

Loop-de-Loop

Nested iteration



Write out the whole plan

PAUSE AND TRY

A coach runs 3 rounds of exercise. Each round is 4 jumping jacks.

THE COACH CALLS OUT

"Round 1!"

"Jack 1, jack 2, ..."

...

Write everything the coach says, in order.
How many times does the coach say
"jack"? Which count starts over?

By the end of class you can...

- 1 Spot a nested loop and name its outer loop and inner loop.
- 2 Explain why the inner loop starts over and finishes before the outer loop moves on.
- 3 Trace nested loops and predict their exact output.
- 4 Use print and println in nested loops to print rows and columns.
- 5 Predict what happens when the inner loop uses the outer loop's variable.

1

PART 1

A loop inside a loop

The inner loop lives in the outer loop's body

```
for (int table = 1; table <= 2; table++) {  
    System.out.println("Table " + table);  
    for (int seat = 1; seat <= 3; seat++) {  
        System.out.println("    seat " + seat);  
    }  
}
```

OUTPUT

```
Table 1  
    seat 1  
    seat 2  
    seat 3  
Table 2  
    seat 1  
    seat 2  
    seat 3
```



A **nested loop** is a loop inside the body of another loop. The one outside is the **outer loop**; the one inside is the **inner loop**.

The inner loop finishes first

```
for (int a = 1; a <= 2; a++) {  
    for (int b = 1; b <= 3; b++) {  
        System.out.println(a + " " + b);  
    }  
}
```

OUTPUT

```
1 1  
1 2  
1 3  
2 1  
2 2  
2 3
```

Each outer pass, the inner loop **starts over** and runs **all** its iterations. The outer variable **holds still** meanwhile.

a	b	Check	Prints
1	1	b <= 3 true	1 1
1	2	b <= 3 true	1 2
1	3	b <= 3 true	1 3
1	4	b <= 3 false	
2	1	b <= 3 true	2 1
2	...	...	...

b goes back to 1

Don't reuse the outer loop's name



The outer variable still exists inside the outer body. The inner loop needs **its own name**.

DOES NOT COMPILE

```
for (int i = 1; i <= 2; i++) {  
    for (int i = 1; i <= 3; i++) {  
        System.out.println(i);  
    }  
}
```

COMPILER SAYS

```
error: variable i is already defined  
in method main(String[])
```

Fix: for (int j = 1; j <= 3; j++)

Count down, then count up

PAUSE AND TRY

Write the exact output.

```
for (int n = 3; n >= 2; n--) {  
    for (int k = 1; k <= 2; k++) {  
        System.out.println(n + " x " + k + " = " + (n * k));  
    }  
}
```

Hint: which variable changes on every line?

A large, white, rounded square containing the number '2' in a bold, teal, sans-serif font.

2

PART 2

Rows and columns

Inner loop: one row. Outer loop: every row.

```
for (int row = 1; row <= 3; row++) {  
    for (int col = 1; col <= 4; col++) {  
        System.out.print("[ ]");  
    }  
    System.out.println();  
}
```

OUTPUT

```
[ ][ ][ ][ ]  
[ ][ ][ ][ ]  
[ ][ ][ ][ ]
```

Outer loop

how many rows

Inner loop + print

one row, column by column

println() after the inner }

ends each row once

Print the inner variable, or the outer one?

PRINT(C): INNER VARIABLE

```
for (int r = 1; r <= 2; r++) {  
    for (int c = 1; c <= 5; c++) {  
        System.out.print(c);  
    }  
    System.out.println();  
}
```

OUTPUT

```
12345  
12345
```

PRINT(R): OUTER VARIABLE

```
for (int r = 1; r <= 2; r++) {  
    for (int c = 1; c <= 5; c++) {  
        System.out.print(r);  
    }  
    System.out.println();  
}
```

OUTPUT

```
11111  
22222
```

The inner variable changes across a row. The outer variable changes only from row to row.

println() inside the inner loop



Inside the inner braces means **once per column**. Ending a row belongs after the inner }.

WANTED 2 ROWS OF [][][]

```
for (int row = 1; row <= 2; row++) {  
    for (int col = 1; col <= 3; col++) {  
        System.out.print("[]");  
        System.out.println();  
    }  
}
```

OUTPUT

```
[]  
[]  
[]  
[]  
[]  
[]
```

Where do the X's land?

PAUSE AND TRY

Write the exact output, one row per line.

```
for (int r = 1; r <= 3; r++) {  
    for (int c = 1; c <= 4; c++) {  
        if (c == r) {  
            System.out.print("X");  
        } else {  
            System.out.print(".");  
        }  
    }  
    System.out.println();  
}
```



3

PART 3

Inner depends on outer

A different count on every pass

```
for (int r = 1; r <= 5; r++) {  
    for (int c = 1; c <= r; c++) {  
        System.out.print("o");  
    }  
    System.out.println();  
}
```

OUTPUT

```
O  
OO  
OOO  
OOOO  
OOOOO
```

r	Inner check	Inner runs
1	c <= 1	1
2	c <= 2	2
3	c <= 3	3
4	c <= 4	4
5	c <= 5	5
Total		15

Inner condition uses the outer variable

Two inner loops in one row

```
for (int r = 1; r <= 3; r++) {  
    for (int s = 1; s <= 3 - r; s++) {  
        System.out.print(" ");  
    }  
    for (int c = 1; c <= r; c++) {  
        System.out.print("*");  
    }  
    System.out.println();  
}
```

OUTPUT

```
  *  
 * *  
* * *
```

r	Spaces	Stars
1	2	1
2	1	2
3	0	3



An inner loop whose condition is false at the start runs **zero times**. That is not an error.

An off-by-one repeats on every row



A mistake in the inner loop's condition happens **again on every row**.

WANTED 1 TO 4 O'S

```
for (int r = 1; r <= 4; r++) {  
    for (int c = 1; c < r; c++) {  
        System.out.print("o");  
    }  
    System.out.println();  
}
```

OUTPUT

(empty first line)

o
oo
ooo

Fix: `c <= r`

Where does the inner loop start?

PAUSE AND TRY

Write the exact output. How many stars in all?

```
for (int r = 1; r <= 3; r++) {  
    for (int c = r; c <= 5; c++) {  
        System.out.print("*");  
    }  
    System.out.println();  
}
```

Every letter against every later letter

```
String w = "banana";
int pairs = 0;
for (int i = 0; i < w.length(); i++) {
    String a = w.substring(i, i + 1);
    for (int j = i + 1; j < w.length(); j++) {
        if (a.equals(w.substring(j, j + 1))) {
            System.out.println(a + " at " + i + " and " + j);
            pairs++;
        }
    }
}
System.out.println("Pairs: " + pairs);
```

OUTPUT

```
a at 1 and 3
a at 1 and 5
n at 2 and 4
a at 3 and 5
Pairs: 4
```

INDEXES

0	1	2	3	4	5
b	a	n	a	n	a

Words to know

Nested loop

A loop written inside the body of another loop.

Nested iteration

Repeating a task inside each repetition of another task.

Outer loop

The loop whose body holds another loop. Its variable changes only after the inner loop finishes.

Inner loop

The loop inside the outer loop's body. It starts over and runs completely on every outer pass.

Who moves first?



Explain to your partner the order in which a nested loop runs. Use the seating example or your own.

SENTENCE STARTER

Each time the outer loop runs, the inner loop ____ and ____ before the outer variable ____.

Next: the Loop-de-Loop puzzle worksheet.