

LESSON 1

Introduction to Algorithms

40 minutes

Pattern Play Without Devices

WHAT THIS LESSON DOES

In this lesson, you'll explore what an algorithm is through a fun, hands-on activity. Follow step-by-step instructions to program a robot in a sandwich-making task, learning the importance of clear, precise directions for successful results.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the concept of an algorithm as a set of step-by-step instructions.
- Recognise the importance of precision and clarity in creating effective instructions.
- Develop skills in identifying and correcting errors through debugging.
- Apply logical thinking to create a clear and sequential set of instructions.
- Appreciate the impact of well-structured algorithms on achieving successful outcomes.

BEFORE THE LESSON

- This is an unplugged lesson: no devices are needed.
- Gather sandwich ingredients and tools: plate, knife, bread, butter and jam.
- Clear floor space so pupils can sit in a circle facing you.
- Have a whiteboard or large sheet of paper ready to record the rewritten algorithm.

Introduction to Algorithms

HOW THE LESSON RUNS

What is an Algorithm?

Programming the Robot

The First Attempt

Debugging the Program

Writing a Clear Algorithm

The Second Attempt

Success!

TEACHING MOVES

- 1. What is an Algorithm? Frame an algorithm as recipe-like steps. Tell pupils the point of the lesson is how much precision those steps need, because a tiny gap can change the result.
- 2. Programming the Robot. Seat pupils in a circle with the ingredients in front of you. State that they are the programmers and you are a robot with no common sense. Use a monotone voice and rigid movements.
- 3. The First Attempt. Demand the first instruction, then obey it exactly. Exaggerate gaps: whole loaf on the plate, whole jar smeared, lid pulled not unscrewed. Keep going so the cost of vague wording is obvious.
- 4. Debugging the Program. Stop and ask what went wrong and why the program failed. Name the process debugging. Stress that the idea was fine; the wording of the steps was not.
- 5. Writing a Clear Algorithm. Co-write a new algorithm on the board, one step per line. Prompt tiny details: a single slice, unscrew the lid, use the knife to spread. Keep the sequence visible.
- 6. The Second Attempt. Follow only the written steps, exactly as recorded. Let the sandwich form correctly so pupils see the payoff of the clearer algorithm.
- 7. Success! Celebrate the finished sandwich. Ask what made the second attempt better and steer answers toward precise, ordered instructions.

WHAT TO WATCH FOR

- Pupils assume a human helper will fill in gaps, so vague commands like "open the jar" or "spread the jam" feel complete. Stay rigidly literal on the first run. After each failure, ask what detail was missing before accepting a revised step.
- Pupils treat debugging as scrapping everything and starting again, rather than finding and fixing specific broken steps. Revisit the failed run step by step. Circle only the unclear lines, rewrite those, and keep any steps that already worked.
- Pupils think step order is flexible as long as all the actions are mentioned somewhere. On the board version, number each line and briefly test what happens if two steps swap, so sequence feels non-negotiable.

DIFFERENTIATION

- Emerging: Invite a single concrete first step from them, or pair them with a classmate to suggest one action while you stay in robot mode. Prompt with choices: "Do I pick up the loaf or one slice?"
- Developing: Ask them to spot one unclear line from the first attempt and rephrase it with a missing detail. Have them read the next board step aloud before you act, checking it sounds complete.
- Proficient: Challenge them to find any remaining ambiguity in the board algorithm before the second run. Ask them to add a necessary step the class has skipped, still in clear sequential language.