

Search...

[C++ Data Types](#)[C++ Input/Output](#)[C++ Arrays](#)[C++ Pointers](#)[C++ OOPs](#)[C++ STL](#)[Sign In](#)

Decision Making in C (if , if..else, Nested if, if-else-if)

Last Updated : 13 May, 2025

In C, programs can choose which part of the code to execute based on some condition. This ability is called **decision making** and the statements used for it are called **conditional statements**. These statements evaluate one or more conditions and make the decision whether to execute a block of code or not.

For example, consider that there is a show that starts only when certain number of people are present in the audience. So, you can write a program like as shown:

```
1  #include <stdio.h>
2
3  int main() {
4
5      // Number of people in the audience
6      int num = 100;
7
8      // Conditional code inside decision making
      statement
9      if (num > 50) {
10         printf("Start the show");
11     }
12
13     return 0;
14 }
```

Output

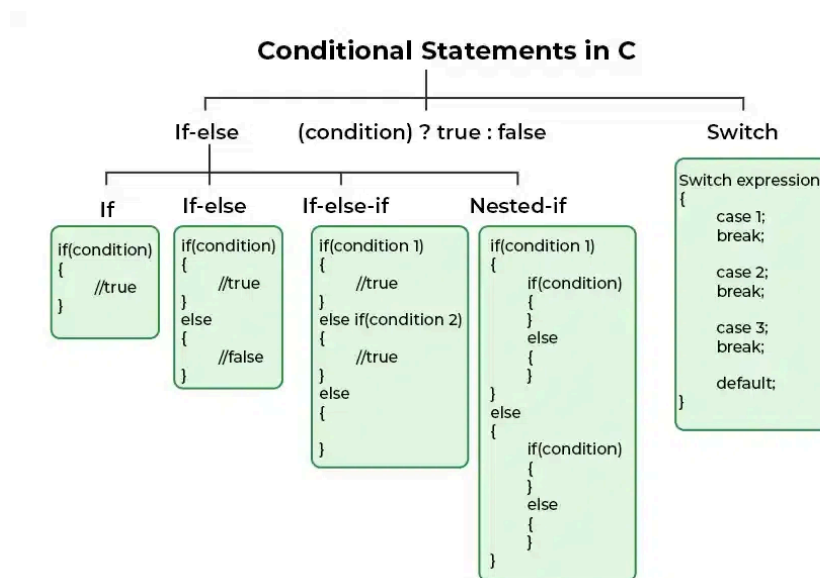
We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

In the above program, the show only starts when the number of people is greater than **50**. It is specified in the **if statement** (a type of conditional statement) as a condition (**num > 50**). You can decrease the value of **num** to less than **50** and try rerunning the code.

Types of Conditional Statements in C

In the above program, we have used if statement, but there are many different types of conditional statements available in C language:



1. if in C

The [if statement](#) is the simplest decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statements is executed otherwise not.

A **condition** is any expression that evaluates to either a true or false (or values convertible to true or false).

Example

```

1  #include <stdio.h>
2
3  int main() {
4

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

```
9         printf("Eligible for vote");
10     }
11 }
```

Output

Eligible for vote

The expression inside **() parenthesis** is the **condition** and set of statements inside **{ } braces** is its **body**. If the condition is true, only then the body will be executed.

*If there is only a single statement in the body, **{ } braces** can be omitted.*

2. if-else in C

The **if** statement alone tells us that if a condition is true, it will execute a block of statements and if the condition is false, it won't. But what if we want to do something else when the condition is false? Here comes the C **else** statement. We can use the **else** statement with the **if** statement to execute a block of code when the condition is false. The [if-else statement](#) consists of two blocks, one for false expression and one for true expression.

Example

```
1  #include <stdio.h>
2
3  int main() {
4      int i = 10;
5
6      if (i > 18) {
7          printf("Eligible for vote");
8      }
9      else {
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

```
13 }
```

Output

Not Eligible for vote

The block of code following the *else* statement is executed as the condition present in the *if* statement is false.

3. Nested if-else in C

A nested if in C is an if statement that is the target of another if statement. Nested if statements mean an if statement inside another if statement. Yes, C allow us to nested if statements within if statements, i.e, we can place an if statement inside another if statement.

Example

```
1  #include <stdio.h>
2
3  int main(){
4      int i = 10;
5
6      if (i == 10) {
7          if (i < 18)
8              printf("Still not eligible for vote");
9          else
10             printf("Eligible for vote\n");
11     }
12     else {
13         if (i == 20) {
14             if (i < 22)
15                 printf("i is smaller than 22
16 too\n");
17             else
18                 printf("i is greater than 25");
19         }
20     }
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Output

Still not eligible for vote

4. if-else-if Ladder in C

The [if else if statements](#) are used when the user has to decide among multiple options. The C if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the C else-if ladder is bypassed. If none of the conditions is true, then the final else statement will be executed. if-else-if ladder is similar to the switch statement.

Example

```
1  #include <stdio.h>
2
3  int main() {
4      int i = 20;
5
6      // If else ladder with three conditions
7      if (i == 10)
8          printf("Not Eligible");
9      else if (i == 15)
10         printf("wait for three years");
11     else if (i == 20)
12         printf("You can vote");
13     else
14         printf("Not a valid age");
15
16     return 0;
17 }
```

Output

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

The [switch case statement](#) is an alternative to the if else if ladder that can be used to execute the conditional code based on the value of the variable specified in the switch statement. The switch block consists of cases to be executed based on the value of the switch variable.

Example

```
1  #include <stdio.h>
2
3  int main() {
4
5      // variable to be used in switch statement
6      int var = 18;
7
8      // declaring switch cases
9      switch (var) {
10     case 15:
11         printf("You are a kid");
12         break;
13     case 18:
14         printf("Eligible for vote");
15         break;
16     default:
17         printf("Default Case is executed");
18         break;
19     }
20
21     return 0;
22 }
```

Output

Eligible for vote

Note: The switch expression should evaluate to either integer or character. It cannot evaluate any other data type.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

The [conditional operator](#) is used to add conditional code in our program. It is similar to the if-else statement. It is also known as the ternary operator as it works on three operands.

Example:

```
1  #include <stdio.h>
2
3  int main() {
4      int var;
5      int flag = 0;
6
7      // using conditional operator to assign the
      value to var
8      // according to the value of flag
9      var = flag == 0 ? 25 : -25;
10     printf("Value of var when flag is 0: %d\n",
        var);
11
12     return 0;
13 }
```

Output

Value of var when flag is 0: 25

7. Jump Statements in C

These statements are used in C for the unconditional flow of control throughout the functions in a program. They support four types of jump statements:

A) break

This loop control statement is used to terminate the loop. As soon as the [break](#) statement is encountered from within a loop, the loop iterations stop there, and control returns from the loop immediately to the first statement after the loop.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

```
2
3  int main() {
4      int arr[] = { 1, 2, 3, 4, 5, 6 };
5      int key = 3;
6      int size = 6;
7
8      for (int i = 0; i < size; i++) {
9          if (arr[i] == key) {
10             printf("Element found at position: %d",
11                    (i + 1));
12             break;
13         }
14     }
15
16     return 0;
17 }
```

Output

Element found at position: 3

B) continue

This loop control statement is just like the break statement. The [continue statement](#) is opposite to that of the break *statement*, instead of terminating the loop, it forces to execute the next iteration of the loop. As the name suggests the continue statement forces the loop to continue or execute the next iteration. When the continue statement is executed in the loop, the code inside the loop following the continue statement will be skipped and the next iteration of the loop will begin.

Example:

```
1  #include <stdio.h>
2
3  int main() {
4      for (int i = 1; i <= 10; i++) {
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).


```
8         // without printing
9         if (i == 6)
10            continue;
11        else
12            printf("%d ", i);
13    }
14
15    return 0;
16 }
```

Output

1 2 3 4 5 7 8 9 10

C) goto

The [goto statement](#) in C also referred to as the unconditional jump statement can be used to jump from one point to another within a function.

Examples:

```
1  #include <stdio.h>
2
3  int main() {
4      int n = 1;
5      label:
6      printf("%d ", n);
7      n++;
8      if (n <= 10)
9          goto label;
10     return 0;
11 }
```

Output

1 2 3 4 5 6 7 8 9 10

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

The [return](#) in C returns the flow of the execution to the function from where it is called. This statement does not mandatorily need any conditional statements. As soon as the statement is executed, the flow of the program stops immediately and returns the control from where it was called. The return statement may or may not return anything for a void function, but for a non-void function, a return value must be returned.

Example:

```
1  #include <stdio.h>
2  int SUM(int a, int b) {
3      int s1 = a + b;
4      return s1;
5  }
6
7  int main()
8  {
9      int num1 = 10;
10     int num2 = 10;
11     int sum_of = SUM(num1, num2);
12     printf("%d", sum_of);
13     return 0;
14 }
```

Output

20

Comment

More info

Campus Training Program

Next Article

C - if Statement

Similar Reads

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

15+ min read

Lua Decision Making

In Lua, you can control your program's flow using if, elseif, and else statements, along with logical operators like and, or, and not. These tool...

15+ min read

C++17 new feature : If Else and Switch Statements with initializers

In many cases, we need to check the value of something returned by a function and perform conditional operations based on this value. So our...

15+ min read

Execute both if and else statements in C/C++ simultaneously

Write a C/C++ program that executes both if-else block statements simultaneously. Syntax of if-else statement in C/C++ language is: if...

12 min read

Implicit Type Conversion in C with Examples

Prerequisite: Data Types, Type Conversion Implicit Type Conversion is also known as 'automatic type conversion'. It is done by the compiler on its...

15+ min read

Data type of case labels of switch statement in C++?

In C++ switch statement, the expression of each case label must be an integer constant expression. For example, the following program fails in...

10 min read

Difference Between if-else and switch in C

In C programming both switch statements and if-else statements are used to perform decision-making and control the flow of the program accordin...

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

C if else if ladder

In C, if else if ladder is an extension of if else statement used to test a series of conditions sequentially, executing the code for the first true...

15+ min read

C++ if else if Ladder

In C++, the if-else-if ladder helps the user decide from among multiple options. The C++ if statements are executed from the top down. As soon...

15+ min read

Nested if in C++

If is a type of condition checking in which a condition turns out to be true a block of statements is executed. Syntax: // if base_condition is true // ever...

15+ min read



Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Legal
Privacy Policy
Careers
In Media
Contact Us
GfG Corporate Solution
Placement Training Program

GfG Weekly Contest
Offline Classroom Program
DSA in JAVA/C++
Master System Design
Master CP
GeeksforGeeks Videos

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Python Tutorial

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview Questions
Competitive Programming

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
NodeJs
Bootstrap
Tailwind CSS

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

SQL
MYSQL
PostgreSQL
PL/SQL
MongoDB

Preparation Corner

Company-Wise Recruitment Process
Aptitude Preparation
Puzzles
Company-Wise Preparation

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets

Machine Learning/Data Science

Complete Machine Learning & Data Science Program - [LIVE]
Data Analytics Training using Excel, SQL, Python & PowerBI -
[LIVE]
Data Science Training Program - [LIVE]
Data Science Course with IBM Certification

Programming Languages

C Programming with Data Structures
C++ Programming Course
Java Programming Course
Python Full Course

Clouds/Devops

DevOps Engineering
AWS Solutions Architect Certification
Salesforce Certified Administrator Course

GATE 2026

GATE CS Rank Booster
GATE DA Rank Booster
GATE CS & IT Course - 2026
GATE DA Course 2026
GATE Rank Predictor

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).