

NAME _____

DATE _____

PERIOD _____

UNIT 1 · TOPIC 1.3 · WORKSHEET

Cursor and Calculator

Chase the cursor, untangle escape sequences, and make integer division confess what it threw away.

About 30 minutes

10 questions

Cumulative through 1.3

TOPIC

1.3

**DIRECTIONS**

Work each puzzle in pencil. When you write output, copy it exactly, including spaces, and show where each new line starts. Code shown without a class goes inside `main`. Some puzzles include pieces you won't need.

PART A · 2 QUESTIONS

A

Warm up your brain

1**PENCIL CHECK**

Write the value of each expression. Watch the types: if the value is a `double`, write it the way Java displays it.

Expression	Value
<code>19 / 4</code>	
<code>19 % 4</code>	
<code>19 / 4.0</code>	
<code>3 % 7</code>	
<code>5 * 2.0</code>	
<code>2 + 6 / 2</code>	

2 YOU'RE THE JVM

You are the computer running this code. Write exactly what appears on the console, line by line.

```
1 System.out.print("Hop");
2 System.out.print("per");
3 System.out.println();
4 System.out.println("is");
5 System.out.print("here");
6 System.out.println("!");
```

CONSOLE OUTPUT**WATCH OUT**

`printIn` with a capital I looks almost like `println` in many fonts. Java is case-sensitive, so `System.out.printIn("Hi");` is a syntax error (cannot find symbol). It's p-r-i-n-t-l-n: "print line."

**THINK FIRST**

`1 / 2 * 4.0` and `4.0 * 1 / 2` use the same three numbers and the same two operators. Do they have the same value? Guess in the margin; question 9 settles it.

PART B · 5 PUZZLES

B Puzzles

3 CODE TILES

Hopper shares 17 stickers among 5 friends. Stickers can't be cut. Put tiles in order so the code displays the output below, with spaces exactly as shown. Two tiles are decoys.

<code>System.out.println(17 / 5.0);</code>	<code>System.out.print("Each: ");</code>
<code>System.out.println(17 % 5);</code>	<code>System.out.print("Left:");</code>
<code>System.out.println(17 / 5);</code>	<code>System.out.print("Left: ");</code>

Target output

Each: 3

Left: 2

YOUR CODE

4 YOU'RE THE COMPILER

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

Snippet	Compiles?	Output, or why not
1 <code>System.out.println("Say \"cheese\"");</code>		
2 <code>System.out.println("C:\data");</code>		
3 <code>System.out.println(10 / 4);</code>		
4 <code>System.out.println>Hello);</code>		
5 <code>System.out.println(8 / 0);</code>		
6 <code>System.out.println(9 % 3 + 2);</code>		



FAIR QUESTION

Why doesn't integer division round to the nearest whole number?

Because it isn't trying to approximate. It answers "how many whole groups fit?" Twenty-nine students in vans that hold 8: three full vans, not "3.625 vans" or "4 vans." The 5 students left over are what % reports.

5 SNIPPET POOL

Hopper got 85 and 90 on two quizzes. Fill each hole with an expression from the pool so the code displays the output below. Each snippet is used at most once, and some aren't needed.

```
int quiz1 = 85;
int quiz2 = 90;
System.out.print("Total: ");
System.out.println();
System.out.print("Average: ");
System.out.println();
System.out.print("Half of quiz 1: ");
System.out.println();
```

Target output

Total: 175

Average: 87.5

Half of quiz 1: 42

THE POOL

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

quiz1 + quiz2	(quiz1 + quiz2) / 2	(quiz1 + quiz2) / 2.0
quiz1 + quiz2 / 2.0	quiz1 / 2	quiz1 / 2.0
quiz1 % 2		

6 MATCH-UP

Four statements look almost the same. Match each one to what it displays. Each letter is used once.

Statement	Code	Displays
1	<code>System.out.println("A\nB");</code>	
2	<code>System.out.println("A\\nB");</code>	
3	<code>System.out.println("A/nB");</code>	
4	<code>System.out.println("\"A\" B");</code>	

- **P** A\nB on one line
- **Q** "A" B on one line
- **R** A on one line, then B on the next
- **S** A/nB on one line

7 FACE-OFF

Tonight: Divide (/) and Remainder (%) argue about 17 cookies shared by 5 friends. Fill in the blanks.

DIVIDE With two `int` values I give whole groups only. `17 / 5` is _____, and I never _____, so `19 / 5` is still 3.

REMAINDER You throw the leftovers away. I keep them: `17 % 5` is _____.

DIVIDE When the first number is smaller, I give up: `3 / 5` is _____.

REMAINDER And I hand back the whole first number: `3 % 5` is _____.

DIVIDE People see you and think _____. You're not.

REMAINDER At least nobody calls me with 0 on the right. When they do that to you, Java throws an _____.

PART C · 3 QUESTIONS

C

Catch the mistake

8

ONE-TOKEN FIX

This code displays `Lunch: 12Snack: 2` on one line. It should display `Lunch: 12` on one line and `Snack: 2` on the next. Change exactly one token to fix it.

```
1 System.out.print("Lunch: ");
2 System.out.print(4 * 3);
3 System.out.print("Snack: ");
4 System.out.println(2);
```

YOUR FIX, AND WHY IT WORKS

9

TWO TRUTHS AND A LIE

Two claims about this code are true and one is a lie. Circle the lie, then rewrite it so it's true.

```
1 System.out.println(1 / 2 * 4.0);
2 System.out.println(4.0 * 1 / 2);
```

- **1** Line 1 displays `0.0`.
- **2** Line 2 displays `2.0`.
- **3** Both lines contain the double `4.0`, so Java does every operation in both lines with double values, and both lines display the same thing.

THE LIE, REWRITTEN

10 WHO AM I?

Some ideas from today are playing a party game. Each gives a clue about itself. Name the idea.

Clue	I am...
I display a value and leave the cursor right after it.	
I display a value and then move the cursor to the start of the next line.	
I'm two characters inside a string literal, and I stand for a new line.	
I report what's left over after dividing one whole number by another.	
I'm the rule that makes Java do * before +.	
I'm thrown when a running program divides an <code>int</code> by the <code>int 0</code> .	