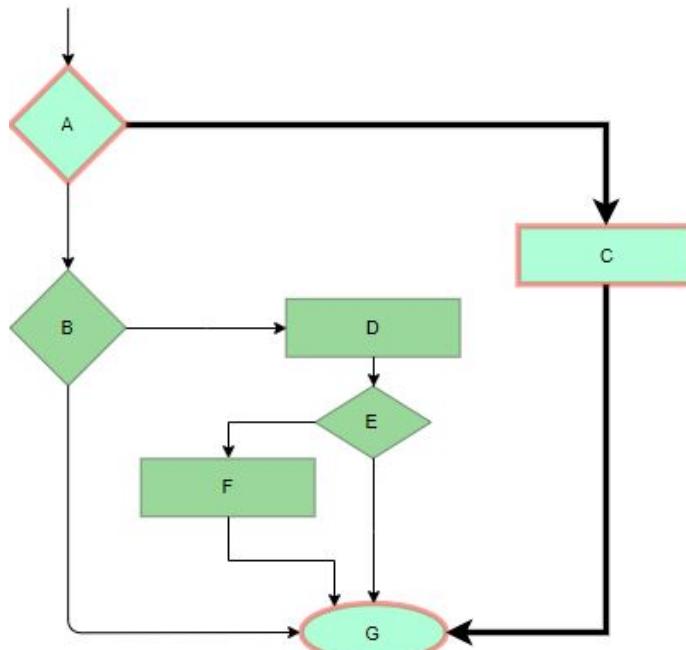
**White box testing** techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

- **Input:** Requirements, Functional specifications, design documents, source code.
- **Processing:** Performing risk analysis for guiding through the entire process.
- **Proper test planning:** Designing test cases so as to cover 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.

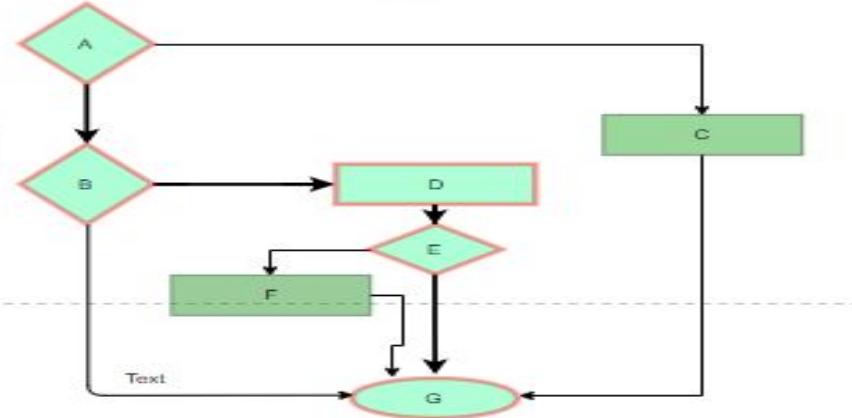
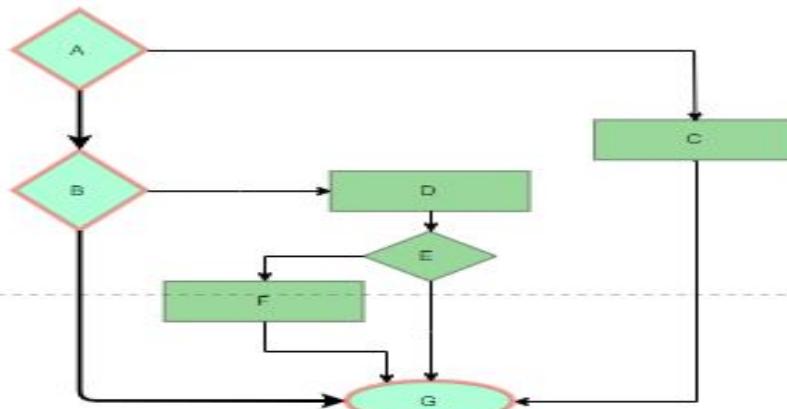
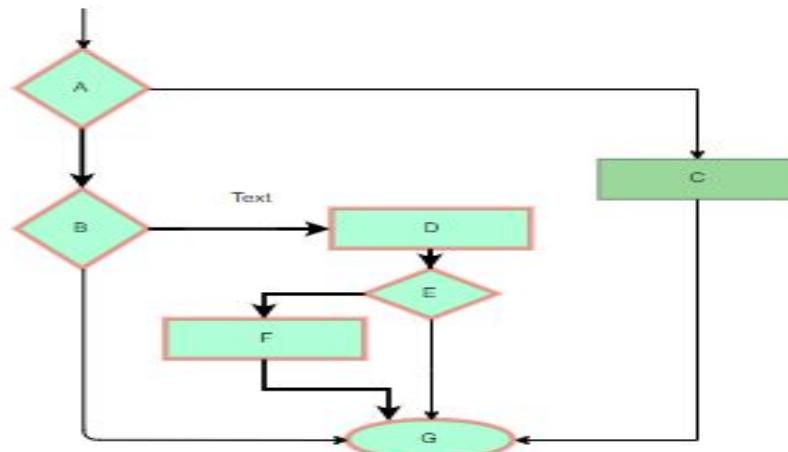
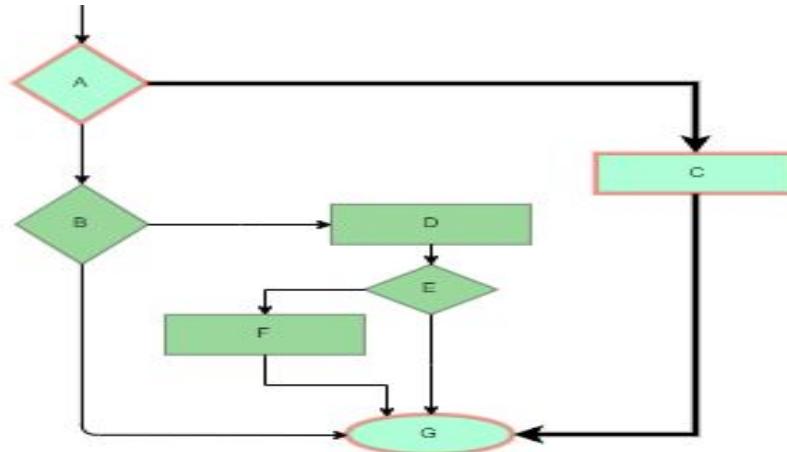


## Testing techniques:

- **Statement coverage:** In this technique, the aim is to traverse all statement at least once. Hence, each line of code is tested. In case of a flowchart, every node must be traversed at least once. Since all lines of code are covered, helps in pointing out faulty code.



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



- **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'
- In this example, there are 2 conditions: X == 0 and Y == 0. Now, test these conditions get TRUE and FALSE as their values. One possible example would be:
  - #TC1 – X = 0, Y = 55
  - #TC2 – X = 5, Y = 0



- **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

- Hence, four test cases required for two individual conditions.

Similarly, if there are n conditions then  $2^n$  test cases would be required.



**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.

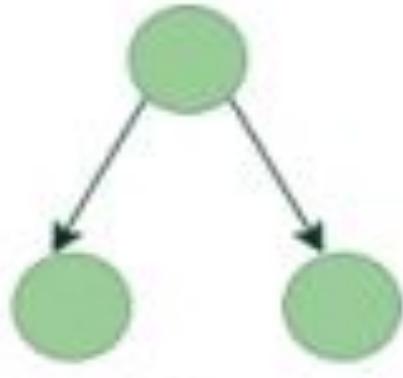
**Steps:**

1. Make the corresponding control flow graph
2. Calculate the cyclomatic complexity
3. Find the independent paths
4. Design test cases corresponding to each independent path

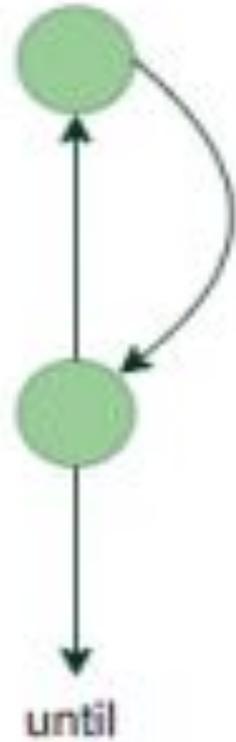
**Flow graph notation:** It is a directed graph consisting of nodes and edges. Each node represents a sequence of statements, or a decision point. A predicate node is the one that represents a decision point that contains a condition after which the graph splits. Regions are bounded by nodes and edges.



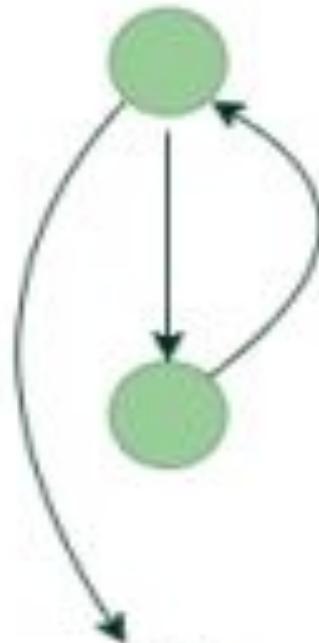
## Flow graph notation:



if-else



until



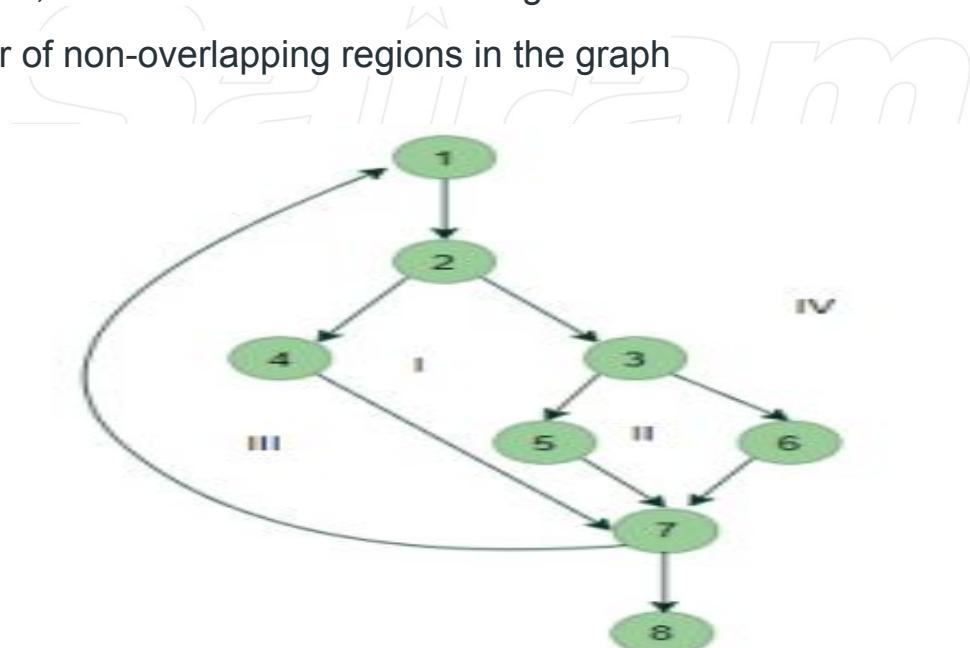
while



**Cyclomatic Complexity:** It is a measure of the logical complexity of the software and is used to define the number of independent paths. For a graph  $G$ ,  $V(G)$  is its cyclomatic complexity.

Calculating  $V(G)$ :

1.  $V(G) = P + 1$ , where  $P$  is the number of predicate nodes in the flow graph
2.  $V(G) = E - N + 2$ , where  $E$  is the number of edges and  $N$  is the total number of nodes
3.  $V(G) = \text{Number of non-overlapping regions in the graph}$



- **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.

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

- Skip the loop entirely
- Only one pass through the loop
- 2 passes
- m passes, where  $m < n$
- $n-1$  ans  $n+1$  passes

2. **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.
3. **Concatenated loops:** Independent loops, one after another. Simple loop tests are applied for each.

If they're not independent, treat them like nesting.



**Advantages:**

1. White box testing is very thorough as the entire code and structures are tested.
2. It results in the optimization of code removing error and helps in removing extra lines of code.
3. It can start at an earlier stage as it doesn't require any interface as in case of black box testing.
4. Easy to automate.

**Disadvantages:**

1. Main disadvantage is that it is very expensive.
2. Redesign of code and rewriting code needs test cases to be written again.
3. Testers are required to have in-depth knowledge of the code and programming language as opposed to black box testing.
4. Missing functionalities cannot be detected as the code that exists is tested.
5. Very complex and at times not realistic.

