

While(keyword)

Repeats execution

Syntax: while (<expression>) <statement>

<statement> is executed repeatedly as long as the value of <expression> remains non-zero. The test takes place before each execution of the <statement>.

You'll find that **while** statements are just as easy to use as if statements. For example:

```
while (a < b)
{
    cout << a;
    a = a + 1;
}
```

This causes the two lines within the braces to be executed repeatedly until **a** is greater than or equal to **b**. The while statement in general works as illustrated to the right.

The **for loop** in C++ is simply a shorthand way of expressing a while statement. For example, suppose you have the following code in C++:

```
x=1;
while (x<10)
{
    X=x+1;
}
```

You can convert this into a for loop as follows:

```
for(x=1; x<10; x=x+1)
{
}
```

Note that the while loop contains an initialization step (**x=1**), a test step (**x<10**), and an increment step (**x++**). The for loop lets you put all three parts onto one line, but you can put anything into those three parts. For example, suppose you have the following loop:

```
a=1;
b=6;
while (a < b)
{
    a=a+1;
    cout << a;
}
```

You can place this into a for statement as well:

```
for (a=1,b=6; a < b; a=a+1,cout <<a);
```

It is slightly confusing, but it is possible. The **comma operator** lets you separate several different statements in the initialization and increment sections of the for loop (but not in the test section). Many C++ programmers like to pack a lot of information into a single line of C++ code; but a lot of people think it makes the code harder to understand, so they break it up.

Q) Write a program to print number from 1..5 in ascending order

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int i;
    clrscr();

    i=1;
    while (i<=5)
    {
        cout <<i<<" ";
        i=i+1;
    }

    cout << "\n\nHit any key to continue";
    getch();
    return 0;
}
```

Q) Write a program to print number from 5..1 in descending order

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int i;
    clrscr();

    i=5;
    while (i>=1)
    {
        cout <<i<<" ";
        i=i-1;
    }

    cout << "\n\nHit any key to continue";
    getch();
    return 0;
}
```

Q) Write a program to search for first number 6 between 5 different numbers

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int n,i;
    clrscr();

    cout <<"Number:=";
    cin>> n;
    i=1;
    while ((n!=6) && (i<=4))
    {
        i=i+1;
        cout <<"Number:=";
        cin>> n;
    }

    if (n==6)
        cout <<" --- Number 6 is found ---";
    cout << "\n\nHit any key to continue";
    getch();
    return 0;
}
```

Q) Write a program to find the summation of numbers of group of numbers and stop when the result exceed 50 :

```
#include <iostream.h>
#include <conio.h>
int main()
{
int sum=0,n;
clrscr();

while (sum<=50)
{
cout <<"Number:=";
cin>> n;
sum=sum+n;
}
cout <<"Sum.:="<<sum;
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to find the summation of the following series and stop when the result exceed 7 :

$$sum = 4 + \frac{1}{2} + \frac{3}{3} + \frac{5}{4} \dots$$

```
#include <iostream.h>
#include <conio.h>
int main()
{
float sum=4;
int i=1,j=2;
clrscr();

while (sum<=7)
{
sum=sum+(float )i/j;
cout <<"j:="<<j<<" i="<<i<<"\n"; // for test
i=i+2;
j=j+1;
}
cout <<"Sum.:="<<sum;
cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Notes:

While (condition)

```
{  
    While (condition)  
    {  
    }  
}
```

// or

While (condition)

```
{  
    for ( expression_1; expression_2; expression_3 )  
    {  
    }  
}
```

// or

```
for ( expression_1; expression_2; expression_3 )  
{  
    While (condition)  
    {  
    }  
}
```

Do...while loop

Syntax: do <statement> while (<expression>);

<statement> is executed repeatedly as long as the value of <expression> remains non-zero.

The test takes place AFTER each execution of the <statement>.

Example1:

```
i = 1; n = 1;
do {
    n =n* i;
    i=i+1;
} while (i <= factorial);
```

Example2:

```
do
{
    cout << a;
    a = a + 1;
}
while (a < b);
```

Q) Write a program to print number from 1..5 in ascending order

```
#include <iostream.h>
#include <conio.h>
int main()
{
int i;
clrscr();

    i=1;
    do
    {
        cout <<i<<" ";
        i=i+1;
    } while (i<=5);

cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to print number from 5..1 in descending order

```
#include <iostream.h>
#include <conio.h>
int main()
{
int i;
clrscr();

    i=5;
    do
    {
        cout <<i<<" ";
        i=i-1;
    } while (i>=1);

cout << "\n\nHit any key to continue";
getch();
return 0;
}
```

Q) Write a program to calculate the area or circumference of rectangle using menu, add choice 3 for exit

```
#include <iostream.h>
#include <conio.h>
int main()
{
int a;
int l,w,r;
clrscr();
cout << "Length:=";
cin >> l;
cout << "Width:=";
cin >> w;

do
{
clrscr();
cout << "*****\n";
cout << "* 1: Area of rectangle * \n";
cout << "* 2: Circumference of rectangle * \n";
cout << "* 3: Exit * \n";
cout << "*****\n\n";

cout << "\nChoice:=";
cin >> a;
switch (a)
{
case 1:
    r=l*w;
    cout << "\n Area of rectangle:="<<r;
    break;
case 2:
    r=(l+w)*2;
    cout << "\n Circumference of rectangle:="<<r;
    break;
case 3:
    cout << "\n\nGood bye at exit";
    break;
}
cout << "\n\nHit any key to continue";
getch();

} while (a!=3);
return 0;
}
```

Exit

exit terminates the program

Remarks:

exit terminates the calling process. Before termination, exit does the following:

- closes all files
- writes buffered output (waiting to be output)
- calls any registered "exit functions" (posted with at exit)

Q) Write a program to calculate the area or circumference of rectangle using menu, add choice 3 for exit

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
int main()
{
int a;
int l,w,r;
clrscr();
cout << "Length:=";
cin >> l;
cout << "Width:=";
cin >> w;
for (;;)
{
clrscr();
cout << "*****\n";
cout << "* 1: Area of rectangle * \n";
cout << "* 2: Circumference of rectangle * \n";
cout << "* 3: Exit * \n";
cout << "*****\n\n";
cout << "\nChoice:=";
cin >> a;
switch (a)
{
case 1:
r=l*w;
cout << "\n Area of rectangle:="<<r;
break;
case 2:
r=(l+w)*2;
cout << "\n Circumference of rectangle:="<<r;
break;
case 3:
cout << "\n\nGood bye at exit, hit any key to continue";
getch();
exit (0);
break;
}
cout << "\n\nHit any key to continue";
getch();
}
}
```

Break (keyword)

Passes control

Syntax: break ;

The break statement causes control to pass to the statement following the innermost enclosing while, do, for, or switch statement. It causes a loop to terminate early. When it is encountered, the loop stops and the program jumps to the statement immediately following the loop.

Q) Write a program to search for first number 6 between 5 different numbers

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int n;
    clrscr();

    for (int i=1;i<=5;i=i+1)
    {
        cout <<"Number:=";
        cin>> n;
        if (n==6)
        {
            cout <<" --- Number 6 is found ---";
            break;
        }
    }
    cout << "\n\nHit any key to continue";
    getch();
    return 0;
}
```

Continue (keyword)

Syntax: continue ;

Causes control to pass to the end of the innermost enclosing while, do, or for statement, at which point the loop continuation condition is re-evaluated. It causes a loop to stop its current iteration and begin the next one. When continue is encountered, all the statements in the body of the loop that appear after it are ignored, and the loop prepares for the next iteration.

Q) Write a program to find the summation of 4 different numbers except number 3 or 5

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int n;
    clrscr();
    for (int i=1,sum=0;i<=4;i=i+1)
    {
        cout <<"Number:=";    cin>> n;
        if ((n==3) || (n==5))
            continue;
        sum=sum+n;
    }
    cout<<"\n\nSum.:"<<sum;
    cout << "\n\nHit any key to continue"; getch();
    return 0;
}
```