

NAME _____

DATE _____

PERIOD _____

UNIT 2 · TOPIC 2.11 · WORKSHEET

Loop-de-Loop

Trace a loop that lives inside another loop, build rows and columns, count the passes, and catch the counter that forgot to start over.

About 35 minutes

10 questions

Cumulative through 2.11

TOPIC

2.11

**DIRECTIONS**

Work each puzzle in pencil. For code, write exactly what prints. Remember: `print` stays on the same line and `println` ends the line. Some puzzles include pieces you won't need.

PART A · 2 QUESTIONS

A

Warm up your brain

1

YOU'RE THE JVM

Run this code in your head. Every `print` stays on one line. What prints?

```
for (int x = 1; x <= 2; x++) {  
    for (int y = 1; y <= 3; y++) {  
        System.out.print(x);  
        System.out.print(y);  
    }  
    System.out.print("|");  
}
```

- A** 111213|212223|
- B** 112131|122232|
- C** 11|12|13|21|22|23|
- D** 111213212223|

2 PENCIL CHECK

This inner loop's condition uses the outer loop's variable. Fill in the table, one row for each value of r .

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

r	Times the inner body runs	Line printed
1		
2		
3		
4		
All four rows		



THINK FIRST

An inner while loop counts k up from 1 to 3. When the outer loop comes around again, does k go back to 1 by itself? Guess in the margin; question 9 settles it.

PART B · 4 PUZZLES

B Puzzles

3 CODE TILES

The outer loop is still on the fridge, but everything inside it fell off. Put the tiles back inside the outer loop's braces so the program prints the target output. One tile is a decoy and isn't used.

```
for (int row = 1; row <= 2; row++) {
    // your tiles go here
}
```

System.out.println();	System.out.print("***");
System.out.println("***");	}
System.out.print("Row " + row + ": ");	for (int k = 1; k <= 3; k++) {

Target output

Row 1: ***

Row 2: ***

YOUR PROGRAM

4 SNIPPET POOL

Fill each numbered hole with a snippet from the pool so the program prints the output below. Each snippet is used at most once.

Shrink.java

```
for (int row = 1; (1); row++) {
    for (int col = (2); col <= 4; col++) {
        System.out.print(col);
    }
    (3);
}
```

Target output

1234

234

34

THE POOL

Each snippet can be used only once, and not every snippet is needed.

row <= 3	1	System.out.print(row)
row < 3	row + 1	System.out.println()
row		

5 MATCH-UP

Each version below runs between `int count = 0;` and `System.out.println(count);`.

Version 1 is the original; each other version changes one thing. Match each version to the number it prints. Each output is used once.

Version	Code	Prints
1	<pre>for (int i = 1; i <= 4; i++) { for (int j = 1; j <= 3; j++) { count++; } }</pre>	
2	<pre>for (int i = 1; i <= 4; i++) { for (int j = 1; j <= i; j++) { count++; } }</pre>	
3	<pre>for (int i = 1; i <= 4; i++) { for (int j = 1; j <= 3; j++) { } count++; }</pre>	
4	<pre>for (int i = 1; i <= 4; i++) { for (int j = i; j <= 3; j++) { count++; } }</pre>	

- **P** 4
- **Q** 10
- **R** 6
- **S** 12



FAIR QUESTION

Can a while loop go inside a for loop?

Yes. Any kind of loop can go inside any kind of loop: `for` in `for`, `while` in `for`, `for` in `while`, `while` in `while`. The rule is the same for all of them. The inner loop is just one more statement in the outer loop's body, so it runs from start to finish every time that body runs.

6 FACE-OFF

Tonight: the Outer Loop and the Inner Loop argue about who's really in charge. Fill in the blanks.

OUTER LOOP	I'm the boss. While you run, my variable stays _____. I don't move until you're done.
INNER LOOP	Because every time you come around, I start _____ and run _____ of my iterations before you get to move on.
OUTER LOOP	So if I run 5 times and you run 2 times on each of my passes, your body runs _____ times.
INNER LOOP	Only if my condition uses a fixed number. If it uses _____ variable, I can run a different number of times on each pass.
OUTER LOOP	People say I'm like the slow digit on a car's odometer.
INNER LOOP	And I'm the _____ one. I roll through every one of my values before you tick up by one.



WATCH OUT

Indenting a line doesn't put it inside a loop; braces do. To see which loop a statement belongs to, find the } it comes after, not how far it's pushed in.

PART C · 4 QUESTIONS

C

Catch the mistake

7

YOU'RE THE COMPILER

You are the compiler. For each snippet, decide whether it compiles. If it does, write what it prints. If it doesn't, say why in a few words.

Snippet	Compiles?	Output, or why not
<pre>1 for (int i = 1; i <= 2; i++) { for (int i = 1; i <= 3; i++) { System.out.print("x"); } }</pre>		
<pre>2 for (int a = 1; a <= 2; a++) { for (int b = 1; b <= 2; b++) { System.out.print(a + b); } }</pre>		
<pre>3 for (int a = 1; a <= 2; a++) { for (int b = 1; b <= 2; b++) { System.out.print(b); } System.out.print(b); }</pre>		
<pre>4 int n = 1; while (n <= 2) { for (int k = 0; k < n; k++) { System.out.print(n); } n++; }</pre>		

8 TWO TRUTHS AND A LIE

This code counts pairs of matching letters in a word. Two of the claims are true and one is a lie. Circle the lie, then rewrite it so it's true.

```
String w = "tooth";
int same = 0;
for (int i = 0; i < w.length(); i++) {
    for (int j = i + 1; j < w.length(); j++) {
        if (w.substring(i, i + 1).equals(w.substring(j, j + 1))) {
            same++;
        }
    }
}
```

- **1** While the inner loop runs with `i` equal to 1, `w.substring(i, i + 1)` stays "o" the whole time.
- **2** When `i` is 4, the inner loop's body doesn't run at all.
- **3** `same` ends at 4, because each matching pair gets counted twice: once from each letter.

THE LIE, REWRITTEN

9 ONE-LINE MOVE

Three boxes each hold 4 cookies, so this program should print 12. It prints 4. Move exactly one line (cut it and paste it somewhere else, without changing it) so it prints 12.

```
1  int total = 0;
2  int box = 1;
3  int cookie = 1;
4  while (box <= 3) {
5      while (cookie <= 4) {
6          total++;
7          cookie++;
8      }
9      box++;
10 }
11 System.out.println(total);
```

WHICH LINE MOVES, WHERE IT GOES, AND WHY IT PRINTS 4 NOW

10 WRAP-UP**REFLECTION**

In one sentence: why does a for inner loop's variable start over on every outer pass, while the outer loop's variable keeps going?

YOUR ANSWER