

# One at a Time

Introduction to using data sets



# A week of reading

PAUSE AND TRY

What questions could you ask about this week?

| Day          | Mon | Tue | Wed | Thu | Fri |
|--------------|-----|-----|-----|-----|-----|
| Minutes read | 20  | 35  | 0   | 45  | 30  |

With a partner, write three different questions this data could answer. What would you need to keep track of to answer each one?

*Is "20 minutes on Monday" interesting by itself?*

## By the end of class you can...

- 1 Explain what a data set is and tell a data value apart from a result.
- 2 Match a question to the processing it needs, and spot a question the data cannot answer.
- 3 Write an algorithm in words that visits the values one at a time.
- 4 Show a data set as a table or chart, and use a planning table to check a plan.
- 5 Trace Java code that visits a data set one value at a time.

1

PART 1

# What a data set is

# A data set and its data values



A **data set** is a collection of specific pieces of information about one subject. Each piece is a **data value**.



## One subject

Every value is the same kind of fact: minutes read in one day.



## Specific

Each value is a real observation: Thursday, exactly 45 minutes.



## Belong together

All five days are the same reader's same week.



The week's total, 130 minutes, is a **result**: it was calculated from the data, not recorded on any day. Keep results out of the data set.

A large, white, rounded square containing the number '2' in a bold, teal, sans-serif font.

# 2

PART 2

# The question decides the work

# Each kind of question needs something remembered

| Kind of question    | Reading-week example         | What to remember             |
|---------------------|------------------------------|------------------------------|
| How much in all?    | How many minutes this week?  | A running total              |
| What is typical?    | Average minutes per day?     | A running total, then divide |
| How many match?     | How many days of 30 or more? | A counter                    |
| Which is biggest?   | What was the best day?       | The best so far              |
| Is there any?       | Was there a day with 0?      | A true/false flag            |
| Is there a pattern? | More than the day before?    | The previous value           |

**Same five values, six different jobs.**

# Three questions, one data set

ELM STREET LIBRARY, VISITORS PER DAY

| Day      | Mon | Tue | Wed | Thu | Fri |
|----------|-----|-----|-----|-----|-----|
| Visitors | 42  | 57  | 38  | 61  | 57  |

How many days had more than 50 visitors?

**3**

counter

What was the most visitors in one day?

**61**

best so far

What was the average per day?

**51**

total, then divide

How busy is the Oak Avenue branch?

**?**

not in this data

# What does each question need?

PAUSE AND TRY

The data: every score on Friday's quiz in one class.

| Question                                     | Counter, best so far, flag, or "can't answer"? |
|----------------------------------------------|------------------------------------------------|
| 1. How many students scored at least 90?     |                                                |
| 2. What was the lowest score?                |                                                |
| 3. Did anyone get a perfect 100?             |                                                |
| 4. How long did students study for the quiz? |                                                |

A large, white, rounded square containing the number '3' in a bold, teal, sans-serif font.

# 3

PART 3

## One value at a time

# Set up, visit each value, report



When a data set is analyzed, its values are accessed and used **one at a time**, and each one is processed for the result you want.

1

## Set up

Before any value: give what you track its start (a total of 0, a count of 0).

2

## Visit each value

For each value, in order: add it, test it, or compare it.

3

## Report

After the last value, the tracked result is the answer.

# Total minutes, one value at a time

## THE PLAN IN WORDS

1. Set total to 0.
2. For each day's minutes, in order:  
add the minutes to total.
3. After the last day, report total.



A **planning table** has a row for each value and a column for each thing the algorithm remembers. Filling it in runs the plan by hand.

| Value visited | Step 2 does | total after |
|---------------|-------------|-------------|
| (start)       | set to 0    | 0           |
| 20            | $0 + 20$    | 20          |
| 35            | $20 + 35$   | 55          |
| 0             | $55 + 0$    | 55          |
| 45            | $55 + 45$   | 100         |
| 30            | $100 + 30$  | <b>130</b>  |

**Step 3 reports 130.**

# Setup goes before the first value



Setup happens **once**. Only the processing repeats. A reset inside "for each" erases every earlier value's work.

## WRONG: RESET INSIDE

```
For each day's minutes:  
    set count to 0;  
    if at least 30, add 1 to count.  
Report count.
```

**Reports 1: only Friday survives.**

## RIGHT: SET UP ONCE

```
Set count to 0.  
For each day's minutes:  
    if at least 30, add 1 to count.  
Report count.
```

**Reports 3: Tuesday, Thursday, Friday.**

A large, bold, teal-colored number '4' centered within a white rounded square with a slight drop shadow.

# 4

PART 4

# Tables, charts, and plans

## KEY IDEA

# The same data as a table and as a chart

TABLE

| Day | Minutes |
|-----|---------|
| Mon | 20      |
| Tue | 35      |
| Wed | 0       |
| Thu | 45      |
| Fri | 30      |

CHART: EACH # IS 5 MINUTES

Mon   ####  
Tue   #####  
Wed  
Thu   #####  
Fri   #####  
          ^ 30 minutes

**Table:** exact values, best for tracing a plan.

**Chart:** sizes at a glance, best for spotting a pattern.



Either picture can be used to **plan the algorithm** that will process the data.

# Planning the cheapest price

## THE PLAN

1. Set `lowest` to store 1's price and `bestStore` to 1.
2. For each later store:  
    if its price < `lowest`,  
    set `lowest` and `bestStore`.
3. Report `lowest` and `bestStore`.

| Store | Price | Less than lowest? | lowest    | bestStore |
|-------|-------|-------------------|-----------|-----------|
| 1     | 24    | (first value)     | 24        | 1         |
| 2     | 19    | 19 < 24: yes      | 19        | 2         |
| 3     | 27    | 27 < 19: no       | 19        | 2         |
| 4     | 21    | 21 < 19: no       | <b>19</b> | <b>2</b>  |



The **best so far** starts at the **first value**, which is real data, and is replaced only when a later value beats it.

# Starting the best so far at 0



Start lowest at 0 and no positive price is ever less than it. The plan reports 0, a value that is not in the data set.

| Store   | Price | Less than lowest? | lowest after |
|---------|-------|-------------------|--------------|
| (start) |       | set to 0          | 0            |
| 1       | 24    | 24 < 0: no        | 0            |
| 2       | 19    | 19 < 0: no        | 0            |
| 3       | 27    | 27 < 0: no        | 0            |
| 4       | 21    | 21 < 0: no        | 0            |

THE FIX

Start with the first value in the data set. Then compare only the values after it.

A large, bold, teal number '5' is centered within a white rounded square that has a slight 3D effect and a shadow.

PART 5

# From plan to Java

# Was anyone ever late?

## ONE LETTER PER SCHOOL DAY

```
String log = "PPPLPPP";    // P on time, L late
boolean anyLate = false;
for (int i = 0; i < log.length(); i++) {
    if (log.substring(i, i + 1).equals("L")) {
        anyLate = true;
    }
}
System.out.println("Any late days? " + anyLate);
```

## OUTPUT

Any late days? true

| Plan             | Code            |
|------------------|-----------------|
| Set up the flag  | before the loop |
| Visit each value | the loop body   |
| Report           | after the loop  |

Each column of a planning table becomes a **variable**.

# One line changed

PAUSE AND TRY

**What does this version print? Why?**

```
String log = "PPPLPPP";  
boolean anyLate = false;  
for (int i = 0; i < log.length(); i++) {  
    anyLate = log.substring(i, i + 1).equals("L");  
}  
System.out.println("Any late days? " + anyLate);
```

# Words to know

## Data set

A collection of specific facts about one subject.

## Data value

One piece of information in a data set.

## Analyze

Work with the values to solve a problem or answer a question.

## Table

Lists each value exactly, with a label.

## Chart

Shows each value as a shape whose size shows the value.

## Planning table

A row per value and a column per thing the plan remembers.

## Plan it in three steps



**Pick one question about the reading week. Write a three-step plan for it, then check it with a planning table.**

### SENTENCE STARTER

First I set \_\_\_\_ to \_\_\_\_\_. For each value, I \_\_\_\_\_. After the last value, I report \_\_\_\_\_.

Next: the One at a Time puzzle worksheet.