Progression Thread

Computing – Programming EYFS – KS2

National Curriculum

Aims

The national curriculum for computing aims to ensure that all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.

Teach Computing

Progression

Progression across key stages

All learning objectives have been mapped to the National Centre for Computing Education's taxonomy of ten strands, which ensures that units build on each other from one key stage to the next.

Progression across year groups

Within the Teach Computing Curriculum, every year group learns through units within the same four themes, which combine the ten strands of the National Centre for Computing Education's taxonomy (see table, right).

This approach allows us to use the spiral curriculum approach (see the 'Spiral curriculum' section for more information) to progress skills and concepts from one year group to the next.

Primary themes	Computing systems and networks	Programming	Data and information	Creating media
Taxonomy strands	Computer systems Computer networks	Programming Algorithms Design and development	Data and information	Creating media Design and development
	Effective use of tools			
	Impact of technology			
	Safety and security			

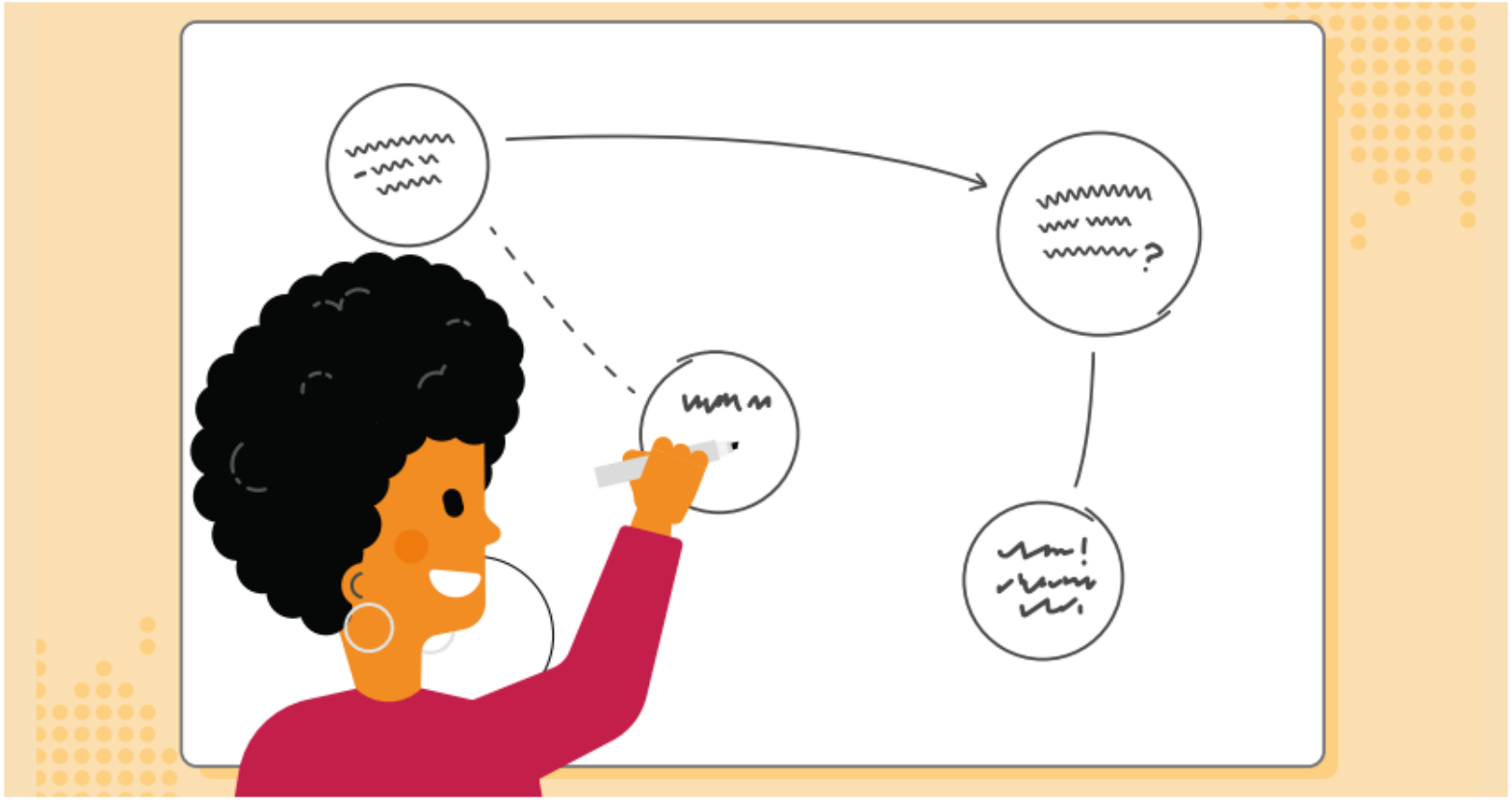
Spiral Curriculum

The units for key stages 1 and 2 are based on a spiral curriculum. This means that each of the themes is revisited regularly (at least once in each year group), and pupils revisit each theme through a new unit that consolidates and builds on prior learning within that theme. This style of curriculum design reduces the amount of knowledge lost through forgetting, as topics are revisited yearly. It also ensures that connections are made even if different teachers are teaching the units within a theme in consecutive years

Progression within a unit — learning graphs

Learning graphs are provided as part of each unit and demonstrate progression through concepts and skills. In order to learn some of those concepts and skills, pupils need prior knowledge of others, so the learning graphs show which concepts and skills need to be taught first and which could be taught at a different time.

The learning graphs often show more statements than there are learning objectives. All of the skills and concepts learnt are included in the learning graphs. Some of these skills and concepts are milestones, which form learning objectives, while others are smaller steps towards these milestones, which form success criteria. Please note that the wording of the statements may be different in the learning graphs than in the lessons, as the learning graphs are designed for teachers, whereas the learning objectives and success criteria are age-appropriate so that they can be understood by pupils.



In each year group, there are two 'Programming' units of work, but only one 'Programming' learning graph. The second 'Programming' unit builds on the content that was taught in the first 'Programming' unit so closely that there is no specific divide where one ends and the other begins.

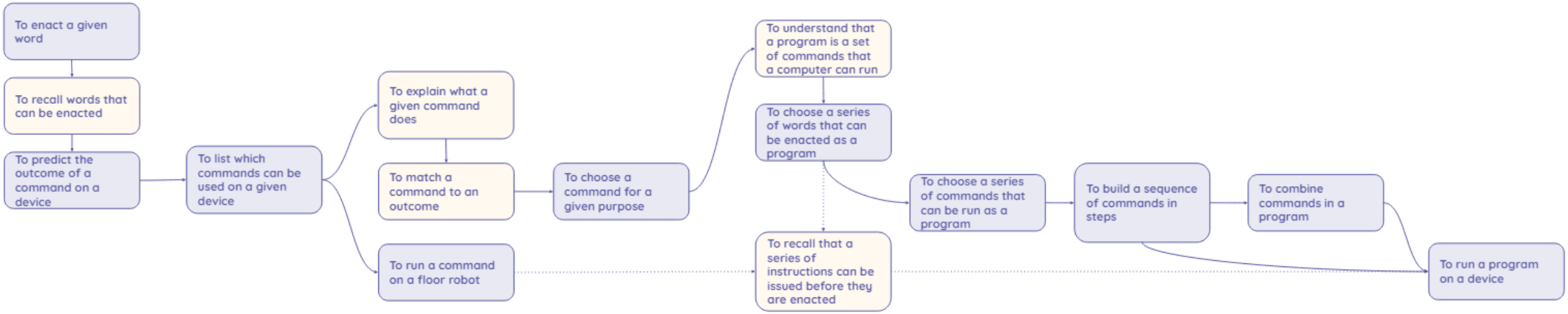
This means that times the software programming and hardware programming happens in a different order, depending on the year group.

Learning Graph Example – Year 1

National Centre for Computing Education

Raspberry Pi

Learning graph
Year 1 – Moving a robot



Structure of the units of Work - Programming

	Programming A	Programming B
Year 1	**Moving a robot** Writing short algorithms and programs for floor robots, and predicting program outcomes.	**Programming animations** Designing and programming the movement of a character on screen to tell stories.
Year 2	**Robot algorithms** Creating and debugging programs, and using logical reasoning to make predictions.	**Programming quizzes** Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.

	Programming A	Programming B
Year 3	**Sequencing sounds** Creating sequences in a block-based programming language to make music.	**Events and actions in programs** Writing algorithms and programs that use a range of events to trigger sequences of actions.
Year 4	**Repetition in shapes** Using a text-based programming language to explore count-controlled loops when drawing shapes.	**Repetition in games** Using a block-based programming language to explore count-controlled and infinite loops when creating a game.

	Programming A	Programming B
Year 5	**Selection in physical computing** Exploring conditions and selection using a programmable microcontroller.	**Selection in quizzes** Exploring selection in programming to design and code an interactive quiz.
Year 6	**Variables in games** Exploring variables when designing and coding a game.	**Sensing** Designing and coding a project that captures inputs from a physical device.

Key Stage 1 – National Curriculum

Pupils should be taught to:

- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- create and debug simple programs
- use logical reasoning to predict the behaviour of simple programs
- use technology purposefully to create, organise, store, manipulate and retrieve digital content
- recognise common uses of information technology beyond school
- use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

Key Stage 2 – National Curriculum

Pupils should be taught to:

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

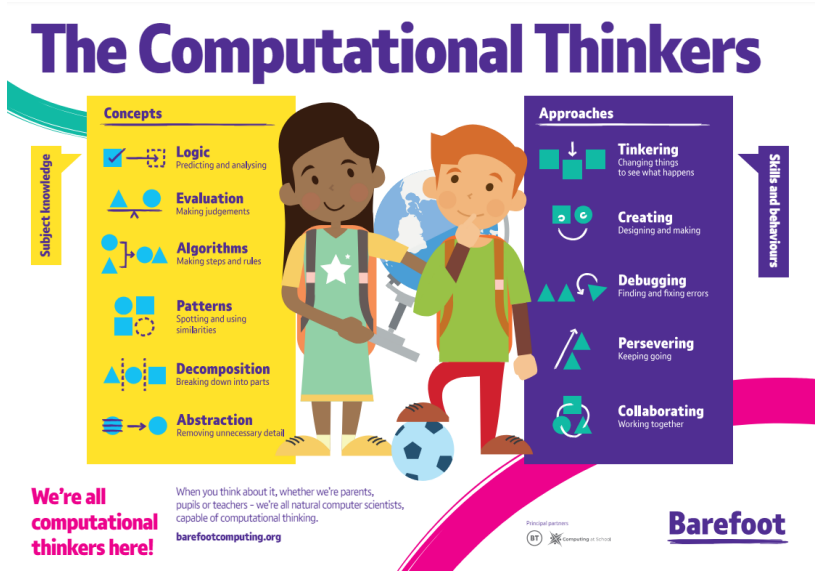
National Curriculum Coverage — Years 1 and 2	1.1 Technology around us	1.2 Digital painting	1.3 Moving a robot	1.4 Grouping data	1.5 Digital writing	1.6 Programming animations	2.1 Information technology around us	2.2 Digital photography	2.3 Robot algorithms	2.4 Pictograms	2.5 Making music	2.6 Programming quizzes
Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions			✓			✓			✓			✓
Create and debug simple programs			✓			✓			✓			✓
Use logical reasoning to predict the behaviour of simple programs			✓			✓			✓			✓
Use technology purposefully to create, organise, store, manipulate, and retrieve digital content	✓	✓		✓	✓		✓	✓		✓	✓	✓
Recognise common uses of information technology beyond school	✓		✓				✓	✓				
Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies	✓			✓	✓		✓	✓	✓	✓		

National curriculum coverage - Years 3 and 4	3.1 Connecting computers	3.2 Stop-frame animation	3.3 Sequencing sounds	3.4 Branching databases	3.5 Desktop publishing	3.6 Events and actions in programs	4.1 The internet	4.2 Audio production	4.3 Repetition in shapes	4.4 Data logging	4.5 Photo editing	4.6 Repetition in games
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts			✓			✓			✓			✓
Use sequence, selection, and repetition in programs; work with variables and various forms of input and output	✓		✓			✓			✓	✓		✓
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs			✓			✓			✓			✓
Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration	✓						✓					
Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content					✓		✓	✓			✓	
Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact		✓		✓			✓	✓			✓	

National curriculum coverage - Years 5 and 6	5.1 Sharing information	5.2 Video production	5.3 Selection in physical computing	5.4 Flat-file databases	5.5 Vector drawing	5.6 Selection in quizzes	6.1 Internet communication	6.2 Webpage creation	6.3 Variables in games	6.4 Introduction to spreadsheets	6.5 3D modelling	6.6 Sensing
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts			✓			✓	✓		✓			✓
Use sequence, selection, and repetition in programs; work with variables and various forms of input and output			✓			✓			✓			✓
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs			✓			✓			✓			✓
Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration	✓						✓					
Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content		✓		✓				✓				
Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact	✓	✓						✓	✓		✓	

Although not a part of the new EYFS curriculum, we embed computational thinking through:

EYFS		
Three and four year olds	Communication and Language	• Use a wider range of vocabulary. • Understand a question or instruction that has two parts, such as: "Get your coat and wait at the door" • Understand 'why' questions, like: "Why do you think the caterpillar got so fat?" • Be able to express a point of view and to debate when they disagree with an adult or a friend, using words as well as actions.
	Personal, Social and Emotional Development	• Increasingly follow rules, understanding why they are important. Remember rules without needing an adult to remind them.
	Physical Development	• Match their developing physical skills to tasks and activities in the setting.
	Mathematics	• Begin to describe a sequence of events, real or fictional, using words such as 'first', 'then…'
	Understanding the World	• Explore how things work.
Reception	Communication and Language	• Learn new vocabulary. • Connect one idea or action to another using a range of connectives. • Use talk to help work out problems and organise thinking and activities, and to explain how things work and why they might happen.
	Personal, Social, Emotional Development	• Show resilience and perseverance in the face of challenge. • Know and talk about the different factors that support their overall health and wellbeing: • sensible amounts of 'screen time'
	Physical Development	• Develop their small motor skills so that they can use a range of tools competently, safely and confidently.
	Expressive Arts and Design	• Explore, use and refine a variety of artistic effects to express their ideas and feelings.
Early Learning Goals	Personal, Social and Emotional Development	Managing Self • Be confident to try new activities and show independence, resilience and perseverance in the face of challenge. • Explain the reasons for rules, know right from wrong and try to behave accordingly.
	Expressive Arts and Design	• Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function.



Barefoot
Computing at School

Early Years

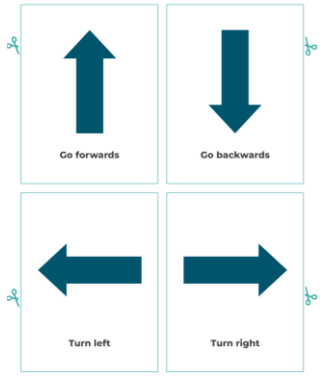
Awesome Autumn

Leaf Labyrinth

Duration: **20 minutes,** *plus time to collect leaves*

Concepts and approaches covered

Logic · Algorithms · Decomposition · Creating

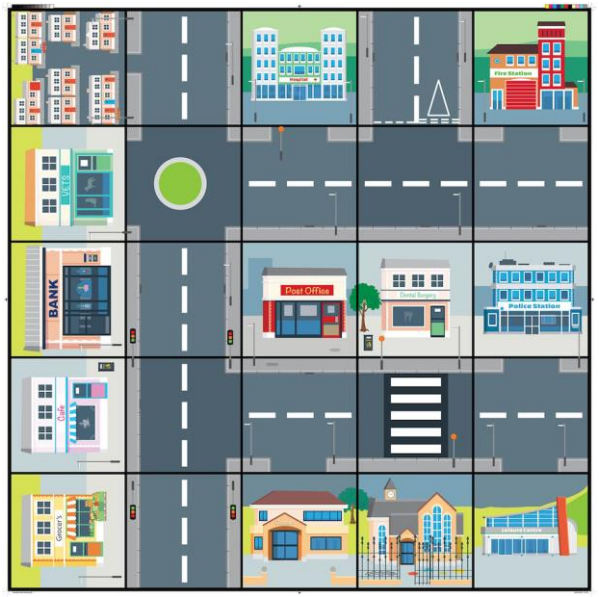


Links to and key questions to encourage Computational Thinking

This section offers lots of opportunities and prompts through the activity supporting the children's Computational Thinking.

Concepts and approaches	Links	Key Questions; to prompt discussion use "I wonder why/how?"
Logic	A key part of logical reasoning is explaining. In this activity, children explain their journey through the maze. They talk about which directions and turns they need to take, and explain the thinking behind their decisions. If there are multiple routes through the maze, or multiple start points, they use logical reasoning to predict which route through the maze will be quickest / best and explain their reasons.	▪ Where will you start? ▪ Which path will you take? Why? ▪ Which path would be best? ▪ How do you know?
Algorithms	An algorithm is a sequence of instructions or a set of rules to get something done. In this activity, children can order or sequence movements needed to navigate through the maze. They can use the direction cards (provided) to support this, and plan paths for a friend to follow. Children will show more progress in algorithm design if they plan their paths ahead.	▪ Practitioners can model algorithmic thinking by 'thinking aloud' as they plan their path through the maze, e.g. ▪ Which way shall I go first? ▪ Where could I go next? ▪ Which path shall I take next? ▪ I think this would be best because...
Decomposition	Decomposition is the process of breaking down a task into smaller, more manageable parts. Children consider the directions and turns needed to navigate through the maze by breaking down or decomposing the route into steps, helping children to learn and remember the sequence. Adults model creating the paths in the maze, and using self talk to work out how to decompose the task and work collaboratively. Children could also be recorded at each stage of creating the maze and then review the images and ask children what they did.	▪ How shall we do this bit? ▪ What do we need to do first? ▪ Which part shall we do next? ▪ What do we need to do to create the maze? ▪ What do we need to use? ▪ What shall we start with?
Creating	When creating the maze, children explore how they can work together and express their thoughts and ideas. This fits well with the Building Relationships Early Learning Goal. Some may suggest improvements, perhaps saying how to add paths or different ways of doing things. As children are making the maze, encourage them to test it out, debugging their design.	▪ Where will the entrance / middle be? ▪ I think this path should go here... Do you agree? ▪ Who is going to do this part? ▪ Whose turn is it?

Programming Hardware - EYFS



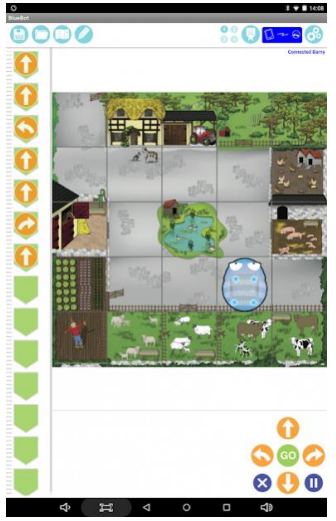
Reception – Exploration of Physical Buttons via Beebots. Exploring the importance of sequencing and the output of each input pressed. Children will be comfortable using Beebots in preparation for Year 1.

Programming Hardware – KS1

Bluebots

Y1 – Learners will be introduced to early programming concepts. Learners will explore using individual commands, both with other learners and as part of a computer program. They will identify what each command for the floor robot does, and use that knowledge to start predicting the outcome of programs. Learners are also introduced to the early stages of program design through the introduction of algorithms.

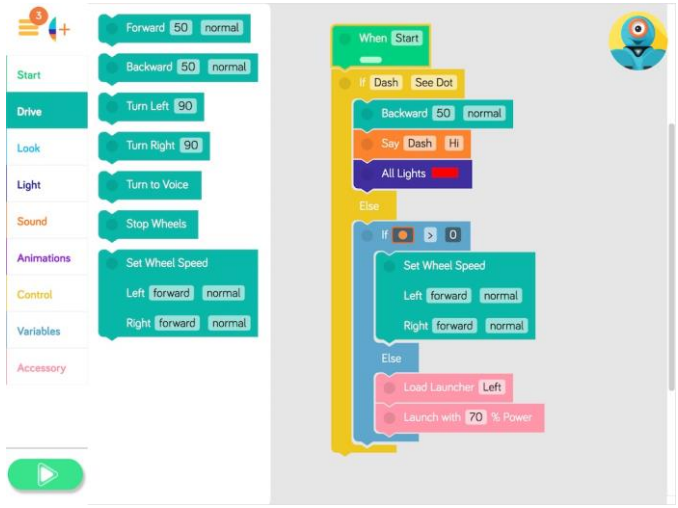
Y2 – This unit develops learners' understanding of instructions in sequences and the use of logical reasoning to predict outcomes. Learners will use given commands in different orders to investigate how the order affects the outcome. They will also learn about design in programming. They will develop artwork and test it for use in a program. They will design algorithms and then test those algorithms as programs and debug them. They will also exploring using Bluetooth to send algorithms to the robot, in preparation for Year 3.



(End of Y2 Topic) On iPads

Programming Hardware – LKS2

Dash



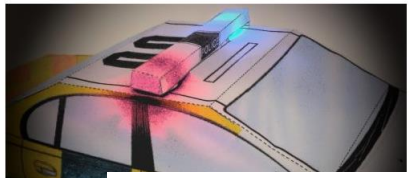
On iPads

Y3 –This unit explores the links between events and actions, while consolidating prior learning relating to sequencing. Learners begin by moving the Dash robot via Bluetooth in four directions (up, down, left, and right). They then explore movement within the context of a maze. The unit concludes with learners designing and coding their own maze-tracing program.

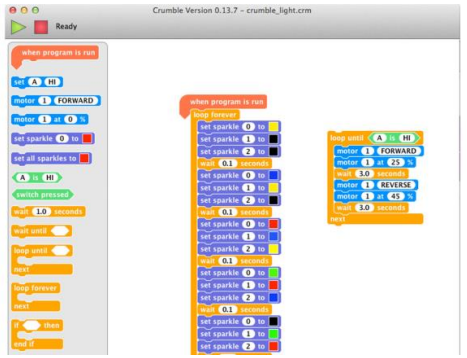
Y4 - Learners will create programs by planning, modifying, and testing commands to create shapes and patterns. This unit is the first of the two programming units in Year 4, and looks at repetition and loops within programming before applying this to Scratch.

Programming Hardware – UKS2

Police Lights Project



Lighthouse Project



On PC

Y5 - learners will use physical computing to explore the concept of selection in programming through the use of the Crumble programming environment. Learners will be introduced to a microcontroller (Crumble controller) and learn how to connect and program it to control components (including output devices — LEDs and motors). Learners will be introduced to conditions as a means of controlling the flow of actions in a program. Learners will make use of their knowledge of repetition and conditions when introduced to the concept of selection (through the 'if…then…' structure) and write algorithms and programs that utilise this concept.

Y6 - This unit is the final KS2 programming unit and brings together elements of all the four programming constructs: sequence from Year 3, repetition from Year 4, selection from Year 5, and variables (introduced in Year 6 – 'Programming A' to create their own, real-world applicable project.

EYFS only to focus on physical programming to embed understanding of computational thinking.

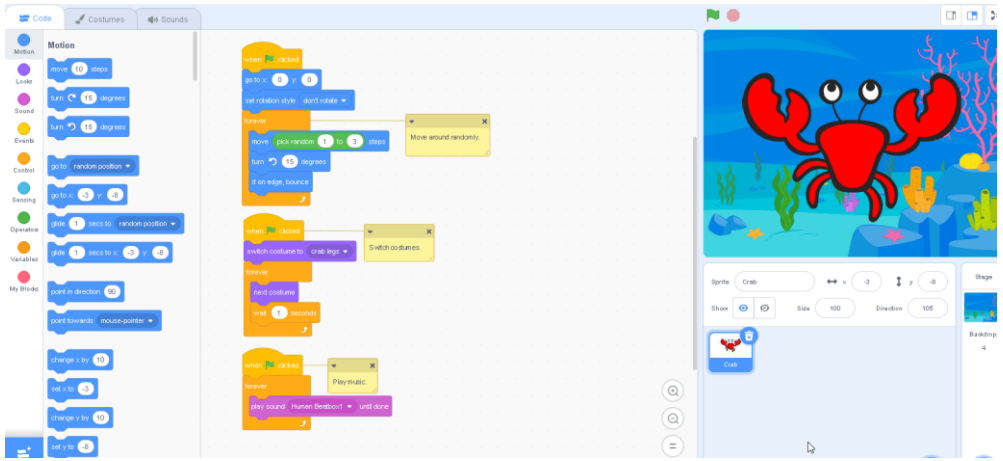
Programming Software – KS1

On iPads

Y1 – Learners will be introduced to on-screen programming through ScratchJr. Learners will explore the way a project looks by investigating sprites and backgrounds. They will use programming blocks to use, modify, and create programs. Learners will also be introduced to the early stages of program design through the introduction of algorithms.

Y2 – Learners begin to understand that sequences of commands have an outcome, and make predictions based on their learning. They use and modify designs to create their own quiz questions in ScratchJr, and realise these designs in ScratchJr using blocks of code.

Programming Software – LKS2

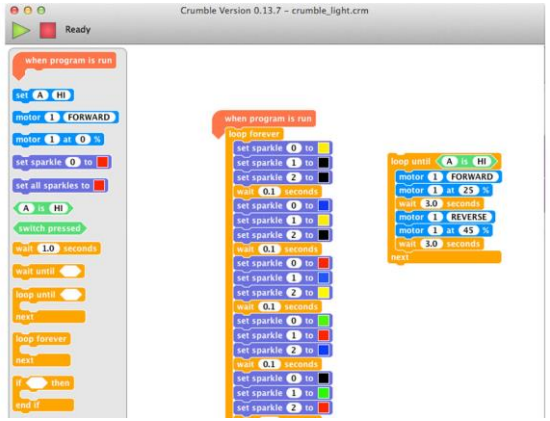


On PC

Y3 – This unit explores the concept of sequencing in programming through Scratch. They will be introduced to a selection of motion, sound, and event blocks which they will use to create their own programs, featuring sequences.

Y4 – Learners look at the difference between count-controlled and infinite loops, and use their knowledge to modify existing animations and games using repetition. Their final project is to design and create a game which uses repetition, applying stages of programming design throughout.

Programming Software – UKS2

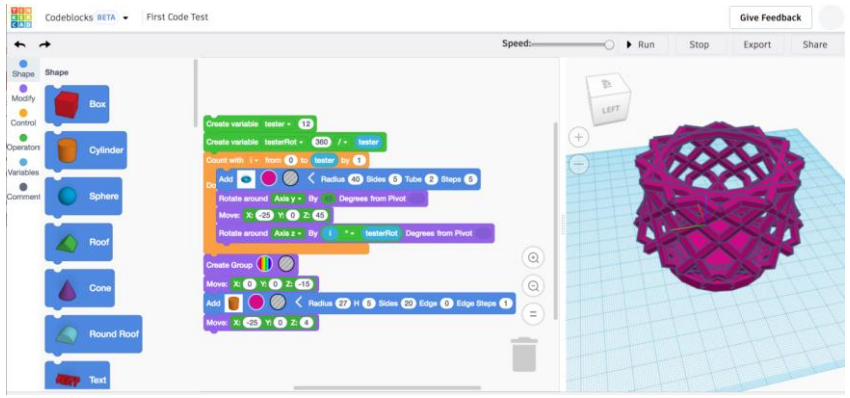


On PC

Y5 – Scratch – Learners will develop their knowledge of 'selection' by revisiting how 'conditions' can be used in programming, and then learning how the 'if… then… else…' structure can be used to select different outcomes depending on whether a condition is 'true' or 'false'.
Y5 – Will applying their block coding to crumble software

Y6 – learners find out what variables are and relate them to real-world examples of values that can be set and changed. Following the Use-Modify-Create model, learners experiment with variables in an existing project, then modify them, before they create their own project. Learners apply their knowledge of variables and design to improve their games in Scratch.
Y6 – Will applying their block coding to crumble software

Programming Software – UKS2



On PC

Y6 – TinkerCAD Codeblocks - Following the Y3-5 lessons using TinkerCAD to model designs, Y6 pupils will use their knowledge of both CAD and coding (both from Scratch and Crumbles) to write code to design a pencil pot. Large focus on real world applicaition of Computing.

Key Focus Points

	Programming A	Programming B	Creating Media – 3D Printing
Year 1 Focus: Directional Language	explore a new programming environment – Bluebots four direction commands to make sequences plan a simple program To find more than one solution to a problem	explore a new programming environment – Scratch Jr identify the effect of changing a value explain that each sprite has its own instructions algorithm to create a program	
Year 2 Focus: Writing programs	what happens when we change the order of instructions predict the outcome of a program create and debug a program that I have written **Using bluetooth**	create a program using a given design change a given design create a program using my own design decide how my project can be improved	
Year 3 Focus: Using codeblocks (inputs and outputs)	explore a new programming environment – Scratch identify that commands have an outcome explain that a program has a start recognise that a sequence of commands can have an order	explore a new programming environment – Dash Robots Using event blocks How to debug code efficiently Adding additional elements to an algorithm	
Year 4 Focus: Loops	decompose a task into small steps create a program that uses count-controlled loops to produce a given outcome explain what 'repeat' means modify a count-controlled loop to produce a given outcome	explain that in programming there are infinite loops and count-controlled loops develop a design that includes two or more loops which run at the same time	
Year 5 Focus: Conditions	explore a new programming environment – Crumbles control a simple circuit connected to a computer design a physical project that includes selection write a program that includes count-controlled loops explain that a loop can stop when a condition is met explain that a loop can be used to repeatedly check whether a condition has been met	explain how selection is used in computer programs relate that a conditional statement connects a condition to an outcome explain how selection directs the flow of a program	
Year 6 Focus: Variables/Sensing	define a 'variable' as something that is changeable choose how to improve a game by using variables Use variables within a project	connect and program sensors attached to crumbles Application of variables when using sensors design a real-world project that uses a controllable device	explore a new programming environment – TinkerCAD Codeblocks 3D Printing Transferable skills Real world application of skills Applying knowledge of CAD software (from years 3-5) and coding knowledge from Scratch and Crumbles