Barefoot

Recommended for **ages 5–7**

Bee-Bots 1,2,3 Activity

An introduction to programming with Bee-Bots

Activity Duration: **30 mins per group + 15 mins for overall introduction & plenary**

Principal partners

BT Computing at School

Concepts and approaches covered

Programming Debugging Algorithms Persevering

@BarefootComp /barefootcomputing barefootcomputing.org

Overview

Pupils create sequences of instructions (an **algorithm**) to draw the shape of a numeral e.g. 3. An algorithm is a sequence of instructions, or a set of rules, for performing a specific task. In this activity, **programming** involves taking the algorithm and using it to program a Bee-Bot to navigate a route, tracing the shape of the numeral.

Pupil objectives

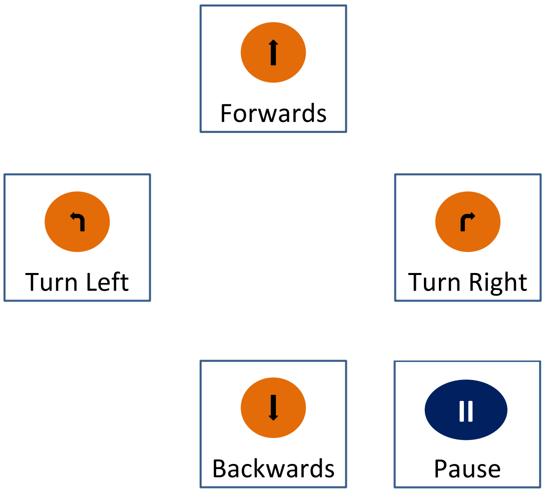
☑ **I can write an algorithm**

☑ **I can program and debug a Bee-Bot to follow my algorithm**

Introduction: whole class (5 mins)

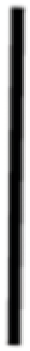
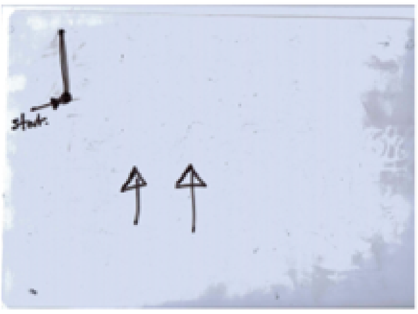
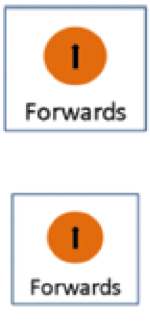
- Be 'bossy' and instruct a child to do something e.g. stand up, go to door, open it, come back to carpet place and sit down
- Say that sequences of instructions are important as they help us to know what to do and how to make things happen
- Explain that the lesson is about programming and how algorithms help us write in a language that computers can understand: code. Remind pupils that algorithms are steps to make something happen and are for people to understand but that programs are for computers
- Show pupils a Bee-Bot. Ask pupils how they think we might program it
- Ask pupils to record on whiteboards or in their books any Bee-Bot command symbols that they can remember from when they **tinkered with the Bee-Bots** (if they have completed this activity)

What programming commands can we use?
What do they do?
How can we record them on our whiteboards?



Bee-Bot group explanation (5 mins)

- Sit your group in a circle and ask them how we could get the Bee-Bot to write the numeral 1. Write the numeral 1 on a whiteboard to show them. They may suggest forward, forward. Some might start at the top of the number, others at the bottom
- Encourage pupils to think about how to write the number 1 (starting at the top) reinforcing number formation
- Ask pupils how they could record their plan. Ask them what they call this plan. Hopefully they remember from earlier work on algorithms that this is an algorithm. At this point let all the pupils have a go at drawing their algorithms on their whiteboards

Show them the A4 numeral card	Show how you can jot your algorithm down on your whiteboard. You could include a jotting of the numeral and where you are going to start (top or bottom).	Some children might find using command cards helps (optional)	Try the algorithm by using a fakebot to walk the algorithm through.
			

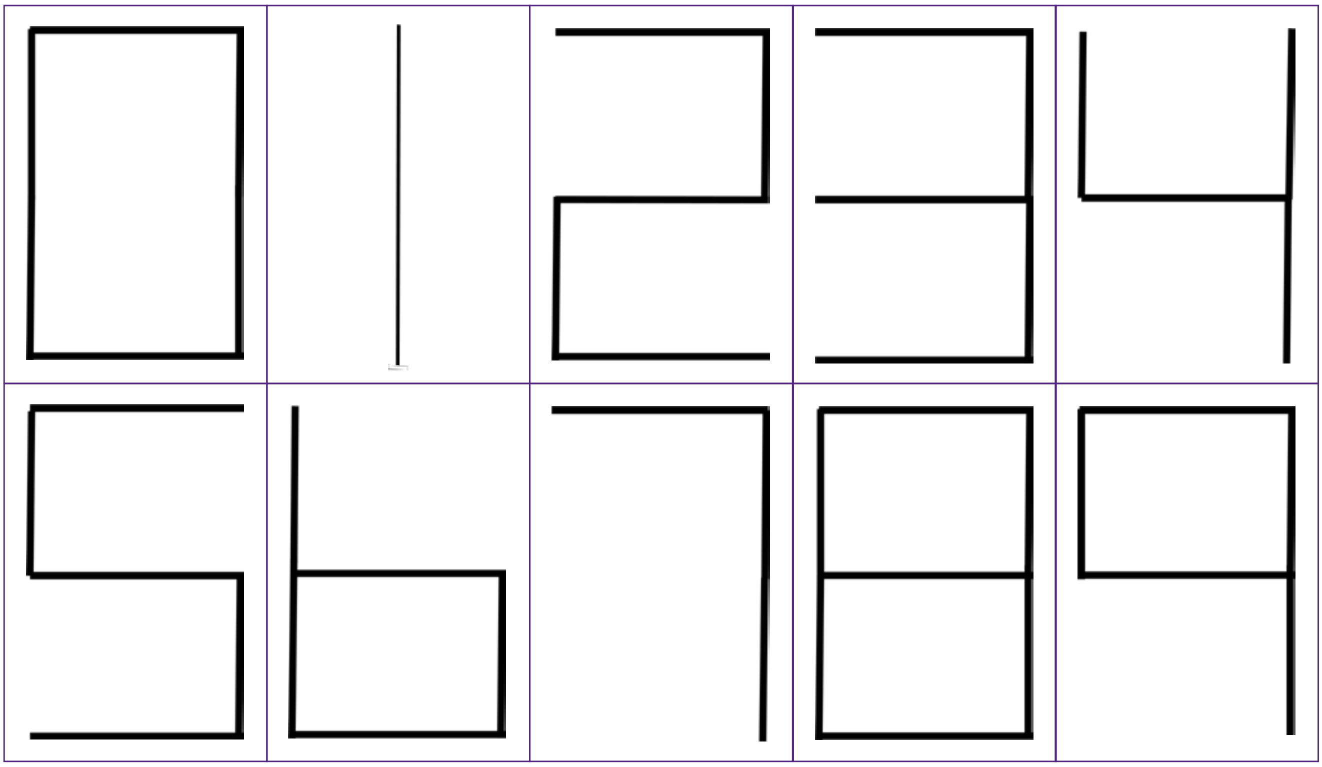
Steps to create and then debug the algorithm

- Explain it is often a good idea to walk through an algorithm before we program it. Stand up and walk through the algorithm and/or use the FakeBot to test the algorithm
- Choose a child to be your partner – your ‘coder’
- Give the coder your algorithm and ask them to use it to type in the commands on the Bee-Bot
- Ask the pupils what went well? (e.g. they remembered to press clear, the Bee-Bot went forward) and what could be improved?

Group activities (20 mins)

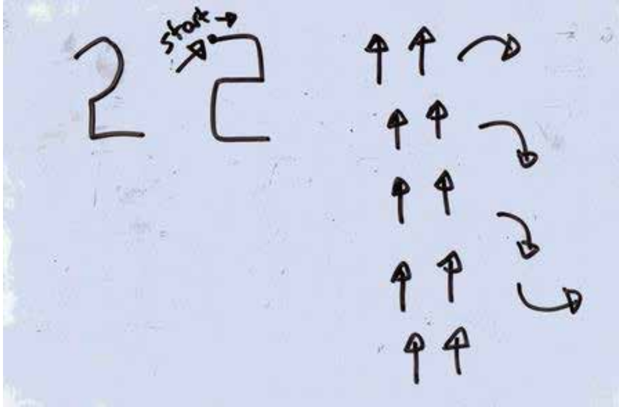
In each pair/group, pupils will take it in turns (and swap over roles) to:

- Write the algorithm to solve the challenge on a pupil’s whiteboard
- Together they will review their success and try to fix any problems
- Show pupils the numeral cards to help them understand how the shapes of numbers need to change for a Bee-Bot to trace them

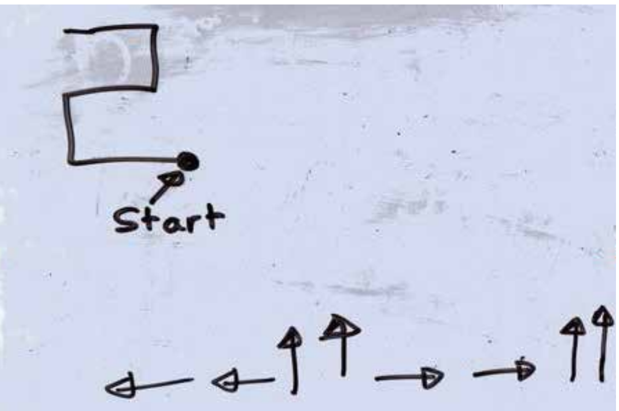


Numeral Cards

- Examples of algorithms you might see:



This first algorithm for the numeral 2 may not do what the child thinks



This may be closer for the numeral 2, but do we need to start at the top of the commands or the bottom when coding this algorithm?

How they record their algorithm is less important than whether they can explain it and use it to program the Bee-Bot.

- Give pupils command cards if they are finding it tricky to work out the sequence of steps. Encourage the pupils to try just 2 or 3 commands before they test it so they work on small bite-size chunks. This is always a useful approach to programming

Plenary (5 mins)

- Ask pairs to share one of their programs with each other explaining what was easy and what was hard about creating the algorithm and programming the Bee-Bot to follow it
- As a whole class, talk about successes, challenges and surprises from the activity

Differentiation

Support:

Pupils may need supportive pairings or adult support, particularly when creating their algorithm. Pupils who find drawing the algorithm pictures difficult can use the Bee-Bot command cards to sequence their steps.

Stretch & Challenge:

Encourage more able pupils to be very precise when creating their algorithms. Encourage pupils to tackle it as a multi-step problem with distinct parts (**decomposing**). Encourage pupils to **debug** as they go along, checking their algorithm.

Assessment opportunities

- Informal teacher assessment of pupils during main task and plenary. Focus on how pupils write their algorithms for the numerals and then program their Bee-Bot to follow this, debugging if required
- Assessment of pupils' algorithms photocopied from whiteboards

Teaching notes

Concepts and approaches

- Pupils use **decomposition** to break down the numerals into sections, making it easier to write the algorithm
- Pupils write an **algorithm** that traces out the shape of each numeral
- Pupils implement their algorithms as **programs** for the Bee-Bot using the inputs on the Bee-Bot
- Pupils **debug** their algorithm or program if there is an error in it
- Pupils **collaborate** together to design the algorithm and create the code for their program
- Pupils **persevere** as they encounter bugs in their work and work through how to fix these

Curriculum links

Computing

- Understand algorithms, how they are implemented as programs on digital devices and that programs work by following clear, precise instructions
- Create simple programs

Maths

Geometry – position and direction:

- Year 1 pupils should be taught to: describe position, direction and movement, including whole, half, quarter and three-quarter turns
- Year 2 pupils should be taught to: use mathematical vocabulary to describe position, direction and movement, including movement in a straight line and distinguishing between rotation as a turn and in terms of right angles for quarter, half and three-quarter turns (clockwise and anti-clockwise)

Resources

- Pupils’ small whiteboards / exercise books and pens
- Bee-Bots
- Bee-Bot A4 numeral cards
- Print some fakebots (a printed image of a Bee-Bot or other programmable toy)
- Optional – Bee-Bot command cards
- Teacher’s IWB or paper flipchart (adapt to your needs)

Further reading

There is a wealth of resources to use with Bee-Bots online:

http://www.tes.co.uk/teaching-resources/primary-40069/ks1-ict-41488/controlling-and-modelling-41489/

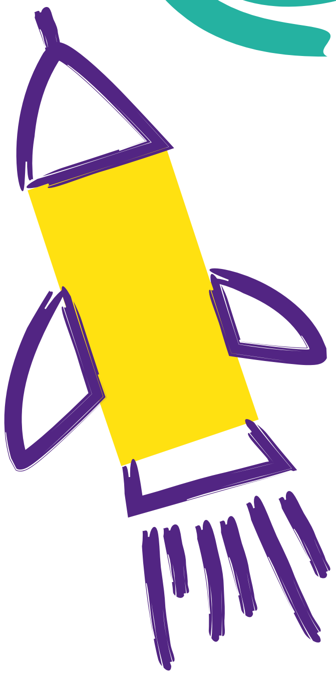
If you are using pro bots:

BBC KS1 How do you program a robot: **http://www.bbc.co.uk/guides/zqnc4wx**

Related activities

KS1 Bee-Bots Tinkering activity

Get more **Barefoot**



Have you had a great Barefoot workshop, or delivered a fun computer science lesson? Send us your comments and pictures via our social channels to help get more teachers involved!

- @BarefootComp
- /barefootcomputing
- barefootcomputing.org

You can start your journey here

Principal partners

BT Computing at School

Supported by

bcs openreach Department for Education Education Scotland Foghlam Alba Gweithio mewn partneriaeth gyda Llywodraeth Cymru Working in partnership with Welsh Government Education

Version 1118   Expires Dec 2021