LESSON 3

Introduction to Algorithms

40 minutes | Foundations of Computer Science | Outcome 1.1 | Outcome 1.3 | Outcome 1.4 | Outcome 1.6 | Outcome 1.7 | Outcome 2.5 | Outcome 2.6

WHAT THIS LESSON DOES

In this lesson, you'll uncover the basics of algorithms, the foundation of computer science. Learn how they function as step-by-step problem-solving guides and gain practical skills by creating flowcharts to visualise processes logically and effectively.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.1	describe a systematic process for solving problems and making decisions
1.3	solve problems by deconstructing them into smaller units using a systematic approach in an iterative fashion
1.4	solve problems using skills of logic
1.6	explain the operation of a variety of algorithms
1.7	develop algorithms to implement chosen solutions
2.5	use pseudo code to outline the functionality of an algorithm
2.6	construct algorithms using appropriate sequences, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the fundamental concept of algorithms as step-by-step instructions for problem-solving in computing.
- Identify the key properties of effective algorithms, including precision, finiteness, effectiveness, and efficiency.
- Recognise the purpose and benefits of using flowcharts to visualise algorithmic processes.
- Develop skills in creating flowcharts to represent simple algorithms using both manual and digital tools.
- Cultivate logical thinking and planning abilities essential for programming and computational problem-solving.

BEFORE THE LESSON

- Open Draw.io yourself first and confirm it loads through the school filter. It runs in the browser with no account needed, but check whether saving requires a Google or OneDrive sign-in and decide how students will keep their work.
- Have blank paper and pens ready for the pen-and-paper flowchart task in the middle of the lesson.
- Have the four flowchart shapes and their meanings on the board: oval start/end, rectangle process, diamond decision, arrows for flow.

Introduction to Algorithms

HOW THE LESSON RUNS

- Introduction
- What is an Algorithm?
- Key Properties of Algorithms
- Introduction to Flowcharts
- Practical Task: Draw a Flowchart using Pen & Paper
- Practical Task: Draw a Flowchart using a Tool
- Wrap Up

TEACHING MOVES

- 2. What is an Algorithm? Run the sandwich recipe live, deliberately doing a step wrong or out of order, so students see that a vague or misordered instruction breaks the result. Let them correct you.
- 3. Key Properties of Algorithms. For precision, use the ambiguous directions example: 'turn left at the big tree' with several trees. Ask students to rewrite one vague instruction so it cannot be misread. Keep finiteness concrete: it must stop.
- 4. Introduction to Flowcharts. Draw one small flowchart on the board while naming each shape as you use it. Stress that a diamond is a yes/no question with two arrows out, which trips people up later.
- 5. Practical Task: Draw a Flowchart using Pen & Paper. Let them choose their own everyday task but hold the sandwich as the fallback. Circulate and check every flowchart starts and ends with an oval before anyone opens Draw.io.
- 6. Practical Task: Draw a Flowchart using a Tool. They recreate the same paper flowchart in Draw.io, not a new one. This keeps the focus on the tool. Show once how to drag a shape and connect it, then let them work.

WHAT TO WATCH FOR

- Students write steps that are correct but too vague, assuming the reader will fill in the gaps the way a person would. Have them read their algorithm aloud to a partner who follows it literally, taking any shortcut the words allow. The gaps show up fast.
- The diamond decision shape is treated as just another box, with one arrow out instead of two labelled branches. State the rule: every diamond asks a yes/no question and must have two arrows leaving it, one for each answer, each labelled.

DIFFERENTIATION

- Emerging: Give them the sandwich task directly rather than asking them to invent one, and pair them with a peer for the pen-and-paper stage. Provide the shapes pre-drawn so they fill in the words rather than deciding shape and content at once.
- Developing: Ask them to add one decision point to their task, for example 'is the bread stale?', so they use a diamond meaningfully.
- Proficient: Challenge them to redraw their flowchart so it loops, for example 'keep adding fillings until the sandwich is full', introducing repetition ahead of loops. Ask them to justify which of the four properties their flowchart best demonstrates and where it could be more efficient.