

NAME

DATE

PERIOD

UNIT 4 · TOPIC 4.6 · WORKSHEET

Pond Log Files

Open a file, read it token by token or line by line, split the records, and catch the reader that runs off the end.

About 30 minutes

10 questions

Cumulative through 4.6

TOPIC

4.6



DIRECTIONS

Work in pencil and write exactly what prints. Unless a question says otherwise, every program imports File, IOException and Scanner, and main has throws IOException.

PART A · 3 QUESTIONS

A

Warm up your brain

1

PENCIL CHECK

Scanner input = new Scanner(new File("pond.txt")); has just run. Each row makes the next call on that same scanner. Write what each call returns.

pond.txt

```
frog 3 2.5
newt 7 true
```

Call	Returns	Call	Returns
1 input.next()		5 input.next()	
2 input.nextInt()		6 input.nextInt()	
3 input.nextDouble()		7 input.nextBoolean()	
4 input.hasNext()		8 input.hasNext()	

2 YOU'RE THE JVM

Run this code on `scores.txt` in your head. What prints?

`scores.txt`

```
Kai 88 true
Lee 92 false
```

```
Scanner input = new Scanner(new File("scores.txt"));
String name = input.next();
int a = input.nextInt();
boolean b = input.nextBoolean();
String name2 = input.next();
int c = input.nextInt();
input.close();
System.out.println(name2 + " " + (a + c));
```

- A** Lee 180

C true 180

B Lee 8892

D An InputMismatchException

3 YOU'RE THE COMPILER

For these snippets only, the imports and headers are as described. Rows 1 to 3 run just `Scanner in = new Scanner(new File("notes.txt"));`. Does it compile? If so, what happens when it runs? If not, why?

Snippet	Compiles?	What happens, or why not
1 All three imports; main has no throws clause.		
2 Imports only <code>Scanner</code> and <code>IOException</code> ; main has throws <code>IOException</code> .		
3 Imports and header correct, but there is no file named <code>notes.txt</code> .		
4 Imports and header correct. <code>Scanner in = new Scanner("notes.txt");</code> <code>System.out.println(in.next());</code>		
5 Imports and header correct. <code>String[] p = "7,8".split(",");</code> <code>int n = p[0];</code>		

PART B · 3 PUZZLES

B Puzzles**4** CODE TILES

Hopper logs laps in `laps.txt`. The program starts with `int total = 0;` and ends with `System.out.println(total);`. In between, build code from five tiles that adds up every number in the file, however many there are, and closes the file. It should print 22. Two tiles are decoys.

`laps.txt``4 7``5 6``input.close();``for (int i = 0; i < 5; i++) {``total += input.nextInt();``Scanner input = new Scanner("laps.txt");``}``while (input.hasNext()) {``Scanner input = new Scanner(new File("laps.txt"));`**YOUR PROGRAM**

5 SNIPPET POOL

Each line of `pets.txt` holds a name, a kind of animal, and an age, separated by commas. Fill each hole from the pool so the program prints `Lady Mochi the cat`, `Pip the frog`, `Big Rex the dog`.

`pets.txt`

```
Lady Mochi,cat,4
Pip,frog,1
Big Rex,dog,9
```

```
Scanner input = new Scanner(new File("pets.txt"));
while (input.hasNext()) {
    String line = input. (1) ;
    String[] parts = line.split( (2) );
    System.out.println(parts[ (3) ] + " the " + parts[1]);
}
input.close();
```

THE POOL

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

<code>next()</code>	<code>", "</code>	<code>1</code>	<code>nextLine()</code>
<code>" "</code>	<code>0</code>	<code>", "</code>	<code>2</code>



FAIR QUESTION

If a line says 4, why can't I add 1 to the piece?

Because `split` always hands back `Strings`: the piece is the text `"4"`, not the number 4, and `"4" + 1` is `"41"`. For now, keep numbers you need for math in files you read with `nextInt` or `nextDouble`. (Topic 4.7 shows how to turn text into a number.)

6 MATCH-UP

String s = "red, green,,blue"; Match each expression to its value from the list below. Each value is used once.

Expression	Value
1 s.split(",").length	
2 s.split(", ").length	
3 s.split(";").length	
4 s.split(",")[2]	

P 1 Q 4 R 2 S "" (the empty string)

PART C · 4 QUESTIONS

C Catch the mistake

7 ONE-TOKEN FIX

It should print Ada 31.5 and Bo 29.0, but it crashes first. Change exactly one token to fix it.

swim.txt

Ada 31.5
Bo 29

```

1 Scanner input = new Scanner(new File("swim.txt"));
2 while (input.hasNext()) {
3     String name = input.next();
4     double time = input.nextInt();
5     System.out.println(name + " " + time);
6 }
7 input.close();

```

LINE, OLD TOKEN, NEW TOKEN, AND THE EXCEPTION IT THREW

**WATCH OUT**

A variable doesn't remember anything from the last time the program ran. Every run starts fresh from the code, so a high score stored only in `int best` is gone when the program ends. Data that must last between runs has to live in a **file**, which persists when no program is running.

8 TWO TRUTHS AND A LIE

`temps.txt` holds 61 64 58. Two claims about this code are true and one is a lie. Circle the lie and rewrite it.

```
Scanner input = new Scanner(new File("temps.txt"));
int count = 0;
while (input.hasNext()) {
    int t = input.nextInt();
    if (t > 60) {
        count++;
    }
}
input.close();
System.out.println(count);
```

- **1** It prints 2.
- **2** If `temps.txt` held 300 numbers, this code would read all 300 without any change.
- **3** Moving `input.close();` inside the loop, just after the `if`, would still print 2.

THE LIE, REWRITTEN

9 FACE-OFF

Tonight: `next()` and `nextLine()` argue about who reads a file better. Fill in the blanks.

NEXT() I skip any _____, then hand you exactly one token. Quick and tidy.

NEXTLINE() I hand you everything up to the end of the _____, spaces and all.

NEXT() Give me the line `Grace Hopper, 1906` and I'll return _____.

NEXTLINE() And you'd leave `Hopper, 1906` behind. I return the whole thing, and `split` can cut it at the _____.

NEXT() Fine. But call you right after `nextInt()` and you can return an _____ string.

NEXTLINE() That's why a smart program uses _____ style on a file, not both.

10 WRAP-UP



REFLECTION

In one sentence: why read a file with `while (input.hasNext())` instead of a `for` loop that runs a set number of times?

YOUR ANSWER