Portfolio Evidence: Boolean Logic Lesson Plan Extract

Subject: IGCSE Computer Science
Topic: Boolean Logic
Class: KS4, 12 students
Focus: Planning and teaching an abstract Computer Science concept through structured modelling, scaffolding, formative assessment and independent application.

This lesson shows how I planned an IGCSE Computer Science lesson on Boolean Logic by building from students' prior knowledge of binary values and TRUE/FALSE conditions. The lesson introduced NOT, AND and OR gates through visual modelling, truth tables, analogies, guided practice, ABCD hinge questioning and independent exercises. It also planned for common misconceptions, bilingual vocabulary support, scaffolding for students who needed structure, and extension tasks for more confident learners.

<table>
<tr><td>Learning Context</td><td colspan="2">In the previous lesson, students completed a unit on Databases and SQL, learning how data is stored, queried, and manipulated. This new topic begins Boolean Logic, focusing on how computers use logic gates (NOT, AND, OR) to process binary data internally.

[Planning for progression] Students already understand: binary values 0 and 1, the idea of digital vs analogue signals, basic use of TRUE/FALSE in conditions

This lesson builds foundational logic needed for upcoming content: logic circuits, truth tables, NAND, NOR, XOR, Boolean algebra for exam-style questions</td></tr>
<tr><td rowspan="2">Aims for learning</td><td>Learning Objectives:</td><td>Evidence of Learning:</td></tr>
<tr><td>By the end of the lesson, students will be able to: [Clear learning goals]
1. Explain what a logic gate is and why computers use them (10.1).
2. Draw and interpret the NOT, AND, OR gates (10.2, 10.3, 10.4).
3. Complete truth tables for NOT, AND, and OR.
4. Write logic expressions such as:
X = NOT A,
X = A AND B,
X = A OR B</td><td>Students will: [Evidence of learning]
• Correctly draw NOT, AND, OR gate symbols.
• Fill in truth tables with 100% accuracy for all three gates.
• Write correct logic expressions in IGCSE notation.
• Answer the hinge question using logic reasoning.
Complete independent practice questions (book-style short exercises).</td></tr>
<tr><td>Anticipated misconceptions and planned responses</td><td colspan="2">[Planning for misconceptions]
Students confuse truth table order Provide structure: A and B always listed in binary count order (00, 01, 10, 11).
Think NOT gate has two inputs Emphasise visually: only one input on left, one output on right.
Mixing up operators (AND = +, OR = .) Use IGCSE notation consistently; show alternative symbols explicitly.
Thinking logic gates perform arithmetic operation. Show and activity which reveals AND behaves like multiplication.</td></tr>
<tr><td>Adaptive and</td><td colspan="2">Supportive Scaffolds [Planning for access and challenge]</td></tr>
</table>

inclusive teaching	• Partially completed truth tables for lower-ability students. • Gate drawings labelled with “1 = HIGH”, “0 = LOW”. • Sentence stems like: “The gate outputs 1 when…” and “For this gate, both inputs must be…” **Challenge (High Achievers)** • Predict gate outputs for longer expressions such as: X = NOT(A AND B) • Explain why AND behaves like multiplication and OR like addition. • Extend: create their own real-life analogy for each gate. **EAL Support** • Use icons (✓ for 1, ✗ for 0). • Keep truth table structure fixed. • Provide bilingual keywords.

Element	**Time**	**What is the learning focus?** What will students be learning – linked to Learning Objectives.	**What will be happening in the classroom?** Outline specific actions you need to take as a teacher as well as what students will be doing.	**How will I check they are learning (formative assessment)?** What strategies will you use?	**Resources / Environment / Classroom Management**
Start of Lesson Routine (creating a climate for learning)	3 min	Recall last topic (SQL). Reset focus to binary logic.	[Activating prior knowledge] 3 quick recall questions using ABCD cards: “What is binary?”, “What does TRUE mean?”, “Which operator compares values?”	Check cards; clarify misconceptions.	ABCD cards, recall slide.
Introduction	2 min	Present today’s learning objectives.	Teacher shows LOs. Students read and briefly discuss why computers need logic gates.	Cold call 1–2 students.	Maxhub or projector.
Main body of lesson	5 min	Understand NOT logic gate.	[Structured modelling] Explain voltage → binary. Show NOT gate symbol. Teacher writes truth table.	Ask: “If A = 0, what is NOT A?”	Board diagrams.

	8 min	Understand AND and OR logic gates.	Teacher explains both gates using guard analogy (tickets). Draw symbols and truth tables on board.	Ask students to predict missing rows before revealing answers.	Diagrams, examples.
	7 min	Construct truth tables cooperatively.	[Guided practice] Students in pairs complete partially blank AND/OR tables. Teacher circulates giving hints.	Check 3–4 tables aloud; correct errors.	Worksheets, board.
	3 min	Check understanding before independent work.	[Formative assessment and responsive teaching] Students answer using ABCD cards: Which of the following truth tables belongs to OR? (Provide four table options).	If <80% correct, re-teach OR vs AND.	ABCD cards, slide.
	8 min	Apply NOT/AND/OR to truth tables & expressions.	[Independent application] Students complete 6–8 short questions: draw gates, fill truth tables, write expressions (X = NOT A, X = A AND B, X = A OR B).	Teacher circulates; mark 1–2 questions quickly.	Exercise book or worksheet.
Plenary	4 min	Consolidate understanding + exit ticket.	[Responsive next-step planning] Students answer a 4-question Microsoft Forms quiz: 1 NOT, 1 AND, 1 OR truth table, 1 expression.	Teacher reviews responses to plan next lesson (NAND, NOR, XOR).	MS Forms link/QR code.

Lesson evaluation

The lesson went well overall. Students were attentive and engaged, and the new content was understood by most of the class. I intentionally checked every worksheet during the independent task to assess individual progress. This allowed me to confirm that most students completed the exercises accurately. I noted particularly strong improvement from one student, who showed clearer reasoning in constructing truth tables. A small number of students will need a short revisit next lesson to ensure they fully understand how outputs are derived in truth tables.

What I learned and next steps

This lesson reinforced the importance of planning abstract Computer Science concepts in small, visible steps. Boolean Logic can be difficult for students because they need to connect binary values, gate symbols, truth tables and logic expressions at the same time. The use of visual modelling, fixed truth table structures and step-by-step examples helped students understand how each gate works.

The guided practice was effective because students were able to complete partially structured truth tables before moving into independent questions. This helped reduce cognitive load while still requiring students to reason about the output of each gate.

The hinge question using ABCD cards was useful because it made understanding visible before students moved into independent practice. It allowed me to check whether students could distinguish OR from AND and identify the correct truth table pattern.

The observation feedback confirmed that modelling, scaffolding, differentiation and student engagement were strengths of the lesson. It also reinforced the value of preparing differentiated tasks in advance so that students could work at an appropriate level of challenge.

For future lessons, I would continue to use visual modelling and hinge questions, but I would ask students to verbalise their reasoning more frequently using sentence stems such as “The output is 1 because…” or “This gate only outputs 1 when…”. I would also use the exit ticket results to decide whether students are ready to move on to NAND, NOR and XOR, or whether they need a short retrieval starter on NOT, AND and OR first.