

Danson Primary School- Computer Science

Computer Science		Digital Literacy				Information Technology	
	EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Computational Thinking	<ul style="list-style-type: none"> I can follow simple oral algorithms I can spot simple patterns I can sequence simple familiar tasks 	<ul style="list-style-type: none"> I can understand what algorithms are I can write simple algorithms I can understand that the sequence of algorithms is important I can debug simple algorithms I can understand that algorithms are implemented as programs on digital devices 	<ul style="list-style-type: none"> I can write algorithms for everyday tasks I can use logical reasoning to predict the outcome of algorithms I can understand decomposition is breaking objects/processes down I can implement simple algorithms on digital devices I can debug algorithms 	<ul style="list-style-type: none"> I can create algorithms for use when programming I can decompose tasks into separate steps to create an algorithm I understand abstraction is focusing on important information I can identify patterns in an algorithm I can use repetition in algorithms 	<ul style="list-style-type: none"> I can use abstraction to focus on what's important in my design I can write increasingly more precise algorithms for use when programming. I can use simple selection in algorithms I can use logical reasoning to detect and correct errors in programs 	<ul style="list-style-type: none"> I can solve problems by decomposing them into smaller parts I can use selection in algorithms I can recognise the need for conditions in repetition within algorithms I can use logical reasoning to explain how a variety of algorithms work I can use logical reasoning to detect and correct errors in algorithms I can evaluate my work and identify errors 	<ul style="list-style-type: none"> I can recognise, and make use, of patterns across programming projects I can write precise algorithms for use when programming I can identify variables needed and their use in selection and repetition I can decompose code into sections for effective debugging I can critically evaluate my work and suggest improvements
Coding and Programming	<ul style="list-style-type: none"> I can access and use simple activities using touch technology with increasing control. I can use a mouse, touch screen or appropriate access device to target and select options on screen I can use a range of control toys and devices. I can input a simple sequence of commands to control a digital device with support (Bee Bot) 	<ul style="list-style-type: none"> I can create a simple program e.g. sequence of instructions for a Bee Bot I can use sequence in programs I can locate and fix bugs in my program 	<ul style="list-style-type: none"> I can understand that programs are executed by following precise and unambiguous instructions I can create programs on a variety of digital devices I can debug programs of increasing complexity I can use logical reasoning to predict the outcome of simple programs 	<ul style="list-style-type: none"> I can design and create programs I can write programs that accomplish specific goals I can use repetition in programs I can work with various forms of input 	<ul style="list-style-type: none"> I can use simple selection in programs I can work with various forms of output I can use logical reasoning to systematically detect and correct errors in programs I can work with various forms of output 	<ul style="list-style-type: none"> I can create programs by decomposing them into smaller parts I can use selection in programs I can use conditions in repetition commands I can work with variables I can create programs that control or simulate physical systems I can evaluate my work and identify errors 	<ul style="list-style-type: none"> I can use a range of sequence, selection and repetition commands combined with variables as required to implement my design I can create procedures to hide complexity in programs I can identify and write generic code for use across multiple projects I can critically evaluate my work and suggest improvements I can identify and use basic HTML tags
Computer Networks (KS2 only)				<ul style="list-style-type: none"> I can understand that computers in a school are connected together in a network I can understand why computers are networked I can understand the difference between the Internet and the World Wide Web (WWW) 	<ul style="list-style-type: none"> I can understand that servers on the Internet are located across the planet I can understand how email is sent across the Internet I can understand how the Internet enables us to collaborate 	<ul style="list-style-type: none"> I can understand how we view web pages on the Internet I can use search technologies effectively I can understand that web spiders index the web for search engines I can appreciate how pages are ranked in a search engine 	<ul style="list-style-type: none"> I can understand what HTML is and recognize HTML tags I know a range of HTML tags and can remix a web page I can create a webpage using HTML

Vocabulary	Instructions, touch screen, digital, device	algorithm, bug, computer, debug, input, logical reasoning, output, program, robot, abstraction, audio, decomposition, edit, frame, narration, pattern, recipe, storyboard, video camera	abstraction, algorithm, bug, code, debug, event, input, output, parallel processing, program, repetition, Scratch, sprite abstraction, algorithm, computational thinking, input, output, parallel processing, pattern recognition, remix, repetition, Scratch, source code, sprite	abstraction, algorithm, bug, code, debug, decomposition, event, abstraction, algorithm, bug, code, debug, event, input, logical reasoning, output, parallel processing, program, repetition, Scratch, sequence, sprite, variable	algorithm, bug, debug, input, output, program, repeat loop, repetition, Scratch, sequence, sprite, variable accelerometer, algorithm, Bluetooth, if/else if/else, JavaScript, LED, Make Code, micro: bit, object code, runtime, simulator, source code, variable	algorithm, background, bug, code, debug, iterative development, logical reasoning, program, Scratch, sprite cipher, codes, cryptanalysis, cryptography, decrypt, encode, encrypt, message, Morse code, semaphore, transmit	accelerometer, Bluetooth, controller, decomposition, edge connector, embedded system, input, interactive, light-emitting diode (LED), Make code, micro: bit, microprocessor, output, simulator, system abstraction, algorithm, binary search, decomposition, divide and conquer, graph, greedy algorithm, linear search, quick shot, search, search algorithm, selection sort, sort
National Curriculum	<p>Development Matters</p> <p>Three and Four-Year-Olds</p> <ul style="list-style-type: none"> •Talk about and identify patterns •Begin to describe a sequence of events <p>Reception</p> <ul style="list-style-type: none"> •Continue, copy and create repeating pattern •Show resilience and perseverance in the face of a challenge. •Develop their small motor skills so that they can use a range of tools competently, safely and confidently. <p>Early Learning Goals</p> <ul style="list-style-type: none"> •Be confident to try new activities and show independence, resilience and perseverance in the face of challenge (ELG - Managing Self). 	<ul style="list-style-type: none"> •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 	<ul style="list-style-type: none"> •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 	<ul style="list-style-type: none"> •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 			

