# Chapter Four
# Conditional Statements

## 4.1. Introduction

At the end of this chapter, we will be able to:

1. Know types of conditional statements.
2. Program any condition contain with the program.

## 4.2. Control Structures

A program is usually not limited to a linear sequence of instructions. During its process, it may bifurcate, repeat code or take decisions. For that purpose, C++ provides control structures that serve to specify what has to be done by our program, when and under which circumstances.

With the introduction of control structures, we are going to have to introduce a new concept: the *compound- statement* or *block*. A block is a group of statements which are separated by semicolons (;) like all C++ statements, but grouped together in a block enclosed in braces: { }:

{ statement1; statement2; statement3; }

Most of the control structures that we will see in this section require a generic statement, as part of its syntax. A statement can be either a simple statement (a simple instruction ending with a semicolon) or a compound statement (several instructions grouped in a block), like the one just described. In the case that we want the statement to be a simple statement, we do not need to enclose it in braces ({}). But in the case that we want the statement to be a compound statement it must be enclosed between braces ({}), forming a block.

## 4.3. Conditional structure

Conditional structure contain three types of write conditional programs.

1. IF – statements
2. IF – Else statements
3. IF – Else – IF statements

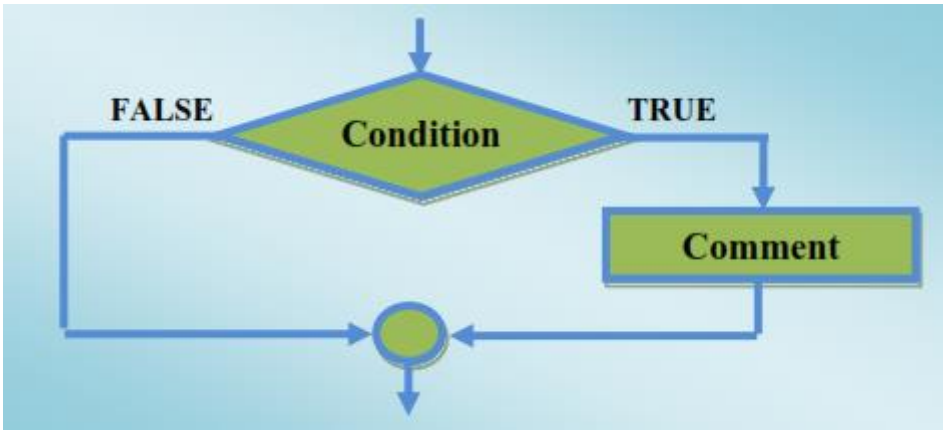
## 4.3.1. IF - Statements

IF- keyword: it is used to execute a statement or block only if a condition is fulfilled. Its form is:

**if (condition) statement**

Where condition is the expression that is being evaluated. If this condition is true, statement is executed. If it is false, statement is ignored (not executed) and the program continues right after this conditional structure.

The structure of execution the statements is as flowchart blow:




***Ex:*** The following code fragment prints x is 100 only if the value stored in the x variable is indeed 100:

```
if (x == 100)
cout << "x is 100";
```

If we want more than a single statement to be executed in case that the condition is true we can specify a block using braces { }:

```
if (x == 100)
{
cout << "x is ";
cout << x;
}
```

***Ex:*** Write a C++ program to enter two Boolean numbers then, print phrase "A And B" if A and B equal to 1, or print phrase "A Or B" if A equal to 1 and B equal to 0.

***Sol:***

```
#include <iostream>
using namespace std;
int main ()
{
bool A,B;
cin >>A ;
cin >>B ;
if ((A==1)&&(B==1))
{cout << "A And B"<<'\n';}
if ((A==1)||(B==0))
{cout << "A or B"<<'\n';}
return 0;
}
```

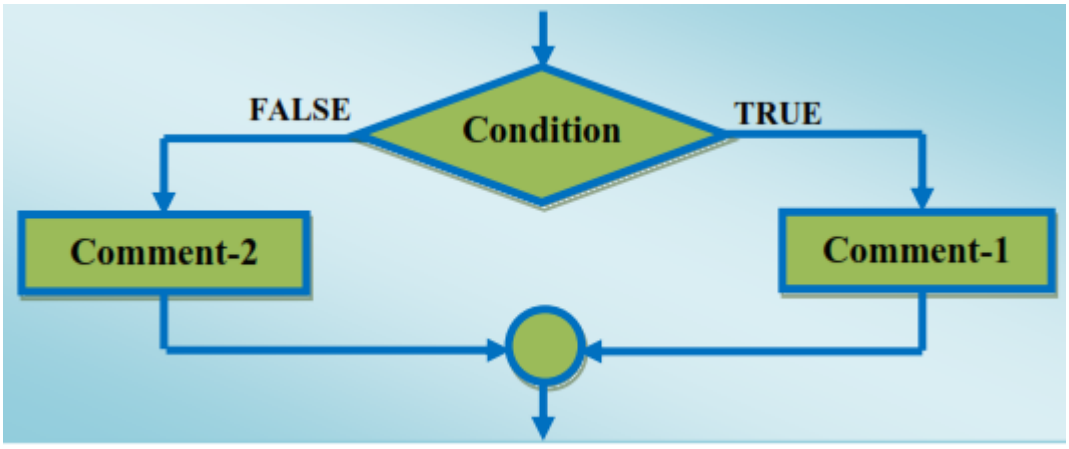
## 4.3.2. IF – Else statements

We can additionally specify what we want to happen if the condition is not fulfilled by using the keyword else. Its form used in conjunction with if is:

```
If (condition)
        statement1
else
        statement2
```

Prints on the screen x is 100 if indeed x has a value of 100, but if it has not -and only if not- it prints out x is not 100.

The (if – else) structures can be concatenated with the intention of verifying a range of values.



***Ex:***

```
if (x == 100)
cout << "x is 100";
else
cout << "x is not 100";
```

***Ex:*** Write a C++ program to find a solution for the following equation:

$$Z = \frac{\sqrt{X+Y}}{X} \qquad Where \quad X+Y \geq 0 \, , \, X > 0$$

Enter (X and Y) and print Z then display the message "Wrong Values", if the two conditions above are not satisfied.

***Sol:***

```
#include <iostream>
using namespace std;
int main ()
{
  float X, Y;
  float Z;
   cin >> X;
   cin>>Y;
   if ((X + Y) >= 0)
    {if (X > 0)
      Z = (sqrt (X + Y)) / X ;
      cout << "The value of Z is:"<< Z;
     }
   else
     {cout << "Wrong Values";
      }
return 0;
}
```

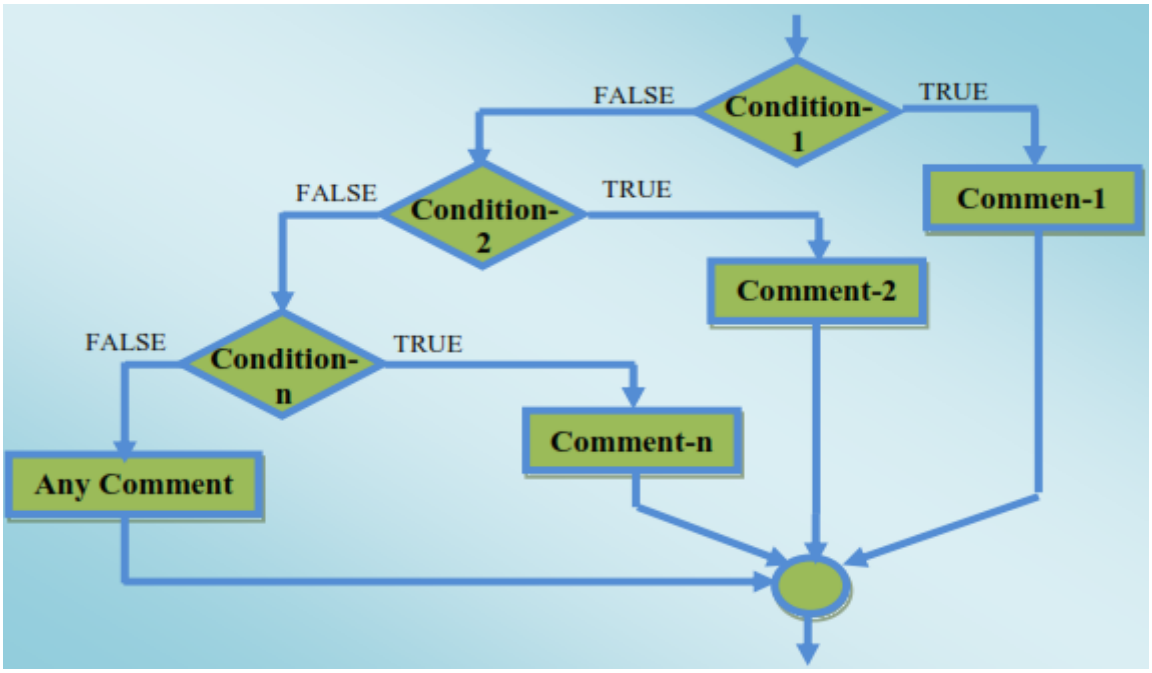
*Remember that in case that we want more than a single statement to be executed, we must group them in a block by enclosing them in braces { }.

## 4.3.4. IF – Else – IF statements

The **If** and **If-else** statement is great when you must test against more than two conditions, however, the **If** and **If-else** becomes difficult to maintain. Although the

logic of the **If-else** statement is simple, the coding is extremely difficult to follow. C++ language supports a statement, called **If-else-If**, which handles such multiple-choice conditions better than **If-else**. Here is the format of the **If-else-If** statement and also its flowchart shown in figure below:

```
If (Condition-1)
   Comment-1;
else If (Condition-2)
   Comment -2;
          .
          .
          .
else If (Condition-n)
   Comment-n;
else
   Any Comment;
```




***Ex:*** Using (if-else-if), Write a C++ program to show that if the value currently stored in x is positive, negative or none of them (i.e. zero):

```
if (x > 0)
cout << "x is positive";
else if (x < 0)
cout << "x is negative";
else
cout << "x is 0";
```

***Ex:*** Write C++ program to enter a number represents a centigrade degree. Find degree in Fahrenheit that generated from the first degree according to the relation:

$$F= (9/5) * C +32.$$

Then display the below phrases according to their equivalent Fahrenheit degree:

1. "Cold" when $F \leq 41$.
2. "Nice" when $41< F \leq 77$.
3. "Hot" when $F >77$.

***Sol:***

```
#include <iostream>
using namespace std;
int main ()
{
   float C,F;
   cin >> C;
   F = (9 / 5) * C + 32;
   cout << "F="<<F<<'\n';
       if (F <= 41)
          {cout << "Cold"<<'\n';}
       else if (F > 41)
            if(F <= 77)
            {cout << " Nice"<<'\n';}
       else if (F > 77)
            {cout << "Hot"<<'\n';}
   return 0;
}
```

## 4.4. Jump statements.

Jump statements divided to three statements:

1. The break statement
2. The continue statement
3. The goto statement

### 4.4.1. The break statement

Using break we can leave a loop even if the condition for its end is not fulfilled. It can be used to end an infinite loop, or to force it to end before its natural end. The

break statement must be used with switch case (selective statement) or with loops. Therefore, we will discuss this statement with switch statement.

### 4.4.2. The continue statement

The continue statement causes the program to skip the rest of the loop in the current iteration as if the end of the statement block had been reached, causing it to jump to the start of the following iteration. The continue statement example will contain loop. Therefore, we will discuss this statement in the next chapter.

### 4.4.3. The goto statement

goto allows to make an absolute jump to another point in the program. You should use this feature with caution since its execution causes an unconditional jump ignoring any type of nesting limitations.

The destination point is identified by a label, which is then used as an argument for the goto statement. A label is made of a valid identifier followed by a colon (:).

This instruction has no concrete use in structured or object-oriented programming aside from those that low-level programming fans may find for it.

***Ex:*** Write C++ program to make countdown loop using goto statement.

***Sol:***

```
#include <iostream>
using namespace std;
int main ()
        {
          int n, b;
           n=10;
        b: n--;
          cout << n << ", ";
        if (n==0)
          { return 0;}
        goto b;
        }
```

## 4.5. The selective statement (switch statement)

The switch statement objective is to check several possible constant values for an expression. Something similar to what we did at the beginning of this section with the concatenation of several if and else if instructions. Its form as the following:

```
switch (expression)
{
case constant1:
group of statements 1; break;
case constant2:
group of statements 2;break;
.
.
.
default:
default group of statements
}
```

***Ex:*** Write a C++ program to choose your country. The program contains a list of countries (Iraq, Germany, Lebanon, Egypt, and France). When we choose any of these countries, the city center of these countries is displayed. The countries will represented from (0) to (3) and fourth country will be any number except these three numbers.

***Sol:***

```
#include <iostream>
using namespace std;
int main ()
        {
        int n;
        cin >> n;
        switch (n)
        {case 0:
                cout<< "Baghdad";
        break;
        case 1:
                cout<< "Berlin";break;
        case 2:
                cout<< "Beirut";break;
        case 3:
                cout<< "Cairo";break;
        defoult:
                cout<< "Paris";break;
```

```
        }
return 0;
}
```

It works in the following way: switch evaluates expression and checks if it is equivalent to constant1, it executes group of statements 1 until it finds the break statement. When it finds this break statement the program jumps to the end of the switch selective structure.

# Questions of Chapter Four

**Q1.** Write a C++ program to enter a number then display the message "Even Number" if the entered number is even and the message "ODD Number" if it is odd.

**Q2.** Write a C++ program to enter a string then display the message boxes "Greater than 6 characters", "Less than 6 characters", and "Equal to 6 characters" if this number was greater, less, and equal to six characters respectively.

**Q3.** Write a C++ program to find W from the equations:

$$W = \begin{cases} X+Y & : X > 0 \, , \, Y < 0 \\ X^2Y^3 + 5X & : X = 0 \, , \, Y > 0 \\ \frac{2X}{3Y} + 4X & : X < 0 \, , \, Y = 0 \end{cases}$$

Print W for each input values of X and Y.

**Q4.** Write a C++ program to find the roots ($X_1$, $X_2$) for the quadratic equation: $ax^2+bx+c$ using the formula:

$$X_1, X_2 = \frac{-b \mp \sqrt{b^2 - 4ac}}{2a}$$

Enter the constants (a, b, c) then check the following points:

1. if a= 0 display the message "Divided by zero".
2. if b2 < 4ac display the message "No real roots".
3. if b2 = 4ac display the message "Equal roots" then find the roots from the formula above.
4. if b2 > 4ac display the message "real roots" then find the roots from the formula above.

**Q5.** Suppose the random bank offers 9% interest on balances of less than $5000, 12% for balances of $5000 or more but less than $10000, and 15% for balances of $10000 or more. Write a C++ program to enter a person balance then calculates a customer's new balance after one year.

**Q6.** Write a C++ program to enter a character represent the person gender (M: for male, F: for female) and a number represents person length ((L) in inch) then find and print its perfect weight ((W) in pound) according to the following relations:

For male (M) : W = (L × 4) - 125

For female (F): W = (L × 3.5) – 108

**Q7.** Write a C++ program to enter a number represents a person age then display the following on a message boxes:

1. "Wrong age" if the age less than or equal to 0 years.
2. "Child" if the age less than 8 years.
3. "Boy" if the age greater than or equal to 8 years.
4. "Young" if the age greater than or equal to 18 years.
5. "Old" if the age greater than or equal to 35 years.
6. "Very Old" if the age greater than or equal to 65 years.

**Q8.** Write a C++ program to find z from the below equations.

$$z = \begin{cases} 2k - \sin(k) & : k = 1 \\ 2k^3 - 3 & : k = 2 \\ k - 2 & : k = 3 \\ k & : k < 1 \; or \; k > 3 \end{cases}$$

Print z for each input value of k.

**Q9.** Write a C++ program to find x from the below equations according to your choice entry from (1 to 4).

1. $x = \sin(t) + \tan(t)$
2. $x = \cosh(t^3 + 2t)$
3. $x = \sec^2(4t) + t^2$
4. $x = 0.25\,t^4 + t^2$

Print x for each input value of t.

**Q10.** Write a C++ program to find and print T from the below equations where a and b are input variables.

$$T = \begin{cases} a^2 + ab + \sqrt{ab} & \mathbf{a < b} \quad \textit{AND} \quad \mathbf{a < 10} \\ ab^2 - 2a + 5b & \mathbf{a = b} \quad \textit{OR} \quad \mathbf{b > 10} \end{cases}$$

**Q11.** Define f(x) as follows:

$$f(x) = \begin{cases} 3 - x & \text{if } x < 2 \\ 2 & \text{if } x = 2 \\ x/2 & \text{if } x > 2 \end{cases}$$

Write a C++ program to calculate f(x) from the above equations. Then, print f(x) for any input value of x.