LOOPS IN JAVA

Java programmes are executed sequentially from top to bottom. Apart from conditional logic, that is the use of if, else and switch statements, one can interrupt the program flow by using loops.

A programming loop is one that forces the program to go back up again. As it forces execution to go back up again, we can execute lines of code repeatedly.

There are basically 3 kinds of loops in Java: for loop, while loop, and do while loop.

For loop

The structure of the for loop is this:

```
for (start_value; end_value; increment_number)
{
      //code to repeat;
}
```

You begin the for loop by writing the keyword '`for`'. Thereafter, you have a pair of round brackets. Inside the round brackets you need three things: the start value of the loop, the end value for the loop, and a way to get from one number to another. This is called the increment number, and is usually 1, but it doesn't have to be. After the round brackets you need a pair of curly brackets. The curly brackets are used to section off the code that you want to execute repeatedly.

For example:

```
public static void main (String [] args) {
      int a;
      for (a = 1; a<=8; a++) {
            System.out.println (a);
      }
}
```

While Loop

Another type of loop you can use in Java is called the while loop. Here is what it looks like:

```
while (condition) {
      // code to repeat;
}
```

You start with the word 'while' in lowercase. The condition you want to test for goes between round brackets. A pair of curly brackets comes next and the code you want to execute goes between the curly brackets. For example:

```
int loopVal = 0;
while (loopVal<5) {
      System.out.println ("Printing some
text");
      loopVal++;
}
```

Do … while loop

Related to the while loop is the do … while loop. It looks like this:

```
int loopVal = 0;
do {
      System.out.println (“Printing some
text”);
      loopVal++;
}
while (loopVal < 5);
```

Again Java will loop round and round until the end condition is met. This time, the ‘`while`’ part is at the bottom. But the condition is the same – keep looping while the value inside of the variable called loopVal is less than 5. The difference between the two is the code between the curly brackets of do … while will get executed at least once. With the while loop, the condition could already be met. Java will then just bail out of your loop, and not even execute your curly bracket code.