

NAME

DATE

PERIOD

UNIT 1 · TOPIC 1.1 · WORKSHEET

Plans, Programs, and Bugs

Tighten a vague recipe, snap a program together, and figure out which kind of mistake you just caught.

About 30 minutes

10 questions

Cumulative through 1.1

TOPIC

1.1



DIRECTIONS

Work each puzzle in pencil. For Java code, write exactly what appears on the screen, one line per `System.out.println`. In Java, `/` means divide, as on a calculator. Some puzzles include pieces you won't need.

PART A · 3 QUESTIONS

A Warm up your brain

1 PENCIL CHECK

A literal follower does exactly what is written and fills in nothing. Could a literal follower carry out each step without guessing? Write *precise* or *vague*.

Step	Precise or vague?
Add some sugar to the lemonade.	
Put 2 slices of bread in the toaster.	
Cook the pasta until it looks right.	
Walk to the corner of Oak Street, then turn left.	
Stir a bit.	

2 SAME ALGORITHM?

Hopper drew the plant-watering algorithm as a flowchart. Which written list is the **same** algorithm?

```

(Start) --> [Fill the watering can] --> [Carry it to the garden]
                                         |
(Stop) <-- [Put the can away] <-- [Water each plant]
  
```

- A** 1. Carry the can to the garden. 2. Fill the watering can. 3. Water each plant. 4. Put the can away.
- B** 1. Fill the watering can. 2. Carry it to the garden. 3. Put the can away.
- C** 1. Fill the watering can. 2. Carry it to the garden. 3. Water each plant. 4. Put the can away.
- D** 1. Fill the watering can. 2. Carry it to the garden. 3. Put the can away. 4. Water each plant.

Choice A uses exactly the right four steps. Why is it still a different algorithm?

YOUR ANSWER

3 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("Go team!")</code>		
2 <code>System.out.println("Go team!!");</code>		
3 <code>system.out.println("Go team!");</code>		
4 <code>System.out.println("10 / 0");</code>		
5 <code>System.out.println("Go team!");</code>		
6 <code>System.out.println(10 / 0);</code>		



FAIR QUESTION

If dividing by zero is obviously wrong, why doesn't the compiler catch it?

The compiler checks that code follows Java's rules, and `10 / 0` is written correctly: a number, a slash, a number. The problem only shows up when the computer actually tries the division, and that happens while the program runs, after the compiler is finished.

PART B · 5 PUZZLES

B

Puzzles

4

CODE TILES

Hopper's hot cocoa program got knocked off the fridge. Put the tiles back in order so the program displays the output below. Two of the tiles have a syntax error. Leave those out.

<code>System.out.println("Stir in cocoa.");</code>	<code>System.out.println("Fill a mug.");</code>
<code>System.out.println("Heat the milk.");</code>	<code>System.out.Println("Sip slowly.");</code>
<code>System.out.println("Sip slowly.");</code>	<code>System.out.println("Stir in cocoa.");</code>

Target output

Heat the milk.

Stir in cocoa.

Fill a mug.

Sip slowly.

YOUR PROGRAM, AND WHAT IS WRONG WITH EACH TILE YOU LEFT OUT



THINK FIRST

A program has five lines that display text. The very last line is missing its semicolon. Does the first line still appear on the screen? Guess in the margin; question 6 settles it.

5 YOU'RE THE JVM

You are the computer running this program. It compiles. What appears on the screen?

```
1 System.out.println("Pizza party!");
2 System.out.println("8 slices, 0 guests.");
3 System.out.println(8 / 0);
4 System.out.println("More pizza for me.");
```

- A** Nothing. The compiler reports an error on line 3.
- B** Pizza party! and 8 slices, 0 guests., then an ArithmeticException message.
- C** All three text lines, then an ArithmeticException message at the end.
- D** Pizza party!, 8 slices, 0 guests., 0, and More pizza for me.

6 MATCH-UP

Four versions of the bus program each change one thing. Write the letter of what appears on the screen and the kind of error: *syntax*, *logic*, *run-time*, or *none*. Each letter is used once.

Version	Code	Screen	Kind of error
1	System.out.println("Line up."); System.out.println("Board the bus."); System.out.println("Find a seat.");		
2	System.out.println("Line up."); System.out.println("Board the bus."); System.out.println("Find a seat.")		
3	System.out.println("Line up."); System.out.println("Find a seat."); System.out.println("Board the bus.");		
4	System.out.println(3 / 0); System.out.println("Board the bus."); System.out.println("Find a seat.");		

- **P** Nothing runs. The compiler reports ';' expected.
- **Q** Only an ArithmeticException message
- **R** Line up. Find a seat. Board the bus.
- **S** Line up. Board the bus. Find a seat.



WATCH OUT

"The compiler said nothing, so my program works." A quiet compiler only means no rule was broken. A program that prints "Brush your teeth" before "Eat breakfast" follows every rule. Decide what *should* happen first, then run the program and compare.

7 FACE-OFF

Tonight: the Compiler and the Tester argue about who catches more mistakes. Fill in the blanks.

COMPILER	I read the whole program _____ it runs. If one rule is broken, _____ runs at all.
TESTER	You only check Java's _____. A program can follow every one and still show its steps in the wrong order.
COMPILER	That's a _____ error. Not my department.
TESTER	Right. I run the program with specific data and compare what it does to the _____ outcome.
COMPILER	And when someone's 10 / 0 slips past me?
TESTER	That's a _____ error. The program had already started, so the lines above the division _____ appear.

8 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'm one application where you type, compile, and run your code.	
I check the whole program for some errors and translate it before it can run.	
I break Java's rules, like a missing semicolon. The compiler catches me.	
I compile and run all the way to the end, but my result is wrong. Only testing finds me.	
I wasn't caught by the compiler. I interrupt a running program when it tries something impossible.	
I'm the order a process's steps are completed in, one at a time.	

PART C · 2 QUESTIONS

C

Catch the mistake

9

THE TEST THAT LIED

An algorithm is supposed to **triple** a number. Riley wrote: "1. Write down the number. 2. Add the number to itself." Riley tested it with the number 0, got 0, and decided it works.

Give a better test: the number you'd use, the expected outcome, and the actual result. Then explain why the test with 0 hid the mistake, and name the kind of error.

Test number	Expected outcome	Actual result

WHY 0 HID THE MISTAKE, AND THE KIND OF ERROR

10

TWO TRUTHS AND A LIE

This program is supposed to display the steps for planting a seed. Two claims about it are true and one is a lie. Circle the lie, then rewrite it so it's true.

```

1 System.out.println("Dig a small hole.");
2 System.out.println("Cover the seed with soil.");
3 System.out.println("Drop in the seed.");
4 System.out.println("Water it.");

```

- 1 It compiles with no errors.
- 2 When it runs, all four lines appear, in the order they are written.
- 3 The compiler found no errors, so the program does what its author wanted.

THE LIE, REWRITTEN