# for Loop

In this lesson, the concept and implementation of for loops and nested for loops in Java is explained.

**We'll cover the following**



## for loop: syntax #

The `for` loop is a loop that lets a programmer control exactly how many times a loop will *iterate*.

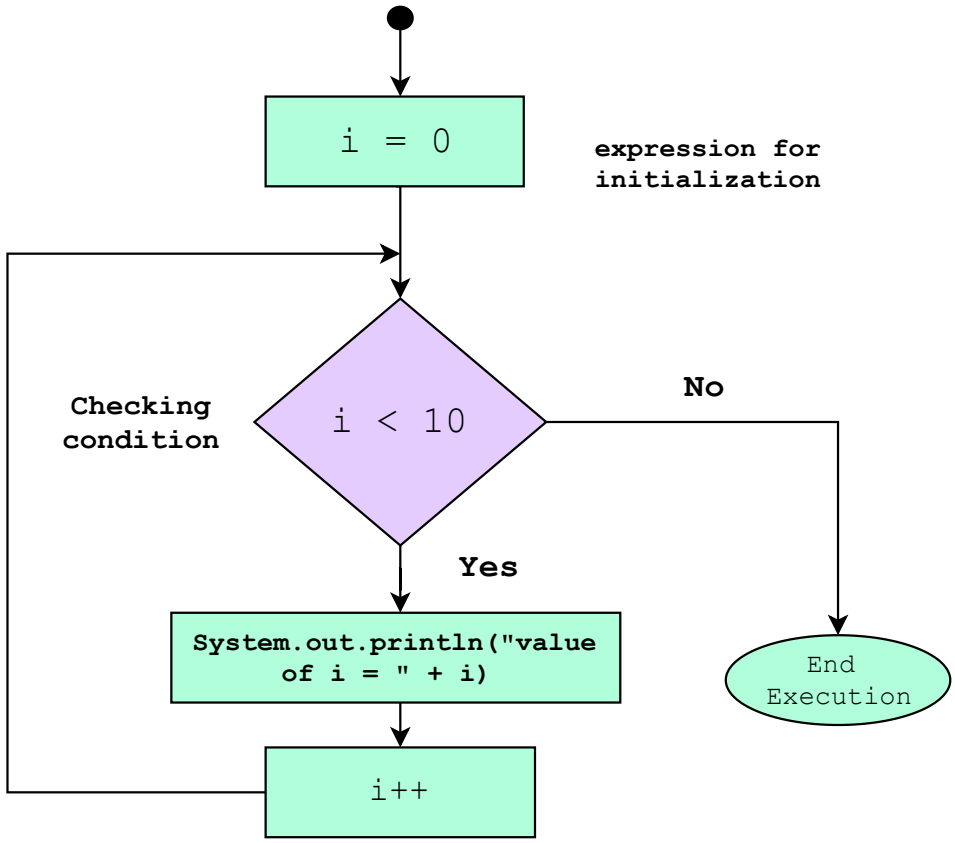
The *syntax* is as follows:

```
for (expression for initialization ; expression for testing ; expression for updating) {
  //body
}
```

Here is an example of how the `for` loop works:

```
class Loops {
    public static void main(String args[]) {
        for (int i = 0; i < 10; ++i) {
            // for loop iterates 10 times
            System.out.println("value of i = " + i);
        }
    }
}
```

Take a look at the illustration below to understand the code above more clearly.




Flow Chart for For Loop

The `for` loop code above and the `while` below are more or less equivalent.

```
class Loops {
    public static void main(String args[]) {
        int i = 0;
        while (i < 10) // while loop runs till i is less then 10 just like in for loop
        {
            System.out.println("value of i =  " + i); //prints value of i
            i++;
        }
    }
}
```

## What does the for loop do?

- Prior to the *first* iteration, it sets the value of `i` to **0**.
- Next, it tests (like a normal `while` loop) if `i` is *less* than **10**.
- If the conditional statement evaluates to `true`, the body of the loop is

executed, and the program will *print* the value of `i` to the console.

- Once all the statements in loop body are executed, `i` is incremented (by **1**), as specified in the update statement, and the conditional is tested again.

So, this loop will run a total of **10** times, printing the **"i"** value each time. You've just taught your program to count! **Wow!**

## Nested for loops #

It is possible to *nest* `for` loops. *Nesting* means including one `for` loop in another `for` loop.

The syntax for a **nested** `for` loop is as follows:

```
for (expression for initialization ; expression for testing ; expression for updating ) {
   for (expression for initialization ; expression for testing ; expression for updating) {
      //body
   }
   //body
}
```

### Example of nested for loop #

Let's take a look at an example code to understand *nesting* of `for` loops better.

```
class HelloWorld {
    public static void main(String args[]) {
        int input = 5;

        System.out.println("How many missiles will you fire?");
        System.out.println("I will fire: " + input + " missiles");

        for (int i = 0; i < input; i++) { // outer for loop
            for (int j = 3; j > 0; j--) { // inner for loop
                System.out.println(j + " ");
            }
            System.out.println("Missile " + (i + 1) + " has launched.");
        }

        System.out.println("All missiles have been launched.");
    }
}
```

In a nested `for` loop, for a single value of the **outer** loop, in this case, `i`, the inner (*nested*) `for` loop will iterate over all its values, that is, for example for `i=0` the

inner (*nested*) loop will run from `j = 3` to `j=1`. After this is done, `i` will be

incremented to **1**, and the inner loop will again iterate over all its values against this value of `i`. The process continues until all values of `i` are iterated over.

Look at the **illustration** below, which will help you visualize this and help you understand this concept more clearly.

## How does nested `for` loop work?

```
main
```

**1** of 25

```
main

input = 5
```

**2** of 25

```
main

input = 5

How many missiles will you fire?
```



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles
```



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = ?
```

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = ?
```



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = 3
```

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = 3
```

**3**



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = 3
```

**3**

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = 2
```

**3**



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = 2
```

**3 2**

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = 1
```

**3 2**



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0

j = 1
```

**3 2 1**

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0
```

**3 2 1**



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 0
```

**3 2 1**

**Missile 1 has launced.**

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 1
```



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 1

j = 3
```

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 1

j = 3
```

**3**



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 1

j = 2
```

**3**

```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 1

j = 2
```

**3 2**



```
main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 1

j = 1
```

**3 2**

main

input = 5

How many missiles will you fire?
I will fire: 5 missiles

i = 1

j = 1

**3 2 1**



main

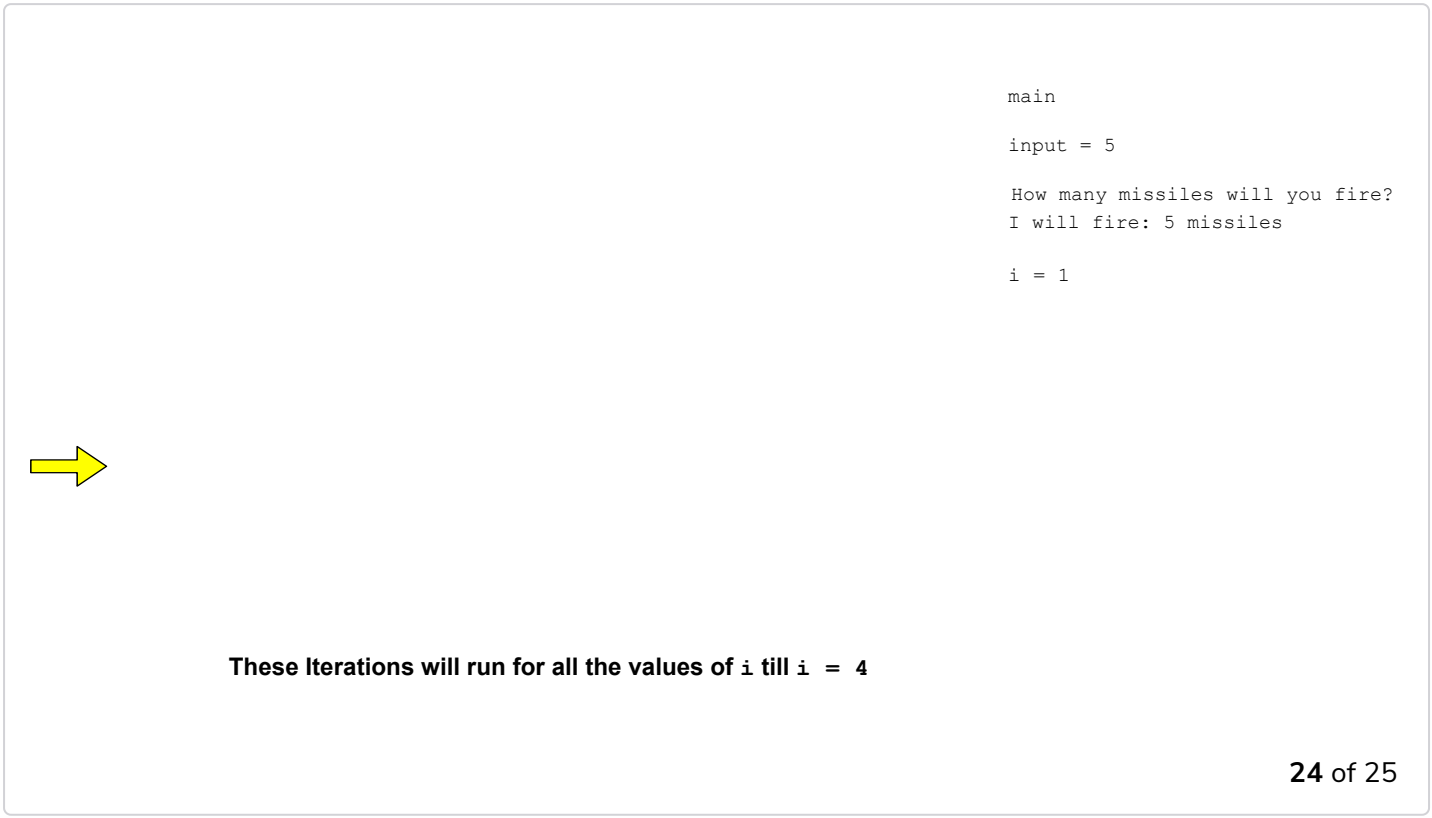
input = 5

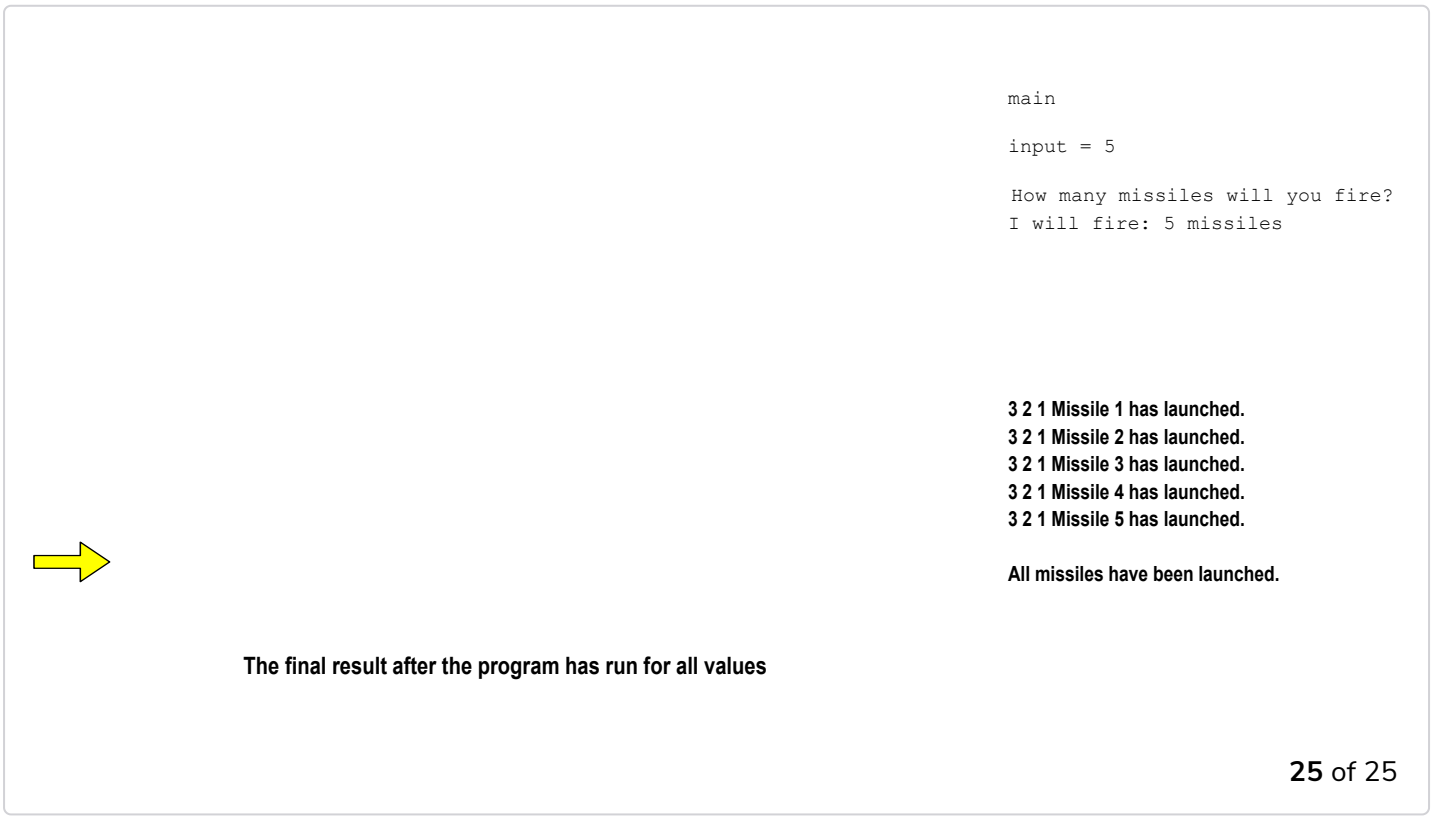
How many missiles will you fire?
I will fire: 5 missiles

i = 1

**3 2 1**

**Missile 2 has launched.**




> **Note:** Just like the `for` loop, any other loop can be nested in the same way.

---

Exciting, right? Now that the concept of `for` loops and *nested* `for` loops are clear, let's look at infinite loops in the next lesson.

let's look at infinite loops in the next lesson.