**Looping statements-**

Loops are used to execute a set of instructions repeatedly while some condition is true. There are three types of loops in Java.

1. For loop

2. While loop

3. Do while loop

**1 .For loop-**

When to use-

If the number of iteration is fixed, it is recommended to use for loop.

**Syntax**

|  |
| --- |
| *For (initialization; condition; increment/decrement) {*  *Statement 1;*  *Statement 2;*  *}* |

Example- Different way to write the loops in java.

Example-1

**public** **class** Looping {

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

**for**(**int** i=1; i<=5;i++) {

System.***out***.println("value of i>>"+i);

}

}

}

Output :  
value of i >> 1

value of i >> 2

value of i >> 3

value of i >> 4

value of i >> 5

Example-2

**public** **class** Looping {

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

**for**(**int** i=1; i<5;i++) {

System.***out***.println("value of i>>"+i);

}

}

}

Output :

value of i >> 1

value of i >> 2

value of i >> 3

value of i >> 4

Example-3

**public** **class** Looping {

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

**int** i = 1;

**for** (; i < 5; i++) {

System.***out***.println("value of i>>" + i);

}

}

}

Output :  
value of i >> 1

value of i >> 2

value of i >> 3

value of i >> 4

Example-4

**public** **class** Looping {

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

**int** a[]= {10,20,30};

**for** (**int** i :a) {

System.***out***.println("value of i>>" + i);

}

}

}

Output :  
value of i >> 10

value of i >> 20

value of i >> 30

Example-5

**public** **class** Looping {

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

**for** (**int** j = 5; j >0; j--) {

System.***out***.println("j>>" + j);

}

}

}

Output :  
 j >> 5

j >> 4

j >> 3

j >> 2

j >> 1

**2 . While loop-**

The while loop evaluates a certain condition. If the condition is true, the code is executed. This process is continued until the specified condition turns out to be false.

When to use-

If the number of iteration is not fixed, it is recommended to use while loop.

**Syntax-**

Initialization;

While (condition) {

Increment/decrement operator;

}

Example-1

**public** **class** Looping {

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

**int** i=1;

**while**(i<=5) {

System.***out***.println("value of i>>" + i);

i++;

}

}

}

Output :  
value of i >> 1

value of i >> 2

value of i >> 3

value of i >> 4

value of i >> 5

Example-2

**public** **class** Looping {

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

**int** i=5;

**while**(i>0) {

System.***out***.println("value of i>>" + i);

i--;

}

}

}

Output :  
value of i >> 5

value of i >> 4

value of i >> 3

value of i >> 2

value of i >> 1

**3.Do while loop-**

The do-while loop is similar to the while loop, the only difference is that the condition in the do-while loop is evaluated after the execution of the loop body. This guarantees that the loop is executed at least once

When to use-

If the number of iteration is not fixed, but you want execute statement or group of code at least once it is recommended to use do while loop.

**Syntax**-

Initialization;

do {

increment or decrement;

} condition;

Example-

**public** **class** Looping {

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

**int** i = 1;

**do** {

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

i++;

} **while** (i <= 5);

}

}

Output :  
 1

2

3

4

5

**Jumping statements-**

Jump statements cause an unconditional jump to another statement elsewhere in the code

1. Break Statement
2. Continue Statement
3. **Break Statement :**

Break statement can be used inside the loop or inside switch case.

When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.

**Example :**

**public** **class** BreakExample {

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

//using for loop

**for**(**int** i=1;i<=10;i++){

**if**(i==5){

//breaking the loop

**break**;

}

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

}

}

}

Break statement is also used inside the switch case statement.

|  |
| --- |
| Switch case without break:- |
| ***public******class*** *Sample{*  ***public******static******void*** *main(String args[])*  *{*  ***int*** *ch=2;*  ***switch****(ch)*  *{*  ***case*** *1:*  *System.****out****.println("I am case 1");*  ***case*** *2:*  *System.****out****.println("I am case 2");*  ***case*** *3:*  *System.****out****.println("I am case 3");*  ***case*** *4:*  *System.****out****.println("I am case 5");*  ***default****:*  *System.****out****.println("I am default case");*  *}*  *}*  *}* |
| *Output:-*  *I am case 2*  *I am case 3*  *I am case 5*  *I am default case* |

|  |
| --- |
| **Switch case with break:-** |
| **public** **class** Sample {    **public** **static** **void** main(String args[])  {  **int** ch=2;  **switch**(ch)  {  **case** 1:  System.***out***.println("I am case 1");  **case** 2:  System.***out***.println("I am case 2");  **break**;  **case** 3:  System.***out***.println("I am case 3");  **break**;  **case** 4:  System.***out***.println("I am case 5");  **break**;  **default**:  System.***out***.println("I am default case");  }  }  } |
|  |

1. **Continue Statement :**

If you want to skip current iteration and go to next iteration of the loop then use continue statement. The code after continue statement is not executed.

**Example :**

**public** **class** ContinueExample {

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

//for loop

**for**(**int** i=1;i<=10;i++){

**if**(i==5){

//using continue statement

**continue**;//it will skip the rest statement

}

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

}

}

}