

1. The `if` Statement

The `if` statement executes a block of code only when its condition is true.

Syntax

```
if (condition)
{
    // Code executes if the condition is true
}
```

Example 1: Check Marks

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

int main()
{
    int marks = 85;

    if (marks >= 80)
    {
        cout << "Excellent Performance!";
    }

    return 0;
}
```

Understand the code

- `int marks = 85;` stores the student's marks.
- `marks >= 80` checks whether marks are at least 80.
- Because the condition is true, the message is displayed.

Output:

```
Excellent Performance!
```

If the marks are 65, the condition is false, so nothing is displayed.

Modify the code

1. Change marks from `85` to `75`. Observe the output.
2. Change the condition from `marks >= 80` to `marks >= 70`.
3. Change the message to `"You are doing well!"`.

Try it: Write a program that displays "Eligible for admission" if a student's marks are 50 or above.

3. The `if-else` Statement

The `if-else` statement provides two possible paths:

- `if`: executes when the condition is true.
- `else`: executes when the condition is false.

Example 2: Pass or Fail

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

int main()
{
    int marks;

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

    if (marks >= 50)
    {
        cout << "Congratulations! You passed.";
    }
    else
    {
        cout << "You failed. Work harder.";
    }

    return 0;
}
```

Understand the code

- `cin >> marks;` accepts marks from the user.
- If marks are 50 or above, the student passes.
- Otherwise, the program displays the failure message.

Example output for input 65:

```
Enter your marks: 65
Congratulations! You passed.
```

Modify the code

1. Change the passing marks from 50 to 40.
2. Replace the messages with "Pass" and "Fail".

3. Change the program to check whether a number is positive or not positive.

Try it: Write a program that checks whether a student is eligible to vote in a student election if the minimum age is 18.

4. The `else-if` Ladder

Use `else-if` when you need to check multiple conditions.

Example 3: Student Grade Notification

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

int main()
{
    int marks;

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

    if (marks >= 80)
    {
        cout << "Grade A - Excellent!";
    }
    else if (marks >= 70)
    {
        cout << "Grade B - Very Good!";
    }
    else if (marks >= 60)
    {
        cout << "Grade C - Good!";
    }
    else if (marks >= 50)
    {
        cout << "Grade D - Pass!";
    }
    else
    {
        cout << "Fail - Improve your performance.";
    }

    return 0;
}
```

Understand the code

The program checks conditions from top to bottom. When it finds the first true condition, it executes that block and skips the remaining conditions.

Marks	Notification
90	Grade A - Excellent!
75	Grade B - Very Good!
65	Grade C - Good!
55	Grade D - Pass!
35	Fail - Improve your performance.

Important: Write the highest range first. If `marks >= 50` comes first, a student with 90 marks will receive the Grade D notification because the first condition is already true.

Modify the code

1. Change the Grade A threshold from 80 to 85.
2. Add a new grade, "Grade A+", for marks of 90 or above.
3. Change the messages to notifications suitable for university students.
4. Add a notification "Invalid marks" when marks are below 0 or above 100. Check invalid marks before the grade conditions.

Try it: Write a program that displays a scholarship notification:

- 90–100: Full Scholarship
- 80–89: Half Scholarship
- 70–79: Quarter Scholarship
- Below 70: No Scholarship

Also handle marks outside the range 0–100.

5. Nested `if` Statement

A nested `if` means an `if` statement inside another `if` statement.

It is useful when a decision depends on more than one condition, particularly when the second condition should be checked only after the first condition is true.

Real-life example: A student can register for a course only if they are admitted and have paid the required fee.

Example 4: University Registration Notification

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

int main()
{
    int admitted, feePaid;

    cout << "Are you admitted? Enter 1 for Yes, 0 for No: ";
    cin >> admitted;
```

```
cout << "Have you paid the fee? Enter 1 for Yes, 0 for No: ";
cin >> feePaid;

if (admitted == 1)
{
    if (feePaid == 1)
    {
        cout << "You can register for courses.";
    }
    else
    {
        cout << "Please pay your semester fee.";
    }
}
else
{
    cout << "Your admission is not confirmed.";
}

return 0;
}
```

Understand the code

First condition: Is the student admitted?

Yes (1)

Second condition

Has the student paid the fee?

Yes → Course registration allowed

No → Pay semester fee

No (0)

Notification

Admission not confirmed

The inner **if** checks the fee only within the admitted student's branch. If admission is not confirmed, the program displays the admission message.

Modify the code

1. Change the registration message to "You are eligible for enrollment."
2. Add a third variable, **documentsVerified**.
3. Allow registration only if the student is admitted, the fee is paid, and the documents are verified.
4. Display a separate notification if the documents are not verified.

Try it: Write a program for a university library:

- First, check whether the student has a valid university ID.
- If the ID is valid, check whether the student has returned all overdue books.
- Allow book borrowing only if both conditions are satisfied.

6. Difference Between Conditional Statements

Statement	Purpose	Example
<code>if</code>	Executes code if a condition is true	Check whether marks are 80 or above
<code>if-else</code>	Chooses between two alternatives	Pass or fail
<code>else-if</code>	Chooses among multiple alternatives	Assign a grade
Nested <code>if</code>	Checks a condition inside another condition	Check admission, then fee payment

Examples

Example 5

✓ Question Statement

Write a C++ program that stores the value 2 in a variable `x` and checks whether the value is greater than 5, equal to 5, or less than 5. Display the appropriate message based on the condition.

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

int main() {
    // Store a number in the variable x
    int x = 2;

    // Check if x is greater than 5
    if (x > 5) {
        cout << "x is greater than 5." << endl;
    }
    // Check if x is equal to 5
    else if (x == 5) {
        cout << "x is equal to 5." << endl;
    }
    // If both above conditions are false, x must be less than 5
    else {
        cout << "x is less than 5." << endl;
    }

    return 0;
}
```

✓ Example 6: Finding the Maximum of Three Numbers

✓ Question Statement

Write a C++ program that takes three numbers as input from the user and determines which of the three is the largest. Display the maximum number using an if-else.

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

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

    // Input three numbers
    cout << "Enter first number: ";
    cin >> a;

    cout << "Enter second number: ";
    cin >> b;

    cout << "Enter third number: ";
    cin >> c;

    if (a >= b && a >= c) {
        cout << "The maximum number is: " << a << endl;
    }
    else if (b >= a && b >= c) {
        cout << "The maximum number is: " << b << endl;
    }
    else {
        cout << "The maximum number is: " << c << endl;
    }

    return 0;
}
```

7. Coding Exercises

Task 1: Positive, Negative or Zero

Write a C++ program that takes a number as input and checks whether it is positive, negative, or zero. Display the appropriate message based on the entered number.

Hint: Use an if-else-if ladder.

Task 2: Check Even or Odd

Write a C++ program that takes an integer as input and determines whether it is even or odd. Display an appropriate message based on the result.

Hint: Use the modulus operator (%) to check the remainder when the number is divided by 2.

Task 3: Check Eligibility Based on Marks

Write a C++ program that takes a student's marks as input. If the marks are 50 or above, display "Eligible"; otherwise, display "Not Eligible".

Hint: Use an if-else statement to check the marks.

Task 4: Student Grade Calculator**

Write a C++ program that takes a student's marks as input and displays the corresponding grade according to the following criteria:

- **A:** 80–100 marks
- **B:** 70–79 marks
- **C:** 60–69 marks
- **D:** 50–59 marks
- **Fail:** Below 50 marks

If the entered marks are less than 0 or greater than 100, display "**Invalid Marks**".

Hint: Use an `if-else-if` ladder to check the marks and assign the appropriate grade. First, validate that the marks are between 0 and 100.

Task 5: Find the Larger Number

Write a C++ program that takes two numbers as input and compares them. Display the larger number. If both numbers are equal, display "Both numbers are equal."

Hint: Use an if-else-if ladder to compare the two numbers and handle all three possible cases.

Task 6: Login Authentication System

Write a C++ program that takes two inputs: a user verification flag and a password verification flag. Display "Login Successful" only if both flags indicate that the username and password are correct. Otherwise, display "Login Failed".

Hint: Use nested if statements to check the user verification flag first and then the password verification flag.

Task 7: Scholarship Eligibility Checker

Write a C++ program that takes a student's marks and family monthly income as input. The student is eligible for a scholarship only if both of the following conditions are met:

- The student's marks are 80 or above.
- The family's monthly income is below Rs. 60,000.

Display "Scholarship Awarded" if the student meets both conditions. Otherwise, display "Not Eligible for Scholarship".

Hint: Use nested if statements to check the marks first and then the family income.

Course Registration

Task 8: Student Registration Eligibility*

Write a C++ program that takes three inputs to check whether a student can register:

- **Admission Status:** Admission is confirmed.
- **Fee Status:** Required fee has been paid.
- **Document Verification Status:** All required documents have been verified.

Display "**Registration Allowed**" only if all three requirements are satisfied. Otherwise, display "**Registration Not Allowed**".

Hint: Use nested `if` statements with appropriate `else` branches to check each requirement one by one.

8. Challenge: Modify and Improve a Program

Start with this program and make the requested changes.

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

int main()
{
    int marks = 75;

    if (marks >= 80)
    {
        cout << "Excellent!";
    }
    else
    {
        cout << "Keep working hard.";
    }

    return 0;
}
```

Tasks

1. Change `marks` to 90 and observe the notification.
2. Add an `else-if` condition for marks of 70 or above.
3. Add another condition for marks of 60 or above.
4. Display "`Fail`" for marks below 50.
5. Add a nested `if` to check whether a student has paid the semester fee.
6. Test your program with marks of 45, 60, 75, and 90.