Computing DISCIPLINARY knowledge progression overview

Disciplinary knowledge enables our pupils to embed their substantive knowledge by thinking, questioning and building knowledge. In our Computing curriculum at Hollybrook our children, use programs, debug, create and program in computing.

The children know how to…

	Year R	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Use 	• Explore using different digital devices e.g., tablets, computers, beebots	• Use a keyboard to type • Use the keyboard to delete and edit text • Use the shape tool and the line tools • Use a computer on their own to paint a picture and to label	• Use a program to input data for a pictogram • Use a digital device to take a photograph both landscape and portrait • Use tools to change an image	• Use a digital device to record sound • Add content to a desktop publishing publication • Change the layout and page settings • Use a digital device to collect/show data	• Use digital devices to create video content • Use a digital device to collect data automatically • Use data collected over a long duration to find information • Use a data logger.	• Use tools to achieve a desired effect • Use a form to record information in a flat-file database • Use a digital device to capture video. •	• Use search engines • Use a computer to create and manipulate three-dimensional (3D) digital objects • Construct a digital 3D model of a physical object • Use a spreadsheet to plan an event • Apply formulas to data, including duplicating
Debug 	• Identify when something doesn't work and attempt a simple fix or ask for help by turning them on or off	• I can execute, check and debug simple codes	• Follow simple steps to fix errors in a sequence of instructions • Use logical reasoning to debug programs and fix errors	• To be able to debug and fix a few mistakes in a range of programs	• To be able to debug and fix a number of mistakes in a variety of programs	• Debug and fix a number of mistakes in a wider variety of programs	• Apply systematic debugging techniques to resolve complex algorithms

Create 	• Explore and use technology to create simple digital content (e.g., mark-making apps)	• Create digital pictures • Create labels for a picture • Begin to create and edit digital content (text, images, sound) and save work with support	• Create categories to use in a pictogram. • Create and edit digital content (text, images, sound) and save work. • Create a musical rhythm by experimenting with images, pitch and musical rhythm	• Create questions with yes/no answers • Create text, changing layout, font style, size, and colours for a given purpose. • Create animated movement with a sequence of images	• Create a sound wave, trim a recording and arrange multiple sounds to create an effect • Create different types of audio that can be combined and played together	• Create a paper-based database • Create a vector drawing by combining shapes • Create and manipulate three-dimensional (3D) digital objects.	• Construct a digital 3D model of a physical object • Create spreadsheets and charts • Create multiple web pages for their site and use hyperlinks to link them together
Program 	• Follow simple instructions and sequences in play-based activities (e.g. using remote control toys, beebots…) • Understand cause and effect in relation to technology (e.g., pressing a button)	• Create simple algorithms using logical reasoning • Use block-based coding to create simple programs • Use a code to make objects move when they are clicked on	• Create algorithms using logical reasoning • Use a code to make objects move, change direction and disappear • Combine start events and click events to make a simple game • Create their own program using their own design	• Write a computer program where different pieces of code execute in a particular sequence • Write a code that uses a timer to create a sequence of events • Create a program to move a sprite in four directions • Develop a program by adding features	• Design and create a project that includes repetition • Design and implement more complex algorithms with loops and conditions • Use multiple different variables and to set the value of a variable	• Set values in code to control the speed of an object, including the use of friction • Design and create a program which uses selection	• Change position by changing co-ordinates. • Use Python to make simple calculations and recognise symbols for multiplication (*) and division (/) • Use an conditional statement to compare a variable to a value • Programme a controllable device to use inputs and outputs