

# Lesson Plan: Spot the Pattern

---

**Year Group:** 8 | **Duration:** 50 minutes | **Topic:** Pattern Recognition

---

## 1. Overview

**Core Concept:** Pattern recognition — identifying similarities across different problems so a single generalised solution can be written once and reused many times.

### Learning Objectives:

- Identify common patterns across different real-world scenarios
- Write a generalised algorithm using placeholders that works across multiple situations
- Explain how pattern recognition reduces code duplication and increases reusability

### Key Vocabulary:

Term	Definition
Pattern recognition	Finding similarities between problems to reuse solutions
Generalisation	Writing a solution that works for many specific cases
Placeholder	A stand-in word (like [ITEMS]) used in a general algorithm
Reuse	Applying the same code or algorithm to multiple situations
Template	A general structure with blanks to fill in for each specific case

---

## 2. Before the Lesson

### Print:

- [resource-scenario-cards.md](#) — print and cut, 1 set per group of 3 (one card per person)
- [worksheet-general-algorithm.md](#) — 1 per student

**Room Setup:** Groups of exactly 3 (one card per person). Individual work first, then group comparison.

---

## 3. Timed Lesson Flow

0–5 min — Hook: What's the Pattern?

1. Write on board: **14, 7, 21, 35, 28** → "What's the pattern?" (multiples of 7)
2. Write: **cat, bat, hat, mat** → "What's the pattern?" (rhyming -at words)
3. Write: **login page, shopping cart, booking form** → "What might these share?" (all validate user input)
4. Key point: **Pattern recognition is a core human skill — and a core CS skill. Spotting patterns lets us write one solution that works everywhere.**

## 5–10 min — Introduce the Task

1. Each student in a group gets ONE scenario card. Read it alone.
2. Write a complete solution algorithm for YOUR scenario only.
3. Don't share your scenario yet — solve it independently.

## 10–25 min — Individual Work

Students write their specific algorithm. Circulate — ensure they write actual numbered steps, not descriptions.

## 25–35 min — Group Comparison

Groups share their three solutions. What steps appear in all three? Complete the comparison table on the worksheet.

## 35–45 min — Write the General Algorithm

Groups write ONE algorithm that works for all three, replacing specific words with placeholders like **[ITEMS]**, **[CAPACITY]**, **[CRITERIA]**.

## 45–50 min — Class Debrief

Groups share general algorithms. How many lines did writing one general solution save vs three specific ones? Real-world connection: sorting algorithms, form validation libraries.

---

# 4. Teacher Facilitation Notes

### What to look for:

- Students writing narrative descriptions rather than steps — prompt: *"What is the first action? What's next?"*
- Groups who find similarities but can't write the general version — prompt: *"Replace 'people' with [ITEMS]. What word covers 'exit', 'shelf', and 'subject'?"*

### Common misconceptions:

- "My scenario is unique" — push them past surface differences to structural similarities
  - A general algorithm is vaguer and therefore worse — the placeholders are filled at runtime; generality is a strength
- 

# 5. Extension Tasks

1. Find a fourth real-world scenario that fits the same general algorithm.
  2. Write the general algorithm in pseudocode with FOR EACH and IF syntax.
  3. Research: what is a sorting algorithm? Why does the same algorithm sort names, numbers, and dates?
-

## 6. Key Takeaway

**Pattern recognition means solving a problem once and reusing the solution. That's how libraries, frameworks, and standard algorithms work — one generalised solution, applied everywhere.**