**Loops**

→ Sometimes we need to execute a certain piece of code repeatedly, so the brute force/naive approach becomes lengthier and increases the lines of code due to which the readability, maintainability of the code becomes worst. So in this case it’s alsways better to go for the loop.

→ Loops are working based on the iteration and the same set of code can be executed with lesser lines of code.

→ In java we’ve different types of loops

A. while loop

B. do-while loop

C. for loop

D. Advanced for loop or for-each loop

**1. while loop:**

Syntax:

while(condition) {

//Code to execute

}

→ Here the control will enter inside the loop only when the condition is true and will execute until the condition remains true.

**2. do-while loop:**

Syntax:

do {

//Code to execute

}while(condition);

→ Here the loop will execute once must, then after the codes inside the loop will execute based on the condition

**Note: When we use any loop, we have to be very much aware that the loop should stop with a certain condition or else the loop will lead towards the infinite loop**

**3. for loop:**

Syntax:

for (initialization; condition; increment/decrement) {

//Code to execute

}

→ While we use the for loop, the initialization inside the for loop is optional in the loop signature but the initialization must have done prior to the usage of the for loop

→ Before entering inside the loop, first the condition is being checked then only it enters into the loop if in case the condition satisfies else the loop will get exited immediately. The section is quite mandatory to be there in the loop signature

→ The increment/decrement section is also optional in the loop signature but if we don’t mention in the signature then we must have to deal the same thing inside the loop body

→ If we consider the loop parameters optional then the loop looks like as below

initialization

for ( ; condition ; ) {

//Code to execute

//Increment/Decrement

}