

# Programming in C++

## Lecture Notes 3

### Loops (Repetition)

## Structures

- Sequential
- Branching
- Repeating

## Loops

- Repetition is referred to the ability of repeating a statement or a set of statements as many times this is necessary.

3

## The teacher of physical education said:

- Run around the football-field until I tell you to stop.
- Run around the football-field five times.

4

## Loops

while()

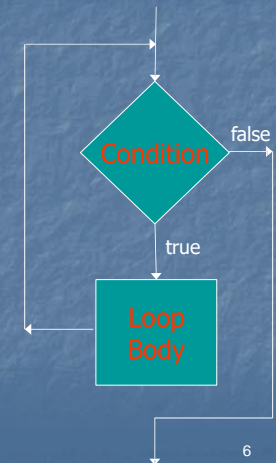
do – while()

for( ; ; )

5

## The while () Loop

```
while (<Condition>)  
<Loop body> ;
```



6

## The while() Loop

```
while (<Condition>
    <Statement> ;
```

```
while (<Condition>
{
    <Statement 1> ;
    <Statement 2> ;
    . . .
}
```

7

```
cout << " * \n";
cout << " *** \n";
cout << "*****\n";
cout << " * \n";
cout << " * \n";
cout << endl;
```

```
*
***
*****
*
*
```

```
*
***
*****
*
*
*
***
*****
*
*
*
***
*****
*
*
```

8

## Increment – Decrement Operators

- ++** Increase the value of a variable by one.
- Decrease the value of a variable by one.

### Post-increment

```
x++; // Increases x by 1, i.e. x = x + 1;
```

### Post-decrement

```
x--; // Decreases x by 1, i.e. x = x - 1;
```

### Pre-increment

```
++x; // Increases x by 1, i.e. x = x + 1;
```

### Pre-decrement

```
--x; // Decreases x by 1, i.e. x = x - 1;
```

## Increment – Decrement Operators

The difference is when they are used inside expressions.

Pre-operators: will increase/decrease the variable and then evaluate the expression.

Post-operators: will evaluate the expression and then increase/decrease the variable.

```
int x, a = 5;
x = a++;
x = 5, a = 6
```

```
int x, a = 5;
x = ++a;
x = 6, a = 6
```

```
int x, a = 5;
x = a--;
x = 5, a = 4
```

```
int x, a = 5;
x = --a;
x = 4, a = 4
```

```

int i = 1;
while (i <= 4)
{
    cout << " * \n";
    cout << " *** \n";
    cout << "***** \n";
    cout << " * \n";
    cout << " * \n";
    cout << endl;
    i++;
}

```

Variable  
i  
5

Screen

```

*
***
*****
*
*
***
*****
*
*
*
***
*****
*
*

```

11

```

int i = 1;
while (i < 11)
{
    i += 3;
    cout << i << endl;
}

```

Variable  
i  
13

Screen

```

4
7
10
13

```

12

## Examples

```
int i = 1;
while (i<=5) {
    cout << i << ' ';
    i++;
}
```

1 2 3 4 5

```
int i = 1;
while (i<=5) {
    cout << 'i' << ' ';
    ++i;
}
```

i i i i i

12

## More Examples

```
int i = 1;
while (i<=5)
    cout << i << ' ';
    i++;
```

1 1 1 1 1 1 ...

```
int i = 1;
while (i<=5)
    cout << i++ << ' ';
```

1 2 3 4 5

13

## More Examples

```
int x;
while (cin >> x, x!=-999)
    cout << x << ' ';
```

```
2 3 -6 -999
2 3 -6
```

```
int x;
while (cin >> x, x!=-999);
    cout << x << ' ';
```

```
2 3 -6 -999
-999
```

15

## More Examples

```
int i = 1;
while (i++ < 5)
    cout << i << ' ';
```

```
2 3 4 5
```

```
int i = 1;
while (++i < 5)
    cout << i << ' ';
```

```
2 3 4
```

16



## Exercises

### 1. What is the output of the following segments?

```
(a) int i = -3;
while (i != 3)
{
    cout << i << " ";
    i = i + 1;
}
```

```
(b) int i = 0, sum = 0;
while (i <= 10)
{
    sum += i;
    i++;
}
cout << "Sum = " << sum;
```

```
(c) int i = 1;
while (i++ <= 5)
    cout << i << " ";
```

```
(d) int i = 10;
while (i > 3)
{
    cout << i << endl;
    i = i - 2;
}
```

```
(e) int i = 6;
while (i-- > 1)
    cout << i << '\n';
```

```
(f) int i = 0;
while (++i < 8)
    cout << i;
```

```
(g) int i = -3;
while (++i <= 3);
    cout << 'i';
```

17

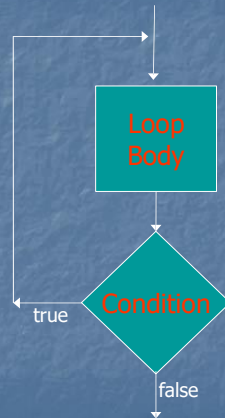
## Exercises

- Write a program to display the numbers from 1 to 100 inclusive.
- Write a program to display all the letters of the Latin alphabet.
- Write a program to calculate the average of the integer numbers between 15 and 25, inclusive.
- Write a program to display the odd numbers from 1 to 101, and also to display their sum.
- Write a program to read the radian of a circle, check if it is bigger than zero, and if it is to calculate and display the perimeter of the circle using the formula  $P=2*3.14*R$ , where P is the perimeter and R is the radius of the circle. Otherwise it should prompt for the radius again until it is bigger than zero.
- The powers of 2 are: 1, 2, 4, 8, 16, 32, ... . Write a program to display the first power of 2, which is bigger than 1000.
- Write a program to calculate the sum of:  $1^2 + 2^2 + 3^2 + \dots + N^2$ .

18

## The do-while() Loop

```
do
  <Loop body> ;
while(<Condition>);
```



19

## The do - while() Loop

```
do
  <Statement> ;
while (<Condition>) ;
```

```
do {
  <Statement 1> ;
  <Statement 2> ;
  . . .
} while (<Condition>) ;
```

20

```

int i = 1;
do {
    cout << " * \n";
    cout << " *** \n";
    cout << "*****\n";
    cout << " * \n";
    cout << " * \n";
    cout << endl;
    i++;
} while(i <= 4);

```

Variable  
i  
5

Screen

```

*
***
*****
*
*
***
*****
*
*
***
*****
*
*
***
*****
*

```

21

```

int i = 1;
do
{
    i += 3;
    cout << i << endl;
} while(i < 11);

```

Variable  
i  
13  
0

Screen

```

4
7
10
13

```

22

## The `for( ; ; )` Loop

```
for (<init> ; <condition> ; <change>)  
<Loop body> ;
```

**init** is usually an assignment to give a loop counter an initial value. Executed ONLY when entering the loop.

**condition** is any statement returning an integral value and for as long as it is true, the **statement** will be executed. Executed at every pass.

**change** is a statement normally to modify the loop counter, so that eventually it will make **condition** false and the loop will terminate. Executed at every pass after the execution of the loop body.

23

## Example 1

```
int i;  
for(i=1; i<7; i++)  
    cout << i << " ";
```



1 2 3 4 5 6

24

## Example 2

```
int i;  
for(i=1; i<7; ++i)  
    cout << i << " ";
```

1 2 3 4 5 6

25

## Example 3

```
int i;  
for(i=1; i++<7; )  
    cout << i << " ";
```

2 3 4 5 6 7

26

## Example 4

```
int i;  
for(i=1; ++i<7; )  
    cout << i << " ";
```

2 3 4 5 6

27

## Example 5

```
int i;  
for(i=9; i>5; --i)  
    cout << i << " ";
```

9 8 7 6

28

## while equivalent of for

```
for(e1; e2; e3) s;
```

same as

```
e1;
while(e2) { s; e3; }
```

### Example

```
for (i=0; i<5; i++)
  cout << i;
```

same as

```
i=0;
while (i<5) {
  cout << i;
  i++;
}
```

29

## Nested For-Loops Example

```
int i, j, n = 7;

for(i=1; i<=n; i++){
  for(j=1; j<=i; j++)
    cout << j;
  cout << endl;
}
```

```
1
12
123
1234
12345
123456
1234567
```

30

## The power of C++

```
int x;
for(cout << "Enter x: "; cin >> x, x!=-999; );
```

```
Enter x: 6
3
7
-999
```

```
int x;
for( ; cout << "Enter x: ", cin >> x, x!=-999; );
```

```
Enter x: 6
Enter x: 3
Enter x: 7
Enter x: -999
```

31

## Endless Loop

```
for( ; ; ) ;
```

32



## The break and continue statements

- A **break** statement is used to “break” out of a loop or a switch statement. When it is executed, it causes the flow of control to immediately exit the innermost switch statement or loop.
- A **continue** statement can only be used inside loops and it causes the execution to skip to the end of the loop, ready to start a new iteration.

33

## Break & Continue statements

```
for(int i=1; i<8; i++){
    if (i==4) break;
    cout << i << " ";
}
```

1 2 3

```
for(int i=1; i<8; i++){
    if (i==4) continue;
    cout << i << " ";
}
```

1 2 3 5 6 7

34