

Control Structures in C Part - II

FAQs

1. Explain the while structure and the do-while structure in C.

The while loop is used when a statement is to be executed repeatedly until a given condition is satisfied. The loop is terminated when the expression yields false value.

The basic format of the while loop is :

```
while (expression)
{
    // body of the loop
}
```

The do-while construct is an exit control loop structure in which the loop statement consists of a block of code and a boolean condition. First the code block is executed, and then the condition is evaluated. If the condition is true, the code block is executed again and again until the condition becomes false.

The normal syntax of do-while construct is :

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

2. Explain the general form of for structure in C. How is it executed?

The 'for loop structure' allows to specify the following three particulars about a loop in a single line:

1. Setting a loop counter to an initial value.
2. Testing the loop counter to determine whether its value has reached the number of repetitions desired.
3. Increasing the value of loop counter each time the program segment within the loop has been executed.

General form of the for loop is :

```
For (initialization; test-condition; increment)
{
    Body of the loop
}
```

The execution of the **for** loop is as follows:

- 1 **Control variable initialization**: control variables are initialized using assignment operator.
- 2 **Test condition checking** : the value of the control variable is tested using the test condition . usually relational expressions are used for conditional checking.
3. Body part is executed and the control is transferred back to the for statement. Now the control variable is incremented using an assignment statement and the new value of the control variable is tested again to

see whether the condition is satisfied and the process is continued until the condition becomes successful

3. What is the purpose of break statement?

The **break statement** allows us to jump out of a loop instantly, without any conditional test. When **break** is encountered inside any loop or in a switch-case control structure it terminates the execution at that point and transfers execution control to the statement immediately following the loop or switch-case construct. Thus it indulges an early exit from any loop or switch-case construct statement.

The general syntax is

```
break;
```

4. Where can we use continue statement.

The **continue statement** can be used to take the control to the beginning of the loop, bypassing the statements inside the loop, which have not yet been executed. When **continue** is encountered inside any loop, control automatically passes to the beginning of the loop. That means it is used within loop structures to end the execution of the current iteration and proceed to the next iteration.

Format is

```
continue;
```

5. Explain the use of label in a goto statement.

The goto construct causes an unconditional transfer of execution and its general format is:

```
label:
```

```
goto label;
```

where label is an identifier and not a number.

The identifier following goto is a statement label and it is not declared. The name of the statement label can also be used as a variable name in the same program if it is declared properly. The compiler identifies this name as a label if it appears in a goto statement and as a variable if it appears in an expression.