# **Lesson** 5**:** Algorithm design

## Introduction

In this lesson, pupils will design algorithms to move their robot around the mats that they designed in Lesson 4. As part of the design process, pupils will outline what their task is by identifying the starting and finishing points of a route. This outlining will ensure that pupils clearly understand what they want their program to achieve.

## Learning objectives

To design an algorithm

* I can explain what my algorithm should achieve
* I can create an algorithm to meet my goal
* I can use my algorithm to create a program

## Key vocabulary

Algorithm

## Preparation

**Subject knowledge:**

* You should have a good understanding of the term ‘algorithm’. An algorithm is a precise set of ordered instructions, which can be turned into code. In this lesson, pupils will create algorithms by drawing symbols (forwards, left, right, backwards). When pupils press the corresponding buttons on the robot, this creates a program that the robot follows.
* In the ‘Algorithm decisions’ activity, pupils will think through the task that they want their program to address. Identifying and fully understanding the task is a key step in program design.

**You will need:**

* [Slides](https://ncce.io/pg2a-5-s) (ncce.io/pg2a-5-s)
* Mats and obstacles from Lesson 4
* Dry wipe boards or paper and pens

## Assessment opportunities

**Activity 1:** You can assess whether pupils can select a ‘start’ and ‘end’ square and plan a route between the two.

**Activity 2:** You can assess whether pupils can draw an algorithm for the route that they have identified.

**Activity 3:** You can assess whether pupils can test their algorithm as a program on the floor robot.

## Outline plan

Please note that the slide deck labels the activities in the top right-hand corner to help you navigate the lesson.

**Note:** For the activities in this lesson, pupils should ideally work in pairs, or if necessary, groups of three.

*\*Timings are rough guides*

|  |  |
| --- | --- |
| Introduction(Slides 2–3)5 mins | **This way and that way**Display slide 2 and introduce the lesson objectives.Move on to slide 3 and remind pupils of Lesson 4, in which they produced mats that had places to visit and obstacles to avoid. Tell pupils that in this lesson, they will create algorithms and programs for different routes. |
| **Activity 1**(Slide 4)5 mins | **Algorithm decisions**In this activity, pupils will think about the task in their program design. Display slide 4 and explain the instructions. Click through the animations on the slide to illustrate each step:1. Ask pupils to choose a square at each end of their mat.
2. Ask pupils to decide which way their robot should face at the start.
3. Ask pupils to determine the route that they want to program the robot to follow.
4. Ask pupils to describe the route to a partner. **Note:** The aim of asking pupils to verbalise their intention is to encourage them to commit to a specific task.
 |
| **Activity 2**(Slide 5)15 mins | **Algorithm design**Display slide 5. Tell pupils that they now need to draw the algorithm that will make the robot follow the route that they have selected and explained to their partner. Ask pupils to draw arrow symbols showing the steps for the robot on a dry wipe board or piece of paper. Explain that these symbols indicate the commands that will need to be used in a program later. Once pupils have drawn their route, ask them to carefully follow it through to check it, either by pointing or using a paper-bot (as they did in Lesson 3).**Note:** If pupils find it more difficult to relate the arrows to the orientation of the floor robot, encourage them to use the paper-bot, as this will support them.Finally, ask pupils to talk through their algorithm with a partner. |
| **Activity 3**(Slide 6)15 mins | **Programming**Display slide 6. Tell pupils that they now need to try their algorithms as programs on the floor robot. Tell them that they should take it in turns to try their algorithms. Before they start, remind pupils that it’s really important to start the robot on the square that they chose in the ‘Algorithm decisions’ activity. It’s also crucial that the robot face in the direction that they selected. Tell pupils to work together to ensure that programs are entered as carefully as possible: one pupil should press the buttons on the robot, while the other pupil keeps track along the algorithm. Explain that, to keep track, pupils can tick under the commands, or cover the commands, but they should not cross out or erase the commands. Also, remind pupils that before they enter their program, they need to clear any previous programs that the robot had by pressing X. Tell pupils that when they run a program, they should follow the robot’s movement and point at the corresponding command in their algorithm. Tell pupils that if the robot doesn’t move to the square that they expected, they should first try clearing the robot’s memory and re-entering the program. If the robot still doesn’t get to the expected destination, pupils should try to identify where their algorithm may be incorrect. If pupils can identify the issue, they should try to correct it.  |
| **Plenary**(Slide 7)5 mins | **Debugging**Ask pupils to reflect on the activities that they have completed in the lesson, and whether they found anything that didn’t go as expected. Display slide 7 and introduce the term ‘debugging’. Explain that programmers don’t always get things right the first time, and when a programmer finds a problem in their program or algorithm, they fix it. |
| **Summary**(Slides 8–9)5 mins | Display slide 8. Ask pupils to reflect on the lesson and indicate how confident they feel about what they have learnt.Move on to slide 9. Summarise what pupils did in this lesson, and introduce what they will do in the next lesson. |

This resource is available online at [ncce.io/pg2a-5-p](http://ncce.io/pg2a-5-p). Resources are updated regularly — please check that you are using the latest version.

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](https://ncce.io/ogl).