Looping statement

Loop:-**we will discuss all about loops. To understand the need of loops, let's begin with an example, suppose we ask you to display "hello sir" four times on the desktop. Easy! you'd say:**

```
#include<stdio.h>

using namespace std;

int main()

{

        printf("Hello sir\n");

        printf("Hello sir\n");

        printf("Hello sir\n");

        printf("Hello sir\n");

        return 0;

 }
```

Indeed. This does exactly what was asked. But now we up the bet and ask you to do the same job for 100 or 1000 or 3425. Whatever it takes you to realize this isn't a sensible method! What you really need is some way of repeating the printf statements without having to write it out each time. The solution to this problem is the loop. So, when a set of statements in a program has to be used repeatedly then looping of the repetitive structure has to be done. In such looping, a group of statement are executed until some condition has been satisfied. In 'C' the loop statement available are

I. *while*
II. *do while*

III. *for*

IV. *nested for*

I. ***While Statement:-*** **while loop has one control condition, and executes as long the condition is true. The condition of the loop is tested before the body of the loop is executed, hence it is called an *entry-controlled* loop.**

Syntax:-

```
while(condition)
{
    statement 1;
    statement 2;
    .....
    statement n;
}
```

Example 1:- ***Write a program to put any one number and print table using while.***

Solution:-

```
#include<stdio.h>
using namespace std;
int main()
{
    int a,i=1,t;
    printf("Enter any one number for a table");
```

```
        scanf("%d",&a);
        while(i<=10)
        { t=i*a;
        printf("a=%d\n",t);
        i++;
        }
        return 0;
    }
```

Output:-

Enter any one number for a table =4 // first time run you enter 4

4
8
12
16
20
24
28
32
36
40

Enter any one number for a table =19 // second time run you enter 19

19
38
57
76
95

114
133
152
171
190

II. ***do while:-*** **It is a variant of while loop where the condition isn't checked at the top but at the end of the loop, known as exit controlled loop. This means statements inside do-while loop are executed at least once and exits the loop when the condition becomes false or break statement is used. The condition to be checked can be changed inside the loop as well.**

Syntax:-

```
do
{
   Statement 1;
    ... ... ...
    Statement n;
}while (condition);
```

Example 1:- ***Write a program to put any one number and find factorial using do while.***

Solution:-

```
#include<stdio.h>
using namespace std;
int main()
```

```
{
    int a,i=1;
    printf("enter one number=");
    scanf("%d",&a);
    do
    {
        i=a*i;
        printf("i=%d\n",i);
        a--;
    }while(a>=1);
    return 0;
}
```

Output:- enter one number= 5

```
i=5
i=20
i=60
i=120
i=120.
```

III. ***for statement:-*** **When you know exactly how many times you want to loop through a block of code, use the for loop instead of a while loop.**

Syntax:-

```
for(variable; condition; increment/decrement)
        {
        statement 1;
        statement 2;
        ....
        statement n;
        }
```

Example 1:- ***Write a program to print 1 to 10 series.***

Solution:-

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1;i<=10;i++)
{
        printf("%d\n",i);
}
return 0;
}
```

Output:-

```
1
2
```

3
4
5
6
7
8
9
10.

IV. **nested for:-** **Nested loop means a loop statement inside another loop statement. That is why nested loops are also called as *loop inside loop*.**

Syntax:-

```
for(variable; condition; increment/decrement)
    {
    for(variable; condition; increment/decrement)
        {
            statement 1;
            statement 2;
            ....
            statement n;
        }

    statement 1;
    statement 2;
    ....
```

```
statement n;
}
```

Example 1:-Write a program to print a 2D matrix of 3×3 using *nested for loop.*

Solution:-

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1;i<=3;i++)
{
        for(int j=1;j<=2;j++)
        {
        printf("%d",j);
        }
        printf("%d",i);
        printf("\n");
}
return 0;
}
```

Output:-

```
121

122

123
```

Created by:-Ajay kumar verma