**Looping Statement in Java**

**Looping statement** are the statements execute one or more statement repeatedly several number of times. In java programming language there are three types of loops; while, for and do-while.

**Why use loop ?**

When you need to execute a block of code several number of times then you need to use looping concept in Java language.

**Advantage with looping statement**

* Reduce length of Code
* Take less memory space.
* Burden on the developer is reducing.
* Time consuming process to execute the program is reduced.

**Difference between conditional and looping statement**

Conditional statement executes only once in the program where as looping statements executes repeatedly several number of time.

**While loop**

In **while loop** first check the condition if condition is true then control goes inside the loop body otherwise goes outside of the body. while loop will be repeats in clock wise direction.

**Syntax**

while(condition)

{

Statement(s)

Increment / decrements (++ or --);

}

**Example while loop**

**class** whileDemo

{

**public** **static** **void** main(String args[])

{

**int** i=0;

**while**(i<5)

{

System.**out**.println(+i);

i++;

}

**Output**

1

2

3

4

5

**for loop**

**for loop** is a statement which allows code to be repeatedly executed. For loop contains 3 parts Initialization, Condition and Increment or Decrements

**Syntax**

for ( initialization; condition; increment )

{

statement(s);

}

* **Initialization:**This step is execute first and this is execute only once when we are entering into the loop first time. This step is allow to declare and initialize any loop control variables.
* **Condition:**This is next step after initialization step, if it is true, the body of the loop is executed, if it is false then the body of the loop does not execute and flow of control goes outside of the for loop.
* **Increment or Decrements:**After completion of Initialization and Condition steps loop body code is executed and then Increment or Decrements steps is execute. This statement allows to update any loop control variables.

**Flow Diagram**

**Control flow of for loop**

* First initialize the variable
* In second step check condition
* In third step control goes inside loop body and execute.
* At last increase the value of variable
* Same process is repeat until condition not false.

Improve your looping concept[For Loop](http://www.tutorial4us.com/cprogramming/c-for-loop)

**Display any message exactly 5 times.**

**Example of for loop**

**class** Hello

{

**public** **static** **void** main(String args[])

{

**int** i;

**for** (i=0: i<5; i++)

{

System.**out**.println("Hello Friends !");

}

}

}

**Output**

Hello Friends !

Hello Friends !

Hello Friends !

Hello Friends !

Hello Friends !

**do-while**

A **do-while** loop is similar to a while loop, except that a do-while loop is execute at least one time.

A do while loop is a control flow statement that executes a block of code at least once, and then repeatedly executes the block, or not, depending on a given condition at the end of the block (in while).

**When use do..while loop**

when we need to repeat the statement block **at least one time** then ues do-while loop. In do-while loop post-checking process will be occur, that is after execution of the statement block condition part will be executed.

**Syntax**

do

{

Statement(s)

increment/decrement (++ or --)

}while();

In below example you can see in this program i=20 and we chech condition i is less than 10, that means conditon is false but do..while loop execute onec and print Hello world ! at one time.

**Example do..while loop**

**class** dowhileDemo

{

**public** **static** **void** main(String args[])

{

**int** i=20;

**do**

{

System.**out**.println("Hello world !");

i++;

}

**while**(i<10);

}

}

**Output**

Hello world !

**Example do..while loop**

**class** dowhileDemo

{

**public** **static** **void** main(String args[])

{

**int** i=0;

**do**

{

System.**out**.println(+i);

i++;

}

**while**(i<5);

}

}

**Output**

1

2

3

4

5