

NAME _____

DATE _____

PERIOD _____

UNIT 1 · TOPIC 1.5 · WORKSHEET

Cast, Chop, and Wrap

Chop decimals on purpose, round the right way, and catch an int that wrapped around the edge.

About 30 minutes

10 questions

Cumulative through 1.5

TOPIC

1.5

**DIRECTIONS**

Work each puzzle in pencil. For code, write exactly what prints, one line per `println`. A `double` that holds a whole number prints with `.0`, like `3.0`. 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?

```
double temperature = 98.7;
int whole = (int) temperature;
System.out.println(whole);
```

A 98**B** 99**C** 98.7**D** 98.0**THINK FIRST**

Does `(double) (7 / 2)` give you 3.5? Guess in the margin; question 2 settles it.

2 PENCIL CHECK

Write the value of each expression exactly as `println` would show it. Watch which value each cast applies to.

Expression	Value
<code>(int) 6.99</code>	
<code>(int) -6.99</code>	
<code>4 + 1.5</code>	
<code>(double) 7 / 2</code>	
<code>(double) (7 / 2)</code>	
<code>(int) (6.5 + 0.5)</code>	



WATCH OUT

A cast makes a new value; it doesn't change the variable. After `int dollars = (int) price;`, the variable `price` still holds 9.99. Only an assignment *to* `price` could change it.



FAIR QUESTION

Why does an `int` even have a biggest value?

Because every `int` is stored in the same fixed amount of memory: 4 bytes, which is 32 bits (0s and 1s). There are only so many different patterns of 32 bits, so there are only so many different `int` values. About half the patterns are used for negative numbers and the rest for zero and the positives, so the range has a bottom and a top.

PART B · 3 PUZZLES

B Puzzles

3 CODE TILES

Hopper hopped a total distance over 4 days. The program reads the total in kilometers, rounds it to the nearest whole kilometer, then prints the average per day. The user types 17.8. Put the tiles in order under the lines shown so the program prints the output below. Two tiles are decoys and aren't used.

```
import java.util.Scanner;
...
Scanner console = new Scanner(System.in);
```

System.out.println(perDay);	int total = (int) (km + 0.5);
double km = console.nextDouble();	double perDay = (double) (total / 4);
double perDay = (double) total / 4;	int total = (int) km + 0.5;

Target output

4.5

YOUR PROGRAM

4 MATCH-UP

Four lines try to show the average of 15 points over 4 games. Each changes one thing. Match each line to what it prints, from the list below. Each output is used once.

```
int sum = 15;
int n = 4;
```

Line	Code	Prints
1	<code>System.out.println(sum / n);</code>	
2	<code>System.out.println((double) sum / n);</code>	
3	<code>System.out.println((double) (sum / n));</code>	
4	<code>System.out.println((int) ((double) sum / n + 0.5));</code>	

- **P** 3.0
- **Q** 4
- **R** 3
- **S** 3.75

5 FACE-OFF

Tonight: `int` and `double` argue about which one is more trustworthy. Fill in the blanks.

INT	Every whole number I hold is stored _____. Add me, subtract me, multiply me: no tiny errors.
DOUBLE	Until you go past _____. Add 1 to that, and you _____ around to the most _____ int, with no error message.
INT	At least I'm not fuzzy. You print <code>0.1 + 0.2</code> as <code>0.30000000000000004</code> .
DOUBLE	I keep only a limited number of digits, so a tiny _____ error sneaks in. That's the price of holding decimals.
INT	That's why programs that handle money count in whole _____ with me.
DOUBLE	And when someone squeezes me into you with <code>(int)</code> , you don't round me. You _____, and my decimal part is gone.

PART C · 5 QUESTIONS

C

Catch the mistake

6

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>double price = 12; System.out.println(price);</code>		
2 <code>int score = 89.9;</code>		
3 <code>int score = (int) 89.9; System.out.println(score);</code>		
4 <code>int big = 3000000000;</code>		
5 <code>double half = 7 / 2; System.out.println(half);</code>		

7

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.

```

1  int top = Integer.MAX_VALUE;
2  int past = top + 1;
3  int bottom = Integer.MIN_VALUE;
4  System.out.println(past);
5  System.out.println(bottom);

```

- 1** Lines 4 and 5 print the same number.
- 2** The program crashes on line 2, because `top + 1` is too big for an `int`.
- 3** `past` holds a negative number, even though `top` is positive.

THE LIE, REWRITTEN

8 ONE-TOKEN FIX

A stock's price changed by -4.7 dollars. This code should print the change rounded to the nearest whole dollar, -5, but it prints -4. Change exactly one token on line 2 to fix it.

```
1 double change = -4.7;
2 int nearest = (int) (change + 0.5);
3 System.out.println(nearest);
```

OLD TOKEN, NEW TOKEN, AND WHY

9 WHO AM I?

Today's ideas are playing a party game. Each gives a clue about itself. Name it.

Clue	I am...
I turn an <code>int</code> into a <code>double</code> automatically, and the amount doesn't change.	
I drop every digit after the decimal point, which moves a value toward zero.	
I'm the largest value an <code>int</code> can hold.	
I happen when an <code>int</code> calculation goes past its limits, and nobody gets an error message.	
I'm the tiny error a <code>double</code> gets when its result has more digits than it can keep.	

10 WRAP-UP**REFLECTION**

In one sentence: a store's program adds up thousands of prices. Why should it store each price as an `int` number of cents instead of a `double` number of dollars?

YOUR ANSWER