3. Control Flow Statements

1. What are conditional statements in C++? Explain the if-else and switch statements.

Conditional statements are used to make decisions in a program.

They allow the program to execute different blocks of code based on conditions.

The most common conditional statement is the if statement.

The if statement checks whether a condition is true or false.

If the condition is true, the code inside the if block executes.

The if-else statement provides an alternative block if the condition is false.

Multiple conditions can be checked using else-if ladder.

Another conditional statement is the switch statement.

The switch statement is used when there are multiple choices.

It works with integer or character values.

The switch statement compares the value with different cases.

If a match is found, that block of code executes.

The break statement is used to exit the switch block.

Conditional statements are important for decision-making programs.

They make programs dynamic and intelligent.

Without conditional statements, programs would execute sequentially only.

Therefore, conditional statements are essential in programming.

2. What is the difference between for, while, and do-while loops in C++?

Loops are used to execute a block of code repeatedly.

The for loop is used when the number of iterations is known.

It contains initialization, condition, and increment in a single line.

The while loop is used when the number of iterations is not known.

It checks the condition before executing the loop body.

If the condition is false initially, the while loop will not execute.

The do-while loop executes the loop body at least once.

It checks the condition after executing the code.

The for loop is compact and easy for counting loops.

The while loop is suitable for condition-based loops.

The do-while loop ensures minimum one execution.

All loops help reduce repetition of code.

Loops make programs efficient and shorter.

Choosing the correct loop depends on the problem.

Understanding loops is important for solving programming problems.

3. How are break and continue statements used in loops? Provide examples.

The break and continue statements are used inside loops.

The break statement is used to terminate the loop immediately.

When break is executed, control moves outside the loop.

It is also used inside switch statements.

The continue statement skips the current iteration.

It moves control to the next iteration of the loop.

Break is useful when a certain condition is met.

Continue is useful when some values need to be skipped.

For example, break can stop a loop when a number is found.

Continue can skip even numbers in a loop.

These statements improve control over loops.

They help in writing efficient programs.

Break avoids unnecessary iterations.

Continue avoids executing unwanted code.

Both statements are important in loop control.

They increase flexibility in program execution.

4. Explain nested control structures with an example.

Nested control structures mean placing one control structure inside another.

For example, a loop inside another loop is called nested loop.

Similarly, an if statement inside another if statement is called nested if.

Nested loops are commonly used for pattern printing.

They are also used in matrix operations.

Nested structures increase program complexity.

They allow solving advanced problems.

For example, printing a triangle pattern requires nested loops.

The outer loop controls rows.

The inner loop controls columns.

Nested if statements check multiple conditions.

They help in detailed decision-making processes.

Proper indentation is important in nested structures.

Without proper structure, the program may become confusing.

Nested control structures are powerful tools in C++.

They are widely used in real-life programming tasks.