

NAME _____

DATE _____

PERIOD _____

UNIT 2 · TOPIC 2.5 · WORKSHEET

And, Or, Not

Snap conditions together, guard a risky division, and catch the operator that's in the wrong job.

About 35 minutes

10 questions

Cumulative through 2.5

TOPIC

2.5

**DIRECTIONS**

Work each puzzle in pencil. For code, write exactly what prints. Some puzzles include pieces you won't need.

PART A · 3 QUESTIONS

A

Warm up your brain

1

YOU'RE THE JVM

You are the Java Virtual Machine. Run this code in your head. What prints?

```
boolean a = true;
boolean b = false;
boolean c = false;
boolean first = a || b && c;
boolean second = (a || b) && c;
System.out.println(first + " " + second);
```

A true false**B** false false**C** true true**D** false true

2 PENCIL CHECK

Hopper is deciding whether to go to the pond. Use these values to fill in the value of each expression: true or false.

```
int temp = 72;
boolean raining = false;
boolean windy = true;
```

Expression	Value
!raining	
temp > 60 && raining	
temp > 60 raining	
!(temp > 80)	
temp < 50 temp > 90	
windy && !raining && temp >= 72	

3 FACE-OFF

Tonight: && and || argue about who's easier to please. Fill in the blanks.

&&	I have standards. I'm true only when _____ of my sides are true.
	I'm easygoing. _____ true side is enough for me, and two is fine too.
&&	Efficient, too. If my left side is _____, I don't even look at my right side.
	Same trick, except I stop early when my left side is _____.
&&	And when we share a line with no parentheses, I go _____.
	Fine. But neither of us outranks _____. It flips its value before either of us gets a turn.

PART B · 3 PUZZLES

B Puzzles

4 CODE TILES

The Lily Pad Plunge has a rule: you can ride alone if you're at least 48 inches tall **and** at least 10 years old. If you're tall enough but not old enough, you ride with an adult buddy. The first line stayed on the fridge, but the rest of the program fell off. Put the tiles back in order so it prints the output below. One tile is a decoy and isn't used.

```
int age = 9;
```

```
boolean buddy = tall && !alone;
```

```
System.out.println(alone + " " + buddy);
```

```
boolean alone = tall || age >= 10;
```

```
int height = 50;
```

```
boolean alone = tall && age >= 10;
```

```
boolean tall = height >= 48;
```

Target output

```
false true
```

YOUR PROGRAM



THINK FIRST

When the left side of && is already false, does Java still bother to check the right side? Guess in the margin; question 5 settles it.

5 SNIPPET POOL

Hopper's grade book prints Honor roll when there is at least one score **and** the average is at least 80. Fill each numbered hole from the pool so the code works in all three cases below.

- total 270, count 3: prints Honor roll
- total 150, count 2: prints Not yet
- total 0, count 0: prints Not yet and does **not** crash

HonorRoll.java

```
// total and count already hold the scores
if ( (1) (2) (3) ) {
    System.out.println("Honor roll");
} else {
    System.out.println("Not yet");
}
```

THE POOL

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

total / count >= 80	count == 0	
count > 0	total / count > 90	&&



FAIR QUESTION

Isn't skipping the right side a little lazy?

It's safe, not lazy. Once the left side of && is false, nothing on the right can make the whole thing true, and once the left side of || is true, nothing can make it false. Java skips only a side that can't change the answer, and that's exactly what lets a guard protect the risky part.

6 YOU'RE THE COMPILER

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

Snippet	Compiles?	Output, or why not
1 int level = 15; System.out.println(10 <= level <= 20);		
2 int month = 12; System.out.println(month == 12 1);		
3 int lives = 3; System.out.println(!lives > 0);		
4 int n = 0; System.out.println(n != 0 && 12 / n > 2);		
5 int n = 0; System.out.println(12 / n > 2 && n != 0);		

PART C · 4 QUESTIONS

C Catch the mistake

7 ONE-TOKEN FIX

The pond shop's weekend sale runs on day 6 (Saturday) and day 7 (Sunday). This program prints `Regular price` even on day 7. Change exactly one token so it prints `Weekend sale!` on days 6 and 7 and `Regular price` on every other day.

```

1  int day = 7;
2  if (day == 6 && day == 7) {
3      System.out.println("Weekend sale!");
4  } else {
5      System.out.println("Regular price");
6  }

```

LINE, OLD TOKEN, NEW TOKEN, AND WHY THE OLD ONE NEVER WORKS



WATCH OUT

English "and" can trick you. "Kids under 5 and seniors ride free" sounds like `age < 5 && age >= 65`, but nobody is both, so nobody rides free. Ask: must one person pass **both** tests (`&&`), or is passing **either** test enough (`|`)?

8 TWO TRUTHS AND A LIE

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

```
boolean hasPass = false;
boolean weekend = true;
boolean rainy = true;
boolean swim = hasPass || weekend && !rainy;
```

- 1 swim holds false.
- 2 Java reads the last line as `(hasPass || weekend) && !rainy`.
- 3 If rainy were false instead, swim would hold true.

THE LIE, REWRITTEN

9 WHO AM I?

Some logic ideas are playing a party game. Each gives a clue about itself. Name the operator or idea.

Clue	I am...
I take one boolean and hand back the opposite.	
I'm true only when both of my sides are true.	
One true side is all I need.	
I'm why the right side of <code>&&</code> sometimes never runs.	
I sit on the left of <code>&&</code> so the risky part on the right runs only when it's safe.	
Put me around part of a condition, and that part goes first.	

10 WRAP-UP**REFLECTION**

In one sentence: how do you decide whether a condition needs `&&` or `|`? Use an example from school.

YOUR ANSWER