

NAME

DATE

PERIOD

UNIT 2 · TOPIC 2.9 · WORKSHEET

The Loop Toolbox

Sum it, count it, average it, split it into digits, and find the best of the bunch.

About 35 minutes

10 questions

Cumulative through 2.9

TOPIC

2.9



DIRECTIONS

Work each puzzle in pencil. When a question asks for output, write exactly what prints, including spaces. Some puzzles include pieces you won't need.

PART A · 2 QUESTIONS

A

Warm up your brain

1

PENCIL CHECK

The `int` variable `n` holds 5308. Write the value of each expression.

Expression	Value
<code>n % 10</code>	
<code>n / 10</code>	
<code>n / 10 % 10</code>	
<code>n / 1000</code>	
<code>n % 4 == 0</code>	
<code>n % 3 == 0</code>	
<code>24 % 36</code>	

2 YOU'RE THE JVM

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

```
1 int sum = 0;
2 int count = 0;
3 for (int k = 1; k <= 4; k++) {
4     sum += k * 3;
5     count++;
6 }
7 System.out.println(sum / count + " " + (double) sum / count);
```

A 7 7.5

B 7.5 7.5

C 7 7.0

D 7.5 7.0



FAIR QUESTION

What if there's nothing to average?

Then the count is 0, and an `int` divided by 0 crashes with an `ArithmeticException`. Check before you divide: `if (count > 0)`, and print something like `No data` otherwise.



THINK FIRST

To find the warmest day, a programmer starts `warmest` at 0, because sums start at 0. Can you think of a week where that gives the wrong answer? Guess in the margin; question 5 settles it.



WATCH OUT

Start an accumulator **once**, before the loop. This one is reset on every pass, so the last thing it prints is 5, not the running total 15:

```
for (int k = 1; k <= 5; k++) {
    int total = 0;
    total += k;
    System.out.println(total);
}
```

Fix: declare `total` before the loop, and print it once, after the loop.

PART B · 6 PUZZLES

B Puzzles

3 CODE TILES

Hopper wants the sum of the digits of 4096. Put the tiles in order, below the lines already written, so the program prints the target output. One tile is a decoy.

```
int n = 4096;
int sum = 0;
```

n /= 10;	System.out.println("Sum: " + sum);
while (n > 0) {	sum += n / 10;
}	sum += n % 10;

Target output

Sum: 19

YOUR PROGRAM

4 SNIPPET POOL

This program reads five temperatures and prints the lowest; for 12 9 7 15 4 it prints Low: 4. **Two** different fills work for every input. Find both.

```
import java.util.Scanner;
Scanner console = new Scanner(System.in);
int lowest = (1);
for (int day = (2); day <= 5; day++) {
    int temp = console.nextInt();
    if ((3)) {
        lowest = temp;
    }
}
System.out.println("Low: " + lowest);
```

THE POOL

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

0	console.nextInt()	Integer.MAX_VALUE
Integer.MIN_VALUE	1	2
temp < lowest	temp > lowest	

YOUR SECOND FILL: (1), (2), (3)

5 TWO TRUTHS AND A LIE

This code looks for the warmest of four winter temperatures. Two claims are true and one is a lie. Circle the lie, then rewrite it so it's true.

```
1  int warmest = 0;
2  for (int day = 1; day <= 4; day++) {
3      int temp = day * 3 - 15;
4      if (temp > warmest) {
5          warmest = temp;
6      }
7  }
8  System.out.println(warmest);
```

- **1** The temperatures are -12, -9, -6, and -3.
- **2** It prints -3, the warmest of the four.
- **3** If every temperature were above 0, this code would print the right answer.

THE LIE, REWRITTEN

6 FACE-OFF

Tonight: the division operator `/` and the remainder operator `%` argue about who matters more. Fill in the blanks.

<code>/</code>	Give me <code>4072 / 10</code> and I hand back _____. I keep the whole part and toss the rest.
<code>%</code>	And <code>4072 % 10</code> gives the _____, which is _____. the last digit. You toss the good part.
<code>/</code>	When someone asks "does 7 divide 84 evenly?" they should ask you, I admit: <code>84 % 7</code> should be _____.
<code>%</code>	Thank you. And put 84 on the _____. <code>7 % 84</code> is just 7.
<code>/</code>	But to visit every digit you still need me. You take the last digit, I _____ it, and we repeat until the number is _____.
<code>%</code>	Fine. We're a team. Right to left, every time.



FAIR QUESTION

Do these tricks work with negative numbers?

Java will compute % with negatives, but the sign rules are outside this course. Every digit and divisibility algorithm here assumes the number is 0 or positive and that you divide by a positive number, which is the usual case for counts, years, and IDs.

7

MATCH-UP

Each version of this program uses a different condition. Match each condition to the number the program prints, from the list below.

```
int count = 0;
for (int n = 1; n <= 20; n++) {
    if (CONDITION) {
        count++;
    }
}
System.out.println(count);
```

Version	CONDITION	Prints
1	<code>n % 3 == 0</code>	
2	<code>n % 3 != 0</code>	
3	<code>n % 3 == 0 && n % 2 == 0</code>	
4	<code>n % 3 == 0 n % 2 == 0</code>	

- P 13
- Q 3
- R 14
- S 6

8 YOU'RE THE JVM

Two algorithms share one loop. Fill in the trace table for each pass, then write the exact output.

```

1  int evens = 0;
2  int biggest = Integer.MIN_VALUE;
3  for (int k = 1; k <= 5; k++) {
4      int v = k * 4 % 7;
5      if (v % 2 == 0) {
6          evens++;
7      }
8      if (v > biggest) {
9          biggest = v;
10     }
11 }
12 System.out.println(evens + " " + biggest);

```

k	v	evens	biggest
1			
2			
3			
4			
5			

Output: _____

PART C · 2 QUESTIONS

C Catch the mistake

9 ONE-TOKEN FIX

This code should count the zero digits in 50070 and print 3. It prints 1. Change exactly one token to fix it.

```
1 int n = 50070;
2 int zeros = 0;
3 while (n > 0) {
4     if (n % 10 == 0) {
5         zeros++;
6     }
7     n /= 10;
8 }
9 System.out.println(zeros);
```

WHY IT PRINTS 1, AND YOUR FIX (LINE, OLD TOKEN, NEW TOKEN)

10 WHO AM I?

The tools in the loop toolbox are playing a party game. Each gives a clue about itself. Name the tool.

Clue	I am...
I start at 0 before the loop and add a value on every pass.	
I start at 0 and go up by 1 only when an if statement says yes.	
I'm a special input that means "stop reading." I never get added.	
Give me any positive int and I'll give you its last digit.	
Every int is less than or equal to me, so I'm a safe start for a minimum.	
Put me in front of sum in <code>sum / count</code> , and the average keeps its decimals.	