

## **2.3**

# **The for Loop**

---

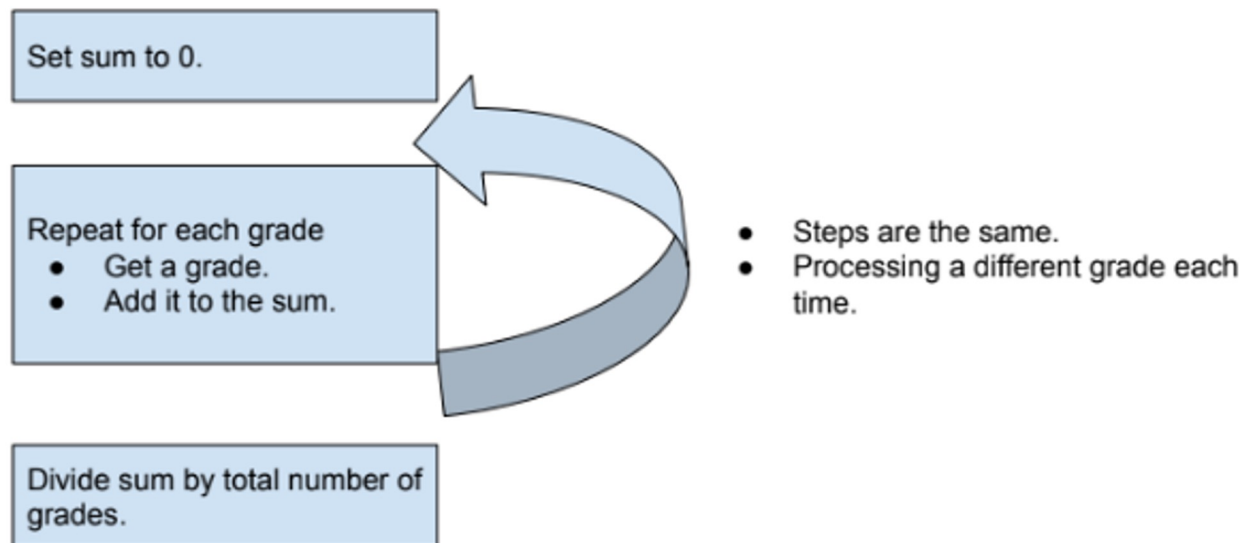
# Loops are AWESOME!

This is reasonable:

```
19 // We love Loops
20 System.out.println("The numbers 1 - 6: ");
21 System.out.println("1");
22 System.out.println("2");
23 System.out.println("3");
24 System.out.println("4");
25 System.out.println("5");
26 System.out.println("6");
```

But, what if we needed to print thousands of numbers?  
What would we need to know?

# For Loop Concept



# Control Structures

- The Loop is our first Control Structure. Control Structures alter the default top-to-bottom execution of program statements.
- Control Structures:
  - Loops
  - Branching (decisions)
  - continue and break (not used in this course)

# In-Class Practice Answers

## Question 1

24 asterisks in one line

## Question 2

```
38     int factorial = 1;
39     int num = 4;
40     for (int i=1; i<=num; i++) {
41         factorial *= i;
42     }
43     System.out.println (num + "! = " + factorial);
```

# Nested For Loops

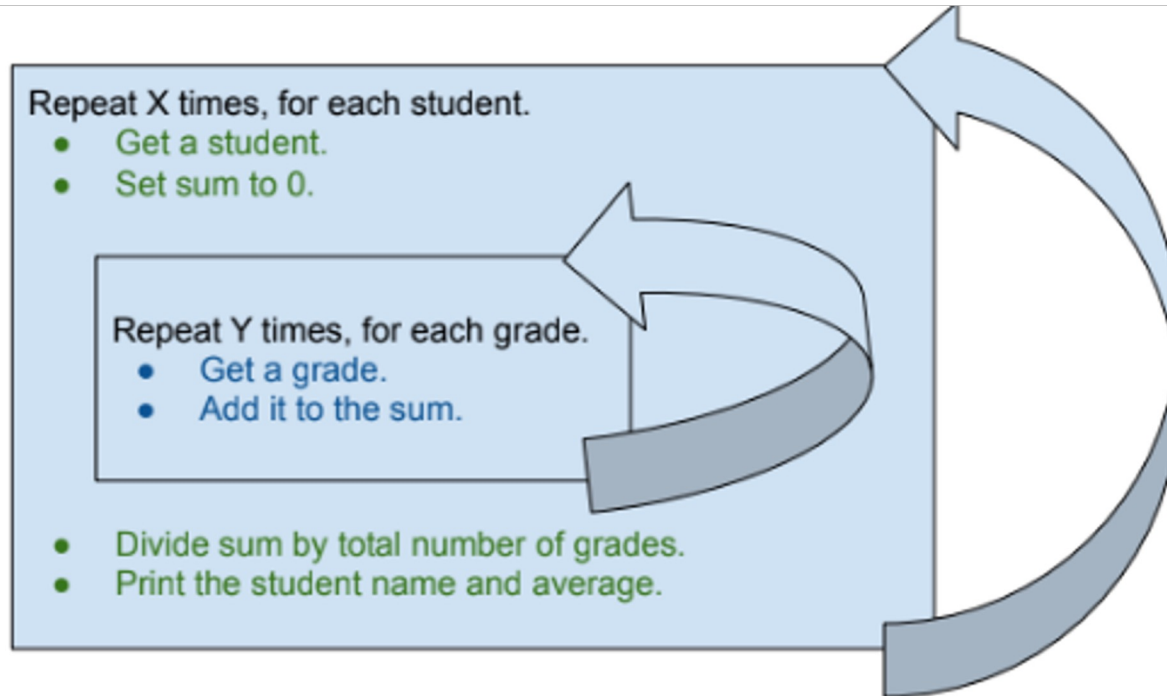
# What About Rows and Columns?

What if I want to print this?

```
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

A for loop could print the first line. Would I need 10 for loops?

# Nested For Loop Concept



- Lines of code in green execute X times.
- Lines of code in blue execute X \* Y times.



# Variable Scope and the for() loop

**scope:** The part of a program where a variable exists. A variable exists from its declaration to the end of the `}` in which it was declared.

79

80

81

82

```
for (int i = 0; i < 10; i++) {  
    System.out.print(i + " ");  
}  
System.out.println(i);
```

# Blueprint - Nested For Loops

Two different counters, typically named i, j, k..., write down what each one means.

Typically, both counters used within the loop.

```

66 int numRows = 5, numCols = 3;
67 for (int i=0; i<numRows; i++) {
68     for (int j=0; j<numCols; j++) {
69         System.out.print("R" + (i + 1) + "C" + (j + 1) + " ");
70     }
71     System.out.println();
72 }
    
```

Output		
R1C1	R1C2	R1C3
R2C1	R2C2	R2C3
R3C1	R3C2	R3C3
R4C1	R4C2	R4C3
R5C1	R5C2	R5C3

Loop	i	j
1	0	0
2	0	1
3	0	2
4	1	0
5	1	1
6	1	2
7	2	0
8	2	1
9	2	2
10	3	0
11	3	1
12	3	2
...	...	...

# Blueprint - For Loop

1. Initialize: declare and initialize counter variable, traditionally named *i*, happens one time

2. Test: a true/false expression, loop stops when false, happens at the top of each loop

3. Increment: change the counter value, happens at end of each loop

```

8   int size = 10;
9   for (int i = 0; i < size; i++) {
10      System.out.println ("Loop # = " + (i + 1) + ", i = " + i);
11  }

```

Humans count starting at 1.

Computers count starting at 0.

Statements in body of loop enclosed by curly braces.

Output
Loop # = 1, i = 0
Loop # = 2, i = 1
Loop # = 3, i = 2
Loop # = 4, i = 3
Loop # = 5, i = 4
Loop # = 6, i = 5
Loop # = 7, i = 6
Loop # = 8, i = 7
Loop # = 9, i = 8
Loop # = 10, i = 9

1. counter starts at 0.

2. last loop when  $i = 3$ .

3. Add one to counter.

```
14  
15  
16
```

```
for (int i = 0; i < 4; i++) {  
    // statements  
}
```

You will use this loop a zillion times, only changing the number of times you want the loop to execute.