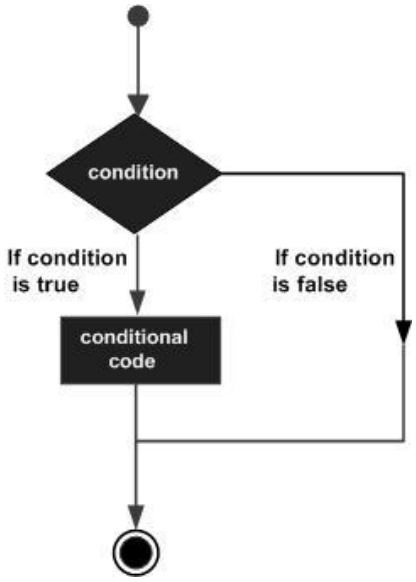
Decision Control Structure

In a 'C' program are executed sequentially. This happens when there is no condition around the statements. If you put some condition for a block of statements, the flow of execution might change based on the result evaluated by the condition. This process is referred to as decision making in 'C.' The decision-making statements are also called as control statements. Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Show below is the general form of a typical decision making structure found in most of the programming languages −



C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value.

There are the following variants of control statement in C language.

- **A.** If statement
- **B.** If-else statement
- **C.** If else-if ladder
- **D.** Nested if
- **E.** Switch statement
- **F.** Nested Switch Statement
- **G.** Conditional operator

Simple IF Statement

An **if** statement consists of a Boolean expression followed by one or more statements.

Syntax

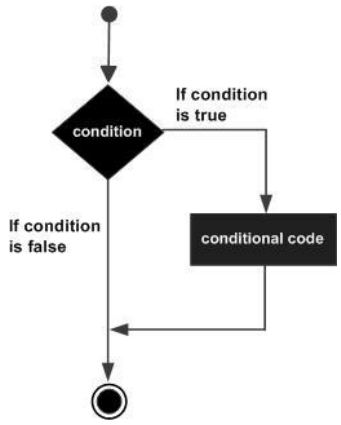
The syntax of an 'if' statement in C programming language is −

```
if(boolean_expression) {
   /* statement(s) will execute if the boolean expression is true */
}
```

If the Boolean expression evaluates to **true**, then the block of code inside the 'if' statement will be executed. If the Boolean expression evaluates to **false**, then the first set of code after the end of the 'if' statement (after the closing curly brace) will be executed.

C programming language assumes any **non-zero** and **non-null** values as **true** and if it is either **zero** or **null**, then it is assumed as **false** value.

Flow Diagram



Example

```
#include <stdio.h>
int main () {
   /* local variable definition */
   int a = 10;
    /* check the boolean condition using if statement */
   if( a < 20 ) {
      /* if condition is true then print the following */
      printf("a is less than 20\n" );
   }
   printf("value of a is : %d\n", a);
   return 0;
}
Output:
a is less than 20;
value of a is : 10
```

Program to find a number is even or not

```
1. #include<stdio.h>
2. int main(){
3. int number=0;
4. printf("Enter a number:");
5. scanf("%d",&number);
6. if(number%2==0){
7. printf("%d is an even number",number);
8. }
9. printf("%d is not an even number",number);
10. return 0;
11. }
```

```
Output:
Enter a number:4
4 is an even number
enter a number:5
5 is not an even number
```

Program to find the largest number of the three.

```
1. #include <stdio.h>
2. int main()
3. {
4.     int a, b, c;
5.      printf("Enter three numbers?");
6.     scanf("%d %d %d",&a,&b,&c);
7.     if(a>b && a>c)
8.     {
9.         printf("%d is largest",a);
10.    }
11.    if(b>a  && b > c)
12.    {
13.        printf("%d is largest",b);
14.    }
15.    if(c>a && c>b)
16.    {
17.        printf("%d is largest",c);
18.    }
19.    if(a == b && a == c)
20.    {
21.        printf("All are equal");
22.    }
23. }
```

```
Output:
Enter three numbers?
```

```
12 23 34
34 is largest
```

If-Else Statement

An **if** statement can be followed by an optional **else** statement, which executes when the Boolean expression is false.

Syntax

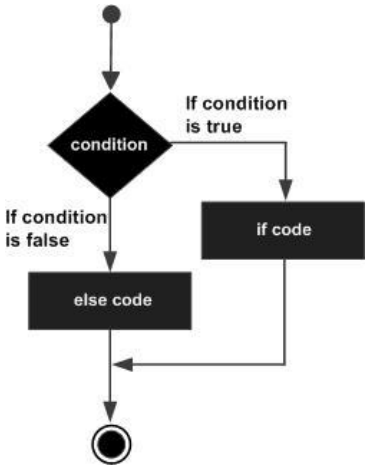
The syntax of an **if...else** statement in C programming language is −

```
if(boolean_expression) {
   /* statement(s) will execute if the boolean expression is true */
} else {
   /* statement(s) will execute if the boolean expression is false */
}
```

If the Boolean expression evaluates to **true**, then the **if block** will be executed, otherwise, the **else block** will be executed.

C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value.

Flow Diagram



Example

```
#include <stdio.h>

int main () {

   /* local variable definition */
```

```
   int a = 100;

   /* check the boolean condition */
   if( a < 20 ) {
      /* if condition is true then print the following */
      printf("a is less than 20\n" );
   } else {
      /* if condition is false then print the following */
      printf("a is not less than 20\n" );
   }

   printf("value of a is : %d\n", a);

   return 0;
}
```

Output:

```
a is not less than 20;
value of a is : 100
```

Program to find a number is even or odd.

```
1. #include<stdio.h>
2. int main(){
3. int number=0;
4. printf("enter a number:");
5. scanf("%d",&number);
6. if(number%2==0){
7. printf("%d is even number",number);
8. }
9. else{
10. printf("%d is odd number",number);
11. }
12. return 0;
13. }
```

Output

```
enter a number:4
4 is even number
enter a number:5
5 is odd number
```

Program to check whether a person is eligible to vote or not.

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

```
5.    printf("Enter your age?");
6.    scanf("%d",&age);
7.    if(age>=18)
8.    {
9.       printf("You are eligible to vote...");
10.   }
11.   else
12.   {
13.      printf("Sorry ... you can't vote");
14.   }
15. }
```

Output

```
Enter your age?18
You are eligible to vote...
Enter your age?13
Sorry ... you can't vote
```

Example of nested if..else

```
#include <stdio.h>
int main()
{
   int var1, var2;
   printf("Input the value of var1:");
   scanf("%d", &var1);
   printf("Input the value of var2:");
   scanf("%d",&var2);
   if (var1 != var2)
   {
        printf("var1 is not equal to var2\n");
        //Nested if else
        if (var1 > var2)
        {
               printf("var1 is greater than var2\n");
        }
        else
        {
               printf("var2 is greater than var1\n");
        }
   }
   else
   {
        printf("var1 is equal to var2\n");
   }
   return 0;
}
```

Output

```
Input the value of var1:12
Input the value of var2:21
```

```
var1 is not equal to var2
var2 is greater than var1
```

If Else-If ladder

An **if** statement can be followed by an optional **else if...else** statement, which is very useful to test various conditions using single if...else if statement.

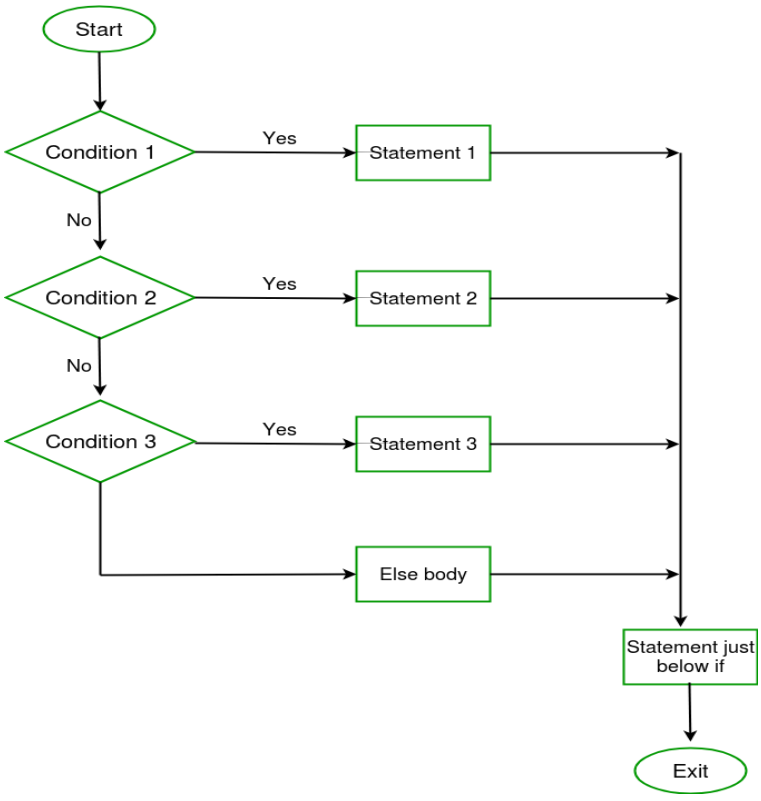
When using if...else if..else statements, there are few points to keep in mind −

- An if can have zero or one else's and it must come after any else if's.
- An if can have zero to many else if's and they must come before the else.
- Once an else if succeeds, none of the remaining else if's or else's will be tested.

Syntax

The syntax of an **if...else if...else** statement in C programming language is −

```
if(boolean_expression 1) {
   /* Executes when the boolean expression 1 is true */
} else if( boolean_expression 2) {
   /* Executes when the boolean expression 2 is true */
} else if( boolean_expression 3) {
   /* Executes when the boolean expression 3 is true */
} else {
   /* executes when the none of the above condition is true */
}
```



Example

```
#include <stdio.h>

int main () {

   /* local variable definition */
   int a = 100;

   /* check the boolean condition */
   if( a == 10 ) {
      /* if condition is true then print the following */
      printf("Value of a is 10\n" );
   } else if( a == 20 ) {
      /* if else if condition is true */
      printf("Value of a is 20\n" );
   } else if( a == 30 ) {
      /* if else if condition is true  */
      printf("Value of a is 30\n" );
   } else {
      /* if none of the conditions is true */
      printf("None of the values is matching\n" );
   }

   printf("Exact value of a is: %d\n", a );

   return 0;
}
```

Output

```
None of the values is matching
Exact value of a is: 100
```

Program to calculate the grade of the student according to the specified marks.

```
1. #include <stdio.h>
2. int main()
3. {
4.     int marks;
5.     printf("Enter your marks?");
6.     scanf("%d",&marks);
7.     if(marks > 85 && marks <= 100)
8.     {
9.         printf("Congrats ! you scored grade A ...");
10.    }
11.    else if (marks > 60 && marks <= 85)
12.    {
13.        printf("You scored grade B + ...");
14.    }
15.    else if (marks > 40 && marks <= 60)
16.    {
17.        printf("You scored grade B ...");
18.    }
```

```
19.    else if (marks > 30 && marks <= 40)
20.    {
21.        printf("You scored grade C ...");
22.    }
23.    else
24.    {
25.        printf("Sorry you are fail ...");
26.    }
27. }
```

Output

```
Enter your marks?10
Sorry you are fail ...
Enter your marks?40
You scored grade C ...
Enter your marks?90
Congrats ! you scored grade A ...
```

Nested If Statement

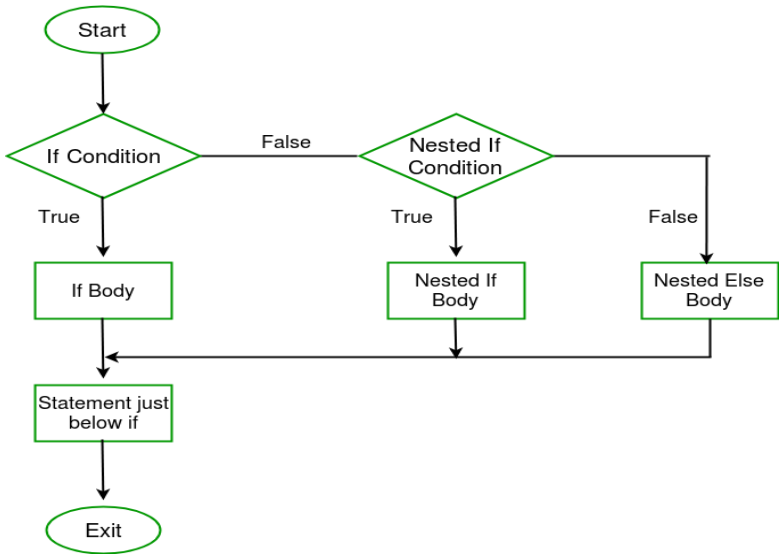
It is always legal in C programming to **nest** if-else statements, which means you can use one if or else if statement inside another if or else if statement(s).

Syntax

The syntax for a **nested if** statement is as follows −

```
if( boolean_expression 1) {

   /* Executes when the boolean expression 1 is true */
   if(boolean_expression 2) {
      /* Executes when the boolean expression 2 is true */
   }
}
```



You can nest **else if...else** in the similar way as you have nested *if* statements.

Example

```
#include <stdio.h>

int main () {

   /* local variable definition */
   int a = 100;
   int b = 200;

   /* check the boolean condition */
   if( a == 100 ) {

      /* if condition is true then check the following */
      if( b == 200 ) {
         /* if condition is true then print the following */
         printf("Value of a is 100 and b is 200\n" );
      }
   }

   printf("Exact value of a is : %d\n", a );
   printf("Exact value of b is : %d\n", b );

   return 0;
}
```

Output

```
Value of a is 100 and b is 200
Exact value of a is : 100
Exact value of b is : 200
```