Loops in Programs

There are many circumstances where we require a loop (set of conditions to be repeated a number of times). It's not feasible to write those lines of code again and again so better is to create a loop statement which can execute the statements contained in the loop a specified number of times based on conditions.

There are a number of looping statements which can be used to create loops.

For example: **for, while, do…while**

for loop

for statement is the most common type of looping statement which we often use in our programs. It has a very simple format and all the three tasks i.e. **initialization, condition and updating** in a single line, all of which are written within for (…) statement.

Format:

for (initialization; condition; updating)

{

Statements to be repeated

}

Here initialization means initializing a value (e.g. i=1), condition like i<=10, and updating like i++ so after each iteration (loop), i increases by 1. Thus, the loop runs for 10 times based on the condition.

while loop

while loop works a little differently and is an alternative to for loop. The main difference lies in the fact that in while loop, all three statements are written on different lines unlike for loop where all three statements came on same line.

Format:

Initialization;

while (condition)

{

Statements to be executed

Updating

}

In this first we initialize outside loop like i=1 then we go for while statement which holds the condition which checks after every iteration. Then, we do the updating part inside the while loop only.

do…while loop

It is another alternative for while loop. It has a little different way of working than while loop which makes its work different in some cases. It first initializes, then runs the loop, updates the loop variable and then checks for condition. It can give different results in some scenarios due to this fact so it finds specific uses in many places.

Format:

Initialize;

do {

Statements to be executed

Updating the loop variable

}

while (condition)