

NAME

DATE

PERIOD

UNIT 2 · TOPIC 2.2 · WORKSHEET

True or False?

Ask Java yes-or-no questions, catch the comparison that's lying, and settle who really compares Strings.

About 35 minutes

10 questions

Cumulative through 2.2

TOPIC

2.2

**DIRECTIONS**

Work each puzzle in pencil. For code, write exactly what prints, one line per `println`. You may use the Java Quick Reference. 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 coins = 8;
boolean rich = coins >= 10;
coins += 5;
System.out.println(rich + " " + (coins >= 10));
```

A false true**B** true true**C** false false**D** true false**THINK FIRST**

Which is greater: `-9` or `-4`? Guess in the margin before you look at question 2; it settles the question.

2 PENCIL CHECK

Here `int a = 9;` and `int b = 4;`. Write the value of each Boolean expression: `true` or `false`.

Expression	Value
<code>a < b</code>	
<code>a % b == 1</code>	
<code>a / b >= 3</code>	
<code>b * 2 <= a - 1</code>	
<code>-a < -b</code>	
<code>a != 9</code>	
<code>"cat".compareTo("dog") < 0</code>	



FAIR QUESTION

Can I use `==` on two booleans?

Yes. `boolean` is a primitive type, so `==` and `!=` compare the actual values. If `lightOn` holds `true` and `doorOpen` holds `false`, then `lightOn != doorOpen` is `true`. Only the size operators (`<`, `>`, `<=`, `>=`) are off-limits, because `true` isn't bigger or smaller than `false`.

PART B · 3 PUZZLES

B Puzzles

3 CODE TILES

Hopper's quiz app passes a score of *at least* 70, then takes 5 points off for turning it in late. Someone knocked the program off the fridge. Put the tiles back in order so it prints the output below. One tile is a decoy and isn't used.

<code>System.out.println(passed);</code>	<code>score -= 5;</code>
<code>boolean passed = score > 70;</code>	<code>int score = 70;</code>
<code>System.out.println(score);</code>	<code>boolean passed = score >= 70;</code>

Target output

65

true

YOUR PROGRAM



WATCH OUT

Math writes a range as $0 < t < 100$. Java can't: it works out $0 < t$ first, gets `true` or `false`, and then can't ask whether a `boolean` is less than 100, so it won't compile. For now, write each comparison on its own. Joining two comparisons into one test comes in Topic 2.5.

4 SNIPPET POOL

Fill each numbered hole with a snippet from the pool so the program prints `true`, `true`, `true`, `false`, one per line. Each snippet is used at most once.

RideCheck.java

```
int height = 48;
int seats = 14;
String name = new String("Hopper");
String typed = new String("Hopper");
System.out.println(height  48);
System.out.println(seats  4 == 2);
System.out.println(name);
System.out.println(seats / 4  3);
```

THE POOL

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

>	.equals(typed)	%
=>	>=	/
.compareTo(typed)		

5 FACE-OFF

Tonight: `==` and `equals` argue about who should compare Strings. Fill in the blanks.

== Give me two `int`s and I compare the actual _____. Quick, simple, done.

EQUALS Sure. But give you two `String` variables and you only compare their _____.

== So? I tell you whether both variables point to the very same _____.

EQUALS Two separate new `String("frog")` objects hold the same text, and you still say _____.

== Fine, you're the one who checks every _____. But you can't tell me which word comes first.

EQUALS That's `compareTo`'s job. It returns an `int`, and you compare that with _____.

PART C · 5 QUESTIONS

C

Catch the mistake

6

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.

```
double sum = 0.1 + 0.2;
boolean exact = sum == 0.3;
boolean close = Math.abs(sum - 0.3) < 0.0001;
int cents = 10 + 20;
boolean exactCents = cents == 30;
```

- 1 exact holds false.
- 2 close holds true.
- 3 exactCents holds false, for the same round-off reason as exact.

THE LIE, REWRITTEN

7

ONE-TOKEN FIX

A coupon works when the total is *at most* \$25.00. For a total of 25.0 this prints Coupon: false, but it should print Coupon: true. Change exactly one token on line 2 so the test is right for every total, not just this one.

```
1 double total = 25.0;
2 boolean coupon = total < 25.0;
3 System.out.println("Coupon: " + coupon);
```

OLD TOKEN, NEW TOKEN, AND WHY

8 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
1 <code>int guess = 4; boolean right = guess = 7;</code>		
2 <code>System.out.println(7 == 7.0);</code>		
3 <code>String a = "ant"; String b = "bee"; System.out.println(a < b);</code>		
4 <code>int n = 6; System.out.println("Even: " + n % 2 == 0);</code>		
5 <code>System.out.println(10 / 4 >= 2.5);</code>		

9 WHO AM I?

Some Java operators, methods, and types are playing a party game. Each gives a clue about itself. Name it.

Clue	I am...
I have exactly two values, and every comparison hands you one of them.	
I ask "same?" For objects, I only check whether two references match.	
I'm true exactly when two values are different.	
I'm true when the left number is bigger or the two are the same.	
I look inside two Strings and check every character.	
I return an <code>int</code> , and you compare me with <code>0</code> to learn alphabetical order.	

10 WRAP-UP**REFLECTION**

In one sentence: why is `==` a risky way to check whether a typed password matches a saved one?

YOUR ANSWER