

# 1. What Are Control Structures?

Control structures are statements in a programming language that control the order in which instructions are executed in a program.

In C++, control structures are divided into three main types:

## 1. Sequential Structure

Executes statements one after another in order.

Example: Input marks, calculate percentage, display result.

## 2. Conditional Structure

Makes decisions based on conditions.

Example: Check whether a student has passed or failed.

## 3. Iterative Structure

Repeats a block of statements multiple times.

Example: Display numbers from 1 to 10.

# 2. Sequential Structure

A sequential structure executes instructions one by one, from top to bottom, in the order in which they are written.

[Understanding the Flow of Control in Python | PythonForAll](#)

[pythonforall.com](https://pythonforall.com)

Example: Calculate the Sum of Two Numbers

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, sum;

    cout << "Enter two numbers: ";
    cin >> a >> b;

    sum = a + b;

    cout << "Sum = " << sum;

    return 0;
}
```

## Understand the Code

1. Declare three integer variables.
2. Take two numbers from the user.
3. Calculate their sum.
4. Display the result.

Example output:

```
Enter two numbers: 10 20
Sum = 30
```

All statements execute in sequence without making a decision or repeating a block.

## 3. Conditional Structure

A conditional structure allows a program to make decisions. It checks a condition and executes the appropriate statement or block of code.

C++ provides the following conditional statements:

### if

Executes a block when a condition is true.

### if-else

Selects one of two alternatives.

### else-if

Selects from multiple alternatives.

### Nested if

Checks a condition inside another conditional statement.

### switch

Selects a branch based on the value of an expression.

Example: Check Pass or Fail

```
#include <iostream>
using namespace std;

int main()
{
    int marks;

    cout << "Enter marks: ";
    cin >> marks;
```

```
if (marks >= 50)
{
    cout << "Pass";
}
else
{
    cout << "Fail";
}

return 0;
}
```

Example output:

```
Enter marks: 65
Pass
```

Understand the condition:

- If marks are 50 or above, the student passes.
- Otherwise, the student fails.

The program selects the appropriate block based on the condition.

## 4. Iterative Structure

An iterative structure, also called a loop, repeats a block of code while a condition or repetition rule allows it.

C++ provides three main types of loops:

### [C++ Loops - Scaler Topics](#)

#### 1. **while** loop

Checks the condition before executing the loop body. It may execute zero times.

[Vòng lặp trong C++ là gì? Các vòng lặp thường gặp trong C++](#)

#### 2. **do-while** loop

Executes the loop body first and checks the condition afterward. It executes at least once.

[for Loop in C++ - GeeksforGeeks](#)

#### 3. **for** loop

Commonly used when the number of repetitions is known in advance.

Example: Display Numbers from 1 to 5

```
#include <iostream>
using namespace std;

int main()
{
    for (int i = 1; i <= 5; i++)
    {
        cout << i << endl;
    }

    return 0;
}
```

Output:

```
1
2
3
4
5
```

## Understand the Code

- `int i = 1` — starts counting from 1.
- `i <= 5` — continues while `i` is 5 or less.
- `i++` — increases `i` by 1 after each iteration.
- `cout << i << endl;` — displays the current value.

The loop repeats until the condition becomes false.

## 5. Comparison of the Three Structures

| Feature           | Sequential                 | Conditional                                                  | Iterative                                                     |
|-------------------|----------------------------|--------------------------------------------------------------|---------------------------------------------------------------|
| Purpose           | Execute in order           | Make decisions                                               | Repeat instructions                                           |
| Main concept      | Sequence                   | Condition                                                    | Repetition                                                    |
| C++ examples      | Input, calculation, output | <code>if</code> , <code>if-else</code> , <code>switch</code> | <code>for</code> , <code>while</code> , <code>do-while</code> |
| Repetition        | No automatic repetition    | Not inherently repetitive                                    | Yes                                                           |
| Real-life example | Follow recipe steps        | Decide whether to take an umbrella                           | Exercise 10 times                                             |

Remember: Sequential means follow the steps, conditional means make a decision, and iterative means repeat the steps. A single C++ program can use all three structures together.