National Curriculum Aims:

- 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

National Curriculum Aims Key Stage 1:

- **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.**

	Year One	Year Two
Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions.	• Understand what algorithms are • Understand how algorithms are implemented on digital devices	• Understand how algorithms are implemented on digital devices • Understand that programs execute by following precise and unambiguous instructions
Create and debug simple programs.	• Write own simple algorithm • Program a device • Reorder a sequence of instructions to debug an algorithm	• Create programs on a variety of digital devices (e.g. BeeBots, iPads) • Use sequencing to write own simple algorithm • Identify and fix simple bugs • Say what works and what doesn't work about my algorithm
Use logical reasoning to predict the behaviour of simple programs	• Read through simple instructions • Say what it might do and give a reason	• Read through instructions • Say what it might do and give a reason
Computational thinking concepts	Understand and use computational thinking concepts: • Patterns • Algorithm •	Understand and use computational thinking concepts: • Patterns • Algorithm
Example activities and resources	• Algorithm dance moves activity (Barefoot) • Bee-Bot tinkering activity (Barefoot) • Bee-Bot basics activity (Barefoot) • Lego building algorithm activity (Barefoot) • ScratchJr tinkering activity (Barefoot) • Run a Race (ScratchJr) • Daisy the Dinosaur app	• Crazy character algorithms (Barefoot) • Bee-Bots 1,2,3 programming (Barefoot) • Jam sandwich bot (http://code-it.co.uk/unplugged/jamsandwich) • ScratchJr Knock-Knock joke (Barefoot) • Scratch Jr activities (http://scratchjr.org/teach/activities)

National Curriculum Aims Key Stage 2

- **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
- Appreciate how [search] results are selected and ranked

	Year Three	Year Four	Year Five	Year Six
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts	• To create a sequence of commands to produce a given outcome (Scratch musical instrument, maze program)	• To use indefinite loops and count-controlled loops to produce a given outcome (program, game)	• Design and write a simulation • Understand some situations in which a simulation might be used • Give some examples of physical systems • *Control a physical system (micro:bit)* • Solve problems by decomposing them into smaller parts	• Design and write an animation with two or more sprites • Design and write a game • 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	• To explain what a sequence is • To identify that a program needs a sequence of commands • To order commands in a program • To explain that the order of commands can affect the outcome	• To identify tasks that include repetition as part of a sequence • To explain we can use a loop command in a program to repeat instructions • To identify a loop within a program • To use indefinite loops and count-controlled loops to produce a given outcome • To explain the importance of instruction order in a loop	• Use sequence, selection and repetition within a program • Use variables to keep score within a game • *Use a micro:bit to investigate input and output*	• Use sequence, selection and repetition within a program • Use variables to keep score and time limits within a game
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs	Use logical reasoning to: • Detect an error in a simple algorithm	Use logical reasoning to: • Explain how a simple algorithm works • Work systematically to detect and correct errors in own algorithms (debugging)	Use logical reasoning to: • Explain how given algorithms work • Work systematically to detect and correct errors in own and given algorithms and programs	Use logical reasoning to: • Explain how sections of peer and given algorithms work • Detect and correct errors in own, given and peer algorithms and programs
Computational thinking concepts	Understand and use computational thinking concepts: • Algorithms • Abstraction • Decomposition • Patterns	Understand and use computational thinking concepts: • Algorithms • Logic • Patterns • Abstraction • Decomposition	Understand and use computational thinking concepts: • Algorithms • Logic • Patterns • Abstraction • Decomposition • Evaluation	Understand and use computational thinking concepts: • Algorithms • Logic • Patterns • Abstraction • Decomposition • Evaluation
Example activities and resources	NCCE Programming – Sequencing in music NCCE Programming – Events and actions https://teachcomputing.org/curriculum Fossil Formation animation (Barefoot) Animated poem decomposition (Barefoot)	NCCE Programming – Repetition in shapes NCCE Programming – Repetition in games https://teachcomputing.org/curriculum Make Music (Scratch) Maths quiz selection (Barefoot)	• Solar System simulation (Barefoot) • Classroom Sound Monitor (Barefoot) BBC Bitesize (Controlling physical systems) https://www.bbc.co.uk/bitesize/clips/z2qxhyc	• Viking Raid animation (Barefoot) • Make a Game project (Barefoot) • Ghostbusters (Scratch)

- Work with various forms of input and output

Inputs	Outputs
Keyboard	Computer display
Smart screen	BeeBot motor
BeeBot	Speakers
Microphone	WeDo motor
Makey-Makey	
Microsoft Kinect	
Wedo sensor	
Webcam sensor	