

NAME

DATE

PERIOD

UNIT 3 · TOPIC 3.7 · WORKSHEET

One Copy for Everyone

Count the shared copy, lock the constants, and catch the class method that reaches for an object it doesn't have.

About 30 minutes

9 questions

Cumulative through 3.7

TOPIC

3.7



DIRECTIONS

Work in pencil. Lines marked *in main*, *in another class* run in a separate class's *main*, in order. Write output exactly as Java prints it. Some puzzles include pieces you won't need.



THINK FIRST

A game counts the coins *all* players have collected. If every `Player` object kept its own `coins` variable, could any one player know the total? Guess; questions 1 and 2 settle it.

PART A · 2 QUESTIONS

A

Warm up your brain

1

FACE-OFF

Tonight: an instance variable and a class variable argue. Fill in the blanks.

INSTANCE

Every object gets its _____ copy of me. Two lanterns, two different glows.

CLASS

There's only _____ copy of me, and every object _____ it.

INSTANCE

Big deal. What makes you so special?

CLASS

One word before my type: _____. I exist even before any _____ is made.

INSTANCE

Then how does code in another class find you?

CLASS

If I'm public, it writes my class _____, a dot, and my name. No object needed.

2 YOU'RE THE JVM

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

```
class Lantern {
    private int glow;
    private static int lit = 0;

    public Lantern(int start) {
        glow = start;
        lit++;
    }

    public void brighten() {
        glow += 2;
        lit++;
    }

    public int getGlow() { return glow; }
    public int getLit() { return lit; }
}

// in main, in another class
Lantern a = new Lantern(1);
Lantern b = new Lantern(5);
a.brighten();
System.out.println(a.getGlow() + " " + b.getLit());
```

A 3 3**B** 3 1**C** 3 2**D** 7 3**FAIR QUESTION**

Why is lit set to 0 where it's declared, not in the constructor?

A constructor runs once *per object*, so `lit = 0`; there would reset the count for every new lantern. A class variable is set up just once.

PART B · 2 PUZZLES

B Puzzles

3 CODE TILES

The snack kiosk has only class methods, so main never makes a Kiosk object. Arrange tiles in main so the program prints the target output. Two tiles are decoys that don't compile; don't use them.

```
class Kiosk {
    public static final int PRICE = 6;
    private static int sold = 0;

    public static void sell(int n) { sold += n; }
    public static int getSold() { return sold; }
    public static int revenue() { return getSold() * PRICE; }
}
```

System.out.println(Kiosk.revenue());	Kiosk.sell(3);
System.out.println(Kiosk.sold);	System.out.println(Kiosk.getSold());
System.out.println(getSold());	Kiosk.sell(4);

Target output

4

42

YOUR MAIN

4 SNIPPET POOL

needsWater is a class method, so it has no plant of its own to look at. Fill each hole with a snippet from the pool so the program prints true and then 2. Each snippet is used at most once, and some aren't needed.

```
class Plant {
    public static final int MIN = 2;
    private static int plants = 0;
    private int water;

    public Plant(int w) {
        water = w;
        plants++;
    }

    public static int count() { return plants; }
    public static boolean needsWater() {
        return  < MIN;
    }
}

// in main, in another class
Plant fern = new Plant(1);
Plant cactus = new Plant(5);
System.out.println(Plant.needsWater());
System.out.println();
```

THE POOL

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

int p	water	Plant.count()	cactus
Plant p	count()	p.water	fern

PART C · 5 QUESTIONS

C

Catch the mistake

5

YOU'RE THE COMPILER

You are the compiler. Each row is one method added to Lamp. Does it compile? If not, say why.

```
class Lamp {
    private boolean on;
    private static int lampsOn = 0;
    public static final int LIMIT = 3;
    public boolean getOn() { return on; }
    // one method from the table goes here
}
```

Method added to Lamp	Compiles?	If not, why not
1 public static int count() { return lampsOn; }		
2 public static void flip() { on = !on; }		
3 public static boolean check(Lamp x) { return x.on; }		
4 public static void raise() { LIMIT++; }		
5 public void switchOn() { on = true; lampsOn++; }		
6 public static boolean test() { return getOn(); }		

6 ONE-WORD FIX

Every locker should get the next number: 200, 201, 202. Instead the program prints 200, 200, 200. Add exactly one word to fix it.

```

1  class Locker {
2      private int nextNumber = 200;
3      private final int number;
4
5      public Locker() {
6          number = nextNumber;
7          nextNumber++;
8      }
9
10     public int getNumber() { return number; }
11 }
12 // in main, in another class
13 Locker a = new Locker();
14 Locker b = new Locker();
15 Locker c = new Locker();
16 System.out.println(a.getNumber());
17 System.out.println(b.getNumber());
18 System.out.println(c.getNumber());

```

LINE, THE WORD YOU ADD, AND WHY IT WORKS

7 WHO AM I?

Each class idea gives a clue about itself. Name the idea.

Clue	I am...
Every object of my class gets its own copy of me.	
There's one of me for the whole class, shared by every object.	
Once I'm on a variable and it has a value, no one can change it.	
I'm public, static, and final, usually named in capitals.	
Call me by my class name. I see an object's data only if you pass it in.	
I make a variable or method belong to the class, not to one object.	

8 TWO TRUTHS AND A LIE

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

```
class Gym {  
    public static final int CAPACITY = 40;  
    public static int inside = 0;  
    public static void enter(int n) { inside += n; }  
    public static int spaceLeft() { return CAPACITY - inside; }  
}  
// in main, in another class  
Gym.enter(12);  
Gym.enter(9);  
System.out.println(Gym.spaceLeft());  
System.out.println(Gym.inside);
```

- **1** It prints 19 and then 21.
- **2** No Gym object is ever made, but `Gym.inside` still changes.
- **3** `spaceLeft` won't compile: a class variable always needs its class name, as in `Gym.CAPACITY`.

THE LIE, REWRITTEN

9 WRAP-UP**REFLECTION**

In one or two sentences: when you add a new variable to a class, what question tells you whether it should be static?

YOUR ANSWER