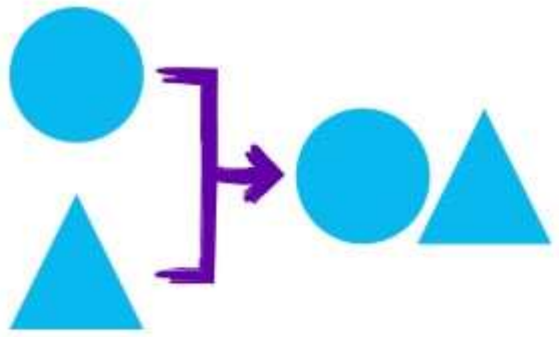
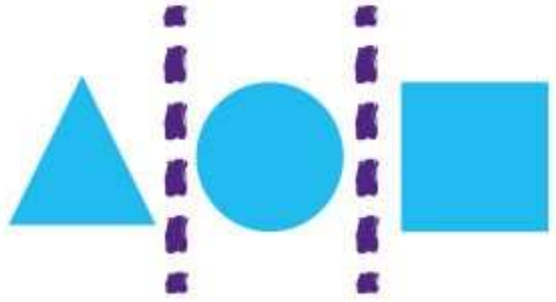
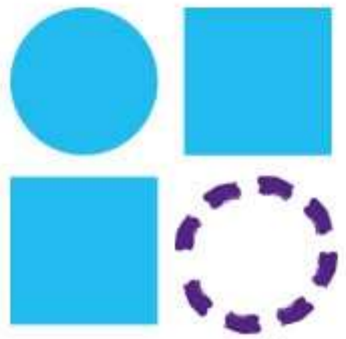
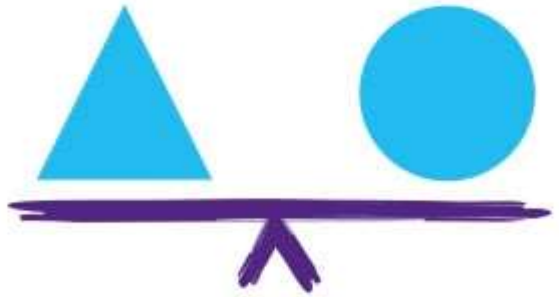
Computational Thinking

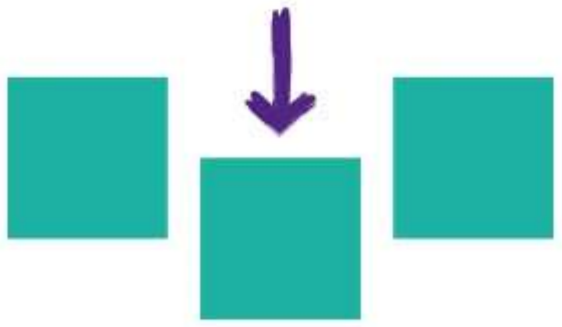
Concept	EYFS	KS1
Logic	• Children start to reason about the world around them. • Children play with mechanical and electronic toys to start forming ideas about how they work. • Provide scenarios for children to predict and test. E.g. they might predict that big things sink and small things float. To test this, we might model trying different objects and then introduce a balloon and a stone.	• Use logical reasoning to predict the behaviour of simple programs, including both their own (perhaps for Scratch or a floor turtle) and other software (such as a game or a painting program).
Examples in other areas of the curriculum	English - children might use it to explain a character's actions in a story so far, and to predict what the character will do next. Science - children should be able to explain how they've arrived at certain conclusions from the results of experiments. History - children should understand how our knowledge is constructed from a variety of sources, and they should be able to discuss the logical connections between cause and effect. Design Technology - children need to reason what material is best suited to each part of a project.	
Algorithms	• Teachers naturally create opportunities for sequencing, which is a key element of algorithms. • Children learn to take turns with others, to tidy up and line up. • Sequencing happens during roleplay activities; for example, the	• There are many opportunities within the school day for children to understand algorithms and create their own. • The algorithms pupils create can often be implemented using programmable toys or "human

	events which occur when we go to post a letter at the Post Office.	robots", and it can be useful for pupils to compare how a square is drawn with a floor turtle and with Logo or ScratchJr. • As the children break down larger tasks into smaller instructions, they also develop their use of decomposition to solve a problem.
Examples in other areas of the curriculum	Instructional writing in English, the method for a science experiment: each can be considered an algorithm. PE – getting dressed for a PE lesson (following a sequence of steps) Maths – children's approach to mental arithmetic might be an implementation of a simple algorithm.	
Decomposition 	• When children label simple diagrams and sequence familiar processes, they start to see that breaking things down into their parts helps us learn about them. • In roleplay, children could think about how to set up a shop – they'll need merchandise, price tags, a till, money for change, etc. • Constructing a model plane, they make the wings, add these to the body, add the wheels: the children think about the parts and then assemble them. • It is important to model these skills and take them a step further by	• Model how to take this further by encouraging children to evaluate whether they've missed aspects and to share their understanding with others.

	showing how to evaluate that all the necessary things are present.	
Examples in other areas of the curriculum	Any task or project will need to be decomposed into smaller, more-manageable parts. Decomposition is everywhere. Humanities - concept maps are more detailed. In exploring detail, children increase their awareness and independence. Science - children should have ongoing opportunities to break things down into their constituents – e.g. a lifecycle and its stages, a diagram and its labelled parts.	
Patterns 	• Children are given practical situations where they can notice patterns, observing and exploring similarities and differences. • They can be presented with sets of items which are sortable in various ways. • For example, they could be given a water tray and assorted objects, some of which float.	• Children continue to engage in practical experiences where similarities and differences can be explored. The range and complexity of these scenarios increase. • Model how to notice patterns, how to think of rules and how to try them out.
Examples in other areas of the curriculum	Children become familiar with repeated phrases in nursery rhymes. Reading - children notice repeated structures in stories. Music - repeating lines in many musical forms Maths - children typically undertake investigations in which they spot patterns and deduce generalised results, look for patterns in more-abstract concepts, including odd and even numbers, negative numbers, multiples and inverse operations. English - children spot more-complex spelling patterns, and they listen for patterns in sounds (phonemes). Science – group and classify – children will notice rules and patterns, for example in animals' appearance and habitat or in the properties of materials, and they'll draw on those patterns to make predictions in other investigations.	

Abstraction	• Opportunities to summarise as children remember events and recount what was important. • In maths, they start to sense the abstraction of number: they can count three bears, three bricks, three friends, etc and formulate an abstraction of ‘three-ness’.	• Starting to identify the important elements and ignoring unnecessary detail.
Examples in other areas of the curriculum	Maths - working with word problems often involves identifying key information and thinking how to represent it in the more abstract language of arithmetic. Music - abstracted to notation. Geography - learning how to add places of interest and to ignore detail; they use world maps and create local maps and so start to see different layers of abstraction. Children can also gain experience of abstraction when playing computer games, appreciating that these interactive simulations are based on real life, but are simpler. History - children consider viewpoints as they roleplay famous people	
Evaluation	• Children can start to develop their evaluation skills as they articulate their judgements and reasons in simple terms, such as “My dog is my favourite pet because she lets me pat her.” • Children to consider different ways to find out things. They can find out about dinosaurs by reading a text, browsing a picture book, using a CD	• Children express preferences more readily and clearly • Children can undertake many different computing activities which include simple evaluation. They can be introduced to the idea of design goals and criteria and may begin to create their own. • Designing algorithms for a Bee-Bot moving between two points, they can evaluate the most effective route, for example the shortest.

	ROM or entering keywords into a search engine.	• They can have criteria for designing a Bee-Bot maze – a start, an end, a minimum number of obstacles – and for the Bee-Bot itself: they might want it to navigate the maze without striking anything. • Children can refer to the design criteria and judge if they've been met.
Examples in other areas of the curriculum	Self- and peer-assessment can help to develop children's evaluation skills, as they make judgements using success criteria and consider potential improvements. PE - a list of 'good' things to aspire to – perhaps certain moves in a routine, perhaps landing on two feet. English - the success criteria for a child's written work might be the correct use of capitals and full stops, or the inclusion of adjectives and adverbs. They may recommend a book to a friend, explaining why they think it will be enjoyed, having made a judgement about what type of books might be favoured. Design Technology - makes use of evaluation as pupils work through the design–make–evaluate cycle.	
Approach	**EYFS**	**KS1**
Tinkering	• Children often demonstrate a love for tinkering. • Early Years environments typically abound with opportunities to try things out and ask open questions such as "Why?" and "How?" • Children tinker with digital devices like remote-control toys, programmable floor turtles, cameras	• Children might tinker with Bee-Bots, Roamer®, ScratchJr, Daisy the Dinosaur, Logo and other programming environments, working out how to use them and thinking about what they might be used for – even noting their similarities and differences – before

	and computers; they explore online games and simulation environments.	using these tools to solve a problem or to create something. • Give children the time to direct their own exploration. • Ask them open-ended questions; challenge them to find out more and to reflect on what they've discovered.
Examples in other areas of the curriculum	When introducing any new concept, start by tinkering and exploring	
Creating 	• Children may create shows with programmable toys or use simple art software. • Encourage pupils to be imaginative, to create simple plans, to invest care in their work and to think how it might be improved.	• Children should be taught to "use technology purposefully, to create, manipulate, organise, store and retrieve digital content." • Provide opportunities for them to create and debug their own programs. • Children could design scripted animations in an online programming environment like ScratchJr, using a timeline or storyboard to work out their algorithm before converting this into code for the sprites. • Challenge them to have programmable toys perform a story or choreographed dance: they'll need to work together to agree characters, make props like mats

		and backdrops – perhaps even clothes – plan the movements, write the code and debug it.
Examples in other areas of the curriculum	Art and Design - They can explore the process of creating in a making area as they assemble, cross-stitch or even bake. Art/Music/DT - ask them to think about creating something and how they'll work through the necessary steps, decomposing a complex process into several planned phases.	
Debugging 	• Children explore (tinker) in the roleplay corner to try things out and learn how things work. • Children continually finding problems and fixing them: essentially, they're debugging. • They might organise a shoe shop, serve a customer and realise they can't find matching pairs of shoes, so they'll reorganise the shoes to solve this 'bug' in their shop display.	Children improve their work and, as part of checking (evaluation), they find and fix their own and other's errors. In computing lessons, with a turtle-graphics program (in Logo or similar), pupils can act out the role of the turtle, walking and turning as they follow the code of the commands line by line – something Seymour Papert described as "playing turtle". hinking back to the original problem, they can check if something has gone wrong. When programming toys to navigate a maze, pupils can test half of their program at a time to see in which half the bug lies. Having identified which, it can again be halved and the testing repeated. This method of homing in on the precise location of a bug is known as the "wolf-fencing algorithm". Deliberately giving pupils 'buggy code' teaches them that bugs are to be expected and enables them to exercise the skills required to locate and fix errors including perseverance.

Examples in other areas of the curriculum	English – A sentence with spelling or grammatical errors to correct	
Persevering	• Encourage patience and persistence. • Arrange quiet areas for puzzles, space for building models, activities that span days if not weeks (such as growing seeds), and storage for half-finished models or drawings to which children can return. • Use opportunities to talk about not giving up, such as reading "Incy Wincy Spider", "The Tortoise and the Hare", "Rosie Revere Engineer" or "Horton Hatches the Egg". • Praise pupils when they have a go and keep going.	• Foster a have-a-go climate in class. • When a child is stuck, encourage independence. • Model a patient, logical approach to problem-solving. • Demonstrate how to decompose problems into manageable parts and how to tackle each in turn. You could even create scenarios where pupils do not complete tasks at the end of a lesson, so that they become accustomed to returning to a problem later. • Teach them the steps of debugging and allow them lesson time to practise looking carefully and persevering. Have them collaborate with a programming partner. • Provide support materials, help sheets and displays. • Make it clear that this is a normal part of programming: finding bugs and working on them should be celebrated just as much as actually fixing them.
Examples in other areas of the curriculum	Carol Dweck's work on "growth mindsets" suggests that hard work and perseverance in the face of difficulties are key to educational outcomes.	

	Encourage pupils to look for strategies they can use when they do encounter difficulties, such as working out exactly what the problem is, using search engines to find a solution, or asking a friend.	
Collaborating	• Encourage pupils to develop their emerging collaborative skills simply by waiting in line, taking turns or sharing resources. • Let them practise explaining and listening to peers. • Give them activities to work on together, with or without staff support. • In computing, have them help each other work out how to use new programs.	• Provide pupils with opportunities to collaborate in computing. • Children can work in pairs or groups, programming toys and designing algorithms. • Encourage the children to explain their ideas, to talk through bugs, to help one another, to set each other challenges and to think carefully about how to give specific feedback.