

Lesson Plan: Grand Challenge

Year Group: 7 | **Duration:** 50 minutes | **Topic:** Synthesis — Algorithms, Variables, Loops, Conditionals

1. Overview

Core Concept: Bringing together all Year 7 concepts — algorithms, variables, loops, and conditionals — in a single problem-solving challenge.

Learning Objectives:

- Apply algorithms, variables, loops, and conditionals together in one solution
- Write a clear, testable algorithm within given constraints
- Evaluate a peer's algorithm by acting as a robot
- Revise an algorithm based on feedback

Key Vocabulary: All Year 7 terms: algorithm, variable, assign, update, loop, repeat, iteration, conditional, IF, ELSE, debug, trace, precise, ambiguous

2. Before the Lesson

Print:

- [resource-challenge-maze.md](#) — 1 per student or pair (A3 strongly preferred — the grid is large)
- [worksheet-solution-sheet.md](#) — 1 per student

Room Setup:

- Pairs or individuals, depending on class ability
 - Ensure students have pencils (they will revise their algorithms after peer testing)
-

3. Timed Lesson Flow

0–5 min — Challenge Reveal

1. Distribute the maze. Students look at it — don't explain anything yet.
2. *"Your challenge: write an algorithm in 20 lines or fewer that navigates from START to EXIT."*
3. Reveal the constraints:
 - **Max 20 lines** (including the content of loops)
 - **Must use at least one REPEAT loop**
 - **Must use at least one IF/ELSE**
 - **Must be peer-testable** — another pair will follow it as a robot

5–10 min — Planning

1. Students study the maze — identify the route, note where loops and conditionals are needed.
2. Mark the planned route on the small planning grid on the worksheet.
3. Ask: *"Where in the maze do you repeat the same move? That's your loop. Where do you have to make a decision? That's your IF/ELSE."*

10–30 min — Write the Algorithm

Students write their solution on [worksheet-solution-sheet.md](#). Circulate:

- Check they're using REPEAT notation (not just "move right 4 times" written out)
- Check IF/ELSE has a condition and both branches
- Remind them of the 20-line limit

30–40 min — Peer Test

1. Swap solution sheets with another pair.
2. The testing pair acts as robots — they follow the algorithm step by step on the maze grid.
3. They record: how many steps succeeded before it failed? What happened?
4. Fill in the Peer Test section of the worksheet.

40–47 min — Read Feedback and Revise

1. Authors get their sheets back and read the peer test results.
2. They write what they changed in the Author's Response section.
3. (Optional: run a second peer test if time allows)

47–50 min — Reflection

- *"Which was harder: writing the algorithm or testing someone else's?"*
- *"What made peer review valuable?"*
- *"Did having a 20-line limit change how you wrote your solution?"*

4. Teacher Facilitation Notes

What to look for:

- Students who write out every step without using loops — ask: *"You move right 4 times. Can you write that as REPEAT 4 TIMES { move right }?"*
- Students whose IF/ELSE has no ELSE — ask: *"What happens if the condition is false?"*
- Students who can't start — ask: *"Where is the robot right now? What single move does it need to make next?"*

How to intervene minimally:

- *"Where is the robot right now?"* — grounds students who are lost
- *"What decision does the robot need to make at that square?"* — prompts IF/ELSE
- *"Could you loop that repeated section?"* — prompts loop usage
- Never tell students the answer — let peer testing reveal the bug

Managing the peer test:

- The testing pair should be genuinely rigorous — encourage them to be "pedantic robots"
 - If a testing pair gets stuck, remind them: *"Follow it exactly. If the instruction is unclear, note that down and stop — don't guess."*
-

5. Extension / Early Finisher Tasks

1. **Optimise:** Can you solve the maze in fewer than 15 lines? 10 lines? How few lines is theoretically possible?
 2. **Add a variable:** Extend your algorithm to track a SCORE variable. Add events that increase or decrease the score based on which route is taken.
 3. **Design your own maze:** Create a new maze that specifically REQUIRES a loop and a conditional to solve efficiently. Swap with a partner.
-

6. Key Takeaway

Real programs combine all concepts — algorithms, variables, loops, and conditionals work together. Writing a correct, efficient solution requires planning, testing, and revision.