

Lesson Plan: Abstract Map

Year Group: 8 | **Duration:** 50 minutes | **Topic:** Abstraction

1. Overview

Core Concept: Abstraction — removing unnecessary detail to focus on what is relevant for a specific purpose.

Learning Objectives:

- Define abstraction and explain its purpose
- Compare different representations of the same thing and explain what each keeps and removes
- Design an abstracted representation for a specific purpose and justify choices
- Relate abstraction to how programmers hide complexity

Key Vocabulary:

Term	Definition
Abstraction	Removing unnecessary detail to focus on what matters for a specific purpose
Representation	A simplified model of something real
Relevant	Useful for the purpose at hand
Irrelevant	Not needed for the purpose at hand
Model	A simplified version of something complex

2. Before the Lesson

Print:

- [worksheet-abstraction.md](#) — 1 per student

Optional for display:

- Satellite view of a local area (Google Maps satellite)
 - A transit/tube/bus map of the same area
 - The school's fire evacuation map
-

3. Timed Lesson Flow

0–5 min — Hook: Two Maps

1. Show (or describe) two views of the same place: a satellite photo and a simple map.
2. "Which would you use to find a classroom? Which to check if the roof needs repair? Why?"
3. Both show the same place — but they serve different purposes, so they include different details.

5–12 min — Introduce Abstraction

1. Abstraction = keeping only what matters, removing everything else.
2. Different purposes → different abstractions. No single "correct" level of detail.
3. Example: a transit map doesn't show accurate distances (irrelevant to passengers) but shows connections and line colours (essential to passengers).

12–20 min — Guided Comparison (Worksheet Part 1)

Students complete the comparison table — analysing satellite photo vs transit map.

20–35 min — Design Challenge (Worksheet Part 2)

1. Each student chooses a purpose: Fire Evacuation, New Student Tour, or Delivery Driver.
2. They draw an abstracted map of the school that serves ONLY that purpose.
3. Must justify every inclusion and exclusion.

35–43 min — Compare Across Purposes

Students pair with someone who chose a different purpose. How do their maps differ?

43–48 min — CS Connection

Where do programmers use abstraction? Function names (hide implementation), classes (hide internal data), APIs (hide how a service works).

48–50 min — Key Takeaway

4. Teacher Facilitation Notes

What to look for:

- Students who include everything "just in case" — challenge: *"Does a fire evacuation map need the canteen menu?"*
- Students who struggle to draw — reassure: labelled boxes and lines are fine. This is about THINKING, not art.
- Students whose maps look too much like a real map — push: *"What can you REMOVE and still serve your purpose?"*

Common misconceptions:

- More detail = better — no, more detail = more cognitive load. Abstraction removes noise.
 - Abstraction means vague — no, an abstraction can be very precise about what it includes.
-

5. Extension Tasks

1. Draw the same room at three levels of abstraction for three different purposes.
2. Research: how does a Class in object-oriented programming use abstraction?

3. Find a real-world example of abstraction that isn't a map (e.g., menu, periodic table, weather forecast).
-

6. Key Takeaway

Abstraction removes unnecessary detail. Good abstraction keeps exactly what the user needs — nothing more, nothing less. The same system can have many valid, different abstractions.