**Looping in Java**

Java provides three repetition statements/looping statements that enable programmers to control the flow of execution by repetitively performing a set of statements as long as the continuation condition remains true. These three looping statements are called for, while, and do...while statements. The for and while statements perform the repetition declared in their body zero or more times. If the loop continuation condition is false, it stops execution. The do...while loop is slightly different in the sense that it executes the statements within its body for one or more times.

There is a common structure of all types of loops, such as:

* There is a control variable, called the loop counter.
* The control variable must be initialized; in other words, it must have an initial value.
* The increment/decrement of the control variable, which is modified each time the iteration of the loop occurs.
* The loop condition that determines if the looping should continue or the program should break from it.

**While Loop**

The while loop allows us to specify that a certain statement is to be executed repetitively until the loop condition is false. Consider a problem to print ten consecutive numbers from 1 to 10. What we do is initialize a control variable outside the while loop construct, state a condition for loop continuation, and then increment the control variable so that at some point of the iteration the condition become false and breaks out of the repetition.



int counter = 1; // Control variable initialized
// Condition for loop continuation
while (counter <= 10) {
 System.out.println(counter);
 counter++; // Increment the control variable
}

**Note:** If such a situation occurs where a program executes infinitely, use a keyboard interrupt, such as **Ctrl+C**; this will terminate the running program.

**For Loop**

The working process of a for loop is similar to the while loop, only the structure is different. Unlike the while loop, the layout of the control variable, loop condition, and the increment of the control statement can be stated in a single line of code, such as in the following example.



int counter;
for( counter = 1; counter <= 10; counter++){
 //... Statements
}
int counter = 1;
for(; counter <= 10; counter++){
 //... Statements
}
int counter = 1;
for(; counter <= 10;){
 //... Statements
 counter++;
}
// Infinite loop
for(;true;);

**Comparing While and For Loops**

Both the for and while loops are entry controlled; in other words, they first check the truth value of the condition and then only enter into the loop.

**Do .. While Loop**

This is an exit controlled looping structure which is different from, yet somewhat similar to, the while statement in the sense that the structure of the loop described, such as initialization of the control variable, is declared outside the loop header, the increment/decrement is written within the body of the loop like any other statements.



int counter = 1; // Control variable initialized
do{
 System.out.println(counter);
 counter--; // Decrements the control variable
}while(counter <= 10); // Condition statement

The significant difference that sets the do...while loop apart from both while and for loop is that the for and while loops are meant to execute zero or more times. This means there is a chance that the loop may not execute at all if the condition is false. The do...while loop, on the other hand, executes at least once or, in other words, one or more times because it never checks any condition during its entry; rather, it evaluates the loop continuation condition statement only at the exit. This makes the body of the loop execute at least once.