Meole Brace Primary School. Computing—Coding overview (Year 2015-2016 NB as teacher and children's knowledge develops, different software will be added) Please be aware that this is a unit which has great importance and should be taught twice in the year.

Year Group	NC Coverage	Unit Objective	Software	Possible Context Please contextualise relative to MTP
EYFS	• understand what algorithms are • create simple programs • use logical reasoning to predict the behaviour of simple programs • use technology purposefully to create. • use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.	Give verbal instructions to complete a task. Control sprite to carry out various tasks. Write the algorithm using pictures and symbols (and spot mistakes if it goes wrong).	Bee-Bot App (Primary) Daisy the Dinosaur (Extension)	• Getting changed for Muddy Mondays • Making porridge for the Bears • Brushing your teeth. • Make Bee-Bot complete tasks. • Make Daisy complete tasks
Year 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 • use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.	Use variables in programmes and control possibilities. Debug simple programmes	Bee-Bot (primary) Remote control Daisy the Dinosaur (Primary)	• Making the bee bot follow routes. • Writing simple algorithms. • Finding mistakes in teachers programming. • Introduce when and repeat tasks.

Year 2	• 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 • use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.	Use variables (repeat) Add backgrounds and change characters Add sounds Test and debug simple programs Repeat and predict patterns to create shapes Make characters talk. Use green flag button.	Scratch Junior App (Primary) Logo (Primary)	• Create a scene where characters interact if they meet (make sound if touching each other) • Create simple shapes and use repeat function to create patterns (Requires knowledge of shape and turn) Logo.
Year 3	• 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	Write a program that makes a sprite move around the screen, switching costumes, change backgrounds and add speech bubbles/extra sprites for interaction. Spot own mistakes in programs and debug them.	Scratch Logo	• Create a maze game. • Create a dance routine to music. • Create simple shapes and use repeat function to create patterns. • Create a fish tank screen saver.
Year 4	• 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a	Write a program that makes a sprite move around the screen, switching costumes, change backgrounds and add speech bubbles/extra sprites for interaction. Introduce use of timer. Spot own mistakes in programs and	Scratch Logo	• Design a game relative to MTP which is controlled by variables. • Use known facts about angle and shape to create patterns of increasing complexity by manipulating simple lines of code into complex instructions. Eg "to hexagon and repeat" to make

	range of ways to report concerns about content and contact.	debug them. Introduce cloning and random events. Add a loop to make character do the same thing. Animate the background Use the IF statement.		flower patterns. If a character touches a tree, it grows.
Year 5	• 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	Design and create an environment with characters and objects. Programme a character to move around the environment. Create, design and debug a world that allows characters to interact with objects and other characters. Collect items.	Kodu	• Create e.g. Mountain environment with rivers and other characters where sprite reacts differently depending on what it touches.
Year 6	• 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	Design, create and debug a game that features time and scores and changing other variables. Extending to multiple levels and health of character. Understand the difference between debugging and refining.	Kodu	• Viking battle game where characters lose health if hit by enemy and scores points if collects treasure. • Race game – changing speed.