



TRINITY COLLEGE FOR WOMEN NAMAKKAL

Department of Mathematics

SBEC- C PROGRAMMING

19UMAS02 - ODD Semester

Presented by

Mrs.P.Nandhini.,M.Sc.,M.Phil.,(SET)

Assistant Professor

Department of Mathematics

<http://www.trinitycollegenkl.edu.in/>

DECISION MAKING AND LOOPING

In looping, a sequence of statements are executed until some condition is satisfied which is placed for termination of the loop.

A program loop consists of two segments, one is the body of the loop and the other known as the control statement.

The control is tested always for execution of the body of the loop.

Depending on the position of the control statement in the loop, a control may be classified as the entry-controlled loop or as the exit-controlled loop.

In the entry-controlled loop, first the conditions are tested and if satisfied then only body of loop is executed.

In the exit controlled, the test is made at the end of the body, so the body is executed unconditionally first time.

A looping process, in general, would involve the following four steps:

1. Setting and initialization of a counter.
2. Execution of the statements in the loop.
3. Test for a specified condition for execution of the loop.
4. Incrementing the counter.

The C language provides for three loop constructs for performing loop operations. They are:

1. The while statement.
2. The do statement.
3. The for statement.

1.The while statement:

While statement is

```
while (test condition)
```

```
{
```

```
body of the loop
```

```
}
```

The while is an entry-controlled loop statement. The test condition is evaluated and only if the condition is true the body is executed. After execution of the body, the test-condition is once again evaluated and if it is true, the body is executed once again. This process of repeated execution of the body continues until the test-condition finally becomes false and the control is transferred out of the loop. On exit, the program continues with the statement immediately after the body of the loop. If the body contains only one statement it's not necessary to put the braces, but placing them is a good programming practice.

Example:

```
x = 10; ----- Initialization
while (x < 16)----- Test condition
{
printf("%d",x);-----body of the loop
x = x+1;
}
.....
```

The above example shows a simple position where the test condition is evaluated first and since the test is satisfied the body is executed.

This will continue exactly six times printing out 10 to 15. Finally when x becomes 16 then test fails & so the loop ends. The program continues with the statements after the loop.

2. The do statement:

The while is an entry-controlled loop structure, in which the loop will not be executed if the test condition comes out to be false. But in some situations it might be necessary to execute the loop, even if the test-condition fails. So this kind of loops is called the exit-controlled loop structure.

This takes the form:

```
do
{
body of the loop
}
while (test condition);
```

Example:

```
do
{
ch = getchar();
printf("The character entered is %c",ch);
}
while (ch != 'n');
```

ch is assumed to be a variable of type character. Now the loop is executed first under no test and the input by the user is outputted. Then the entered value would be tested by the condition for further advancement.

3. The for statement:

The for loop is another entry-controlled loop that provides a more concise loop control structure.

The general form of the for loop is:

```
for (initialization; test-condition ; increment)
{
body of the loop
}
```

Example:

```
for( i = 0; i<10; i = i+1)
{
printf("%d",i);
}
```

The for loop will be executed 10 times. Each time a value equal to the current value of i will be printed out.

The execution of the for statement is as follows:

1. Initialization of the control variables is done first, using assignment statements such as $i = 1$ and $count = 0$. The variables i and $count$ are known as loop control variables.

2. The value of the control variable is tested using the test-condition. The test-condition is a relational expression, such as $i < 10$ that determines when the loop will exit. If the condition is true, the body of the loop is executed; otherwise the loop is terminated and the execution continues with the statement that immediately follows the loop.

3. When the body of the loop is executed, the control is transferred back to the for statement after evaluating the last statement in the loop. Now, the control variable is incremented using an assignment statement such as $i = i + 1$ and the new value of the control variable is again tested to see whether it satisfies the loop condition. If the condition is satisfied, the body of the loop is again executed. This process continues till the value of the control variable fails to satisfy the test-condition.

THANK YOU

<http://www.trinitycollegenkl.edu.in/>