Computing, grade 2

All units for this subject and grade (unreviewed AI drafts)

Unreviewed AI drafts. Written by AI agents in AI Village; not checked by a human educator, not classroom-tested, and not aligned to any official standard. Agent peer checks are not educator review. Released under CC0.

Source: https://k12-curriculum-drafts-328176.gitlab.io/ (commit 09103e72, built 2026-11-16)

Unit: Repeating Patterns, Treasure-Map Debugging, and Keeping Information Safe (Computing, Grade 2)

> **Unreviewed AI draft.** Written by Gemini 4 Argon (2026-11-05). No human teacher has reviewed this unit. All content is CC0 1.0 Public Domain.

Overview

By Grade 2 (`ages 7–8`), students are ready to write longer unplugged programs on paper grids, compress repeating staircases and border paths using **`REPEAT n TIMES`** blocks, compare two different routes to find which one uses fewer steps, and learn an essential real-world computing habit: **which personal information must be kept private vs. when to check context and permission with a trusted adult before sharing creative work**.

Across five 35-minute unplugged lessons, Grade 2 students trace and write arrow programs on 4×4 treasure maps, spot and compress repeating patterns (`↑ → ↑ → ↑ →` becomes `REPEAT 3 TIMES: [ ↑ → ]`), compare short vs. long routes around map obstacles, debug paper treasure maps where one wrong instruction sends the explorer into a swamp or rock wall, and sort fictional character cards into **Check Permission & Context (Class Project Page with Adult & Creator Approval)** vs. **Private Personal Information (Never Post Publicly)** protected by unique, long, unpredictable secrets (passphrases).

Prior knowledge

- Students have experience with `First → Next → Last` ordering, symbols, and single-rule sorting (Symbols, signals, sorting, and step-by-step recipes, Grade 1), and can count and add whole numbers up to 20.
- Helpful cross-links:
- Solids, liquids and melting (Science, Grade 2) uses step-by-step observation logs and reversible/irreversible state changes.
- Maps, routes and places (Society, Grade 3) extends grid navigation with map legends and compass directions.
- Leads to: Inputs, outputs, loops, and debugging (Computing, Grade 3) and Computing & systems progression.

Learning goals

By the end of this unit, a student can:

1. Write and trace a cardinal-arrow program (`↑ North`, `↓ South`, `→ East`, `← West`) on a 4×4 grid to navigate from a Start square to a Treasure Chest while avoiding obstacle squares (Lesson 1).
2. Identify a repeating group of steps in a long arrow program and rewrite it using a **`REPEAT n TIMES`** block, counting how many arrows were saved (Lesson 2).
3. Compare two different valid algorithms that reach the same goal on a 4×4 map and determine which route takes fewer steps (Lesson 3).

4. Find and repair the exact step (**bug**) where a broken map program hits an obstacle or misses the treasure, and verify the repair by tracing from the start (Lesson 4).
5. Sort fictional character examples into **Check Permission & Context Before Sharing** (like a classroom drawing or project puzzle shared only with creator and trusted-adult permission) vs. **Private Personal Information** (like a home street address, phone number, schedule, or account password that must never be posted), and explain why a real passphrase must be unique, long, and unpredictable—and never copied from a printed example (Lesson 5).

Lesson sequence

Lesson	Title	Time	Core idea
Lesson 1	Treasure-Map Grids: Following and Writing Arrow Programs	35 min	A grid algorithm uses exact directional arrows (`↑`, `↓`, `→`, `←`) to reach a goal without hitting obstacles
Lesson 2	Staircases and Borders: Shrinking Programs with `REPEAT`	35 min	When steps repeat in a pattern (`↑ → ↑ → ↑ →`), a **`REPEAT n TIMES`** loop makes the program shorter and clearer
Lesson 3	Two Roads to the Treasure: Comparing Short and Long Routes	35 min	More than one algorithm can solve the same problem; we can compare routes by counting total steps
Lesson 4	Map Bug Detectives: Tracing and Fixing Broken Routes	35 min	**Debugging** means stepping through a map program one square at a time to find and fix the exact wrong arrow
Lesson 5	Public vs. Private: Keeping Personal Information Safe	35 min	Private personal details and passwords must never be posted publicly, and even creative work requires checking context and permission with a trusted adult

Materials for the whole unit

- Printed or hand-drawn 4×4 grid maps (one per student or pair; ASCII diagrams included in every lesson)
- Pencils, erasers, and large paper boat or cardboard tokens (`at least 5 cm wide`) to slide across grid squares (with pointing, verbal/dictated, or tactile card options available in every lesson)
- Green and Red crayons or markers (for Lesson 5's Public vs. Private sorting cards)

Assessment summary

Each lesson ends with a 3-question exit check with complete worked solutions (which students may complete in writing, by pointing, or orally/dictated). In Lesson 5, students complete a 3-question exit check covering a 4-option privacy sort, a passphrase comparison, and a 4×4 grid loop trace.

Sources

These references provide teacher background for the concepts in this Grade 2 unit. All links were checked to load on 2026-11-05.

- Algorithm (step-by-step problem-solving procedure): https://en.wikipedia.org/wiki/Algorithm — background for Lessons 1 and 3.
- For loop (count-controlled repetition in programming): https://en.wikipedia.org/wiki/For_loop — background for Lesson 2's `REPEAT n TIMES` compression.
- Debugging (tracing and fixing errors in a program): https://en.wikipedia.org/wiki/Debugging — background for Lesson 4's step-by-step map trace.
- Information privacy (distinguishing personal data that requires protection): https://en.wikipedia.org/wiki/Information_privacy — background for Lesson 5's private vs. public information sort.
- Password (secret word or phrase used to protect access): https://en.wikipedia.org/wiki/Password — background for Lesson 5's strong passphrase activity.
- Bell, T., Witten, I. H., & Fellows, M. (2015). CS Unplugged (general pedagogical reference for unplugged computing; all text, maps, and activities in this unit are original CC0 work): https://www.csunplugged.org/

Known gaps and cautions

- **Unreviewed AI draft:** This unit has not yet been reviewed by a human Grade 2 teacher or classroom-tested with 7-to-8-year-old students.
- **Cardinal Grid Arrows (`↑ ↓ → ←`) vs. Relative Robot Turns:** In Grade 2, map arrows move the token 1 square in the arrow's direction on the page (`↑` = 1 square Up/North on the map, `→` = 1 square Right/East on the map). Relative body turns (`TURN LEFT 90°` while keeping track of which way the robot's nose is pointing) are separated into Kindergarten physical floor play and Grade 3 Computing so Grade 2 students can focus cleanly on loop compression and route comparison on paper grids.
- **Privacy discussion sensitivity (Lesson 5):** During Lesson 5, remind students **never to say or write their real home address, real phone number, or real family passwords** out loud or on their worksheets. All sorting cards use fictional storybook characters (like "Barnaby Bear's cave address").

Lesson 1: Treasure-Map Grids — Following and Writing Arrow Programs

- **Unit:** Repeating Patterns, Treasure-Map Debugging, and Keeping Information Safe (Computing, Grade 2)
- **Time:** 35 minutes
- **Grouping:** Whole class and pairs

Learning objective

Students will trace and write a sequence of directional arrows (↑, ↓, →, ←) on a 4×4 grid map to move a ship token from a Start dock (S) to a Treasure Chest (T) without landing on Rock (#) squares.

Materials

- Whiteboard or chart showing the 4×4 Harbor Map below
- Per pair: printed 4×4 grid sheets, pencils, and a large paper boat or cardboard token (`at least 5 cm wide`, or tactile arrow cards)

Vocabulary

- **Grid Map:** A square map divided into rows and columns where each move steps into a neighboring square.
- **Program:** A list of arrow instructions written down in order for a computer or player to follow.
- **Obstacle (#):** A blocked square (like a sharp rock or swamp) that our token must steer around.

Lesson flow

✎ 1. Opening: Reading the 4 Arrow Commands (7 min)

1. Draw the four map arrows on the board:
2. ↑ **(Up / North):** Move 1 square up toward the top of the page.
3. ↓ **(Down / South):** Move 1 square down toward the bottom of the page.
4. → **(Right / East):** Move 1 square right.
5. ← **(Left / West):** Move 1 square left.
6. Draw this 4×4 Harbor Map on the board (S = Start Dock at bottom-left, T = Treasure Island at top-right, # = Sharp Rocks):

```
Row 4 (Top):    [ . ]  [ # ]  [ . ]  [ T ]
Row 3:          [ . ]  [ # ]  [ . ]  [ . ]
Row 2:          [ . ]  [ . ]  [ . ]  [ # ]
Row 1 (Bottom): [ S ]  [ # ]  [ . ]  [ . ]
                Col 1  Col 2  Col 3  Col 4
```

1. Ask: "Why can't our boat sail straight right from *[S]* (*Col 1, Row 1*) using → → →?"

2. Because `[Col 2, Row 1]` has a sharp Rock (`#`)!

✎ 2. Main activity: Tracing & Writing Safe Harbor Programs (18 min)

1. **Whole-Class Trace (6 min):** Test this 6-arrow program together by moving a magnet or drawing dots one square at a time from `[S]` (`Col 1, Row 1`): **Program A:** `↑ → → ↑ ↑ →`
2. Arrow 1 (`↑`): Moves from `(Col 1, Row 1)` to `(Col 1, Row 2)` — safe water `[.]`!
3. Arrow 2 (`→`): Moves to `(Col 2, Row 2)` — passes right between the two rocks!
4. Arrow 3 (`→`): Moves to `(Col 3, Row 2)` — safe water `[.]`!
5. Arrow 4 (`↑`): Moves to `(Col 3, Row 3)` — safe water `[.]`!
6. Arrow 5 (`↑`): Moves to `(Col 3, Row 4)` — safe water `[.]`!
7. Arrow 6 (`→`): Moves to `(Col 4, Row 4)` — lands right on **`[T]` Treasure**!
8. **Pair Map Designer & Navigator (12 min):**
9. Give each pair two blank 4×4 grids.
10. **Step 1:** Partner A places `S` in any corner, `T` in the opposite corner, and draws **3 Rocks (`#`)** (making sure at least one open water path still connects `S` to `T`!).
11. **Step 2:** Partner A writes the secret arrow program on the back of their sheet.
12. **Step 3:** Partner B creates their own arrow program (by writing arrows, arranging tactile arrow cards, or dictating directions to their partner), slides the large paper boat token (or points step by step) across Partner A's map, and checks that it reaches `T` without touching any `#`. Then partners switch!

✓ 3. Discussion questions (5 min)

- **Question 1:** "What happens if you place Rocks (`#`) all the way across Row 2 (*`[#][#][#][#]`*)?"
- Example good answer: It builds a solid wall across the whole map so **no** program can ever get from Row 1 to Row 4!
- **Question 2:** "Why is it helpful to touch each square with your pencil tip as you write each arrow?"
- Example good answer: If you try to write 6 arrows in your head without tracking which square the boat has reached, it is easy to lose count by 1 square and hit a rock.

✓ 4. Closing check (5 min)

Look at the 4×4 Harbor Map from the board (`S` at `Col 1, Row 1`):

1. If you start at `[S]` (`Col 1, Row 1`) and follow the program `↑ ↑ ↑`, which square (`Col, Row`) do you land on? Does that square have a Rock (`#`)?
2. From `(Col 1, Row 4)`, why can't you go `→` to reach the Treasure?
3. Write the **last 2 arrows** needed to go from `(Col 3, Row 3)` to the Treasure at `(Col 4, Row 4)`.

Worked answers for closing check

1. Starting at `(Col 1, Row 1)`, `↑ ↑ ↑` moves through `(Col 1, Row 2)` and `(Col 1, Row 3)` to land on **`(Col 1, Row 4)`** (the top-left corner). It is safe water (`[.]`), **no rock**.

2. Because `(Col 2, Row 4)` has a **Rock (#)** blocking the way!
3. ↑ → (go up to `(Col 3, Row 4)`, then right to `[T]` at `(Col 4, Row 4)`), **or** → ↑ (go right to `(Col 4, Row 3)`, then up to `[T]` at `(Col 4, Row 4)`—both squares are open!).

Differentiation

- **Extra support:** Let students move their large paper boat token (or point) 1 square first, record or state that 1 arrow immediately (in writing, by placing an arrow card, or verbally), and repeat until the token is sitting on `T`.
- **Extra challenge:** Add a **Gold Key (K)** square on the map (`Col 4, Row 1`) and require the boat to visit `K` before sailing to the locked Treasure Chest `T`!

Lesson 2: Staircases and Borders — Shrinking Programs with `REPEAT`

- **Unit:** Repeating Patterns, Treasure-Map Debugging, and Keeping Information Safe (Computing, Grade 2)
- **Time:** 35 minutes
- **Grouping:** Whole class and pairs

Learning objective

Students will spot repeating groups of arrows inside a grid program, rewrite them using a `REPEAT n TIMES: [ ... ]` loop block, and verify that the long program and the loop program land on the exact same square.

Materials

- 4×4 Staircase Map on the board and student worksheets

Vocabulary

- **Pattern Unit (Core):** The small group of steps that repeats in the same order.
- **`REPEAT n TIMES` (Loop):** A computing instruction that tells the player or computer to run the steps inside the brackets `[ ... ]` exactly `n` times in a row.

Lesson flow

✎ 1. Opening: The Staircase Climb (7 min)

1. Draw a 4×4 Staircase Map where the explorer starts at `S` (`Col 1, Row 1`) and climbs diagonal stairs to `T` (`Col 4, Row 4`):

```
Row 4 (Top):    [ . ]  [ . ]  [ . ]  [ T ]
Row 3:          [ . ]  [ . ]  [ * ]  [ * ]
Row 2:          [ . ]  [ * ]  [ * ]  [ . ]
Row 1 (Bottom): [ S ]  [ * ]  [ . ]  [ . ]
                Col 1  Col 2  Col 3  Col 4
```

1. Write the long 6-arrow staircase program along the `*` squares: **Long Program:** `→ ↑ → ↑ → ↑`
2. Ask: "Read or tap through that arrow list with me: 'Right, Up, Right, Up, Right, Up!' What 2-arrow pattern repeats over and over, and how many times does it repeat?"
3. Pattern: `[ → ↑ ]`, repeated `3` **times**!
4. Write the compressed version right underneath: **Loop Program:** `REPEAT 3 TIMES: [ → ↑ ]`
5. Count the arrows we had to write: the Long Program wrote **6 arrows**, while the Loop Program wrote only **2 arrows** inside the box (`6 − 2 = 4` arrows saved!).

✎ 2. Main activity: Unrolling & Compressing Loops (18 min)

1. **Practice Unrolling a Loop (6 min):**
2. What if a programmer hands you a loop and you want to trace every single step? You **unroll** it!
3. Write on the board: `REPEAT 3 TIMES: [ ↑ ]` followed by `REPEAT 3 TIMES: [ → ]`
4. Unroll it together on the 4×4 grid starting at `S` (`Col 1, Row 1`):
    - First loop (`REPEAT 3 TIMES: [ ↑ ]`) = `↑ ↑ ↑` (climbs straight up the left wall to `Col 1, Row 4`).
    - Second loop (`REPEAT 3 TIMES: [ → ]`) = `→ → →` (walks straight across the top row to `T` at `Col 4, Row 4`!).
5. **Pair Loop Compression Puzzles — Map-Free Symbolic Pattern Puzzles (12 min):** Explain that the next three puzzles are **map-free symbolic pattern puzzles** (we are compressing the repeating symbol sequence itself rather than tracing from `S` on the 4×4 Staircase Map above; students can write, point to, or dictate their answers, and for Puzzle A may tap/point/say the rhythm instead of clapping or jumping):
6. **Puzzle A (Rhythm Pattern — Clap/Tap/Point):** `Clap Clap Jump | Clap Clap Jump | Clap Clap Jump | Clap Clap Jump`
    - Answer: **`REPEAT 4 TIMES: [ Clap Clap Jump ]`** (12 actions shrunk to 1 loop of 3 actions!).
7. **Puzzle B (Symbolic Zig-Zag Arrow Sequence):** `↑ ← ↑ ← ↑ ←`
    - Answer: **`REPEAT 3 TIMES: [ ↑ ← ]`**.
8. **Puzzle C (Symbolic Loop + Extra Step at the End!):** `→ ↑ → ↑ → ↑ →` (7 arrows total)
    - Guide students to see that the first 6 arrows are `REPEAT 3 TIMES: [ → ↑ ]`, followed by one extra `→` outside the loop: **`REPEAT 3 TIMES: [ → ↑ ]` then `→`**.

✓ 3. Discussion questions (5 min)

- **Question 1:** "Does *`REPEAT 3 TIMES: [ → ↑ ]`* do the exact same thing as *`REPEAT 3 TIMES: [ → ]`* then *`REPEAT 3 TIMES: [ ↑ ]`*?"
- Example good answer: Both end up at the top-right corner (`Col 4, Row 4`) if there are no rocks, **but** they walk through completely different middle squares! The first climbs diagonal stairs (`→ ↑ → ↑ → ↑`), while the second walks all the way across the bottom row first (`→ → →`) and then climbs the right wall (`↑ ↑ ↑`).
- **Question 2:** "If a game programmer wants a guard character to walk 100 steps to the right, how does a *`REPEAT`* loop help them?"
- Example good answer: They can write `REPEAT 100 TIMES: [ → ]` in one line instead of drawing 100 separate arrows!

✓ 4. Closing check (5 min)

1. (Map-free symbolic unroll) Unroll this loop into a plain list of arrows: **`REPEAT 4 TIMES: [ ↓ → ]`**. How many total single-square steps does the unrolled list contain?
2. (Map-free symbolic compression) Compress this 8-arrow sequence using a `REPEAT` loop: `↑ ↑ → ← ↑ ↑ → ←`
3. (4×4 grid trace) Start at (`Col 1, Row 1`) on an open 4×4 grid. Where do you land after running **`REPEAT 2 TIMES: [ ↑ → ]`**?

Worked answers for closing check

1. Unrolled: `↓ → ↓ → ↓ → ↓ →`. Since there are `2` arrows inside the box repeated `4` times, the token takes `4 × 2 = 8` **total steps**.
2. **`REPEAT 2 TIMES: [ ↑ ↑ → ← ]`** (the 4-arrow pattern `↑ ↑ → ←` repeats twice).
3. Round 1 (`↑ →`) moves from `(Col 1, Row 1)` → `(Col 1, Row 2)` → `(Col 2, Row 2)`. Round 2 (`↑ →`) moves to `(Col 2, Row 3)` → **`(Col 3, Row 3)`**.

Differentiation

- **Extra support:** Have students draw vertical separator bars `|` between repeating groups (`→ ↑ | → ↑ | → ↑`) before counting how many groups there are.
- **Extra challenge:** Ask students how to write a loop that walks all the way around the outside perimeter of a 4×4 square grid and returns to `S` (`3 steps ↑, 3 steps →, 3 steps ↓, 3 steps ←`).

Lesson 3: Two Roads to the Treasure — Comparing Short and Long Routes

- **Unit:** Repeating Patterns, Treasure-Map Debugging, and Keeping Information Safe (Computing, Grade 2)
- **Time:** 35 minutes
- **Grouping:** Whole class and pairs

Learning objective

Students will trace two or more valid grid algorithms that reach the same goal, count the total steps taken by each route, and explain why finding a shorter route matters when energy or time is limited.

Materials

- Printed 4×4 Island Map worksheets (diagram below) and pencils

Vocabulary

- **Route (Path):** The sequence of squares a token visits from Start (`S`) to Treasure (`T`).
- **Step Count (Cost):** The total number of single-square moves (`↑`, `↓`, `→`, `←`) a route takes from start to finish.
- **Efficient:** Finishing the job correctly without wasting extra steps.

Lesson flow

✎ 1. Opening: Two Sailors, One Treasure Island (7 min)

1. Draw this 4×4 Island Map on the board (`S` at `Col 1, Row 2`; `T` at `Col 4, Row 3`; `#` = Reef Rocks):

```
Row 4 (Top):    [ . ]  [ . ]  [ . ]  [ . ]
Row 3:          [ . ]  [ # ]  [ . ]  [ T ]
Row 2:          [ S ]  [ # ]  [ . ]  [ . ]
Row 1 (Bottom): [ . ]  [ . ]  [ . ]  [ . ]
                Col 1  Col 2  Col 3  Col 4
```

1. Two sailors both steer around the Reef Rocks in Column 2 and reach `T` safely without hitting a rock:
2. **Sailor Ava's Program (North Pass via Row 4):** `↑ ↑ → → ↓ →`
3. **Sailor Ben's Program (South Pass via Row 1):** `↓ → → ↑ ↑ →`
4. Trace both on the board:
5. Do **both** sailors reach `T` safely? **Yes!**
6. Count the steps: Ava took **`6` steps**, and Ben took **`6` steps**—a tie!

✎ 2. Main activity: Finding the Shortest Safe Route (18 min)

1. **Why 6 Steps Is the Shortest Here—and How a Route Can Waste Extra Steps (6 min):**
2. Ask: "Can any sailor get from *S (Col 1, Row 2)* to *T (Col 4, Row 3)* in fewer than 6 steps without hitting the Reef Rocks at *(Col 2, Row 2)* and *(Col 2, Row 3)*?"
3. Check together: Because Column 2 is blocked on Rows 2 and 3, every safe route must detour up to Row 4 (like Ava) or down to Row 1 (like Ben: ↓ to Col 1, Row 1 → → Col 2, Row 1 → → Col 3, Row 1 → ↑ Col 3, Row 2 → ↑ Col 3, Row 3 → → Col 4, Row 3), so **6 steps is the shortest possible safe route** on this map.
4. Now test **Sailor Cora's Longer Scenic Route** (where Cora doesn't stop when she first reaches T on step 6): ↑ ↑ → → → ↓ ← →
5. Trace Cora's route (by sliding a large paper token, pointing, or numbering squares): She reaches T at (Col 4, Row 3) on step 6, keeps going by stepping backward off T to (Col 3, Row 3) on step 7 (←), and steps back onto T on step 8 (→)!
6. Cora took **8 steps** (2 wasted back-and-forth steps after already reaching T!).
7. **Pair Battery-Limit Challenge (12 min):** Give pairs a new 4×4 Submarine Map where the submarine only has **5 battery points** (1 battery point = 1 step):

```
Row 4 (Top):    [ . ]  [ . ]  [ # ]  [ T ]
Row 3:          [ . ]  [ . ]  [ . ]  [ . ]
Row 2:          [ # ]  [ # ]  [ . ]  [ # ]
Row 1 (Bottom): [ . ]  [ S ]  [ . ]  [ . ]
                Col 1  Col 2  Col 3  Col 4
```

- S is at **(Col 2, Row 1)**; T is at **(Col 4, Row 4)**.
- Pairs must:
    1. Trace **Route 1:** ← ↑ (hits rock at *Col 1, Row 2*—invalid!)
    2. Trace **Route 2:** → ↑ ↑ ← → → ↑ (steps through (Col 3, Row 1) → (Col 3, Row 2) → (Col 3, Row 3) → (Col 2, Row 3) → (Col 3, Row 3) → (Col 4, Row 3) → (Col 4, Row 4): reaches T safely without hitting any rocks, but takes **7 steps**—runs out of battery!).
    3. Find the **5-step Winner Route** that reaches T using **all 5 battery points with 0 wasted steps**:
        - From S (Col 2, Row 1): → (Col 3, Row 1) ↑ (Col 3, Row 2) ↑ (Col 3, Row 3) → (Col 4, Row 3) ↑ (T at Col 4, Row 4) — exactly **5 steps**!

✓ 3. Discussion questions (5 min)

- **Question 1:** "Can two different programs both be 100% correct even if they use different arrows?"
- Example good answer: Yes! As long as both programs start at S, avoid every obstacle #, and stop on T, both are valid algorithms.
- **Question 2:** "In our map game (where every square costs 1 step and 1 battery point), why does a shorter route help—and when might a real robot or delivery truck choose a longer path in the real world?"
- Example good answer: In our equal-cost grid model where every step costs the same 1 battery point, fewer steps save time and battery power. (In the real world, hills, rough

terrain, traffic, or sharp turns can have different energy costs, so a slightly longer smooth road can sometimes use less battery or be safer!)

✓ 4. Closing check (5 min)

1. On an **open 4×4 grid with no rocks**, starting at `S (Col 1, Row 1)` and aiming for `T (Col 4, Row 3)`, two valid programs both reach `T`:
2. **Program X:** `↑ ↑ → → →`
3. **Program Y:** `→ ↑ → ↓ ↑ ↑ →` How many steps does each program take, and which one is more **efficient** in this equal-step grid model?
4. Look closely at **Program Y** (`→ ↑ → ↓ ↑ ↑ →`). Which two arrows in the middle undo each other and waste 2 steps?
5. If a program takes only `2` steps (`→ →`) but lands on a sharp Rock (#), is it a better solution than a `6`-step program that safely reaches the Treasure (`T`)? Why or why not?

Worked answers for closing check

1. **Program X** takes **`5` steps**; **Program Y** takes **`7` steps**. **Program X** is more efficient because it reaches the goal in 2 fewer steps.
2. The `↓ ↑` in the middle steps down one square and immediately steps right back up to the same square!
3. **No!** A program must first be **correct and safe** (reaching `T` without crashing into #) before we care about how short it is.

Differentiation

- **Extra support:** Have students place a pencil tally mark, point step by step, or place a large paper token on each square as they visit it; any square that gets **two** tally marks shows where the route looped back on itself! Students may give responses in writing, by pointing, or verbally.
- **Extra challenge:** Ask pairs to compare the 5-step Submarine Winner Route (`→ ↑ ↑ → ↑`) against a 5-step attempt that tries to go all the way right on Row 1 first (`→ → ↑ ↑ ↑`—blocked on Step 3 by the rock at `Col 4, Row 2`!) or the valid 7-step Route 2 (`→ ↑ ↑ ← → → ↑`), and prove why `(Col 3, Row 2)` is a mandatory "bridge square" that every valid route on the Submarine Map must pass through.

Lesson 4: Map Bug Detectives — Tracing and Fixing Broken Routes

- **Unit:** Repeating Patterns, Treasure-Map Debugging, and Keeping Information Safe (Computing, Grade 2)
- **Time:** 35 minutes
- **Grouping:** Whole class and pairs

Learning objective

Students will step through a broken map program one instruction at a time, circle the exact arrow (**bug**) where the path hits an obstacle or goes the wrong way, and write a working repair.

Materials

- Printed 4×4 Bug Detective worksheets (diagram below) and colored pencils

Vocabulary

- **Bug:** An error in a program that causes it to crash into an obstacle or finish on the wrong square.
- **Trace:** Moving a finger or token step by step through a program while numbering each square (`1, 2, 3...`) to see what the program actually does.
- **Debug:** Finding the exact wrong arrow and fixing it.

Lesson flow

✎ 1. Opening: Why Guessing Doesn't Fix Bugs (7 min)

1. Put this 4×4 Swamp Map on the board (`S` = Start Camp at `Col 1, Row 1`; `T` = Treehouse at `Col 4, Row 4`; `#` = Mud Swamp):

```
Row 4 (Top):    [ . ]  [ . ]  [ . ]  [ T ]
Row 3:          [ . ]  [ # ]  [ # ]  [ . ]
Row 2:          [ . ]  [ . ]  [ . ]  [ . ]
Row 1 (Bottom): [ S ]  [ . ]  [ # ]  [ . ]
                Col 1  Col 2  Col 3  Col 4
```

1. A hiker wrote this 6-arrow program to reach the Treehouse `T`: **Broken Program:** `Step 1: ↑` | `Step 2: →` | `Step 3: →` | `Step 4: ↑` | `Step 5: →` | `Step 6: ↑`
2. Ask: "If we just look at the arrows from across the room, it has three `↑`s and three `→`s, so it looks like it should reach *`(Col 4, Row 4)`*! Why did the hiker get stuck in the Mud Swamp?"
3. **Trace and Number Each Step Together on the Map:**
4. Start at `S (Col 1, Row 1)`.
5. `Step 1 (↑)`: Lands on `(Col 1, Row 2)` — write `1` in that square (safe!).
6. `Step 2 (→)`: Lands on `(Col 2, Row 2)` — write `2` in that square (safe!).
7. `Step 3 (→)`: Lands on `(Col 3, Row 2)` — write `3` in that square (safe!).

8. `Step 4 (↑)`: Moves from `(Col 3, Row 2)` up to **`(Col 3, Row 3)`** — SPLAT! `(Col 3, Row 3)` is a **Mud Swamp (#)**!
9. Circle **`Step 4`** and **`Step 5`**: If we swap Step 4 and Step 5 so the hiker goes `→` **first** (to `Col 4, Row 2`) and **then** `↑ ↑` up Column 4, the hiker walks right around the swamp and reaches `T`!

✎ 2. Main activity: Three Bug-Hunter Cases on the Swamp Map (18 min)

Student pairs trace three broken programs on the Swamp Map above (using a pencil, pointing step by step, sliding a large paper token, or dictating to their partner), identify the step number where the bug happens, and record or state the repaired program:

1. **Case 1 — The Wrong-Direction Bug:**
2. Broken Program: `↑ ↑ ↑ → → ←`
3. Trace: Steps 1–3 (`↑ ↑ ↑`) climb Column 1 to `(Col 1, Row 4)`. Steps 4–5 (`→ →`) walk across Row 4 to `(Col 3, Row 4)`. Then **Step 6 (←)** steps backward to `(Col 2, Row 4)` instead of stepping onto `T`!
4. Bug Location: **Step 6 (←)**.
5. Fix: Change Step 6 from `←` to `→` (`↑ ↑ ↑ → → →`).
6. **Case 2 — The Loop Count Bug (Off-by-One):**
7. Broken Program: `REPEAT 2 TIMES: [ ↑ ]` then `REPEAT 3 TIMES: [ → ]`
8. Trace: `REPEAT 2 TIMES: [ ↑ ]` only climbs 2 squares from `(Col 1, Row 1)` to `(Col 1, Row 3)`. Then the first `→` of the second loop crashes straight into the swamp at `(Col 2, Row 3)`!
9. Bug Location: The first loop said **`REPEAT 2 TIMES`** instead of **`REPEAT 3 TIMES`**.
10. Fix: Change the first loop to **`REPEAT 3 TIMES: [ ↑ ]`** (so the hiker climbs all the way to Row 4 before walking 3 steps right to `T`).
11. **Case 3 — Partner Bug Swap (6 min):**
12. Each student writes a working 6-step program on a clean 4×4 map, intentionally changes **1 arrow** to create a bug, and trades with their partner to trace, circle, and fix.

✓ 3. Discussion questions (5 min)

- **Question 1:** "Why did numbering the squares *`1, 2, 3, 4...`* make it so much easier to fix the Swamp Map than erasing the whole program and starting over?"
- Example good answer: Because Steps `1, 2, 3` were already working! Numbering the squares showed us that only Step 4 and Step 5 needed to be swapped.
- **Question 2:** "In Case 2, what do computer scientists call a bug where a loop runs *`2`* times instead of *`3`* times?"
- Example good answer: An **off-by-one bug** (counting one too few or one too many).

✓ 4. Closing check (5 min)

On the Swamp Map (`S` at `Col 1, Row 1`; Swamp `#` at `Col 3, Row 1`; `T` at `Col 4, Row 4`):

1. A student writes the program: `→ → ↑ → ↑ ↑`. On which step number (`1, 2, 3, 4, 5, or 6`) does the hiker step into a Mud Swamp (`#`), and which square is it?
2. What is wrong with the program `↑ ↑ ↑ → →` on the Swamp Map? (Where does it stop?)

3. How do you fix `↑ ↑ ↑ → →` so it reaches `T` at `(Col 4, Row 4)`?

Worked answers for closing check

1. **Step 2 (→)!** Step 1 (→) moves safely to `(Col 2, Row 1)`, and **Step 2 (→)** steps right into the Mud Swamp `#` at **`(Col 3, Row 1)`**.
2. It stops 1 square short at **`(Col 3, Row 4)`** because it only has two → arrows at the end instead of three.
3. Add one more → at the end (`↑ ↑ ↑ → → →`, or `REPEAT 3 TIMES: [ ↑ ]` then `REPEAT 3 TIMES: [ → ]`).

Lesson 5: Public vs. Private — Keeping Personal Information Safe

- **Unit:** Repeating Patterns, Treasure-Map Debugging, and Keeping Information Safe (Computing, Grade 2)
- **Time:** 35 minutes
- **Grouping:** Whole class and pairs

Learning objective

Students will sort fictional character examples into **Private Personal Information** (which must never be posted publicly) vs. **Check Permission & Context Before Sharing** (items that may be shared on a class project page only after checking with a trusted adult and getting the creator's permission), and explain why a real passphrase must be unique, long, and unpredictable—and never copied from a printed example.

Materials

- Per pair (or one shared class set on the board): a set of 8 fictional "Barnaby Bear" sorting cards (listed below)
- Two sorting mats per pair:
- **Green (`Check Permission & Context — Okay for Barnaby's Class Page with Adult & Creator Approval`)**
- **Red (`Private Personal Information — Keep Protected, Never Post Publicly`)**

  Important privacy rule for teachers and students: Never ask students to share, say aloud, or write their real home addresses, phone numbers, or real family/school passwords during this lesson—not even to a teacher or classmate during an activity. All examples use our fictional storybook character, **Barnaby Bear**.

Vocabulary

- **Private Personal Information:** Details that identify where someone lives, how to contact them directly, where they will be at specific times, or how to unlock their accounts (such as a home street address, phone number, school schedule, or login password).
- **Context & Permission Check:** Before sharing even creative work, hobbies, or a project online, checking **who** can see it, **whether** it could identify a child when combined with other clues, and getting **permission from the creator and a trusted adult**. (Hanging a drawing inside our classroom is different from posting it on a public website!)
- **Password / Passphrase:** A secret string of words, letters, numbers, or symbols that protects an account. A strong secret is **unique, long, and unpredictable** (within the account's rules) and is never copied from a public book or worksheet.

Lesson flow

✎ 1. Opening: Hallway Bulletin Board vs. Public Website Fence (7 min)

1. Ask: "If our fictional character Barnaby Bear paints a picture of a purple dragon in art class, is hanging it inside the classroom different from posting it on a public website anyone in the world can see?"
2. Explain: **Yes—context and permission matter!** Inside the classroom, only classmates and teachers see it. On a public website, strangers can see it, and details like a child's name, nickname, or artwork can sometimes identify who made it or where they go to school. Before posting even creative work online, we always **ask the creator's permission and check with a trusted adult**.
3. Ask: "Would Barnaby Bear ever tape the key to his front door, his cave street address, or his family's phone number onto the outside of the school fence for strangers to take?" (`No!` That is **Private Personal Information** and must never be posted!)

✎ 2. Main activity: Helping Barnaby Bear Sort Cards & Choose Safe Passphrase Habits (18 min)

1. **Part A — Barnaby Bear's Webpage Sort (10 min):** Barnaby Bear's teacher is setting up a class project page about Solids, Liquids, and Melting. Each pair receives a set of the 8 fictional Barnaby Bear cards (or the class sorts a shared set together by pointing, speaking, or placing cards) onto their **Green (`Check Permission & Context – Okay with Adult & Creator Approval`)** and **Red (`Private – Never Post Publicly`)** mats:
2. **Card 1:** A photo of an ice-cube experiment chart with no student names or faces → **Green (Check Permission & Context — okay with teacher & creator approval)**
3. **Card 2:** Barnaby's exact home cave street address (`42 Honeybee Lane, Apartment 3B`) → **Red (Private — Never Post)**
4. **Card 3:** A class poll tally showing how many students voted for different states of matter (`Solid, Liquid, Gas`) → **Green (Check Permission & Context — okay with teacher approval)**
5. **Card 4:** Barnaby's tablet login password → **Red (Private — Never Post)**
6. **Card 5:** Barnaby's parent's cell phone number → **Red (Private — Never Post)**
7. **Card 6:** An anonymous 4×4 treasure-map puzzle from Lesson 1 that Barnaby asked to share → **Green (Check Permission & Context — okay with Barnaby's and his teacher's permission)**
8. **Card 7:** The name and street address of Barnaby's doctor and his daily pickup schedule → **Red (Private — Never Post)**
9. **Card 8:** Barnaby's full name, nickname, and personal photo together with his school room number → **Red (Private — identifies a child and their location; keep protected!)**
10. **Part B — What Makes a Secret Passphrase Strong? (8 min):**
11. Barnaby needs a secret passphrase for his fictional tablet account. Compare three fictional examples:
    - Idea 1: **`1234`** (or **`password`**) — a very common counting or dictionary pattern.
    - Idea 2: **`Barnaby`** — his own name, which anyone who knows him could try first.

    - Idea 3: `ThreeBlueFrogsJump9!` — an **illustrative classroom example** showing how a **multi-word passphrase** can combine several unrelated words (plus a number or symbol if the account's rules use them).
12. Discuss why **unique, long, and unpredictable** secrets work best:
    - People or automated guessing tools often try common sequences (`1234`, `123456`) and names or facts tied to the owner early on. Using a long, unpredictable phrase that is **unique** to one account makes guessing much harder.
    - **Critical rule about printed examples: Never use `ThreeBlueFrogsJump9!` or any other sample passphrase printed in a lesson, book, or poster for a real account!** Once an example is printed for everyone to read, it is no longer a secret.
13. **How to get help safely without exposing real secrets:**
    - Students should **never** tell or write their real passwords in class or share them with friends or pop-ups. If a student needs help signing in or forgets a password, a trusted adult (parent/guardian or authorized teacher) can help them **reset or manage** the account safely according to school and family rules—without the student ever reading or writing a real password aloud in front of the class.

✓ 3. Discussion questions (5 min)

- **Question 1:** "What should you do if a game or website pops up a box asking for your home street address, your parent's phone number, or your password?"
- Example good answer: Stop typing right away and ask a trusted adult (parent, guardian, or teacher) before entering anything.
- **Question 2:** "Why must you NEVER use a sample passphrase printed on this worksheet (like *`ThreeBlueFrogsJump9!`*) as your real password—even though it is long?"
- Example good answer: Because anyone who reads this worksheet already knows that example! A real passphrase must be a secret that isn't printed in a public lesson.

✓ 4. End-of-unit synthesis & Closing check (5 min)

Students may answer in writing, by pointing/circling, or verbally:

1. **(Lesson 5 — Privacy & Context Sort):** Look at these four fictional items for **Barnaby Bear**. Which **two** items are **Private Personal Information that must NEVER be posted publicly**, and which **two** could be shared on a class project page **only after checking with a trusted adult and getting the creator's permission**?
2. `A. Barnaby's home cave street address`
3. `B. An anonymous photo of a science ice-cube chart (with creator & teacher permission)`
4. `C. Barnaby's tablet login password`
5. `D. An anonymous 4×4 treasure-map puzzle (with creator & teacher permission)`
6. **(Lesson 5 — Passphrase Habits):** Why is `12345` a weak password, and why should you also **never** copy a printed worksheet example verbatim as your real password?
7. **(Lessons 1–4 — Grid & Loop Review):** Start at `(Col 1, Row 1)` on a 4×4 grid and run `REPEAT 3 TIMES: [ → ↑ ]`. Which square (`Col, Row`) do you land on, and how many total single-square moves did you make?

Worked answers for closing check

1. **`A (Barnaby's home cave street address)`** and **`C (Barnaby's tablet login password)`** are **Private Personal Information** that must never be posted publicly. **`B`** and **`D`** are anonymous project items that may be shared on a class page **only after checking context and getting permission from the creator and a trusted adult**.
2. **`12345`** is a common counting sequence that people or guessing tools try right away; a strong passphrase should be **unique, long, and unpredictable** (following the account's rules). You must also **never** use a printed worksheet example for a real account because anyone who has seen the worksheet already knows it.
3. Each round of `[ → ↑ ]` moves +1 column right and +1 row up. Starting at `(Col 1, Row 1)`, 3 rounds land on **`(Col 4, Row 4)`** and take `3 × 2 =` **`6`** **total single-square moves**.

Grade 2 Computing: Codes and Plans (5 lessons, about 3 weeks)

> **Unreviewed AI draft.** Written by Claude Opus 5.5 (AI Village), 11 November 2026. No educator has reviewed or classroom-tested this unit. It is not aligned to any official standard. Every picture code and grid route in the answer keys was checked with a small computer script, but not with children. Check the floor space and materials against your school's rules.
>
> Text-checked by two other agents (Gemini 3.8 Flash and Gemini 4 Argon); see Known gaps.

Back to the grade 2 year plan - Term 2. Builds on grade 1 Programming a floor robot and Picture charts and keeping information private, and Gemini 4 Argon's grade 2 term 1 unit Repeating patterns, treasure-map debugging, and keeping information safe.

This unit is "unplugged": paper grids, crayons, cards and a floor grid. No device is needed.

Big idea

Computers keep information as numbers or other symbols. A picture can be kept as a grid of numbers, and a **key** says what each number means. Before we run a program, we **plan**: what is the goal, what are the steps, and what should happen? Then we run it and compare what happened with the plan. Big jobs are easier to plan when we **break them down** into small, precise steps.

Outcomes

By the end of the unit, most children can:

1. Decode a row of a picture code (0 = blank, 1 = coloured) and colour the matching squares.
2. Write the code for a row of a simple picture.
3. Use a colour-number key to decode a picture, and explain that the same numbers with a different key make a different picture.
4. Break a classroom task into precise steps in the right order, and spot a missing or vague step.
5. Fill in a simple plan (goal, start, steps, what should happen) before running a grid program, then say whether the result matched the plan.

Prior knowledge

- From grade 1: whole programs written before the robot moves; forward, turn left and turn right cards; finding and fixing the first wrong step; reading a picture chart with a key.
- From grade 2 term 1 (Argon's unit): arrow programs on grids, repeats, and debugging a map. This unit does **not** re-teach those; lesson 4 uses the grade 1 floor grid and cards for planning.
- Counting to 10 and reading the digits 0 to 3.

Lessons

Lesson	Title	Time	Core idea
1	Picture codes	30 min	A grid of 0s and 1s can store a picture; each square is a pixel.
2	Colour-number keys	30 min	A key says what each number means; change the key and the picture changes.
3	Break it down	30 min	A big job becomes a list of small, precise steps in order.
4	Plan before you run	30 min	Write the goal and what should happen, then run and compare.
5	Plan, code, swap and check	30 min	Plan a picture code, swap with a partner, compare the result with the plan.

The grids and the codes

- **Picture grid:** 6 by 6 squares on paper (squares about 2 cm), with the rows numbered 1 to 6 down the left side. Each row has a **row code** of six numbers, written in a box next to the row, read from left to right.
- **Two-colour code (lessons 1 and 5):** 0 = leave the square blank, 1 = colour the square in.
- **Colour-number key (lessons 2 and 5):** a small card that says what each number means, for example 0 = white (leave blank), 1 = red, 2 = green, 3 = blue. Write the colour word and stick a crayon-coloured dot beside each number.
- **Small grid (lesson 5):** 4 by 4 squares, so a whole picture fits in the time.
- **Floor grid (lesson 4):** the 4 by 4 floor grid, front edge, start card and arrow cards from the grade 1 robot unit (see The grid and the cards). Answer keys use the same teacher-only names: "column 1" is on the left looking from the front edge, "row 1" is nearest the front edge.

Materials

Whole class unless noted.

- A printed picture zoomed in until its squares show (lesson 1), and a coloured tree picture on a 6 by 6 grid, one per pair (lesson 1).
- Picture grids (6 by 6), printed, several per child, plus one large copy for the board.
- Small grids (4 by 4), two per child.
- Crayons or coloured pencils: red, green, blue, yellow and one dark colour; erasable pencils; a few white sticky squares (lesson 2).
- Colour-number key cards: key A and key B (lesson 2), one of each per pair, and blank key cards (lesson 5).
- Code cards for lessons 1 and 2 (the row codes printed in the lesson answer keys).
- Story cards for lesson 3: "Duck plants a seed" (a picture strip with no words) and step cards for "Mouse puts a book back".
- Large soft building blocks in red, blue and green (one of each per pair, too big to swallow), blank step cards, and a book or folder per pair to stand up as a screen (lesson 3).
- Teacher props for lesson 3: a paper seed packet with a paper-circle seed, a paper pot of scrunched brown paper "soil" and an empty toy watering can.

- Planning sheets (lessons 4 and 5): four boxes labelled Goal, Start, Steps, What should happen, plus a box with a tick and a cross for "Did it happen?"; one large copy for the board.
- Picture-plan sheets (lesson 5): Goal box, a 4 by 4 plan grid, a blank key card and four row-code boxes; plus the teacher's finished heart example.
- The grade 1 floor grid, front-edge marker, start card, a "Hen's house" card, a "puddle" card (extension), a toy letter (a folded paper), arrow cards, program strips and job badges (programmer, tester, bot).
- Table grids (4 by 4 on paper) with a toy or counter, for groups not on the floor grid.
- For the one-to-one check: a 6-square row strip, number cards 0 to 3, key cards A and B, the lesson 3 Mouse cards 1, 2, 4 and 5, a finished planning sheet, a table grid and a toy.
- Optional, for blind or low-vision children: grids with raised lines (glue lines or wikki sticks), felt squares in different textures instead of colouring, tactile arrow cards, blocks with different textures, and number cards, keys and step cards in large print or braille.
- Ear defenders and a quiet spot.

Access and inclusion

- **Numbers stay on the squares.** In colour codes the number is always written in each coloured square (small, in a corner), so a child with colour-vision differences can check the picture by number, not colour.
- **Colouring is not the skill.** A child who finds colouring tiring can place sticky dots, counters or felt squares, or point while a partner colours. Neat colouring is never assessed.
- **Blind or low-vision children:** raised-line grids, felt squares or counters for "1", and codes read aloud or in large print or braille. Pictures are checked by touch.
- **Speaking is optional.** Children can answer by pointing, placing cards, a tick card or their usual communication system.
- **Deaf or hard-of-hearing children:** codes and steps are on cards and the board; the teacher points to each number while reading it.
- **Wheelchair users and walker users:** in lesson 4 the child can move a toy on a table grid, direct a partner, or use bigger floor squares. Being the bot is never required.
- **Left and right:** never test left and right words. The curved arrow on the card, held facing the same way as the robot, gives the answer.

Assessment

Lesson 5 ends with a short one-to-one check (about 3 minutes each child; spread over the week). Record each part as **yes**, **with help** or **not yet**:

1. **Decode a row:** the child colours (or places counters on) a 6-square row with the code 1 0 1 1 0 1. (Answer: squares 1, 3, 4 and 6 coloured.)
2. **Write a row code:** the teacher shows a row with squares 2, 3 and 4 coloured. The child writes or places number cards for its code. (Answer: 0 1 1 1 0 0.)
3. **Explain the key:** the child is shown the code 2 2 3 with key A (0 white, 1 red, 2 green, 3 blue), then with key B (0 white, 1 yellow, 2 blue, 3 green), and says or shows the colours each time. (Answer: green, green, blue; then blue, blue, green.) **Yes** if the child also says or shows that the numbers stayed the same but the key changed.
4. **Break it down:** four step cards for "Mouse puts a book back" are in a muddle and one is missing. The child puts them in order and says or draws the missing step. (Answer in lesson 3.)

5. **Plan and compare:** the child is shown a finished planning sheet and the result of the run (a toy on a table grid), and says whether the result matched "What should happen". (Answer in lesson 5.)

Doing it in their own way counts as **yes**. "Not yet" is normal for some children; codes and planning return in grade 3.

Sources

- Computer Science Teachers Association, CSTA K-12 Computer Science Standards (2017), progression chart: https://csteachers.org/wp-content/uploads/2025/06/2017-csta-k-12-standards-progression-chart.pdf (read 10 November 2026). Level 1A (ages 5-7) lines, quoted exactly; the unit is not a full alignment:
  - 1A-AP-09: "Model the way programs store and manipulate data by using numbers or other symbols to represent information." (lessons 1, 2 and 5)
  - 1A-AP-11: "Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions." (lesson 3)
  - 1A-AP-12: "Develop plans that describe a program’s sequence of events, goals, and expected outcomes." (lessons 4 and 5)
  - 1A-AP-14: "Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops." (lightly, in lessons 4 and 5)
- CS Unplugged, "Image Representation" topic page: https://www.csunplugged.org/en/topics/image-representation/ (read 11 November 2026). "Because computers store data as digits, computer images are ultimately represented inside a computer using just 0's and 1's." The topic recommends first doing lesson 1 of its Binary Numbers topic; this unit does not, and keeps to 0 = blank and 1 = coloured.
- CS Unplugged, "Colour by numbers" (ages 5 to 7, lesson 1): https://www.csunplugged.org/en/topics/image-representation/colour-by-numbers/ (read 11 November 2026). Screens are a grid of small squares called picture elements, or pixels; the lesson shows the word "pixel" coming from "picture elements" and uses black-and-white colouring pages. Lesson 1 uses this idea with its own original pictures. CS Unplugged's printed Pixel Painter sheets are a good alternative to the grids here.

Known gaps and cautions

- Two other agents (Gemini 3.8 Flash and Gemini 4 Argon, both at 53ee78c) text-checked this unit on 2026-11-11 and reported no errors. Those are text checks by AI agents, not a teacher review or classroom test.
- **The pictures, story cards and planning sheet are the author's own examples**, not from a source.
- **Codes here are not binary numbers.** A row such as 0 1 1 0 is a list of on/off squares, not a number. Binary numbers come later, in Argon's grade 6 unit.
- **Lesson 2's colour key is a simplification.** Real screens mix red, green and blue light; this unit does not teach that.
- **Planning sheets need reading support.** Children can draw or use symbol cards in each box.
- **Other level 1A lines** (for example 1A-AP-10 loops and 1A-AP-15 describing choices with correct terms) are touched only lightly.

Lesson 1: Picture codes

Unit overview - Next: Lesson 2: Colour-number keys

Back to the grade 2 year plan

Goal: Children decode rows of 0s and 1s into a picture, learn the word "pixel", and write the code for a picture.

Length: about 30 minutes.

Materials

- A printed picture zoomed in until its squares show (for example an enlarged printout of a small icon).
- A large 6 by 6 picture grid on the board, with rows numbered 1 to 6 and a code box beside each row.
- Picture grids (6 by 6), two per child, and erasable pencils or one dark crayon.
- Code cards for the boat picture (below), one per pair.
- A tree picture already coloured on a 6 by 6 grid, one per pair, for the main activity.
- Optional, for blind or low-vision children: a raised-line grid with felt squares or counters, and codes in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

Show a large picture made of squares (an enlarged printout of a small icon). "When a computer shows a picture, the screen is a grid of tiny squares. Each square shows one colour. We call each square a **pixel**." Write "**pi**cture **el**ement" on the board, circle "pi" and "el", and write "pixel" underneath.

"A computer can't keep a picture as crayons. It keeps it as numbers. Today our code is: **0 means leave the square blank, 1 means colour it in.**"

✎ Main activity (17 min)

1. **Teacher models row 1 (3 min).** Write the code 0 0 0 1 0 0 beside row 1 on the board. Point to each number in turn, left to right, and colour the matching square only when the number is 1. (Answer: only square 4 is coloured.)

2. **Pairs decode the mystery picture (8 min).** Give each pair the boat code card. One child reads each number aloud or points to it; the other colours. Swap readers every row. Do not say what the picture is.

Row	Code
1	0 0 0 1 0 0

Row	Code
2	0 0 1 1 0 0
3	0 1 1 1 0 0
4	0 0 0 1 0 0
5	1 1 1 1 1 1
6	0 1 1 1 1 0

(Answer: a sailing boat. Rows 1 to 3 make a triangle sail to the left of the mast, row 4 is the mast, rows 5 and 6 are the boat. Count of coloured squares: 1, 2, 3, 1, 6, 4, so 17 in all.) 3. **Write the code (6 min).** Give each pair the coloured tree picture. Together they write the six row codes in the code boxes. Early finishers swap their codes with another pair and colour a blank grid from the other pair's codes to check.

(Answer for the tree: row 1: 0 0 1 1 0 0; row 2: 0 1 1 1 1 0; row 3: 1 1 1 1 1 1; rows 4, 5 and 6: 0 0 1 1 0 0.)

✓ Discussion questions (with answers) (4 min)

- "What does a 0 tell you? What does a 1 tell you?" (0: leave the square blank. 1: colour it in.)
- "Rows 4, 5 and 6 of the tree have the same code. Why?" (The trunk is the same in each row, so the squares are the same.)
- "If one number in the boat code was wrong, what would happen?" (One square would be wrong in the picture; we could find it by checking each number against each square.)

✓ Closing check (with answers) (4 min)

On the board, draw a single row of six squares and write the code 1 1 0 0 1 1. Children show on their fingers how many squares get coloured, then one child colours them. (Answer: four; squares 1, 2, 5 and 6.)

Then colour squares 3 and 4 of a new row. "Show me the code with number cards, or write it." (Answer: 0 0 1 1 0 0.)

Vocabulary

- **pixel:** one small square of a picture on a screen ("picture element").
- **code:** numbers or symbols that stand for something.
- **row:** a line of squares going across.

Differentiation

- **More support:** use only rows 1 to 4 of the boat, or a 4 by 4 grid; the reader points to each number while the partner colours.
- **More challenge:** children design their own 6 by 6 picture, write its codes and give the codes (not the picture) to a partner.

⚠ Safety

- Usual classroom care with pencils. No screens are needed.

Teacher notes

- Children often read the code right to left, or skip a 0. Ask them to put a finger under each number as they go and move one square across for each number, even when it is 0.
- The idea and the word "pixel" follow CS Unplugged's "Colour by numbers" lesson (see Sources in the unit overview); the boat and tree pictures are the author's own.
- Don't call the rows "binary numbers". At this stage each 0 or 1 is just one square.

Lesson 2: Colour-number keys

Unit overview - Previous: Lesson 1: Picture codes - Next: Lesson 3: Break it down

Back to the grade 2 year plan

Goal: Children use a key to decode a picture made of numbers 0 to 3, change one number and see the picture change, and find out that the same numbers with a different key make a different picture.

Length: about 30 minutes.

Materials

- The large 6 by 6 grid on the board, and key A drawn large beside it.
- Picture grids (6 by 6), two per child.
- Crayons: red, green, blue, and yellow (for key B).
- Key cards A and B, one of each per pair. Each number has its colour word and a coloured dot:
  - **Key A:** 0 = white (leave blank), 1 = red, 2 = green, 3 = blue.
  - **Key B:** 0 = white (leave blank), 1 = yellow, 2 = blue, 3 = green.
- Code cards for the Little Hill picture (below), one per pair.
- Optional, for blind or low-vision children: a raised-line grid, felt squares in different textures for each number, and the key and codes in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

Recap lesson 1 with one row on the board: 0 1 1 0 0 1. "Last time 0 meant blank and 1 meant coloured. What if we want more colours?" Show key A. "A **key** tells us what each number means. The numbers are the code; the key tells us how to read it."

Model one square: the number 3 means blue, so the teacher colours the square blue **and writes a small 3 in the corner**. "We always write the number in the square, so anyone can check it, even if colours are hard to tell apart."

✎ Main activity (17 min)

1. **Decode Little Hill with key A (9 min).** Pairs take the code card and key A. One reads, one colours; swap readers each row. Each coloured square keeps its small number.

Row	Code
1	3 0 0 3 3 3
2	3 3 3 3 3 3
3	3 3 1 3 3 3

Row	Code
4	3 3 2 2 3 3
5	3 2 2 2 2 3
6	2 2 2 2 2 2

(Answer with key A: a green hill at the bottom, a blue sky, a small white cloud (two blank squares in row 1) and a red flower on top of the hill (row 3, square 3). Counts: 2 white, 1 red, 12 green, 21 blue; 36 squares in all.) 2. **Change one number (3 min).** "Mayor Hen wants a bigger cloud." On the board, change row 1 to 3 0 0 0 3 3. Pairs find the one square that changes on their picture and leave it blank (they can rub out or cover it with a white sticky square). (Answer: row 1, square 4 becomes white; the cloud is now three squares.) 3. **Same numbers, new key (5 min).** Each pair colours rows 4, 5 and 6 again on a fresh grid, using **key B**. Then they put the two pictures side by side. (Answer with key B: the hill turns blue and the sky around it turns green. The numbers did not change; only the key did.)

✓ Discussion questions (with answers) (4 min)

- "The codes were the same. Why do the pictures look different?" (We used a different key, so the same number meant a different colour.)
- "If I send you only the numbers, what else do you need to make my picture?" (The key.)
- "Mayor Hen changed one number. How many squares changed?" (One.)

✓ Closing check (with answers) (4 min)

On the board write the code 1 3 3 and show key A. Children hold up the crayons for the three squares in order. (Answer: red, blue, blue.) Then show key B for the same code. (Answer: yellow, green, green.)

Vocabulary

- **key:** a list that says what each number or symbol means.
- **code:** numbers or symbols that stand for something.
- **pixel:** one small square of a picture (from lesson 1).

Differentiation

- **More support:** use only rows 4 to 6 of the picture with key A; or a key with just two colours.
- **More challenge:** children make their own key C (for example 1 = brown, 2 = yellow, 3 = black) and colour rows 4 to 6 with it, then explain what the picture now looks like.

⚠ Safety

- Usual classroom care with crayons and scissors (if cutting white sticky squares).

Teacher notes

- Keep the small number in each square. It is how children with colour-vision differences check their work, and it shows that the numbers are what is really stored.
- Real screens make colours by mixing red, green and blue light. That is not taught here; keys A and B are made-up keys for a game.
- The Little Hill picture and both keys are the author's own examples, not from a source.

Lesson 3: Break it down

Unit overview - Previous: Lesson 2: Colour-number keys - Next: Lesson 4: Plan before you run

Back to the grade 2 year plan

Goal: Children break a job into small, precise steps in the right order, and see why a vague step doesn't work for a robot.

Length: about 30 minutes.

Materials

- Teacher props for "Duck plants a seed": a paper seed packet with a paper-circle seed inside, a paper pot filled with scrunched brown paper as "soil", and an empty toy watering can.
- The "Duck plants a seed" picture strip (six pictures, no words), one large copy.
- Step cards for "Mouse puts a book back" (five cards, words and a small picture on each), one set per pair.
- Large soft building blocks in three colours (red, blue, green), one of each per pair; blocks too big to swallow.
- Blank step cards (strips of paper) and pencils.
- A book or folder per pair to stand up as a screen.
- Optional, for blind or low-vision children: blocks with different textures or shapes for each colour, and step cards read aloud or in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

"Today I am a robot. I do **exactly** what you say, and nothing else." Hold up the props and say: "Duck wants to plant a seed. Tell me what to do." Take the first instruction literally. For "Plant the seed!", put the whole closed packet on top of the pot. For "Put it in the pot!", ask "Put what in the pot?" and wait.

"A big job like 'plant a seed' is too big for one step. We need to **break it down** into small steps, in the right order, and each step must be **precise**: it says exactly what to do."

✎ Main activity (17 min)

1. **Duck's steps together (5 min).** Show the six-picture strip. Children say or point to a step for each picture; the teacher writes it on the board and does it, robot-style, with the props:
    1. Open the packet.
    2. Take out one seed.
    3. Make a hole in the soil with one finger.
    4. Put the seed in the hole.
    5. Cover the seed with soil.

6. Water the pot.

(Answer: those six steps, in that order. If children swap steps 4 and 5, the robot covers the hole first and then drops the seed on top; ask "What went wrong?") 2. **Mouse puts a book back (5 min).** Pairs get the five step cards in a muddle and put them in order: 1. Pick up the book. 2. Walk to the shelf. 3. Find the gap where the book goes. 4. Turn the book so its spine faces out. 5. Slide the book into the gap.

(Answer: that order. Steps 3 and 4 can swap and still work; the rest can't. Ask: "Which steps could swap? Which couldn't?") 3. **Tower steps (7 min).** One child is the **planner**, the other the **builder**. The planner secretly decides a tower order (for example red at the bottom, then blue, then green on top) and writes or draws three precise steps on blank cards, such as "Put the red block on the table", "Put the blue block on the red block", "Put the green block on the blue block". The planner sits behind the book screen and passes one card at a time; the builder does exactly what the card says. Then they lift the screen and compare. Swap jobs.

✓ Discussion questions (with answers) (4 min)

- "Why didn't 'Plant the seed!' work for the robot?" (It was too big and not precise; the robot didn't know the small steps.)
- "What happens if the steps are in the wrong order?" (The job can go wrong, like covering the hole before the seed is in.)
- "Was 'Put a block on' a precise step?" (No. Which block, and on what? A precise step names the block and where it goes.)

✓ Closing check (with answers) (4 min)

Read out: "Put the blue block on the table. Put a block on it." Children show a thumbs up if it is precise and thumbs down if not, then say or show how to fix it. (Answer: thumbs down for the second step; for example "Put the red block on the blue block".)

Vocabulary

- **break down (decompose):** split a big job into small steps.
- **precise:** says exactly what to do, with nothing to guess.
- **step:** one small instruction.

Differentiation

- **More support:** the Mouse cards with only four steps (leave out step 3), or a two-block tower.
- **More challenge:** a four-block tower, or a step that uses "repeat" (for example "Repeat 2 times: put a block on top").

⚠ Safety

- Props are paper; no real soil, seeds or water, so there are no allergy or spill risks.

- Blocks are too big to swallow; builders keep towers low (three or four blocks) so nothing falls far.

Teacher notes

- **For the one-to-one check in lesson 5:** give the child the Mouse cards 1, 2, 4 and 5 in a muddle, without card 3. (Answer: order 1, 2, 4, 5, and the missing step is "Find the gap where the book goes", or any step that says where on the shelf the book goes. An order with the turning step anywhere before sliding the book in also counts.)
- Children sometimes think "precise" means "long". A short step can be precise if it names what and where.
- Being the robot is fun because it is silly; keep the literal acting kind, and never use a child's own instructions to make fun of them.
- Duck, Mouse and the steps are the author's own examples.

Lesson 4: Plan before you run

Unit overview - Previous: Lesson 3: Break it down - Next: Lesson 5: Plan, code, swap and check

Back to the grade 2 year plan

Goal: Children fill in a plan (goal, start, steps, what should happen) before running a floor-grid program, then run it and compare what happened with the plan.

Length: about 30 minutes.

Materials

- The 4 by 4 floor grid with the marked front edge, start card, a "Hen's house" card and a toy letter (folded paper), from the grade 1 robot unit.
- Arrow cards (forward, turn left, turn right) and a program strip per group of three.
- Job badges: programmer, tester, bot (one set per group).
- Planning sheets, one per group: boxes for **Goal**, **Start**, **Steps** and **What should happen**, and a box with a tick and a cross for **Did it happen?**
- A large planning sheet on the board.
- Table grids (4 by 4 on paper) with a toy or counter, for groups not on the floor grid.
- Optional, for blind or low-vision children: tactile arrow cards, a raised-edge table grid and planning-sheet boxes with symbol cards.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

"Duck has a letter for Hen. Before the robot moves, programmers make a **plan**." Fill in the large planning sheet with the class:

- **Goal:** take the letter to Hen's house.
- **Start:** column 1, row 1, facing the back of the grid (show the start card and point).
- **Steps:** (left empty for now).
- **What should happen:** "The robot stops on Hen's house holding the letter."

"Writing what should happen **before** we run the program means we can check it afterwards."

✎ Main activity (17 min)

1. **A plan with a bug (5 min).** Put the start card on (column 1, row 1) and Hen's house on (column 3, row 3). Write these steps on the board plan: forward, forward, turn left, forward, forward. A volunteer bot holds the letter and the tester shows each card. Stop the bot as soon as a step would take it off the grid. "Did it happen?" (Answer: no. After forward, forward the bot is on (column 1, row 3). Turn left points it off the left edge, so the next forward leaves the grid.) "Which step is different from our plan?" (Turn left.) Change it to turn right and run again. (Answer: forward, forward, turn right, forward, forward ends on (column 3, row 3), Hen's house. Tick "Did it happen?")

2. **Groups plan and run (10 min).** Groups of three. Start card on (column 4, row 1), bot facing the back; Hen's house on (column 2, row 4). The group fills in Goal, Start and What should happen **first**, then lays out the steps, then runs. The tester ticks or crosses "Did it happen?". If it is a cross, they find the first step that went wrong, fix it and run again. Swap badges after each run.

   (One answer: forward, forward, forward, turn left, forward, forward. Other answers that reach Hen's house also count, for example turn left, forward, forward, turn right, forward, forward, forward.) 3. **Share (2 min).** One group shows its plan and says whether it matched.

✓ Discussion questions (with answers) (4 min)

- "Why do we write 'what should happen' before we run the program?" (So we can check if the program did what we wanted, and spot a bug.)
- "Two groups had different steps but both reached Hen's house. Were both right?" (Yes. Different programs can reach the same goal.)
- "What do we do if 'Did it happen?' is a cross?" (Find the first step that went wrong, fix it and run again.)

✓ Closing check (with answers) (4 min)

On the board: Goal "reach the star". Start (column 2, row 1), facing the back. Steps: forward, forward. What should happen: "stops on (column 2, row 3)". Show the star on (column 2, row 3). Children show tick or cross for "Will it happen?" (Answer: tick. Two forwards from row 1 reach row 3.)

Vocabulary

- **plan:** what we write before we start: the goal, the steps and what should happen.
- **goal:** what the program is for.
- **bug:** a step that makes the program do something we didn't plan.

Differentiation

- **More support:** routes with no turns (Hen's house straight ahead), and a planning sheet with only Goal and What should happen.
- **More challenge:** add a "puddle" card the robot must not step on, for example on (column 4, row 3) in the group route, and plan a route around it. (One answer: forward, turn left, forward, turn right, forward, forward, turn left, forward. It goes round the puddle and ends on (column 2, row 4).)

⚠ Safety

- The bot walks slowly, one square per card, and nobody else stands on the grid during a run.

- Being the bot is never required; a toy on a table grid counts.

Teacher notes

- "Column" and "row" names are for the adult. Children use the start card, the Hen's house card and pointing.
- A common error is turning **and** stepping on a turn card. Remind children that turn cards turn on the spot with no step, as in grade 1.
- Right and left are from the robot's view. If the bot faces the front edge, the cards look reversed to the class; the "stand behind the robot" trick from grade 1 helps.
- Duck, Hen and the routes are the author's own examples; every route was checked with a small script.

Lesson 5: Plan, code, swap and check

Unit overview - Previous: Lesson 4: Plan before you run

Back to the grade 2 year plan

Goal: Pairs plan a small picture and its code, swap codes with another pair, decode, and compare the result with their plan. The one-to-one check for the unit happens this week.

Length: about 30 minutes, plus about 3 minutes per child for the one-to-one check across the week.

Materials

- Small grids (4 by 4), two per child.
- Picture-plan sheets, one per pair: a box for **Goal** ("Our picture is a ..."), a 4 by 4 grid for **What should happen** (the planned picture), a blank key card, and four row-code boxes.
- Crayons: red, green, blue, yellow; erasable pencils.
- Key cards A and B from lesson 2, for pairs who want to use them.
- For the one-to-one check: a 6-square row strip, number cards 0 to 3, key cards A and B, the "Mouse puts a book back" cards 1, 2, 4 and 5, a finished planning sheet, a table grid and a toy.
- Optional, for blind or low-vision children: raised-line small grids, felt squares, and number cards in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

Show a finished picture-plan sheet made by the teacher: Goal "a red heart"; key 0 = white, 1 = red; a planned 4 by 4 picture; and the four row codes:

Row	Code
1	1 0 0 1
2	1 1 1 1
3	1 1 1 1
4	0 1 1 0

"We plan the picture **first**, then write the code. Then another pair gets **only the code and the key**. If their picture matches our plan, our code worked."

✎ Main activity (17 min)

1. **Plan and code (8 min).** Each pair writes its goal, makes a key (two or three colours, or uses key A), draws the planned picture on the plan grid, and writes the four row codes. They keep the planned picture hidden (fold the sheet).

2. **Swap and decode (5 min).** Pairs swap only the key and the codes with another pair, who colour a blank small grid, writing the small number in each square.
3. **Compare and fix (4 min).** The decoded grid goes back to the planners, who unfold their plan and compare square by square. If a square doesn't match, together they find out whether the code had a wrong number or the decoder made a slip, and fix it.

✓ Discussion questions (with answers) (4 min)

- "Did the decoded picture match your plan? How did you check?" (Children compare square by square; it matches if every square is the same.)
- "If a square was wrong, how did you find out why?" (Look at the code number for that square: if the number is wrong, the code has a bug; if the number is right, the decoder made a slip.)
- "Why did we draw the plan before writing the code?" (So we know what should happen and can check the result.)

✓ Closing check (with answers) (4 min)

On the board, show the teacher's heart plan and a decoded picture where row 4 is 1 1 1 0. "Does it match?" (Answer: no.) "Which square is different, and what should its number be?" (Answer: row 4, square 1; it should be 0, white.)

The one-to-one check (about 3 minutes each child, across the week)

Use the five parts in the Assessment section of the unit overview. Extra answers for part 5:

- **Plan:** Goal "reach Hen's house". Start (column 1, row 1), facing away from the child. Steps: forward, forward, turn right, forward. What should happen: "stops on Hen's house on (column 2, row 2)".
- **Run:** move the toy one card at a time. (Answer: it stops on (column 2, row 3), one square past Hen's house, so the result did **not** match the plan. A child who also says there is one forward too many before the turn shows extra understanding, but this is not needed for **yes**.)

Record **yes**, **with help** or **not yet** for each part. Don't record which picture or key a child chose.

Vocabulary

- **plan, goal, key, code, pixel, bug** (from lessons 1 to 4).
- **match:** the result is the same as the plan.

Differentiation

- **More support:** two-colour code only (0 and 1), or a 3 by 3 grid; the teacher writes the codes as the child points.
- **More challenge:** a 6 by 6 picture, or a picture that uses a key with four colours; or write the code for a row with a repeat, such as "1 1 1 1" written as "four 1s".

⚠ Safety

- Usual classroom care with pencils and crayons.

Teacher notes

- Children often blame the decoder first. Model checking the code number before deciding who made the slip, and keep it kind: a bug is normal, and finding it is the skill.
- Planned pictures should be things, patterns or shapes, not people in the class.
- The heart picture and the check examples are the author's own; the route was checked with a small script.

Grade 2 Computing: Same Data, Different Pictures (5 lessons, about 3 weeks)

> **Unreviewed AI draft.** Written by Claude Opus 5.5 (AI Village), 11 November 2026. No educator has reviewed or classroom-tested this unit. It is not aligned to any official standard. Every count, total and comparison in the answer keys was checked with a small computer script, but not with children. No device is needed; if your class does use devices, check logins, accounts and online spaces against your school's rules.
>
> Text-checked by three other agents (Gemini 3.8 Flash, Gemini 4 Carbon and Gemini 4 Argon); see Known gaps.

Back to the grade 2 year plan - Term 3. Builds on grade 1 Picture charts and keeping information private, grade 2 term 2 Codes and plans, and the bar graph in grade 2 mathematics term 3 Measuring, time, money and shapes (lesson 5).

This unit is "unplugged": cards, squared paper, sticky notes and pretend login cards. No device is needed.

Big idea

Computers keep lots of **data** (facts such as counts and answers). The same data can be shown in different ways: a tally, a table, a picture graph or a bar graph. Each way is good for a different job, and they must all tell the same story. A chart can help us spot a **pattern** and make a **prediction**, which is a careful guess, not a promise. When we work with others online we are kind, helpful and truthful, and we keep our login private and log off when we finish.

Outcomes

By the end of the unit, most children can:

1. Turn a set of picture vote cards (up to four kinds, up to 20 cards) into a tally chart in bundles of five, and then into a table of numbers.
2. Show the same data as a picture graph and as a bar graph with a numbered scale, and read the bar graph to say how many, which is most and fewest, how many altogether and how many more.
3. Say which way of showing the data helps most with a given question, and spot and fix a chart that is not fair (a bar that does not start on the line, squares of different sizes, or missing numbers or labels).
4. Spot a pattern in a bar graph and use it to make a prediction, saying that it is a guess, not a promise.
5. Choose a kind, helpful and true comment for someone's work on a pretend class message board, and say what to do if a message is unkind or asks for private information (do not reply; tell a trusted grown-up).
6. Say that login information is kept private (not shared even with friends), and put the steps for logging off a shared device in order.

Prior knowledge

- From grade 1 computing term 3: sorting picture cards, picture charts with one picture per box, tally marks, a pattern and prediction, what is private, picture passwords and logging off. This unit does **not** re-teach those; it adds a table, a numbered scale, choosing a picture for a question, unfair charts, kind comments online and shared devices.
- From grade 2 mathematics term 3 (lesson 5): a bar graph with up to four categories, one square for each item. From grade 2 science term 3 (lesson 2): reading a class bar graph.
- From grade 2 computing term 2: numbers and symbols can stand for information; a key says what they mean.
- Counting to 20 and reading numbers to 20.

Lessons

Lesson	Title	Time	Core idea
1	From vote cards to a table	30 min	Tally the vote cards in fives, then write the totals in a table.
2	Picture graph and bar graph	30 min	The same numbers as a picture graph and a bar graph with a scale; answering questions.
3	Patterns, predictions and unfair charts	30 min	The umbrella-stand bar graph shows a pattern; three charts that are not fair.
4	Kind and careful online	30 min	A paper "class message board": kind, helpful, true comments; what to do about unkind or nosy messages.
5	Shared devices and logging off	30 min	Duck's story; keeping login cards private; logging off in order; one-to-one check.

The data in this unit

All the data here is made up for this unit (not from a source).

- **Story-time vote (lessons 1, 2, 4 and the check).** The 20 children in Little Hill School's class each chose one book for story time, with no names on the cards: Duck's Picnic 6, Bear's Picnic 4, Goat and the Kite 7, Hen's Hat 3 (20 altogether). Each vote card shows the book's animal and the first letter of its title.
- **Umbrella stand (lesson 3).** The number of umbrellas in the Little Hill School umbrella stand at 9 o'clock each school day for two weeks, with a weather symbol under each day: week 1: Monday sunny 2, Tuesday rainy 9, Wednesday sunny 1, Thursday rainy 8, Friday cloudy 4; week 2: Monday rainy 9, Tuesday sunny 2, Wednesday cloudy 4, Thursday sunny 1, Friday rainy 10.
- **Bar graphs** stand up from a **start line**, use squares all the same size (2 cm squared paper), have a label under each bar and a **scale** of numbers up the left side (0 at the start line, then 1, 2, 3 and so on, one number for each square).

Materials

Whole class unless noted.

- Story-time vote cards (20 picture cards, the counts above), one set per pair, plus one large set for the board.

- Tally sheets with four rows (one per book, with the animal picture), one per pair; table sheets with two columns (Book, Number), one per pair; large copies of both for the board.
- Squared paper (2 cm squares) with a start line and a 0 to 10 scale printed up the left side, two sheets per child; small picture stickers or drawn pictures for the picture graph; card squares and glue for raised bars.
- Crayons, erasable pencils and a ruler per pair; small whiteboards and pens (or paper); craft sticks for tallies; counters (about 10 per pair).
- The umbrella-stand bar graph (lesson 3), one per pair and one large copy, with weather symbols under each bar.
- Three "Is it fair?" chart cards (lesson 3), one set per pair, and an "a lot" card and an "a few" card per pair.
- Question cards for lessons 2 and 3 (printed in the lesson answer keys).
- Lesson 4: a large paper "class message board" (a sheet of card with a title box), sticky notes, comment cards (printed in lesson 4), sentence-starter strips and picture symbols, and the "Stop, think, tell" poster.
- Lesson 5: a pretend tablet made of card (a screen with a "Log in" box, a "Log off" button and an "Are you sure?" box), Duck's story cards, a sheet of ten small pictures per child and blank small cards, pretend picture-login cards (one per child, in an envelope with the child's name on the outside only), and log-off step cards.
- For the one-to-one check: a bag of 8 counters (or 8 vote cards), a pencil and paper for tallies, the check bar graph, two small tables, the umbrella-stand bar graph, and comment and login choice cards (printed in lesson 5).
- Optional, for blind or low-vision children: tallies made with craft sticks laid on felt, raised bar graphs with card squares and a raised start line, vote cards with a different texture for each book, and cards and tables in large print or braille.
- Ear defenders and a quiet spot.

⚠ Safety and access

- **No names on data.** Votes are anonymous; nobody has to say which book they chose. The umbrella data is made up, and no child's belongings are counted.
- **Private stays private.** Children never say, write or show a real password, and are never asked for one. The login cards in lesson 5 are pretend, made in class, and go home or in the bin at the end.
- **Online feelings are handled gently.** Lesson 4 uses made-up messages about made-up characters. Children are never asked whether anyone has been unkind to them online. If a child tells you something worrying, follow your school's safeguarding procedure.
- **Colour is never the only clue.** Bars carry labels and the animal picture, not only a colour. The weather symbols use shapes (sun, cloud, raindrops) and the word.
- **Speaking is optional.** Children can answer by pointing, placing cards or counters, a tick card or their usual communication system.
- **Blind or low-vision children:** raised bars, craft-stick tallies and textured vote cards; the child counts squares or sticks by touch and reads the scale by feeling the raised start line and counting up.
- **Deaf or hard-of-hearing children:** every question is on a card or the board; the teacher points while reading.
- **Fine motor:** a child can stick card squares, place counters or point while a partner draws. Neat drawing is never assessed.

Assessment

Lesson 5 ends with a short one-to-one check (about 3 to 4 minutes each child; spread over the week). Record each part as **yes**, **with help** or **not yet**:

1. **Tally** (outcome 1): the child counts 8 counters and makes a tally. (Answer: one bundle of five and three more lines.)
2. **Read a bar graph with a scale** (outcome 2): the check bar graph shows the story-time vote (Duck's Picnic 6, Bear's Picnic 4, Goat and the Kite 7, Hen's Hat 3). "Which book got the most votes?" (Goat and the Kite.) "How many more votes did Goat and the Kite get than Bear's Picnic?" (3.)
3. **Same data** (outcome 2): two small tables are shown next to the check bar graph. Table 1: 6, 4, 7, 3. Table 2: 6, 7, 4, 3. "Which table tells the same story as the bar graph?" (Table 1.)
4. **Which picture, and is it fair?** (outcome 3): "I want to compare the books quickly. Which helps more, the table or the bar graph?" (The bar graph; a child who says the table is good for finding an exact number also shows the idea.) Then show "Is it fair?" chart card 1 (lesson 3). "What is wrong?" (Hen's bar does not start on the line.)
5. **Pattern** (outcome 4): on the umbrella-stand bar graph: "The forecast says next Monday will be rainy. Will there be a lot of umbrellas or a few?" (A lot; yes if the child points to the rainy days or says rainy days have the most. A child who adds "but it's a guess" shows the whole idea.)
6. **Online and logins** (outcomes 5 and 6): comment choice card A and login choice card B (printed in lesson 5). (Answers in lesson 5.)

Doing it in their own way counts as **yes**. "Not yet" is normal for some children; data and online safety return in grade 3.

Sources

- Computer Science Teachers Association, CSTA K-12 Computer Science Standards (2017), progression chart: https://csteachers.org/wp-content/uploads/2025/06/2017-csta-k-12-standards-progression-chart.pdf (read 10 November 2026). Level 1A (ages 5-7) lines, quoted exactly; the unit is not a full alignment:
    - 1A-DA-06: "Collect and present the same data in various visual formats." (lessons 1, 2 and 3)
    - 1A-DA-07: "Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions." (lesson 3)
    - 1A-IC-17: "Work respectfully and responsibly with others online." (lesson 4, on paper)
    - 1A-IC-18: "Keep login information private, and log off of devices appropriately." (lesson 5, on a pretend device)
- Common Core State Standards for Mathematics, grade 2, Measurement and Data, 2.MD.D.10: "Draw a picture graph and a bar graph (with single-unit scale) to represent a data set with up to four categories. Solve simple put-together, take-apart, and compare problems using information presented in a bar graph." https://web.archive.org/web/2025/https://www.thecorestandards.org/Math/Content/2/MD/ (read 10 November 2026; a footnote marker after "problems" is left out). This is why every graph here has up to four bars and one square for each item.

Known gaps and cautions

- Three other agents (Gemini 3.8 Flash, Gemini 4 Carbon and Gemini 4 Argon, all at 11a3d2a) text-checked this unit on 2026-11-11 and reported no errors; Carbon suggested one optional wording fix in lesson 1, made in 6897a8c and re-checked by Argon. Those are text checks by AI agents, not a teacher review or classroom test.
- **The data, stories, comment cards, chart cards and pretend tablet are the author's own examples**, not from a source.
- **CSTA 1A-DA-06, 1A-DA-07 and 1A-IC-18 were also used in the grade 1 term 3 unit.** This unit repeats them a step harder (a table, a numbered scale, four categories, unfair charts, shared devices); it has not been checked against any published grade 2 computing scheme.
- **"Online" is on paper.** Lesson 4's message board and lesson 5's tablet are pretend. Children who use real devices at school will need the school's own rules taught as well.
- **Lesson 3's umbrella pattern is a comparison, not proof** that rain causes umbrellas; the lesson says so in child-friendly words, and treats predictions as guesses.
- **Mathematics overlap.** Grade 2 mathematics term 3 lesson 5 also makes a bar graph. This unit adds the numbered scale, the table and the "same data" idea; if the classes are close together, lesson 2 can be shortened.
- **Other level 1A lines** (for example 1A-DA-05 storing and finding data on a device, and 1A-NI-04 strong passwords) are not taught here.

Lesson 1: From vote cards to a table

Unit overview - Next: Lesson 2: Picture graph and bar graph

Back to the grade 2 year plan

Goal: Children sort 20 picture vote cards, make a tally chart in bundles of five, and write the totals in a table.

Length: about 30 minutes.

Materials

- Story-time vote cards (20 picture cards: Duck's Picnic 6, Bear's Picnic 4, Goat and the Kite 7, Hen's Hat 3), one set per pair, plus one large set for the board.
- Tally sheets with four rows (one per book, with the animal picture), one per pair, and a large copy for the board.
- Table sheets with two columns (Book, Number), one per pair, and a large copy for the board.
- Erasable pencils, and small whiteboards and pens (or paper) for the closing check.
- Craft sticks (about 10 per child who wants them) for laying tallies.
- Optional, for blind or low-vision children: craft sticks and a felt mat for tallies, textured vote cards, and tally and table sheets in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

"Little Hill School's class voted for a story-time book. Each child put one card in a box, with no names. Here are the cards. Computers keep facts like these. Facts we collect are called **data**."

Hold up the four books' cards. "Lots of cards in a muddle are hard to read. This week we will show the **same data** in different ways, and see what each way is good for."

✎ Main activity (17 min)

1. **Teacher models a tally (4 min).** Turn over the first eight board cards one at a time. For each card, draw one line in that book's row. When a row gets its fifth mark, draw it across the other four: "Five makes a bundle. Bundles are quick to count." (Answer for the first eight board cards, laid in this order: Goat, Duck, Goat, Hen, Goat, Bear, Goat, Goat: Goat has one bundle of five; Duck, Hen and Bear have one line each.)

2. **Pairs tally all 20 cards (8 min).** Each pair shuffles its set. One child turns over a card and says or shows the book; the other draws the line. Swap jobs after 10 cards. Then pairs count each row by bundles and lines.

(Answer: Duck's Picnic: one bundle and 1 line, 6. Bear's Picnic: 4 lines, 4. Goat and the Kite: one bundle and 2 lines, 7. Hen's Hat: 3 lines, 3.) 3. **Tally to table (5 min).** Pairs write each total in the Number column of the table. Then they add a last row, "Altogether", and work it out. (Answer: 6, 4, 7, 3; altogether 20.) Early finishers check: "Do we have 20 cards? Does the table say 20?"

✓ Discussion questions (with answers) (4 min)

- "The cards, the tally and the table all show the same data. How do we know?" (They all say Duck's Picnic 6, Bear's Picnic 4, Goat and the Kite 7, Hen's Hat 3, and 20 altogether.)
- "Which is quickest for finding how many voted for Hen's Hat?" (The table: the number is written there. In the tally you count the lines; with the cards you count them all.)
- "Why do we draw a bundle of five?" (Bundles are quicker to count than lots of single lines.)

✓ Closing check (with answers) (4 min)

Show a tally row on the board: one bundle and 4 lines. "Show me how many with your fingers or number cards." (Answer: 9.)

Then: "Draw a tally for 7." Children draw on whiteboards or lay craft sticks. (Answer: one bundle of five and 2 lines.)

Vocabulary

- **data:** facts we collect, such as counts or answers.
- **tally:** a line for each thing counted, in bundles of five.
- **table:** rows and columns with words and numbers.
- **altogether:** all of them added up.

Differentiation

- **More support:** use only three books (leave out Hen's Hat, 17 cards) and give tally sheets with two rows of five dotted boxes for each book, one line in each box (Goat and the Kite needs 7 boxes).
- **More challenge:** "How many more voted for Goat and the Kite than for Duck's Picnic?" (1.) Then pairs make up a fifth book, give it some votes, and change the table and the Altogether row.

⚠ Safety

- Usual classroom care with pencils. No screens are needed. Nobody says which book they would choose; the votes are Little Hill's.

Teacher notes

- Children sometimes draw the fifth line as a separate line, or draw it across after six. Count aloud "one, two, three, four, and five goes across".
- Some children count a bundle as 4, because they count the four upright lines. Ask them to touch every line, including the one across.
- Children who tallied with craft sticks can leave their tally on the mat for lesson 2, or use the board table.
- The vote cards and books are the author's own examples; "Duck's Picnic", "Bear's Picnic" and "Goat and the Kite" are stories from earlier units, and "Hen's Hat" is just a title on a card.

Lesson 2: Picture graph and bar graph

Unit overview - Previous: Lesson 1: From vote cards to a table - Next: Lesson 3: Patterns, predictions and unfair charts

Back to the grade 2 year plan

Goal: Children show the story-time votes as a picture graph and as a bar graph with a numbered scale, check that both tell the same story as the table, and answer questions from the bar graph.

Length: about 30 minutes.

Materials

- The class table from lesson 1 (Duck's Picnic 6, Bear's Picnic 4, Goat and the Kite 7, Hen's Hat 3) on the board, and each pair's own table.
- Squared paper (2 cm squares) with a start line and a 0 to 10 scale printed up the left side, two sheets per child.
- Small picture stickers or drawn pictures for the picture graph.
- Card squares and glue for raised bars (for any child who wants them).
- Crayons, erasable pencils and a ruler per pair.
- Question cards (below), one set per pair.
- Optional, for blind or low-vision children: a raised start line, card squares in a different texture for each bar, and the scale and question cards in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

Point to the table. "Yesterday we made a table. Today we'll draw the **same data** as pictures." Show a large sheet with a start line. "Every bar starts on this **start line**. The squares are all the same size. And up the side is a **scale**: the numbers tell us how tall a bar is without counting every square."

Model one bar: Duck's Picnic, 6 squares coloured up from the start line. Run a finger from the top of the bar across to the scale. "The top of the bar is level with 6. Duck's Picnic got 6 votes."

✎ Main activity (17 min)

1. **Picture graph (5 min).** On the first sheet, each child sticks or draws one picture in each square, starting at the start line, one column for each book. They write the book's animal or first letter under each column. (Answer: columns of 6, 4, 7 and 3 pictures.)
2. **Bar graph (6 min).** On the second sheet, children colour (or stick card squares for) one bar per book, starting at the start line, then write the label under each bar. They check the top of each bar against the scale. (Answer: bar tops level with 6, 4, 7 and 3.)

3. **Question cards (6 min).** Pairs answer from their bar graph, using the scale where they can:

Card	Question	Answer
1	Which book got the most votes?	Goat and the Kite (7)
2	Which book got the fewest votes?	Hen's Hat (3)
3	How many votes did Bear's Picnic get?	4
4	How many more votes did Goat and the Kite get than Bear's Picnic?	3 (7 take away 4)
5	How many votes did Duck's Picnic and Bear's Picnic get together?	10 (6 and 4)
6	How many children voted altogether?	20 (6 + 4 + 7 + 3)

Early finishers try card 7: "If 2 more children voted for Hen's Hat, which books would it be level with?" (None: Hen's Hat would have 5, and no other book has 5. It would be one more than Bear's Picnic and one fewer than Duck's Picnic.)

✓ Discussion questions (with answers) (4 min)

- "Do the table, the picture graph and the bar graph tell the same story?" (Yes: 6, 4, 7 and 3 in each.)
- "Which helps you see the biggest and smallest fastest?" (The picture graph or the bar graph: you can see the tallest and shortest without reading numbers.)
- "How does the scale help?" (You read the number level with the top of the bar instead of counting every square.)
- "Why must every bar start on the start line?" (So the bars can be compared fairly; a bar that starts higher would look bigger than it is.)

✓ Closing check (with answers) (4 min)

Draw a bar on the board, 5 squares tall, next to the scale. "How many does this bar show? Show me." (Answer: 5.)

Then: "Goat and the Kite has 7. Hen's Hat has 3. Show me how many more Goat and the Kite has." (Answer: 4.)

Vocabulary

- **picture graph:** a chart with one picture for each thing counted.
- **bar graph:** a chart where the height of each bar shows how many.
- **scale:** the numbers up the side of a bar graph.
- **start line:** the line every bar starts from.

Differentiation

- **More support:** stick card squares and count them; use cards 1 to 3 only.

- **More challenge:** draw the same bar graph sideways, with the bars going across from a start line on the left and the scale along the bottom. Does it tell the same story? (Yes.)

⚠ Safety

- Usual classroom care with glue and scissors (if cutting card squares, use round-ended scissors). No screens are needed.

Teacher notes

- Children often read the scale number just above or below the top of the bar, or count the start line as 1. Model running a ruler from the top edge of the bar across to the scale.
- Some children colour bars different widths or leave gaps inside a bar. Point out that each square is one vote, so every square must be the same size and next to the last one.
- Grade 2 mathematics term 3 lesson 5 made bar graphs by counting squares. The new step here is reading the numbered scale and comparing the bar graph with the table and picture graph.

Lesson 3: Patterns, predictions and unfair charts

Unit overview - Previous: Lesson 2: Picture graph and bar graph - Next: Lesson 4: Kind and careful online

Back to the grade 2 year plan

Goal: Children read a bar graph with ten bars, spot a pattern, make a prediction and say it is a guess; then they find and fix three charts that are not fair.

Length: about 30 minutes.

Materials

- The umbrella-stand bar graph, one per pair and one large copy for the board: ten bars (two weeks of school days), a 0 to 10 scale, and a weather symbol and word under each bar (week 1: Monday sunny 2, Tuesday rainy 9, Wednesday sunny 1, Thursday rainy 8, Friday cloudy 4; week 2: Monday rainy 9, Tuesday sunny 2, Wednesday cloudy 4, Thursday sunny 1, Friday rainy 10).
- Three "Is it fair?" chart cards (below), one set per pair, and large copies for the board.
- The class table from lesson 1 (Duck's Picnic 6, Bear's Picnic 4, Goat and the Kite 7, Hen's Hat 3).
- Squared paper with a start line and scale, one sheet per pair, and crayons, pencils and a ruler.
- Counters (about 10 per pair) and an "a lot" card and an "a few" card per pair.
- Optional, for blind or low-vision children: raised versions of the umbrella graph and the chart cards (card squares, raised start line, tactile weather symbols), and everything in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

Show the large umbrella-stand bar graph. "Every school day at 9 o'clock, Mouse counted the umbrellas in the Little Hill School umbrella stand, for two weeks. Under each bar is the weather that morning." Read the symbols together: sun, cloud, raindrops.

"Look at the tall bars. Look at the short bars. What do you notice?"

✎ Main activity (17 min)

1. **Find the pattern (6 min).** Pairs put a counter on every rainy day, then every sunny day, then every cloudy day, and read each bar with the scale.

   (Answer: rainy days: 9, 8, 9 and 10, the tallest bars. Sunny days: 2, 1, 2 and 1, the shortest bars. Cloudy days: 4 and 4, in the middle. The pattern: on rainy days there are

lots of umbrellas; on sunny days only a few.) 2. **Make a prediction (3 min).** "The forecast says next Monday will be rainy. Will there be a lot of umbrellas or a few?" Pairs show "a lot" or "a few" with a card, then say why. (Answer: a lot, probably about 8 to 10, because every rainy day so far had 8 to 10.) Then: "Is it certain?" (No. It is a guess from the pattern. Some children might forget their umbrella, or the rain might stop early.) 3. **Is it fair? (8 min).** Give each pair the three chart cards. Each one shows the story-time votes from lesson 1, but something is wrong. Pairs compare each card with the table, say what is wrong, and draw one fixed chart on squared paper.

Card	What it shows	What is wrong (answer)
1	Hen's Hat's bar is 3 squares tall but starts 4 squares above the start line, so its top is level with 7, the same as Goat and the Kite.	The bar does not start on the start line. Hen's Hat looks as big as Goat and the Kite, but it has 3, not 7.
2	Goat and the Kite's bar has 7 squares, but they are half the height of the others, so it looks shorter than Duck's Picnic's 6.	The squares are not all the same size. Goat and the Kite got the most votes but looks smaller.
3	Four bars of the right heights, but no scale numbers and no labels under the bars.	You can't tell which bar is which book, or how many each has without counting. It needs labels and a scale.

✓ Discussion questions (with answers) (4 min)

- "What pattern did you find in the umbrella graph?" (Rainy days have lots of umbrellas; sunny days have a few.)
- "Does the graph prove that rain makes people bring umbrellas?" (Not by itself: it shows they happen together on these ten days. But it helps us make a good guess.)
- "Why might someone make an unfair chart by mistake?" (They might start a bar in the wrong place, draw squares different sizes, or forget the labels.)
- "How can we check a chart is fair?" (Every bar starts on the start line, the squares are all the same size, every bar has a label, and there is a scale.)

✓ Closing check (with answers) (4 min)

On the umbrella graph, point to week 2 Thursday. "What was the weather, and how many umbrellas?" (Answer: sunny, 1.)

Then show chart card 1 again. "Thumbs up if it is fair, thumbs down if not." (Answer: thumbs down; Hen's bar does not start on the line.)

Vocabulary

- **pattern:** something that happens again and again.
- **prediction:** a careful guess about what will happen, using a pattern.
- **fair (chart):** a chart that shows the numbers truly, so the bars can be compared.

Differentiation

- **More support:** use only week 1 (five bars) and only chart card 1.

- **More challenge:** "Mouse counted for a third week. Which days do you think will be tallest?" (Children cannot know without the weather; the best answer is "the rainy days, whichever they are".) Then make a fourth unfair chart for a partner to fix.

⚠ Safety

- Usual classroom care with pencils. No screens are needed. The umbrella data is made up; no child's belongings are counted.

Teacher notes

- Some children think a prediction must be right. Say often: "A prediction is a careful guess. If it turns out wrong, we learn something."
- Some children read week 2 Friday (10) as off the scale. Show that the scale goes up to 10.
- Card 1 is the hardest; children often say "Hen's Hat has 7". Ask them to count the squares in the bar, not where the top is.
- The umbrella data and chart cards are the author's own examples, not from a source.

Lesson 4: Kind and careful online

Unit overview - Previous: Lesson 3: Patterns, predictions and unfair charts - Next: Lesson 5: Shared devices and logging off

Back to the grade 2 year plan

Goal: Children practise kind, helpful and true comments on a pretend class message board, and say what to do if a message is unkind or asks for private information.

Length: about 30 minutes.

Materials

- A large paper "class message board": a sheet of card with the title "Little Hill class board" in a box at the top, and the children's bar graphs from lesson 2 pinned around it (only if each child agrees; a child can pin a copy of the board bar graph instead).
- Sticky notes, several per child, and pencils.
- Printed sentence-starter strips ("I like how…", "Your bars…", "Could you add…?") and picture symbols.
- Comment cards (below), one set per pair, and large copies for the board.
- The "Stop, think, tell" poster: **Stop** (don't reply straight away), **Think** (is it kind? is it private?), **Tell** (a trusted grown-up).
- Optional, for blind or low-vision children: comment cards in large print or braille, read aloud; sticky notes with a raised dot in the corner so the child knows which way up.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

"Lots of people use computers to work together: they share pictures and leave messages for each other. That's being **online**. Today our class board is a pretend online board, made of paper."

"Goat put a bar graph on the Little Hill class board. Three friends left messages." Read the first three comment cards. "Which one would help Goat? Which one might make Goat sad?"

✎ Main activity (17 min)

1. **Sort the comments (7 min).** Pairs sort the comment cards into two piles: "Kind and helpful" and "Not OK". Then they pick one "Not OK" card and say what to do.

Card	Message (for Goat's bar graph)	Answer
1	"Your bars all start on the line. Well done!"	Kind and helpful: it is true and says what is good.
2	"Your graph is rubbish."	Not OK: unkind, and it doesn't help. Don't reply; tell a grown-up if it upsets you.

Card	Message (for Goat's bar graph)	Answer
3	"I think Hen's Hat needs a label."	Kind and helpful: it is polite and helps Goat fix the chart.
4	"LOL" with a laughing face, posted by someone Goat doesn't know.	Not OK: it might be laughing at Goat, and we don't know who it is. Stop, think, tell a grown-up.
5	"Nice graph! What's your password so I can fix it for you?"	Not OK: it asks for private information. Never share a password; tell a grown-up.
6	"I counted your Duck bar. I think it has 7 squares, not 6. Can you check?"	Kind and helpful: it is polite, and asks Goat to check instead of saying "wrong".
7	"Where do you live? I want to send you a better graph."	Not OK: it asks where Goat lives, which is private. Stop, think, tell a grown-up.

(Kind and helpful: cards 1, 3 and 6. Not OK: cards 2, 4, 5 and 7.) 2. **Write a comment (7 min).** Each child chooses one bar graph on the class board (not their own) and writes or draws a sticky-note comment that is **kind, helpful and true**: one thing that is good, and, if they like, one polite idea to make it better. Children who prefer can choose a printed starter: "I like how…", "Your bars…", "Could you add…?" 3. **Read your board (3 min).** Children read the comments on their own graph (or have them read aloud). "Did any comment help you?"

✓ Discussion questions (with answers) (4 min)

- "What makes a comment kind and helpful?" (It is polite, true, and says what is good or how to make it better.)
- "What should you do if a message is unkind?" (Don't reply; tell a trusted grown-up.)
- "What should you do if a message asks for a password or where you live?" (Don't answer; stop, think and tell a trusted grown-up.)
- "Online messages can stay for a long time and lots of people can see them. How does that change what we write?" (We think before we post and only write things we'd be happy for anyone to read.)

✓ Closing check (with answers) (4 min)

Read two new messages. Children show thumbs up for "kind and helpful" and thumbs down for "not OK".

- "Your scale numbers are neat. Could you add a label under the last bar?" (Answer: thumbs up.)
- "Tell me your login and I'll give you a sticker." (Answer: thumbs down; it asks for private information.)

Vocabulary

- **online:** using computers to share and talk with other people.
- **comment:** a message about someone's work.
- **respectful:** kind and polite, treating others well.

- **private:** something we keep to ourselves and trusted grown-ups.

Differentiation

- **More support:** use cards 1, 2, 3 and 5 only; offer sentence starters and picture symbols for the sticky notes.
- **More challenge:** rewrite card 2 so it is kind and helpful. (For example: "Your bars are colourful. Could you make all the squares the same size?")

⚠ Safety

- The messages are about made-up characters. Children are never asked whether anyone has been unkind to them online, or what apps or games they use.
- A child can choose not to pin their own graph. Remove any sticky note that is unkind before children read their boards.
- If a child tells you something worrying, follow your school's safeguarding procedure.

Teacher notes

- Some children think a helpful comment must never suggest a change. Show that card 6 is kind and still points out something to check.
- Some children say "I'd reply and tell them off". Praise the wish to stand up for themselves, then explain that telling a grown-up is safer and works better.
- "Stop, think, tell" is the author's own wording, not from a source. If your school uses a different phrase for online safety, use that instead.
- The comment cards are the author's own examples, not from a source.

Lesson 5: Shared devices and logging off

Unit overview - Previous: Lesson 4: Kind and careful online

Back to the grade 2 year plan

Goal: Children say why login information stays private, even from friends, and put the steps for logging off a shared device in order. The one-to-one check takes place this week.

Length: about 30 minutes, plus the one-to-one check (about 3 to 4 minutes each child, spread over the week).

Materials

- A pretend tablet made of card: a screen with a "Log in" box, a "Log off" button and an "Are you sure?" box with "Yes" and "No".
- Duck's story cards (below), large, for the board.
- A sheet of ten small pictures per child, blank small cards, and crayons or pencils.
- Pretend picture-login cards: one per child, made in this lesson (three small pictures chosen from a sheet of ten, in a row), each in an envelope with the child's name on the outside only.
- Log-off step cards (below), one set per pair.
- For the one-to-one check: a bag of 8 counters (or 8 vote cards), a pencil and paper, the check bar graph (the story-time votes), two small tables (table 1: 6, 4, 7, 3; table 2: 6, 7, 4, 3), "Is it fair?" chart card 1 (lesson 3), the umbrella-stand bar graph (lesson 3), and choice cards A and B (below).
- Optional, for blind or low-vision children: a tablet with a raised "Log off" button, tactile picture choices for the login card, and story and step cards in large print or braille.
- Ear defenders and a quiet spot.

Steps

✎ Opening (5 min)

Tell Duck's story with the cards.

1. Duck uses the class tablet to make a bar graph, and logs in with Duck's picture login.
2. The bell rings. Duck runs out to play and leaves the tablet on, still logged in.
3. Crab picks up the tablet. It still says "Duck". Crab posts "I don't like Goat's graph" on the class board. Everyone thinks Duck wrote it.
4. Goat is sad. Duck is upset: "I didn't write that!"

"What went wrong? How could Duck have stopped it?" (Duck didn't log off, so the tablet still said Duck. Crab should not have posted as Duck; it wasn't fair or kind.)

✎ Main activity (17 min)

1. **Make a pretend login card (5 min).** Each child picks three pictures from the sheet and draws them in a row on a small card. "This is a **pretend** login. A real one would be

kept just as secret." The card goes straight into the envelope. "Who can see it?" (Only me and a trusted grown-up, such as my teacher or family.)

2. **Friends and logins (4 min).** Ask, with thumbs up or down:
    - "Your best friend asks for your login so they can play your game. Do you share it?" (Thumbs down: logins stay private even from friends. A friend could post as you by mistake, or share it with someone else.)
    - "Your teacher helps you log in when you forget." (Thumbs up: a trusted grown-up can help.)
    - "A message on the screen says 'Type your password to win a prize'." (Thumbs down: stop, think, tell a grown-up.)

3. **Log-off steps (8 min).** Pairs put the log-off step cards in order, then act them out on the card tablet: one child is the user, the other the checker who watches each step.

Card	Step (cards are given in a muddle)
A	Save your work.
B	Press "Log off".
C	When it asks "Are you sure?", press "Yes".
D	Check the screen says "Log in" again, not your name.
E	Put the tablet back where it lives.

(Answer: A, B, C, D, E. A, B and C must be in that order; a child who swaps D and E but checks the screen before walking away also shows the idea.)

"What if you forget and see a friend's name on a shared tablet?" (Don't use it as them; log them off or tell a grown-up.)

✓ Discussion questions (with answers) (4 min)

- "Why do we keep our login private, even from friends?" (So nobody else can post or change things as us.)
- "Why do we log off a shared tablet?" (So the next person can't use it as us.)
- "Who can help if we forget our login?" (A trusted grown-up, such as the teacher.)
- "What would have helped in Duck's story?" (Duck logging off; Crab not posting as Duck; Crab telling a grown-up the tablet was still logged in.)

✓ Closing check (with answers) (4 min)

Hold up step card B ("Press 'Log off'"). "Which card comes just before this one?" (Answer: A, save your work.) "Which comes just after?" (Answer: C, press "Yes".)

Then collect the login envelopes. They go home, or into the bin, today; nobody opens anyone else's.

The one-to-one check

Use the six parts in the unit overview (see Assessment there). The cards for part 6:

- **Choice card A (online):** "Goat's graph is on the class board. Which message would you leave?" Pictures with words: (1) "Your graph is silly." (2) "I like your scale. Could you add a label?" (3) "What's your password?" (Answer: 2. **Yes** if the child also says that 3 asks for private information, or that they would tell a grown-up about 1 or 3.)
- **Choice card B (logins):** "You've finished on the class tablet. What do you do?" Pictures with words: (1) Walk away. (2) Save, press "Log off", press "Yes", check it says "Log in". (3) Give your login to a friend so they can carry on. (Answer: 2.)

Vocabulary

- **login:** the name and secret (such as a password or picture code) that tells a computer who you are.
- **log off:** tell the computer you have finished, so nobody else can use it as you.
- **shared device:** a computer or tablet that lots of people use.
- **trusted grown-up:** an adult you know who keeps you safe, like your teacher or family.

Differentiation

- **More support:** use step cards A, B and C only; act out the steps with an adult first.
- **More challenge:** "Write one rule for the class tablet that would have stopped Duck's problem." (For example: "Always log off before you go.") Then put it on the class board.

⚠ Safety

- The login cards are pretend and are never used for anything. Children never say, write or show a real password, and are never asked for one.
- If your class uses real devices, teach your school's own login and log-off steps; they may differ from the cards here.
- Crab's message is about a made-up character. Children are not asked about anything that has happened to them online.

Teacher notes

- Some children think it's fine to share a login with a best friend. Use Duck's story: even a kind friend can make a mistake, and everyone would think it was you.
- Some children think closing the lid is logging off. On many devices it isn't; the screen might still show your name when someone opens it.
- Picture logins were introduced in grade 1; the new step here is shared devices, friends and the order of logging off.
- Duck's story and the step cards are the author's own examples, not from a source.