

NAME \_\_\_\_\_

DATE \_\_\_\_\_

PERIOD \_\_\_\_\_

## UNIT 2 · TOPIC 2.4 · WORKSHEET

# Branches on Branches

Guard a risky check, put a grade chain in order, and catch the else that wandered off to the wrong if.

About 35 minutes

10 questions

Cumulative through 2.4

TOPIC

# 2.4

**DIRECTIONS**

Work each puzzle in pencil. For code, write exactly what prints, one line per `println`. If nothing prints, write *nothing*. Some puzzles include pieces you won't need.

## PART A · 2 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?

```
int points = 0;
int level = 7;
if (level > 5) {
    points += 10;
} else if (level > 3) {
    points += 5;
} else {
    points += 1;
}
System.out.println(points);
```

**A** 10**B** 15**C** 16**D** 1



### THINK FIRST

The outer condition of a nested if is false. Can the *inner* else still run? Guess in the margin; question 2 settles it.

## 2 PENCIL CHECK

A ride sign is set by this code. For each rider, write the value of sign after it runs.

```
String sign = "Too short";
if (height >= 48) {
    if (age >= 10) {
        sign = "Ride alone";
    } else {
        sign = "Ride with a grown-up";
    }
}
```

| height | age | sign |
|--------|-----|------|
| 52     | 12  |      |
| 52     | 8   |      |
| 44     | 12  |      |
| 48     | 10  |      |
| 40     | 6   |      |

## PART B · 3 PUZZLES

### B Puzzles

#### 3 CODE TILES

Hopper's grade app finds the quiz average, but a student with no quizzes yet would make it divide by zero. Put all six tiles where the comment is so the program prints the target output for all three tests, without crashing.

```
int quizzes = 0;    // also tested with 4 quizzes
int total = 0;      // ... totaling 340, then 200
// tiles go here
}
System.out.println("Checked");
```

|                              |                                   |
|------------------------------|-----------------------------------|
| } else {                     | System.out.println("Keep going"); |
| if (total / quizzes >= 80) { | }                                 |
| if (quizzes > 0) {           | System.out.println("Honor roll"); |

| Test                 | Target output         |
|----------------------|-----------------------|
| 0 quizzes, total 0   | Checked               |
| 4 quizzes, total 340 | Honor roll<br>Checked |
| 4 quizzes, total 200 | Keep going<br>Checked |

YOUR TILES, IN ORDER



#### FAIR QUESTION

#### How deep can I nest if statements?

As deep as you like; Java doesn't mind. People do. Every level is one more thing a reader has to hold in their head, so three or four levels deep is usually a sign that the code can be laid out more clearly. You'll meet more tools for that soon.

#### 4 SNIPPET POOL

Scores of 90 and up get an A, 80 to 89 a B, 70 to 79 a C, and anything lower an F. Fill each hole from the pool so the code gives the right grade for *every* score. Hopper tests it with 95, 90, 85, 80, 72, 70, and 65, and expects A A B B C C F.

Grades.java

```
String grade;
if (score ) {
    grade = "A";
} else if (score ) {
    grade = "B";
} else if (score ) {
    grade = "C";
} else {
    grade = "F";
}
System.out.println(grade);
```

#### THE POOL

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

|      |       |       |       |
|------|-------|-------|-------|
| > 80 | >= 70 | >= 60 | >= 90 |
| < 70 | >= 80 | > 90  |       |



#### WATCH OUT

`else if` is two words with a space between them. Java has no `elseif` keyword, so writing `elseif (x > 3)` won't compile. And an `else` never gets a condition of its own.

## 5 FACE-OFF

Tonight: an `else if` chain and a row of separate `if` statements argue about who's better at grading. Fill in the blanks.

|              |                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHAIN        | I check my conditions in order and run the body of the _____ one that's true. Then I skip the rest.                                                  |
| SEPARATE IFs | We never skip. Each of us checks our condition _____ time, so several bodies can run, or _____.                                                      |
| CHAIN        | That's why a score of 95 gets exactly _____ grade from me.                                                                                           |
| SEPARATE IFs | Sure, but you're fussy about order. Check <code>score &gt;= 60</code> before <code>score &gt;= 90</code> , and a 95 never reaches your _____ branch. |
| CHAIN        | And when none of my conditions is true, my trailing _____ still catches the score.                                                                   |

## PART C · 5 QUESTIONS

### C Catch the mistake

#### 6 TWO TRUTHS AND A LIE

This code has no braces, and its indentation is misleading. Two of the claims are true and one is a lie. Circle the lie, then rewrite it so it's true.

```
int x = 5;
if (x > 10)
    if (x > 20)
        System.out.println("A");
else
    System.out.println("B");
System.out.println("C");
```

- 1 It prints only C.
- 2 The `else` belongs to `if (x > 20)`, the nearest `if` that doesn't have an `else` yet.
- 3 Because `x > 10` is false, the `else` runs and prints B.

THE LIE, REWRITTEN

## 7 ONE-TOKEN FIX

A score of 92 should earn only Gold, but this prints Gold and then Silver. Add, change, or delete exactly one token so it prints one medal for every score: Gold for 90 and up, Silver for 75 to 89, Bronze below that.

```
1 int score = 92;
2 if (score >= 90) {
3     System.out.println("Gold");
4 }
5 if (score >= 75) {
6     System.out.println("Silver");
7 } else {
8     System.out.println("Bronze");
9 }
```

LINE, TOKEN, AND WHY IT FIXES IT

## 8 WHO AM I?

Some selection ideas are playing a party game. Each gives a clue about itself. Name it.

| Clue                                                                         | I am... |
|------------------------------------------------------------------------------|---------|
| I'm an if statement living inside another if statement's body.               |         |
| I'm checked only when every condition above me in the chain was false.       |         |
| I'm the last stop in a chain. I run when no condition was true.              |         |
| I'm an outer condition that keeps a risky inner check from crashing.         |         |
| Without braces, I pair with the nearest if above me that has no partner yet. |         |
| I'm an if, else if, else chain that runs at most one body.                   |         |

## 9 YOU'RE THE COMPILER

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

| Snippet                                                                                                                                                                               | Compiles? | Output, or why not |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------|
| <pre>1 int n = 5;   if (n &gt; 8) {     System.out.println("big");   } else {     System.out.println("small");   } else if (n &gt; 3) {     System.out.println("medium");   } }</pre> |           |                    |
| <pre>2 int count = 0;   if (count != 0) {     if (10 / count &gt; 2) {       System.out.println("high");     }   }   System.out.println("safe");</pre>                                |           |                    |
| <pre>3 int count = 0;   if (10 / count &gt; 2) {     if (count != 0) {       System.out.println("high");     }   }   System.out.println("safe");</pre>                                |           |                    |
| <pre>4 int n = 2;   if (n &gt; 8) {     System.out.println("big");   } else if (n &gt; 5) {     System.out.println("medium");   } }</pre>                                             |           |                    |

**10** WRAP-UP**REFLECTION**

In one sentence: how is an `else if` chain different from a row of separate `if` statements?

**YOUR ANSWER**