Introduction to algorithms

Scratch lesson plan – Code Playground

BARCLAYS

Lesson overview

Teach students to understand and create algorithms by building a step-by-step sequence in Scratch. This lesson will help students realise the importance of clear, ordered instructions in coding.

Time	Key learning outcomes	Resources
30 mins	• Understand algorithms as step-by-step instructions • Build a basic sequence in Scratch using movement and action blocks • Recognise how clear, ordered steps are essential for creating successful algorithms.	• Laptops or desktop computers • Access to Scratch website - https://scratch.mit.edu • Projectors or smartboard for demonstration

Content

Lesson introduction (5 minutes)

Algorithms are like recipes. Ask students to think about tasks they do daily, like making a sandwich. Each step, like getting bread, adding ingredients, and putting the slices together, is part of a recipe-just like an algorithms.

Imagine brushing your teeth. You don't start by rinsing until you've brushed. In an algorithm, each step must be in the right order for the task to work correctly.

Show the "**How computers work**" video to the class.

Activity 1 – Physical algorithm exercise (10 minutes)

Reinforce the concept of clear instructions by having students physically guide each other through a task.

Instructions

1. Have students work in pairs, with one student as the '**programmer**' and the other as the '**robot**'
2. Ask the '**programmer**' to give step-by-step directions to the 'robot' for simple actions, like moving forward, turning, or clapping hands
3. After a few minutes, discuss why it's essential to be specific. Explain that computers need every step clearly defined.

Ask the class

- Why do you think programmers need to give such specific instructions? What happens if a step is missed?

Activity 2 – Creating a basic algorithm (15 minutes)

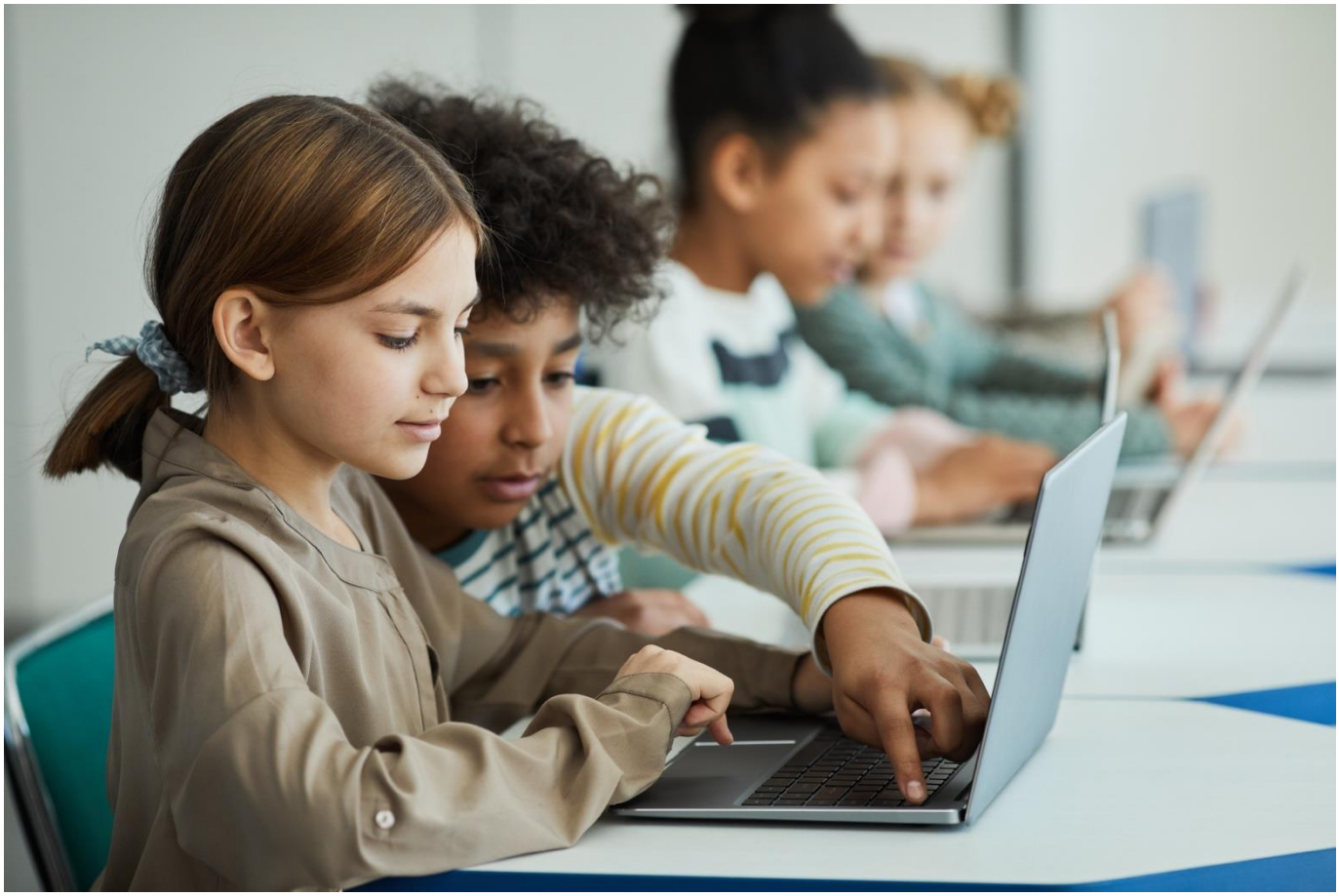
Guide students in creating a simple sequence in Scratch, using blocks to make a sprite move and speak.

Instructions

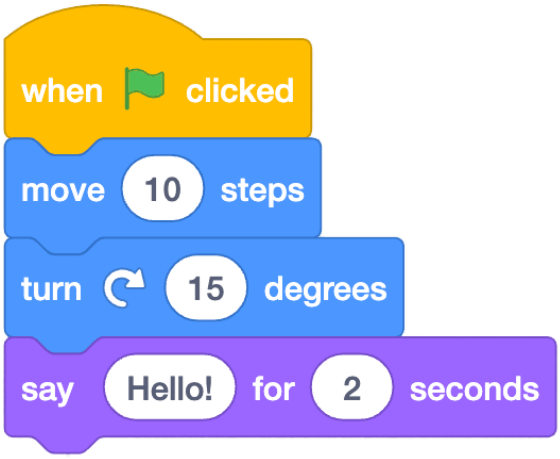
1. Have students open a new Scratch project and choose a sprite
2. Explain that they'll create a series of movements and actions for their sprite, just like they did with their '**robot**'
3. Encourage them to experiment by adding different actions (like moving, turning, and speaking) and to arrange these actions in a specific order.

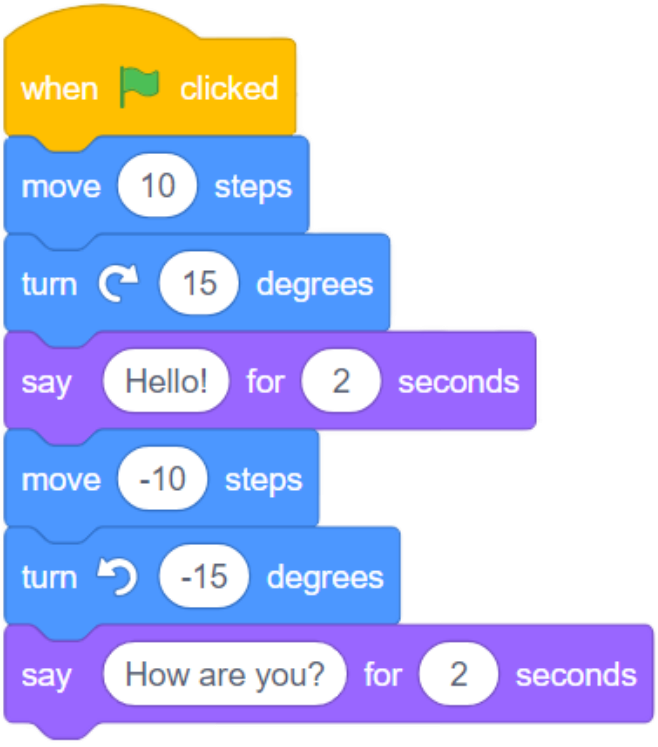
Ask the class

- What do you think will happen if you switch the order of two steps?
- What other actions can you add to your sequence to make it unique?



Code snippet - examples





Summary

The following information is an example of what a child at an expected level would be able to demonstrate when completing these activities with additional examples to demonstrate how this would vary for a child with emerging or exceeding achievements.

Assessment guidance

Differentiation – Lower ability/ASN

- Focus on a short sequence, using just a few blocks like '**move**' and '**say**'. Provide visual aids or screenshots to help with block recognition
- Guide students through acting out each step physically before adding it to their Scratch project. This helps reinforce the sequence before coding.

Differentiation – Higher ability/extension

- Encourage students to add more steps to their sequence, creating patterns or repeating movements
- Challenge advanced students to use '**repeat [x] times**' to create a repeating pattern, such as moving in a square or bouncing.

Plenary

Reinforce the concept of creating clear, ordered steps in an algorithm.

- Invite students to share their sequences and explain the order of their steps. Discuss why each step follows the one before it
- Discuss other tasks that use ordered steps, like getting ready in the morning or tying shoelaces.

Assessment questions

- Can you explain what an algorithm is in your own words?
- Why is the order of steps important in an algorithm?
- What would happen if you moved the last step to the beginning of your sequence? How would that change what your sprite does?