Coding

Subject curriculum intent:	A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

End of KS3 intent/outcome	**End of KS4 intent/outcome**	**End of KS5 intent/outcome**
Pupils will understand how to follow simple instructions. They will be able to put together simple algorithms using physical hardware such as beebots/ blue-bots.	Pupils will have a good awareness of algorithms and how they affect everyday life. They will understanding how to put together a set of instructions to create a basic block coding programme/app. They will manipulate hardware to make it follow certain pathways	N/A

Intent for this topic:	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.
Core vocabulary needed for this subject/topic:	Code, debug, solve, instructions, algorithms, order, change, app, follow, create, directions, movement, plan
Vocabulary pupils will have accessed in other topics or subject areas:	Instructions, plan, solve, change, create, follow

Prior knowledge: what pupils may already have studied

Key stage	Subject	Topic title	Term/year taught	Content/What might pupils already know?
KS3	Computing	Coding	Summer term	Following basic instructions
KS4	Computing	Coding	Summer term	Creating simple algorithmic instructions
KS5	Computing	Coding	Summer term	Creating simple algorithmic block coding animations/apps

Cross Curricular Links: Maths, English, Science

	B2P 5-6	B2P 7-8	B2 Step 1	B2 Step 2	B2 Step 3
Theme-Reading, writing and debugging computer code					
Subject specific knowledge	Understands what an instruction is. Understands the meaning of directional instructions (i.e. arrows, start and stop etc). Understands the meaning of order/sequence. Can identify a digital device (i.e. laptop, iPad, touch screen computer).	Understands what an instruction is. Understands the meaning of directional instructions (i.e. arrows, start and stop etc). Understands the meaning of order/sequence. Can identify a digital device (i.e. laptop, iPad, touch screen computer). Understands the term "switch" and how it relates to input devices (ie, mouse, keyboard, bigmack switch, trackball, joypads).	Understands that computers need exact instructions. Understands the term algorithm: *"A programming algorithm is a computer procedure that is a lot like a recipe (called a procedure) and tells your computer precisely what steps to take to solve a problem or reach a goal. The ingredients are called inputs, while the results are called the outputs."* *https://www.bbc.co.uk/bitesize/topics/z3tbwmn/articles/z3whpv4* Understands the term "switch" and how it relates to input devices (ie, mouse, keyboard, big mack switch, trackball).	Understands that computers need exact instructions. Understands the term algorithm: *"A programming algorithm is a computer procedure that is a lot like a recipe (called a procedure) and tells your computer precisely what steps to take to solve a problem or reach a goal. The ingredients are called inputs, while the results are called the outputs."* Understands the difference between programmer (writes the code), code (code is what we use to write out the algorithm, we will use symbol based coding) and algorithm (the steps you want the computer to take). Understands the term "logical reasoning" when applied to reading and interpreting algorithms. Understands the term debugging (correcting errors within algorithms). Understands that algorithms are present within digital devices and are contained within the software for that device.	

Subject specific skills	Is able to sequence (order) a 3, 4 and 5 step event on a computer screen (i.e. morning routine, a trip to the shops). Is able to identify and correct errors within a 3, 4 and 5 step event. Is able to identify directional symbols (i.e. arrows, start and stop) within a 3x3 and 5x5 matrix on a digital screen. Is able to match paper to digital directional symbols. Is able to follow a 3, 4 and 5 step visual sequence in order to program a Beebot.	Is able to use simple directional instructions in order to direct the movement of a student. Is able to follow simple directional instructions. Is able to sequence directional and programming symbols in order to plan a specific route for a Beebot. Is able to input these symbols into a Beebot. Is able to read a sequence of directional and programming symbols for a Beebot and interpret the route the Beebot will take. Is able to identify and correct errors	Is able to use complex directional instructions in order to direct the movement of a student. Is able to follow complex directional instructions. Is able to recognise the use of non-computer algorithms within our daily life (i.e. recipes, directions, flat pack / lego instructions) Is able to read and interpret a range of non-computer algorithms. Is able to identify and correct errors within non-computer algorithms Is able to write simple non-computer algorithms. Is able to state a definition of an algorithm. Is able to list devices and software that can be controlled by switches (mouse, joypad, keyboard, remote control, Bigmack switch, trackball)	Is able to read and interpret simple symbol based coding. Is able to read and debug simple symbol based algorithms. Is able to write simple symbol based coding in order to create an algorithm to fulfil a specific goal. Is able to discuss the use of algorithms within digital devices and is able to discuss why a computer needs exact instructions. Is able to design an app and illustrate their design process through their	Is able to read and interpret complex (i.e. repetition, and / or, time based coding) symbol based coding. Is able to read and debug complex symbol based algorithms. Is able to write complex symbol based coding in order to create an algorithm to fulfil a specific goal. Is able to discuss the use of algorithms within digital devices and is able to discuss why a computer needs exact instructions. Is able to design an app and illustrate their design process through their planning and writing.

		within the symbols. Is able to use a switch to accurately control events on a screen.		planning and writing.	

Personal development	**Problem solving-** Linked to debugging coding and algorithms. **Teamwork-** Linked to the leadership and collaboration work when students are peer to peer assessing and supporting each other. **Self-management** Linked to the student’s ability to follow a brief when designing and planning an app. **Communication skills-** Asking appropriate questions and listening to responses when troubleshooting ICT issues. **Self-belief-** Never giving up if unable to resolve an issue, seeking out appropriate support. Embracing appropriate feedback.	
Suggested activities **P5-8** -Beebot programming -use of paper symbols / actions to read, interpret, sequence and debug instructions -Giving and following directions / plotting routes on a map -keyword quizzes **Level 1-3** -use of paper symbols / actions to read, interpret, sequence and debug algorithms -Giving and following directions -Espresso tasks -keyword quizzes and presentations -complete a plan and design pack		
Online resources https://central.espresso.co.uk/espresso/coding/lessons.html?username=student22081#/coding/units http://www.edutechpost.com/codemonkey-coding-children/ https://www.kodugamelab.com/ https://www.twinkl.co.uk/resources/keystage2-ks2/ks2-subjects/ks2-ict https://www.bbc.co.uk/bitesize/subjects/zvnrq6f https://www.barefootcomputing.org/primary-computing-resources https://community.computingatschool.org.uk/resources/2616/single https://www.icompute-uk.com/primary-computing-resources.html		

Evidencing Work **All work / evidence sheets need to be printed off (where appropriate levelled in accordance with the rubric), students need to self-assess and work needs to be put in student folders. Practical activities need to be evidenced with an individual picture feedback sheet (see example in curriculum folder).**	

The main software we use is:
https://www.discoveryeducation.co.uk/resources/primary/coding/
This software has various coding levels covering basic coding to more advanced. Students enter it at an appropriate level and work through the modules building on previous levels/work.
For lower level students, they begin with basic sequencing (i.e. sequencing the making of a cup of tea) and move onto programming Beebots (we have bought updated Beebots).