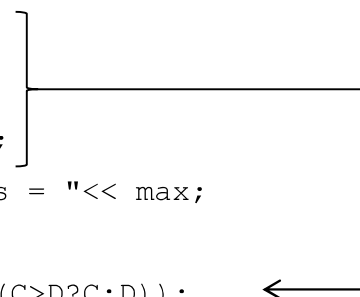


```

#include<iostream.h>
void main( )
{
    int A, B, C, D, max;
    cout<< "Enter four numbers : " ;
    cin>>A>>B>>C>>D ;
    int max1=(A > B) ? A:B ;
    int max2=(C > D) ? C:D ;
    max=(max1 > max2) ? max1:max2 ;
    cout<< "Maximum of four numbers = "<< max;
}
max=( (A>B?A:B)>(C>D?C:D) ? (A>B?A:B) : (C>D?C:D) );

```



H.W Use the selection operator (? :) to find the maximum of six integers.

3. The switch STATEMENT (Multiple Choice Statement)

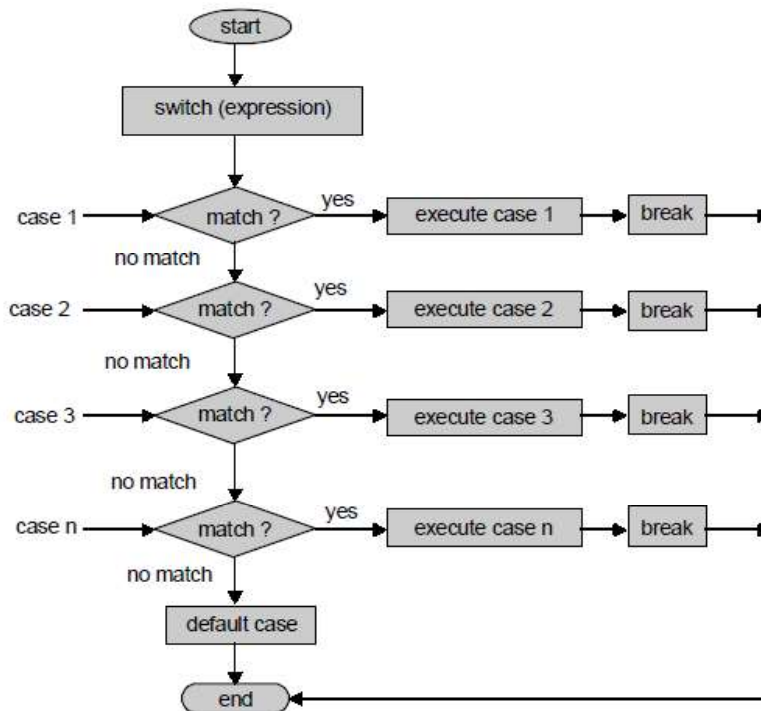
When a multiple selection is required we may use switch statement which is illustrated below:

```

switch (expression or variable )
{
    case value1 : statement1; break;
    case value2 : statement2; break;
    .....
    case value n : statement n; break;
    default : statement;
}

```

During execution of the program, the expression is evaluated and compared with the values mentioned in different cases of switch expression. If the value matches a value of a particular case, the statements in that case are executed. If no case-value matches with the value of the expression the program goes to the last statement which is a default statement as shown in figure below:



Note The word ***break*** means exit from ***switch*** statement.

The following program illustrate the ***switch*** statement.

```

#include<iostream.h>
void main( )
{
    int day;
    cout<<" Enter the week day (1-7)";
    cin>> day;
    switch (day)
    {
        case 1: cout<<"The day is Sunday" ; break;
        case 2: cout<<"The day is Monday" ; break ;
        case 3: cout<<"The day is Tuesday" ; break;
        case 4: cout<<"The day is Wednesday" ; break;
        case 5: cout<<"The day is Thursday" ; break;
        case 6: cout<<"The day is Friday" ; break;
        case 7: cout<<"The day is Saturday" ; break;
        default: cout<<"The number is not in range.";
    }
}

```

Ex: Write a program to receive an arithmetic operator and two integers, the program performs the arithmetic operation on the two numbers (use switch statement).

```
#include<iostream.h>
void main( )
{
    char ch;
    int x,y;
    cout<<"Enter the arithmetic operator : "<<endl;
    cin>>ch;
    cout<<"Enter the two numbers : "<<endl;
    cin>>x>>y;
    switch(ch)
    {
        case '+' : cout<<x + y ; break;
        case '-' : cout<<x - y ; break;
        case '*' : cout<<x * y; break;
        case '/' : cout<<x / y; break;
        case '%' : cout<<x % y; break;
        default : cout<<" Error try again";
    }
}
```

H.W.

Write a program to find the value of y from the following (using switch statement).

$$y = \begin{cases} \sqrt{(x+5)^3} & x = -1 \\ 2x^3 + 5x + 4 & x = 0 \\ x - \sin(x) & x = 1 \\ 5 & \text{otherwise} \end{cases}$$

LOOP AND OTHER CONTROL STATEMENTS

In C++ programming language, there are three loop statements, they are:

1. The while statement (loop) .
2. The do...while statement (loop) .
3. The for statement (loop).

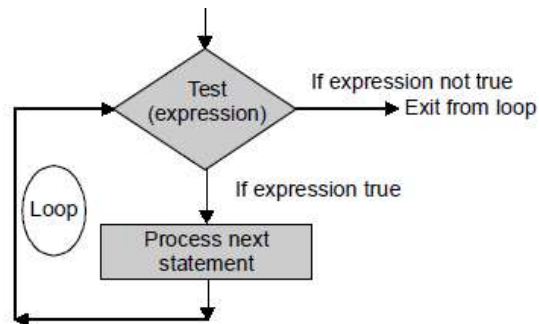
1. THE while STATEMENT.

The while statement or loop is as illustrated below :

```
while (conditional expression)
{
    Block_of_statements ;
}
```

This means that as long as the condition is true, *Block_of_statements* will be executed.

Ex: `while (i <= n)`
`sum +=i++;`



Ex: Use while loop to find the sum of numbers from 1 to 10.

```
#include <iostream.h>
void main( )
{
    int n = 10, i= 0, sum = 0;
    while (i <= n)
        Sum += ++i;
    cout<< "Sum = "<< sum;
}
```

The following loop is called endless loop because the condition is always true:

```
while (true)
    statement;
```

Ex: Write a program to find the sum of squares of integers.

```
#include<iostream.h>
main( )
{
```

```

int i = 1, n, sum = 0;
cout << "Enter a positive integer:";
cin>>n;
while (i <= n)
{
    sum += i*i;
    i++;
}
cout << " sum = " << sum ;
return 0;
}

```

THE NESTED while STATEMENTS

When more than one parameter such as i and j are to be varied in a program, two loops are required. The i loop is the outer loop and j loop is the inner loop. The code may be written is illustrated below.

```

while (int i < n)
{
    while (int j < m)
    {
        statements ;
    }
}

```

The following program illustrates nested while loops by finding the the value of Z :

$$Z = \sum_{i=0}^5 \sum_{j=0}^4 i * j$$

```

#include <iostream.h>
//The program illustrates nested while loop
void main()
{
    int i=0, Z=0;
    while (i<=5) // outer while loop
    {
        int j = 0;

```

```

        while (j<=4) // inner while loop
        {
            Z+=(i*j);
            ++j
        }
        ++i;
    }
    cout<<"Z="<<Z;
}

```

2. THE do...while LOOP

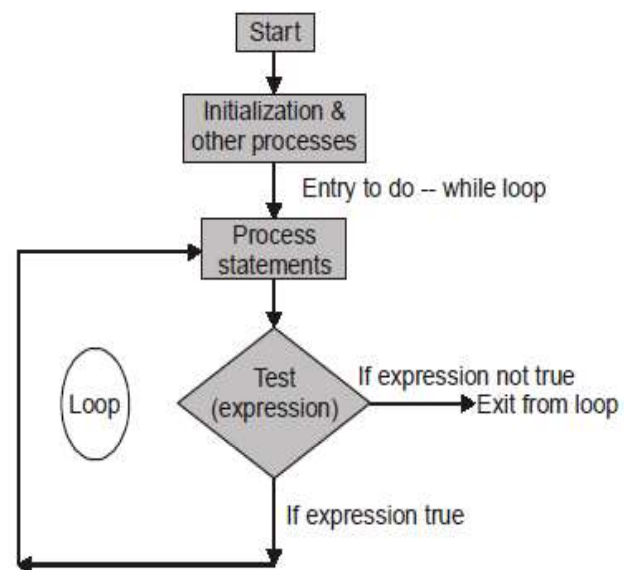
The do...while statement is almost the same as the while statement. Its syntax is:

```

do
{
    Block_of_statements;
}
while (condition);

```

The only difference between while and do...while is that the do...while statement executes the statements first and then tests the condition. These two steps are repeated until the condition becomes false. A do...while loop always iterates at least once, regardless of the value of the condition, because the statement executes before the condition is evaluated.



Ex: Write a program to evaluate the factorial of an integer.

```

#include<iostream.h>
void main()
{
    int n, f = 1;
    cout << "Enter a positive integer: ";
    cin >> n;
}

```

```

do
{
    f *= n;
    --n;
}
while (n > 1);
cout <<"factorial of"<<n<<" is"<< f ;
}

```

3. THE *for* LOOP

The *for* loop is written as given below: -

```

for (initial value ; condition ; increment /decrement)
{
    Block_of_statements ;
}

```

The *for* loop is controlled by three expressions: an *initialization*, a *condition*, and *update* (*increment/decrement*).

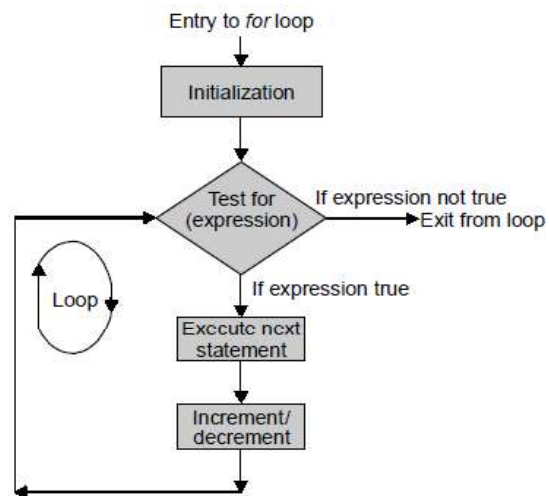
Ex:

```

for (i=0 ; i<=10 ; i++)
    x=i+2 ;

```

The statement ($x=i+2$) is executed repeatedly as long as ($i \leq 10$).



Ex: Write a program to evaluate the factorial of an integer.

```

#include<iostream.h>
void main()
{
    int n, f=1,i;
    cout << "Enter a positive integer: ";
    cin >> n;
    for(i=1;i<=n;++i)

```