Tutorial Link https://codequotient.com/tutorials/Iteration - Introduction and while loop/5a22a3d7c66cfe38f2962269

TUTORIAL

Iteration - Introduction and while loop

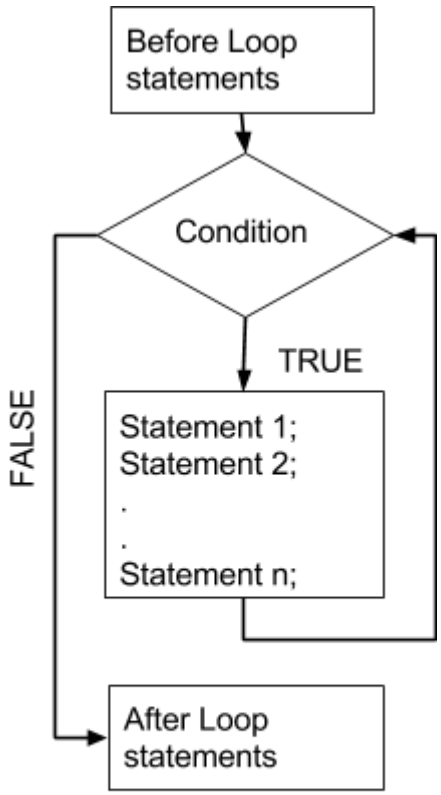
Chapter

1. Iteration - Introduction and while loop

Topics

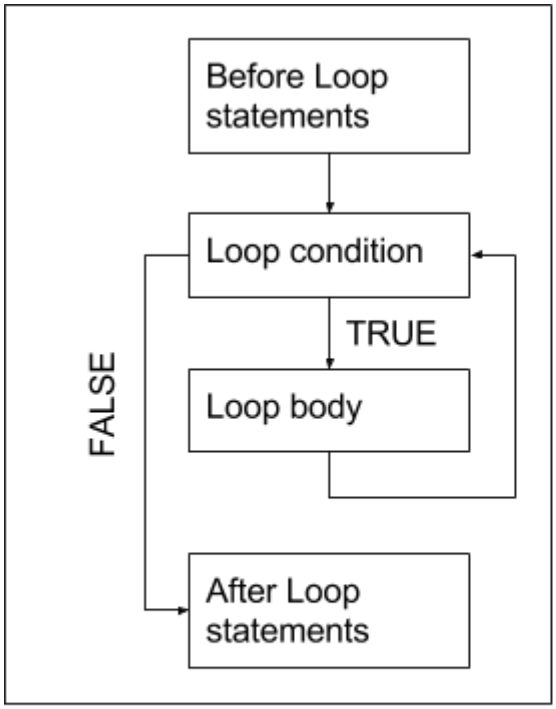
1.2 The while loop

In real life we like to do those things again and again, which makes us happy. The same also applies to programming. Most of the algorithms have some piece of code, which needs to be executed multiple times to solve a particular problem. These repetition can be achieved in multiple ways in every programming language. In C, we have three kind of looping constructs: while, for & do-while. These are also known as iteration statements. Iteration statements allow a set of instructions to be repeatedly executed until a certain condition is reached. Iteration is shown in below figure: -

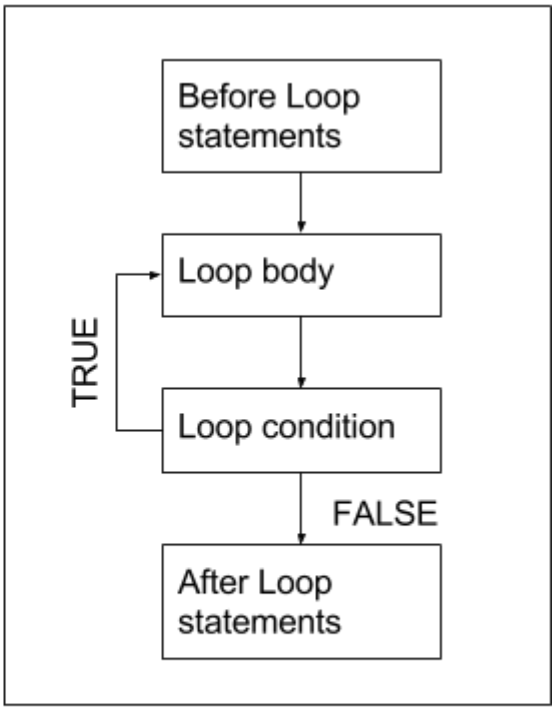


Loop / Iteration

This condition may be predetermined (as in the for loop) or open ended (as in the while and do-while loops). So it is the decision of programmer to check the condition for execution of repeating block before entering the loop body or after exiting the loop body. If we first check the condition and then decide whether to execute the set of statements, it is called entry-controlled loops, and if we execute the set of statements and then check a condition to execute the statements again or not, it is called exit-controlled loops. This can be summarized in following figure: -



Entry Controlled Loop



Exit Controlled Loop

The while loop

The first form of loop is while loop. The general form is:

```
while(condition)
{
   // set of statements known as loop body
}
```

In this case, the control first check the condition, if it evaluates to true control goes inside the loop body, otherwise control will jump to the first statement after the while scope. For example,

```c
#include <stdio.h>

int main()
```

```
{
  int var1;
  var1 = 1;
  while(var1 <= 3)
  {
    printf("Var1 = %d\n",var1);
    var1 = var1 + 1;
  }
  return 0;
}

```

This loop will run three times. It starts from var1 = 1, and run till var1 = 3. After third iteration, the conditions becomes false, and control comes outside the while loop. So the general form of while becomes as shown below:

```
initialise counter ;
while (condition)
{
   // set of statement (loop body)
   manage loop counter;
}
```

while loops can be written in many different forms. Some of them are described below.

```c
#include <stdio.h>

int main()
{
```

```
  int var1;
  var1 = 1;
  while(var1 <= 3)
  {
    printf("Var1 = %d\n",var1);
  }
  return 0;
}

```

It is a program with infinite loop, because in this, the condition always evaluates to true. The value of var1 is 1 and never changing in the program. So generally, the loop counter is modified in loop body to decrease the make the condition meaningful.

Below program is a decrements type while loop: -

```c
#include <stdio.h>

int main()
{
  int var1;
  var1 = 5;
  while(var1 >= 1)
  {
    printf("Var1 = %d\n",var1);
    var1--;
  }
  return 0;
}

```

while loop also test the condition before entering the loop body so if condition evaluates to false in the first go, control will never go inside the loop. For example,

```c
#include <stdio.h>

int main()
{
  int var1;
  var1 = 10;
  while(var1 <= 3)
  {
    printf("var1 = %d\n", var1);
    var1++;
  }
  printf("This is after while loop.");
  return 0;
}

```

This will directly go and execute the statement after the while loop. The loop body will not execute for even once. So while loop is also called entry-controlled loop.