

Introduction:

a *statement* is a part of your program that can be executed. That is, a statement specifies an action. C++ categorizes statements into these groups:

1 – Selection

2 - Iteration

3 - Jump

4 - Label

5 - Expression

6 - Block

Included in the selection statements are **if** and **switch**. (The term *conditional statement* is often used in place of "selection statement.") The iteration statements are **while**, **for**, and **do-while**. These are also commonly called *loop statements*. The jump statements are **break**, **continue**, **goto**, and **return**. The label statements include the **case** and **default** statements (discussed along with the **switch** statement) and the label statement (discussed with **goto**). Expression statements are statements composed of a valid expression. Block statements are simply blocks of code. Remember, a block begins with a { and ends with a }. Block statements are also referred to as *compound statements*.

Selection Statements :

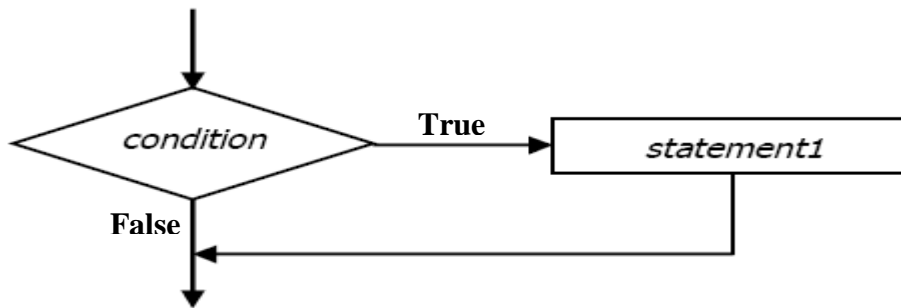
Conditional expressions are mainly used for decision making. C++ provides multiple selection structures: **if**, **if/else**, **if else if ladder**, **nested if** and **switch**.

A) The Single If Statement Structure:

The *if* statement is used to express conditional expression. If the given condition is true then it will execute the statements; otherwise it will execute the optional statements.

The general form of single if statement is :

```
if ( expression or condition )  
    statement ;
```



Example 1: `if (average >= 50)`
 `cout << "Pass";`

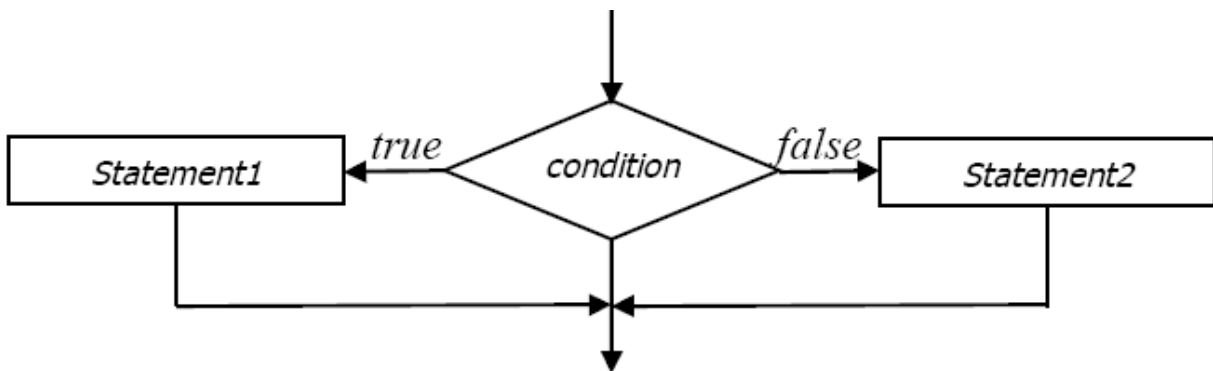
Example 2: `if (x % 2 == 0)`
 `sum = sum + x;`

Example 3: `if (num == 0)`
 `zcount = zcount + 1;`

B) if else :

The general form of the **if ... else** statement is :

```
if ( expression )  
    statement1;  
else  
    statement2 ;
```



where a *statement* may consist of a single statement, a block of statements, or nothing (in the case of empty statements). The **else** clause is optional. If *expression* evaluates to true (anything other than 0), the statement or block that forms the target of **if** is executed; otherwise, the statement or block that is the target of **else** will be executed, if it exists. Remember, only the code associated with **if** or the code associated with **else** executes, never both.

Example 1 :

```
if ( a > b )
    cout<< " a is larger " ;
else
    cout<< " b is larger " ;
```

Example 2 :

```
if ( number % 2 == 0 )
    cout<< " The number is even" ;
else
    cout<< " The number is odd" ;
```

C) The if-else-if Ladder :

A common programming construct is the *if-else-if ladder*, sometimes called the *if-else-if staircase* because of its appearance. Its general form is

```
if (expression)
    statement ;
else
    if (expression)
        statement ;
.
.
else
    statement;
```

Example 1 : Write a C ++ program to Find the largest integer out off three inserted integer numbers.

```
#include<iostream >
main()
{
    int  x , y , z , max;
    cout<<"Enter the three integer numbers";
    cin >>x>>y>>z;
    if( x>y &&  x>z)
        max=x;
    else
        If (y>x  &&  y>z)
            max=y;
        else
            max=z;
    cout<< "the large number is "<<max;
}
```

Example 2: Write a C++ program to read (input) one character then print the type of it (uppercase , lowercase, digit, special) character .

```
#include <iostream>
main()
{
    char c;
    cout<<"Enter the character";
    cin>>c;
    if( c>='A' && c<= 'Z')
        cout<< "the character is uppercase";
    else
        if( c>='a' && c<= 'z')
            cout<< "the character is lowercase";
        else
            if( c>='0' && c<= '9')
                cout<< "the character is digit";
            else
                cout<< "the character is special";
}
```

D) The Switch statement :

C++ has a built-in multiple-branch selection statement, called **switch**, which successively tests the value of an expression against a list of integer or character constants. When a match is found, the statements associated with that constant are executed. The general form of the **switch** statement is :

```
switch (expression)
{
    case constant1:
        statement sequence
        break;
    case constant2:
        statement sequence
        break;
    .....
    .....
    Default :
        statement sequence }
```

The *expression* must evaluate to a character or integer value. Floating-point expressions, for example, are not allowed. The value of *expression* is tested, in order, against the values of the constants specified in the **case** statements. When a match is found, the statement sequence associated with that **case** is executed until the **break** statement or the end of the **switch** statement is reached. The **default** statement is executed if no matches are found. The **default** is optional and, if it is not present, no action takes place if all matches fail.

Example 3: Write a C++ program to read integer number which represent the day No. in week, and print the name of the day in a week.

```
#include<iostream >
void main( )
{
    int day;
    cout << "Enter the day No.";
    cin >> day;
    switch (day)
    {
        case 1: cout << "Sunday"; break;
        case 2: cout << "Monday"; break;
        case 3: cout << "Tuesday"; break;
        case 4: cout << "Wednesday"; break;
        case 5: cout << "Thursday"; break;
        case 6: cout << "Friday"; break;
        case 7: cout << "Saturday"; break;
        default: cout << "Invalid day No.";
    }
}
```

Example 4 : Write C++ program to read two integer numbers, and read the operation to perform on these numbers.

```
#include <iostream >
void main( )
{
    int x, y;
    char op;
    cout << "Enter two numbers"<<endl;
    cin >> x >> y;
    cout << "enter the operation "<<endl;
    cout << "+ for addition"<<endl;
    cout << "- for subtraction "<<endl;
    cout << "* for multiplication"<<endl;
    cout << "/" for division "<<endl;

    cin >> op;
    switch ( op )
    {
        case '+':  cout << a + b;
        break;
        case '-':  cout << a - b;
        break;
        case '*':  cout << a * b;
        break;
        case '/':  cout << a / b;
        break;
        default:  cout<<"Invalid operator";
    }
}
```

Work Sheet

Exercise 1. Write a program that reads two integer numbers and determines and prints if the first is a multiple of the second .

Exercise 2. Write a program that inputs three integer numbers from the keyboard and prints the sum, average, product, smallest and largest of these numbers.

Exercise 3. Write a program that will first allow a user to choose one of two options:

1. Convert a temperature from degrees Celsius (centigrade) to degrees Fahrenheit.
2. Convert a temperature from degrees Fahrenheit to degrees Celsius.

Exercise 4. Write a program that prompts the user to enter the date as three integer values for the day, month and the year. The program should then output the date in the form 31 December 2018 when the user enters 31 12 2018, for example.

Exercise 5. Write a program that will calculate the price for a quantity entered from the keyboard, given that the unit price is \$5 and there is a discount of 10 percent for quantities over 30 and a 15 percent discount for quantities over 50.

Exercise 6. Write a program that will calculate the price for a quantity entered from the keyboard, given that the unit price is \$5 and there is a discount of 10 percent for quantities over 30 and a 15 percent discount for quantities over 50.

Exercise 7. Write a program to read any time (hour , minuet and second) then find and display next time after one second.