

جامعة بغداد / كلية التربية للعلوم الصرفة ابن الهيثم / قسم علوم الحاسبات
المرحلة الاولى / نظري
السنة الدراسية 2024-2025

Structured Programming

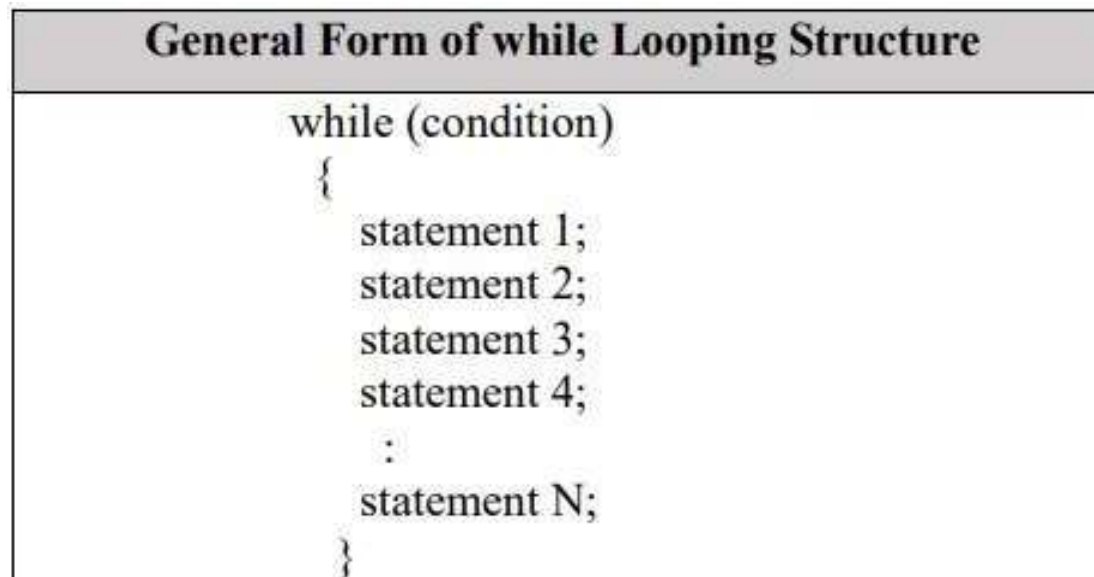
Lecture 8 – Looping Statements

Looping (Repetition) Statements

- Sometimes it is necessary to repeat a set of statements several times. If you want to repeat a set of statements 100 times, you type the set of statements 100 times in the program.
- However, this solution of repeating a set of statements is impractical, if not impossible. Fortunately, there is a better way to repeat a set of statements. C++ has three repetitions, or looping, structures that allow you to repeat a set of statements until certain conditions are met as while, do-while, and for loop.

while Looping (Repetition) Structure

- In C++, while is a reserved word. Of course, the statement can be either a simple or compound statement.
- The expression acts as a decision maker and is usually a logical expression.
- The statement is called the body of the loop. Note that the parentheses around the expression are part of the syntax.



Example 1:

```
i = 0 ;  
while ( i < 10 )  
{  
    cout << i ;  
    i ++ ;  
}
```

Output:

0 1 2 3 4 5 6 7 8 9

Example 2:

```
i = 0 ;  
while ( i < 10 )  
{  
    cout << i ;  
    i += 2 ;  
}
```

Output:

0 2 4 6 8

Example 3: Write a C++ program to read 10 numbers and print only even numbers.

```
#include<iostream.h>
# include<conio.h>
main ( )
{
int x, i ;
    i = 0 ;
    while ( i < 10 )
    {
        cin >> x ;
        if ( x % 2 == 0 )
            cout << x << " is even \n" ;
        i = i + 1 ;
    }
    getch ( ) ;
}
```

Output:

1 3 5 6 8 8 2 9 0 6

6 is even

8 is even

8 is even

2 is even

0 is even

6 is even

Example 4: Write a C++ program to find the summation of the following series.

$$\text{sum} = 1 + 3 + 5 + 7 + \dots + 99.$$

```
#include<iostream.h>
# include<conio.h>
main ( )
{
int i, sum = 0 ;
    i = 1 ;
    while ( i <= 99 )
    {
        sum = sum + i ;
        i = i + 2 ;
    }
    cout << "sum=" << sum ;
getch ( ) ;
}
```

<u>Output:</u> sum=2500

Example 5: Write a C++ program to find the summation of the following series.

$$\sum_{i=1}^n i^2 = 1^2 + 2^2 + 3^2 + \dots + n^2$$

```
#include<iostream.h>
# include<conio.h>
main ( )
{
int i=1, n, sum = 0 ;
cout << "enter positive number" ;
    cin >> n ;
    while ( i <= n )
    {
        sum += i * i ;
        i ++ ;
    }
    cout << "sum is: " << sum << endl ;
    getch ( ) ; }
```

Output:

enter positive number 3
sum is:14

Example 6: Write a C++ program to read a sequence of numbers and print them, the program stops when the number equal to zero.

```
#include<iostream.h>
# include<conio.h>
main ( )
{
int x ;
cout << " Enter number" ;
cin >> x ;
while ( x != 0 )
{
cout << "x is" << "\t" << x << " \n" ;
cin >> x ;
}
getch ( ) ;
}
```

Output:

Enter number

4

x is 4

9

x is 9

7

x is 7

3

x is 3

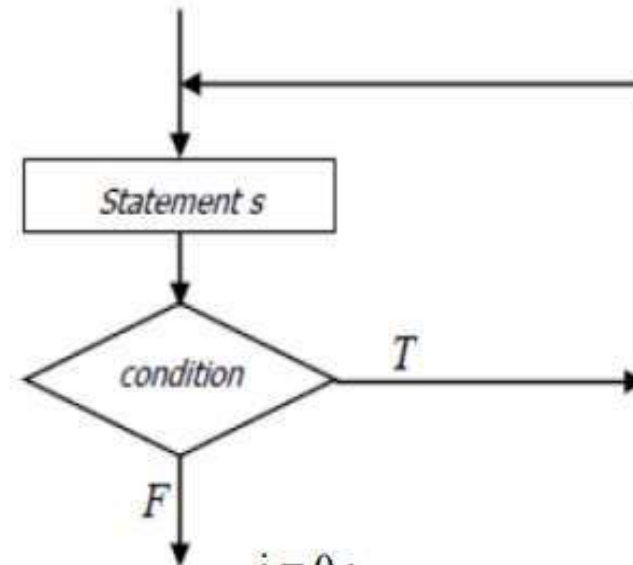
0

Do / While Looping (Repetition) Structure

- do-while loop is evaluated after the execution of statement instead of before, granting at least one execution of statement even if condition is never fulfilled

General Form of do / while Looping Structure

```
do
{
    statement 1;
    statement 2;
    statement 3;
    :
    statement N;
}
while (condition);
```



```
i = 0 ;
do
{
    cout << i ;
    i += 2 ;
}
while ( i < 10 );
```

Output:
0 2 4 6 8

Example 9: Write a C++ program to find the summation of student marks, and its average, assume the student have 8 marks.

```
#include<iostream.h>
# include<conio.h>
main ( )
{
int mark, sum = 0, i ;
float avg;
i = 1 ;
do
{
    cin >> mark ;
    sum += mark ;
    i ++ ;
}
while ( i <= 8 );
cout << "sum= " << sum << "\n " ;
avg = sum / 8 ;
cout << " the average is: " << avg ;
getch ( ) ;
}
```

Output:

```
6 7 3 1 2 2 2 1
sum= 24
the average is: 3
```

Example 10: Write a C++ program to read a list of integer numbers ended by (-100) and count the number of list and the product and the average.

```
#include<iostream.h>
# include<conio.h>
main ( )
{
    int x, k, p ;
    float ave, s ;
    cin >> x ;
    s = 0 ; k = 0 ; p = 1 ;
    do
    {
        s = s + x ;
        k = k + 1 ;
        p = p * x ;
        cin >> x ;
    }
    while ( x != -100 );
    ave = s / k ;
    cout << s << endl << k << endl << p << endl << ave ;
    getch ( ) ;
}
```

Output:

1 2 3 4 2 -100

S=12

K=5

P=48

Ave=2.4

Example 11: Write C++ program to find the sum of positive numbers if the user enters the negative numbers the loop ends.

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
main( ) {
```

```
    int number = 0;
```

```
    int sum = 0;
```

```
    do
```

```
    {
```

```
        cout << "Enter a number: ";
```

```
        cin >> number;
```

```
        sum += number;
```

```
    }
```

```
    while (number >= 0);
```

```
    cout << "The sum is " << sum << endl;
```

```
    getch ( ) ;
```

```
}
```

Output:

Enter a number: 6

Enter a number: 5

Enter a number: 9

Enter a number: 1

Enter a number: 4

Enter a number: -1

The sum is 25

Different between while and Do while loop

while	do-while
Condition is checked first then statement(s) is executed.	Statement(s) is executed atleast once, thereafter condition is checked.
It might occur statement(s) is executed zero times, If condition is false.	At least once the statement(s) is executed.
No semicolon at the end of while. while(condition)	Semicolon at the end of while. while(condition);
If there is a single statement, brackets are not required.	Brackets are always required.
<pre>while(condition) { statement(s); }</pre>	<pre>do { statement(s); } while(condition);</pre>

Different between while and Do while loop

```
// while loop example
int i = 1; // initialize a variable
while (i <= 5) { // check the
condition
    cout << i << " "; // print the
variable
    i++; // increment the variable
}
// output: 1 2 3 4 5
```

```
// do-while loop example
int j = 1; // initialize a variable
do
{
    cout << j << " "; // print the
variable
    j++; // increment the variable
}
while (j <= 5); // check the
condition
// output: 1 2 3 4 5
```

Different between while and Do while loop

```
// while loop example
int i = 6;
while (i <= 5)
{
    cout << i << " ";
    i++;
}
// Output: nothing
```

```
// do-while loop example
int i = 6;
do
{
    cout << i << " ";
    i++;
}
while (i <= 5);
// Output: 6
```