while & do-while Loops

In this lesson, an introduction of the while and do-while loops in Java is provided. It uses coding examples to show their implementation and explain their workings.

We'll cover the following

Loops allow a programmer to execute the same block of code repeatedly.

The while loop

The `while` *loop* is really the only necessary repetition construct. It will keep running the statements inside its body until some condition is met.

The syntax is as follows:

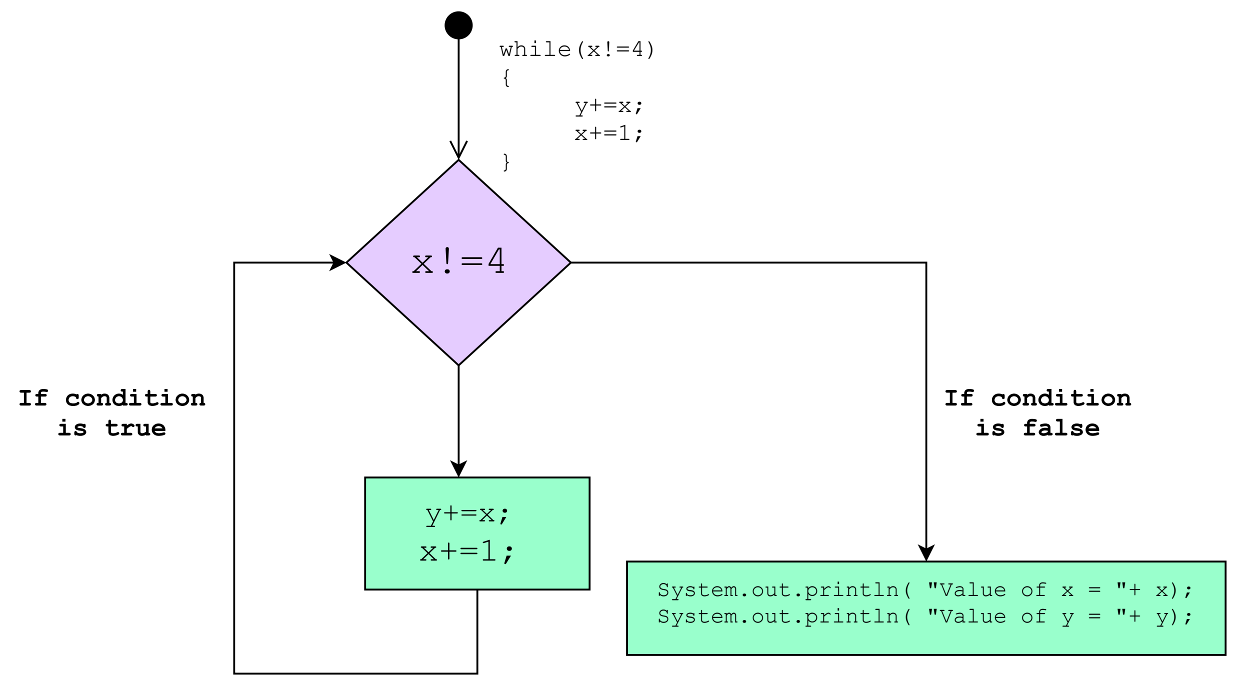
```
while ( condition ) {
    //body
}
```

Again, the **curly braces** surrounding the *body* of the `while` *loop* indicate that *multiple* statements will be executed as part of this *loop*.

Here's a look at the `while` loop code:

```
class HelloWorld {
    public static void main(String args[]) {
        int x = 0;
        int y = 0;
        while (x != 4) { // the while loop will run as long as x==4 condition is being met
            y += x;
            x += 1;
        }
        System.out.println("Value of y = " + y);
        System.out.println("Value of x = " + x);
    }
}
```

Below is an *illustration* of the code above to help you better understand the logic.


while(x!=4)
{
y+=x;
x+=1;
}
x!=4
If condition is true
If condition is false
y+=x;
x+=1;
System.out.println("Value of x = "+ x);
System.out.println("Value of y = "+ y);

Flow Chart For While Loop Code

If the `while` *loop* code looked like this instead, There would be a problem.

You will witness an Execution Timed Out Exception.

```
class Loops {
    public static void main(String args[]) {
        int x = 0, y = 0;
        while (x != 4) //since x is not being changed inside the while loop you will get stuck
        { // in an infinte loop as the condiiton will always be met
            y += x;
        }
        x += 1;
    }
}
```

In the code above, **line number 8** is still there. It will only be run after the while loop ends. But the while loop doesn't end because the terminating condition is never met.

This is a huge problem because the variable involved in the condition `(x)` does

not change, so the condition will always evaluate to *true*, making this an **infinite**

loop.

> **Note:** Be careful with the `while` loop as it has the potential of being endless.

The do...while loop

The `do-while` loop is nearly identical to the `while` loop, but instead of checking the *conditional* statement before the loop starts, the `do-while` loop checks the *conditional* statement **after** the *first* run, then continuing onto another iteration if the condition is `true`.

The *syntax* is as follows:

```
do {
    //body
} while (condition);
```

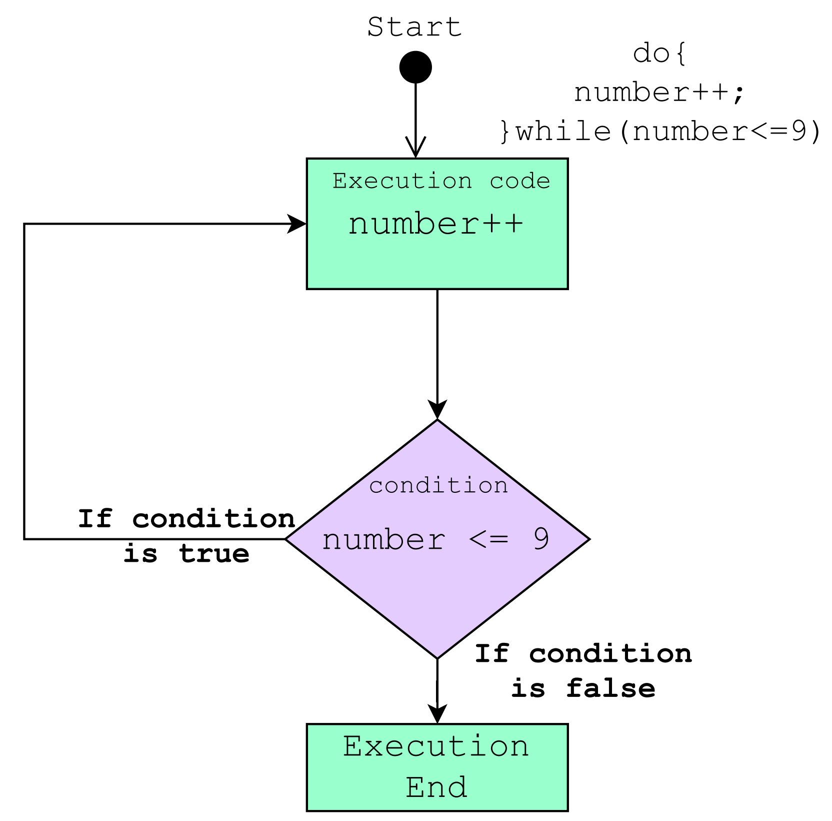
As you can see, it will run the loop **at least** once before checking the conditional statement.

> **Note:** The `do-while` loop is still haunted by *infinite* loops, so exercise the same caution with the `do-while` loop as you would with the `while` loop. Its usefulness is much more limited than the `while` loop, so use this only when necessary.

Below is an example showing how to implement the **do...while** loop in Java.

```
class HelloWorld {
    public static void main(String args[]) {
        int number = 5;
        do {
            System.out.println("Value of number is: " + number);
            number++;
        } while (number <= 9); // the contition is being checked after the first run
    }
}
```

Below is an *illustration* of the code above to help you better understand the logic.



Flow Chart For Do While Loop Code

When is do-while used?

A `do-while` loop is used where your loop should execute **at least one** time even if the given condition is `false`.

For example, let's consider a scenario where we want to take an *integer* input from the user until the user has entered a **positive** number. In this case, we will use a `do-while` as we have to run loop **at-least-once** because we want input from the user at least once. This loop will continue running until the user enters a **positive** number.

That's all the major stuff you needed to know about the workings of `while` and `do..while` loops in Java. Let's learn about `for` loops in the next lesson.