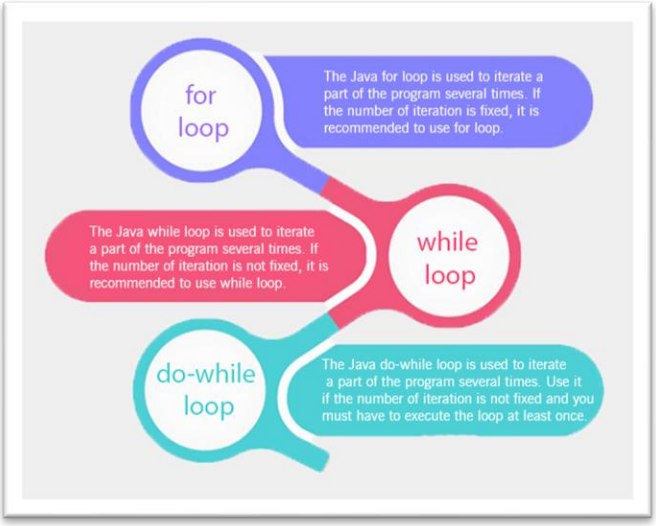
LOOPING STATEMENT

- The Java for **loop** is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop.
- A "**looping statement**" is a programming instruction that lets you execute a block of code repeatedly as long as a certain condition is met.
- **For example**, if you want to show a message 100 times, then rather than typing the same code 100 times, you can use a loop.

There are **three types** of for loops in Java.

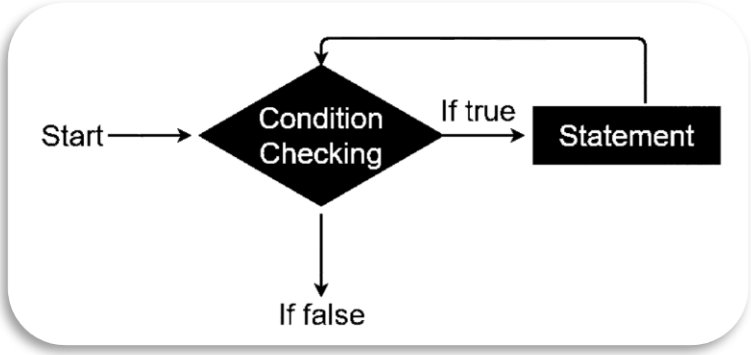
- **For loop**
- **While loop**
- **Do..while loop**



1.While loop:

A **while loop** is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

©Topperworld



Syntex:

```
while (boolean condition)
{
   loop statements...
}
```

Example:

```
import java.io.*;

class Topperworld {
    public static void main (String[] args) {
    int i=0;
    while (i<=10)
    {
        System.out.println(i);
        i++;
    }
    }
}
```

Output:

```
0
1
2
3
4
5
6
7
8
9
10
```


©Topperworld

- While loop starts with the checking of Boolean condition. If it evaluated to true, then the loop body statements are executed otherwise first statement following the loop is executed. For this reason it is also called Entry control loop
- Once the condition is evaluated to true, the statements in the loop body are executed. Normally the statements contain an update value for the variable being processed for the next iteration.
- When the condition becomes false, the loop terminates which marks the end of its life cycle.

2.For loop:

For loop provides a concise way of writing the loop structure. Unlike a while loop, a for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.

Syntax:

```
for (initialization condition; testing condition;increment/decrement)
{
    statement(s)
}
```


©Topperworld

Example:

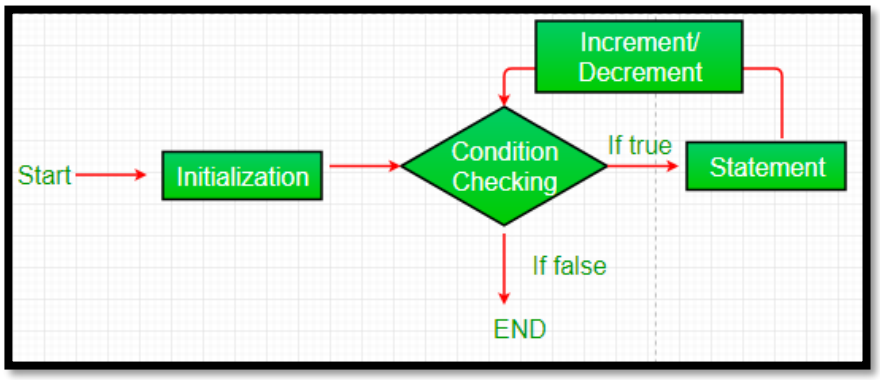
```
import java.io.*;

class Topperworld {
    public static void main (String[] args) {
    for (int i=0;i<=10;i++)
    {
        System.out.println(i);
    }
    }
}
```

Output:

```
0
1
2
3
4
5
6
7
8
9
10
```

Flowchart:



©Topperworld

- **Initialization condition:** Here, we initialize the variable in use. It marks the start of a for loop. An already declared variable can be used or a variable can be declared, local to loop only.
- **Testing Condition:** It is used for testing the exit condition for a loop. It must return a boolean value. It is also an Entry Control Loop as the condition is checked prior to the execution of the loop statements.
- **Statement execution:** Once the condition is evaluated to true, the statements in the loop body are executed.
- **Increment/ Decrement**: It is used for updating the variable for next iteration.
- **Loop termination:** When the condition becomes false, the loop terminates marking the end of its life cycle.

3.do while

do while loop is similar to while loop with only difference that it checks for condition after executing the statements, and therefore is an example of Exit Control Loop.

Syntex:

```
do
{
    statements..
}
while (condition);
```


©Topperworld

Example:

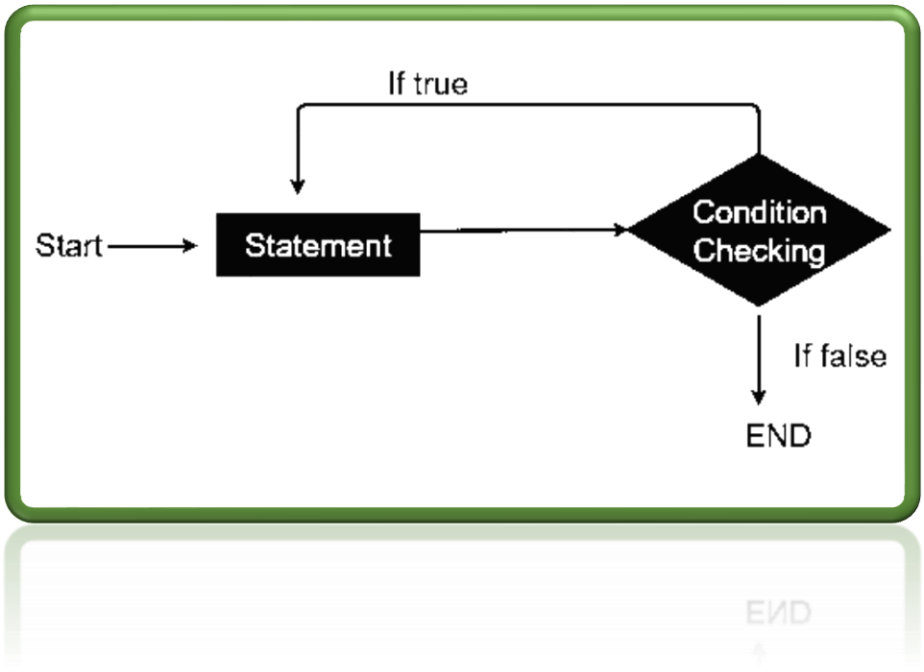
```
import java.io.*;

public class Topperworld {
   public static void main (String[] args) {
   int i=0;
   do
   {
       System.out.println(i);
       i++;
   }while(i<=10);
   }
}
```

Output:

```
0
1
2
3
4
5
6
7
8
9
10
```

Flowchart:



©Topperworld

- do while loop starts with the execution of the statement(s). There is no checking of any condition for the first time.
- After the execution of the statements, and update of the variable value, the condition is checked for true or false value. If it is evaluated to true, next iteration of loop starts.
- When the condition becomes false, the loop terminates which marks the end of its life cycle.
- It is important to note that the do-while loop will execute its statements atleast once before any condition is checked, and therefore is an example of exit control loop.

©Topperworld