**Chapter 7**

**What is control flow?  
Control Flow** means the **order** in which the computer runs the instructions in a program.

Usually, a program runs line by line from top to bottom. But **conditional statements** like if, for, while can **change this order**, so some lines may be skipped or repeated.

**Example:**

int x = 5;

if (x > 0) {

cout << "Positive number";

}

The program starts at the top, sets x = 5.

Then it checks if (x > 0), which is true, so it runs the next line and prints “Positive number”.

**What is CFT (Control Flow Testing)?**

Control Flow Testing is a method used to **check how a program moves** from one line to another, especially when it includes decisions and loops.

**Why is it used?**

* To **find errors** caused by wrong logic in code.
* To detect **dead branches** (code that never runs).
* To test **different paths** through a program.

**How does it work?**

* A **Control Flow Graph (CFG)** is created where:
  + **Nodes** = statements or blocks of code
  + **Edges** = flow of control (like arrows showing movement)
* Testers **select paths** through the program to check if they produce correct results.

**Example:**

int x = 10;

if (x > 5) {

cout << "High";

} else {

cout << "Low";

}

* Path 1: x > 5 → prints "High"
* Path 2: x ≤ 5 → prints "Low"

CFT will **test both paths** to make sure the program works correctly for each condition.

**✅ Summary:**

* CFT = Control Flow Testing
* Checks **paths and decisions** in code
* Helps **find bugs** in logic
* Uses **graphs** to map and test all possible flows

**Assignment Statements**  
These statements **store a value** in a variable.  
Example: x = 2 \* y;

This means **multiply y by 2** and put the result in x.

**Conditional Statements**  
These decide **which part of the program to run next** based on a condition.  
Examples:

* **if statement:**

if (x > 10) {

print("x is greater than 10");

}

This runs the message only if x is more than 10.

* **for loop:**

for (int i = 0; i < 5; i++) {

print(i);

}

This repeats printing i from 0 to 4.

* **while loop:**

while (x > 0) {

x = x - 1;

}

This keeps reducing x by 1 until it reaches 0.

**Control Flow**  
Without conditions, program runs **step by step** from top to bottom.  
With conditional statements, the flow can **jump, repeat, or skip parts** based on conditions.

**Program Path**  
A program path is the **route the program takes** through its statements from start to end.  
For example, with an if statement, the path could be:

* Take the "if" branch if condition is true, or
* Skip the "if" branch if condition is false.  
  Different inputs create different paths. Each path has its own expected output.

**CFG  
CFG is a picture that shows how a program’s control moves from one part to another.**

cfgsymbolsIt helps us understand all the possible paths the program can take while running.

cfgopenfiles1

**Path selection criteria:**

Path selection means choosing which paths in a program to test because many paths exist.

1. **Select all paths:**

* Test every possible path (100% path coverage).
* Finds most errors but hard for big programs.

2. **Select paths to achieve complete statement coverage** (**Complete statement coverage):**

* Run tests so every line of code runs at least once.
* Covers all assignments and conditions.

3. **Select paths to achieve complete branch coverage (Complete branch coverage):**

* Test all decisions (true and false outcomes).
* Covers every branch in the program.

4. **Select paths to achieve predicate coverage (Predicate coverage):**

* Test all true/false combinations of conditions.

These helps test the program fully without repeating the same paths.

**Effectiveness and Limitations of CFT  
Effectiveness:**

* Widely used in unit testing.
* Focuses on testing statements and branches.
* Detects about 50% of bugs during unit testing and 33% of all bugs.
* More effective on unstructured code (with jumps or gotos) than on well-structured code (loops, blocks).

**Limitations:**

* Cannot find problems when different parts of a program don’t work well together (interface issues).  
  *Example:* Two modules sending mismatched data won’t be detected by control flow testing.
* Misses some errors in variable or data initialization.  
  *Example:* Forgetting to initialize a variable before use might not be caught.
* Does not catch mistakes in program design or requirements.  
  *Example:* If the program’s expected behavior is wrong from the start, control flow testing won’t find this.

**Chapter 8**

 **Data Flow Testing** has two types:

* **Static:** checks code without running it to find data problems.
* **Dynamic:** runs the program to track how data moves.

**CFT VS DFT**

 **Similarities:**

* Find program paths and make test cases.

 **Difference:**

* Control flow tests use program structure to pick paths.
* Data flow tests use how data is defined and used to pick paths.