

Sequence of Lessons / Unit	Approx. time req'd	Year A or B	Strand		Knowledge and understanding						Strand: Processes and production skills					
			Digital systems		Representation of data		Collecting, managing and analysing data		Creating digital solutions by:							
			CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	Investigating and defining	Evaluating	Collaborating and managing					
Content Description	Recognise and explore digital systems (hardware and software components) for a purpose (ACTDIK001)	Recognise and explore patterns in data and represent data as pictures, symbols and diagrams (ACTDIK002)	Collect, explore and sort data, and use digital systems to present the data creatively (ACTDIP003)	Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems (ACTDIP004)	Explore how people safely use common information systems to meet information, communication and recreation needs (ACTDIP005)	Create and organise ideas and information using information systems independently and with others, and share these with known people in safe online environments (ACTDIP006)	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #		
An intro to algorithms		F-1	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>		<input type="checkbox"/>		<input checked="" type="checkbox"/>	3	<input type="checkbox"/>		<input type="checkbox"/>			

Years F-2 Achievement Standard	Years 3 and 4 Achievement Standard
By the end of Year 2 <ul style="list-style-type: none"> Students identify how common digital systems (hardware and software) are used to meet specific purposes. (1) They use digital systems to represent simple patterns in data in different ways. (2) Students design solutions to simple problems using a sequence of steps and decisions. (3) They collect familiar data and display them to convey meaning. (4) They create and organise ideas and information using information systems, and share information in safe online environments. (5) 	By the end of Year 4 <ul style="list-style-type: none"> Students describe how a range of digital systems (hardware and software) and their peripheral devices can be used for different purposes. (1) They explain how the same data sets can be represented in different ways. (2) Students define simple problems, design and implement digital solutions using algorithms that involve decision-making and user input. (3) They explain how the solutions meet their purposes. (4) They collect and manipulate different data when creating information and digital solutions. (5) They safely use and manage information systems for identified needs using agreed protocols and describe how information systems are used. (6)

An introduction to algorithms

Students should be provided with opportunities to explore algorithms through guided play, including hands-on, kinaesthetic and interactive learning experiences. Students begin to develop their design skills by conceptualising algorithms as a sequence of steps or procedures for carrying out instructions to solve problems or achieve certain things. These skills could include identifying steps in a process or controlling a Bee-Bot. Provide authentic and meaningful ways to introduce students to simple programming while consolidating concepts across other subject areas.

Flow of activities				
Short text	Following instructions Follow teacher-provided instructions and explore the problem in detail.	Describe a sequence Provide instructions to complete a familiar task, or sequence a story or event.	Instructing a robotic device Carry out instructions using a digital system such as a robotic device.	Algorithmic thinking Conduct a longer term project that incorporates algorithmic thinking.
Questions to guide exploration	<i>Can you follow steps to complete a task?</i>	<i>Can you describe and represent a sequence of steps to solve a problem?</i>	<i>How can I carry out instructions using a robotic device?</i>	<i>Inquiry focus question</i>
AC Alignment	Investigating and defining (ACTDIP004)	Investigating and defining (ACTDIP004)	Investigating and defining (ACTDIP004) Digital systems (ACTDIP001)	Investigating and defining (ACTDIP004)
What’s this about?	<p>Breaking down a problem into smaller parts, or steps, is an important part of computational thinking.</p> <p>When initially defining problems you need to be able to identify the key features of the problem. These features become the scaffold or structure of the problem, and they then help frame the solution.</p> <p>Typically, when we describe to another person a solution to a problem or task, we only mention the key features or instructions. We omit less important ones because we assume that people know what we mean. Focusing on important details and ignoring aspects that are not relevant is part of the process of abstraction.</p> <p>When providing instructions to a computer, it is necessary to give specific instructions about sequences and decisions because a computer cannot make assumptions. It is integral in programming to only focus on important details, ignoring aspects that are not relevant.</p> <p>At this level, students should be able to describe, follow and represent an algorithm, which is a sequence of steps for carrying out instructions to solve a problem or achieve a desired outcome. It is suggested that students begin their learning by following instructions issued by others about completing a task. This will help them to build knowledge and skills in abstraction as well as skills in writing algorithms.</p>	<p>Describing a sequence involves students thinking logically – deciding the order in which instructions need to be followed. Students could give verbal instructions to another student to complete a task.</p> <p>When representing an algorithm, the steps and instructions need to be documented so that the task can be carried out as planned. There are different ways that instructions can be documented; for example, cards or images could be arranged to indicate a sequence; or text instructions could be provided.</p> <p>Some methods are more efficient than others.</p> <p>Students will begin to recognise patterns in their instructions and consider how these could be shown.</p>	<p>At this level, students can carry out instructions using a digital system such as a robotic device.</p> <p>Students develop their design skills by formulating a sequence of steps to solve a problem or achieve a desired outcome. This could be identifying steps to control a Bee-Bot, or using programming cards to provide instructions.</p> <p>The Bee-Bot can be thought of as a piece of hardware and therefore linked into the way digital systems operate with an input and output.</p>	<p>Algorithms are used around us every day. For example, algorithms are used to control traffic lights, the scores in a computer game and the advertisements on websites. Any digital automated process is instructed to behave in a particular way because of the decisions and steps that have been designed and carried out for the solution. Note: An algorithm can be used not only to describe a digital solution, but also to describe an everyday process such as cleaning teeth or making a salad.</p> <p>In this activity, students inquire into an everyday process, and represent the decisions made and steps taken to carry out that process.</p> <p>Students connect algorithms to everyday events; this helps them to recognise the relevance of what they are learning.</p>
The focus of the learning (in simple terms)	<p>The focus of this learning is on students being able to follow instructions.</p> <p>Begin by selecting some students to carry out teacher-provided instructions such as placing a book on another student’s table or putting some rubbish in the bin.</p> <p>Provide a ‘normal’ instruction, one that does not explicitly state every move or action. This illustrates the key features of a solution, but it hides lots of detail.</p> <p>Students can suggest all of the other instructions that were not given, and then discuss that because digital</p>	<p>Students can begin exploring algorithms by looking at familiar activities. Ask students to identify the steps involved in a task and to create instructions for someone to follow. Relevant examples might include cleaning teeth, getting ready for school or setting the table.</p> <p>Write a step on each card, and then put the cards in a logical order. Discuss what happens if you do steps out of order.</p> <p>Simple sequencing can also be used in a linked literacy lesson to discuss the order of events in a familiar story. Students use abstraction to break down a story into its most basic parts when retelling the story.</p>	<p>Introduce a programmable robot such as a Bee-Bot to help students develop computational thinking skills.</p> <p>Set challenges for students. Students will need to program the Bee-Bot by a sequence of commands to complete a particular task.</p> <p>Provide authentic and meaningful ways to introduce students to simple programming while consolidating concepts across other subject areas. For example, you could ask students to spell out words by programming the Bee-Bot to land on letters. Alternatively, students could estimate and</p>	<p>Conduct a longer term project in which students design, represent and follow an algorithm. Growing a seed is a common classroom activity. You could align that activity with computational thinking and algorithm design.</p> <p>Students design a maze using construction blocks. This task can be used to integrate programming. After designing the maze students can use ready-made laminated programming cards to move a character to a desired destination. Working in pairs one student creates the sequence and the other follows it.</p>

	<p>systems are not human they need specific instructions.</p>	<p>Students describe steps for a particular task. They compare different ways to do the same task.</p> <p>Identify parts of the algorithms that are different and informally measure the time taken to complete the same tasks using different algorithms. For example, compare setting the table by getting all the utensils out first and then laying them out as opposed to setting cutlery for one person at a time, and then returning after each to get more cutlery.</p> <p>Support students to understand that different methods can be used to solve problems or complete tasks.</p> <p>Next, introduce the use of arrows as a form of representation.</p> <p>Mark out a grid on the floor; for example, 5x5. Place an item in one of the squares and challenge a student to describe how to move to that square.</p> <p>Creating instructions to move along a grid reinforces directional language: forward, backwards, turn right, turn left and stop.</p> <p>Represent the instructions for the class to follow. Discuss efficiencies as you go. What is the fewest number of steps needed to get to the target?</p>	<p>measure how many moves the Bee-Bot needs to reach a target.</p> <p>Use programming cards to develop programming skills.</p>	
<p>Supporting resources and tools and purpose/ context for use.</p>	<p>Introducing algorithms The learning hook is one possible approach to introducing algorithms.</p> <p>Fairy tale fun Use the slide sorter function to arrange a set of presentation slides in correct sequence to retell a fairy tale.</p>	<p>First to finish Compare algorithms designed to complete the same task, and evaluate each for efficiency.</p> <p>Bot logic This interactive game helps students to develop early programming skills. Students create simple sequences of instructions based on logic.</p> <p>Bee-Bot mobile app This app makes use of Bee-Bot's keypad functionality. Students develop their skills in using directional language and in programming through sequences of forwards and backwards, and left and right turns.</p>	<p>Buzzing with Bee-Bots Follow and describe a series of steps to program a robot.</p> <p>Bee-Bot ruler Explore the concept of sequencing steps, using Bee-Bots to measure length.</p> <p>Skip-counting with Bee-Bots Play a skip-counting game where students program the Bee-Bot to stop at multiples of a set number, eg 2, 4, 5, 10, on a number grid.</p> <p>Spelling bee Write a set of instructions that program a Bee-Bot to move to letters to spell out a word on an alphabet grid.</p> <p>Goldilocks, the three bears and BeeBot Students explore a sequence of steps using Bee-Bots in the context of familiar narratives.</p>	<p>Planting fruit and vegetables Students grow a plant from seed. They capture each step and decision as an algorithmic process and record data for future learning.</p>
<p>Assessment</p>	<p>Suggested approaches Participate in following instructions and contribute to discussions about hidden instructions.</p> <p>Achievement standard Design solutions to simple problems using a sequence of steps and decisions.</p>	<p>Suggested approaches</p> <ul style="list-style-type: none"> Retell stories, with relevant parts of the story sequenced in the correct order. Prepare a set of instructions that are then followed by another student. <p>Achievement standard Design solutions to simple problems using a sequence of steps and decisions.</p>	<p>Suggested approaches</p> <ul style="list-style-type: none"> Demonstrate two instructions carried out by the robotic device. Label a diagram of a robotic device, identifying the hardware and software components. <p>Achievement standard Design solutions to simple problems using a sequence of steps and decisions.</p> <p>Identify how common digital systems (hardware and software) are used to meet specific purposes.</p>	<p>Suggested approaches Present or demonstrate an algorithm to describe a common event.</p> <p>Achievement standard Design solutions to simple problems using a sequence of steps and decisions.</p>

