For Loops

By: Megan Avery
Updated October 2018

Before we begin

- Make sure you have Java & Java JDK downloaded on your computer
    - can run **`java -version`** in terminal to check
- Make sure you have IntelliJ downloaded on your computer

- *Suggested:* Watch previous Java tutorials

Vocab Alert!

loop variable: the variable that keeps track of the loop counter

increment: add to a variable

decrement: subtract from a variable

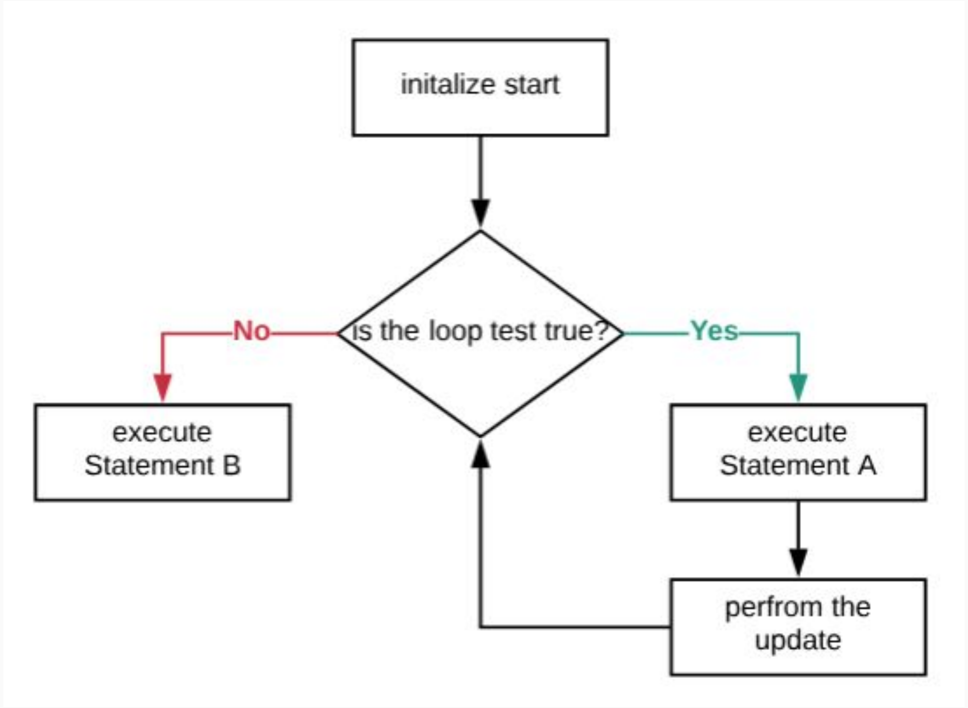
What is a for loop?

A **for loop** is a **control structure** that lets you repeat a piece of code a fixed number of times. This number of times can be explicitly set during the code, based off a parameter, or some property of an Object, like the length of a String.

Code and Diagram of a For Loop

```
for (start; loop test; update) {
     // Statement A
}

// Statement B
```



Other ways to edit variables

```
i++;

++i;

i--;

--i;
```

```
i *= value;

i /= value;

i += value;

i -= value;

i %= value;
```

++i; vs. i++;

++i; will increment that value and return the incremented value.

Example:

```
i = 1;
j = ++i;
// i is 2, j is 2
```

i++ saves off the original value, increments the variable, and returns the original value. It is less efficient.

Example:

```
i = 1;
j = i++;
// i is 2, j is 1
```

Common Issues

- Loops that never run because the loop test is false from the beginning

```
for (int i = 5; i < 4; i++) {
      System.out.println("*");
}
```

- Loops that never run because there is a semi colon after the first line

```
for (int i = 0; i < 4; i++); {
      System.out.println("*");
}
```

- Loops that run forever because the loop condition is always true

```
for (int i = 5; i > 0; i++) {
      System.out.println("*");
}
```

“Break”ing a For Loop

Putting a **break;** inside of a for loop, or any other type of loop, will force the loop to stop executing. This can be good if you are using your loop to find the first occurence of something or if some condition (besides the loop test) should cause your loop to stop.

Returning Inside a For Loop

Returning inside of a for loop will return from the method you are inside of as well. This means anything in your method after of the loop won’t happen if the return inside of your loop gets executed.

When to use a for loop?

If you find yourself writing a piece of code where adjacent lines of code are basically the same you might consider using a for loop there instead of explicitly writing out all the code.

Example:

```java
System.out.println(1);
System.out.println(2);
System.out.println(3);
System.out.println(4);
```

$\longrightarrow$

```java
for (int i = 1; i <= 4; ++i) {
    System.out.println(i);
}
```

Fence Post Problem

A fence post problem is when the output is expected to be different for one of the ends of the output. For example: 1, 2, 3, 4, 5

To solve a fence post problem you print one of the ends outside of the loop and run the rest of the output through a loop. The solution for the example would be:

```
System.out.print(1);
for (int i = 2; i <= 5; ++i) {
    System.out.print(", " + i);
}
System.out.println();
```

Loop Tracing

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i   output

**** keep track of loop variable (i) and output ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i	output
0	

**** initialize i to 0 ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i	output
0	*

**** i is less than 5, print out an asterisk ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i

~~0~~1

output

*

**** increment i by 1, i is now 1 ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i	output
~~0~~1	**

**** i is less than 5, print out an asterisk ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i

~~0~~ ~~1~~ 2

output

```
**
```

**** increment i by 1, i is now 2 ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i

~~0~~~~1~~2

output

```
***
```

**** i is less than 5, print out an asterisk ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i

~~0~~~~1~~~~2~~3

output

```
***
```

**** increment i by 1, i is now 3 ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i

~~0~~~~1~~~~2~~3

output

```
****
```

** i is less than 5, print out an asterisk **

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i

~~0~~~~1~~~~2~~~~3~~4

output

```
****
```

** increment i by 1, i is now 4 **

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i	output
~~0~~~~1~~~~2~~~~3~~4	*****

** i is less than 5, print out an asterisk **

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

<u>i</u>

~~01234~~5

<u>output</u>

```
*****
```

**** increment i by 1, i is now 5 ****

Loop Tracing

```
for (int i = 0; i < 5; i++) {
    System.out.println("*");
}
```

i

~~0~~~~1~~~~2~~~~3~~~~4~~5

output

```
*****
```

**** i is NOT less than 5, stop executing the loop ****

Exercising

How Many Times Does This Loop Execute?

Time Limit: 5 minutes

```
for (int i = 6; i >= 0; --i) {
   System.out.print("*");
}
System.out.print();
```

For Loop Exercise 1

Time Limit: 5 minutes

Write a method to print out the following shape:

```
+-----+
\     /
/     \
\     /
/     \
+-----+
```

**** the height of this figure is 2 ****

Make the height of the figure parameters to your method

For Loop Exercise 2

Time Limit: 5 minutes

Write a method that prints a NASA like countdown to a rocket ship launching. Make the starting number for the countdown a parameter to the method.

Example output:

```
3…
2…
1…
Lift off!
```

For Loop Exercise 3

Time Limit: 5 minutes
Write a method that calculates the power function WITHOUT calling the Math.pow method. The base and power should be parameters to the method, they should both be whole numbers. The return value should be the answer as a whole number.

For Loop Exercise 4

Time Limit: 5 minutes
Write a method that takes in 2 chars, a start and a stop.
Return a String that contains the start to the stop, inclusive, separated by commas.

Example:
Given A and E the method would return: A, B, C, D, E

Practice For Later: Caesar Cipher

Write 2 methods, one that encodes and one that decodes words according to a Caesar cipher with the “shift amount” as a parameter and the word to encode/decode as the other parameter. The methods should return the encoded/decoded words. All answers should be in the original alphabet so wrap any elements that “fall off the end” of the alphabet in either direction.

Caesar Cipher definition: It is a type of substitution cipher in which each letter in the original word is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.

The End