Computing at The Mead Academy Trust

"A high-quality computing education is the pen and paper of our time and is the lens through which we experience our world." *David Warlick*

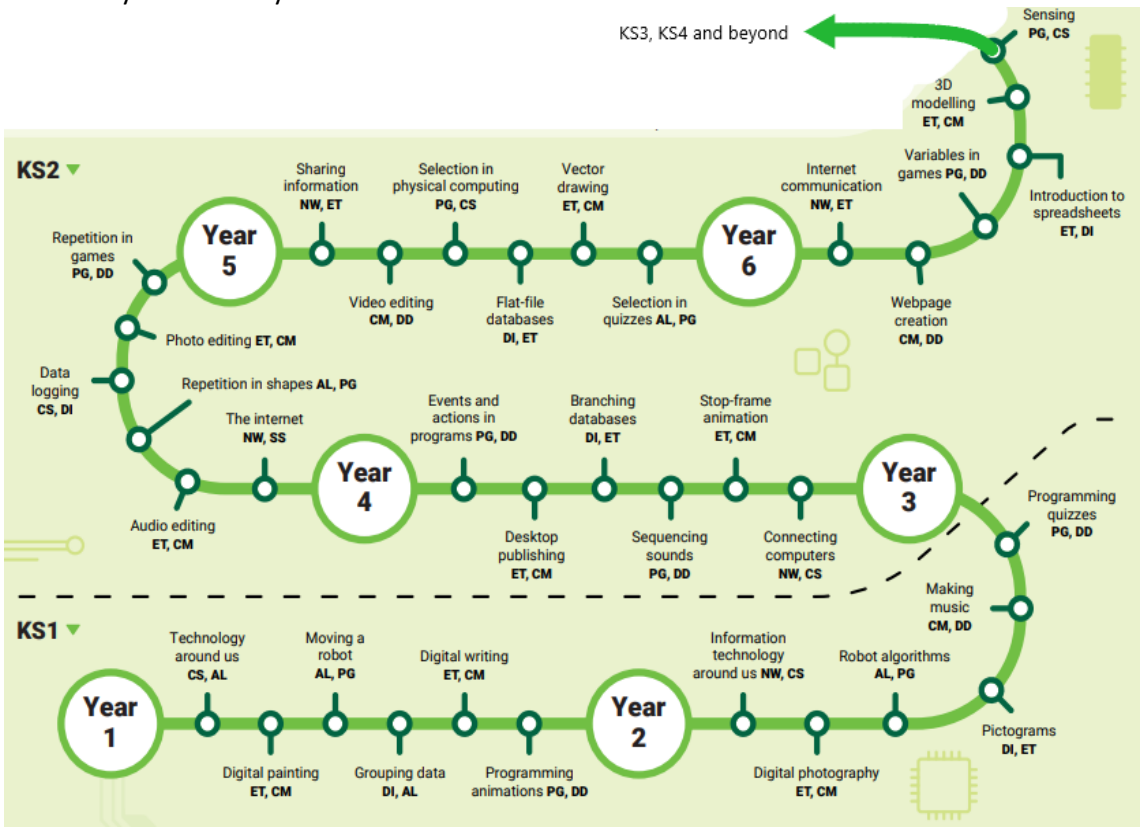
Computing Intent	At The Mead Academy Trust, we develop computational thinkers who can confidently communicate and solve problems, grounded in their own experiences. We cultivate an enthusiasm for digital literacy and enable all children to be members of, and contribute towards, a fast-changing dynamic world. ***National Curriculum Purpose of Study*** A high-quality computing education equips children 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 children 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, children are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that children 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.

Computing Implementation	**We use the Teach Computing curriculum, from the NCCE (National Centre for Computing Education), which uses the following pedagogical principles:** **Lead with concepts** Support children in the acquisition of knowledge, through the use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps, and displays, along with regular recall and revision, can support this approach. **Unplug, unpack, repack** Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach (semantic waves) can help children develop a secure understanding of complex concepts. **Create projects** Use project-based learning activities to provide children with the opportunity to apply and consolidate their knowledge and understanding. Design is an important, often overlooked aspect of computing. Children can consider how to develop an artefact for a particular user or function, and evaluate it against a set of criteria. **Challenge misconceptions** Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion. **Structure lessons** Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and Use-Modify-Create. These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson. **Work together** Encourage collaboration, specifically using pair programming and peer instruction, and also structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding. **Model everything** Model processes or practices — everything from debugging code to binary number conversions — using techniques such as worked example s and live coding. Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away. **Add variety** Provide activities with different levels of direction, scaffolding, and support that promote active learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all children engaged and encourage greater independence. **Make concrete** Bring abstract concepts to life with real-world, contextual examples and a focus on interdependencies with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in children' lives. **Read and explore code first** When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and text-based programming, encourage children to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments children' ability to write code. **Get hands-on** Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides children with a creative, engaging context to explore and apply computing concepts. **Foster program comprehension** Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's Problems. Regular

comprehension activities will help secure understanding and build connections with new knowledge.

The Teach Computing journey is taught using the following concepts and skills, within a coherent and well-sequenced spiral curriculum. Each component provides all children with the knowledge and skills required to progress from one year group to the next.

AL – Algorithms
CS – Computing systems
CM – Creating media
DI – Data and information
DD – Design and development
ET – Effective use of tools
NW – Networks
PG – Programming
SS – Safety and security



Resources	**KS1 Units:**	Laptop/Desktop Chromebook Tablet
	1.1 Technology around us	Paintz.app
	1.2 Digital painting	Microsoft paint or similar
	1.3 Moving a robot	Bee-Bot, Blue-Bot, or other fixed-movement robot
	1.4 Grouping data	Google Slides or Microsoft PowerPoint
	1.5 Digital writing	Google Docs or Microsoft Word
	1.6 Programming animations	ScratchJr
	2.1 Information technology around us	Google slides or Microsoft Powerpoint
	2.2 Digital photography	Digital camera
	2.3 Robot algorithms	Bee-Bot, Blue-Bot, or other fixed-movement robot
	2.4 Pictograms	j2data Pictogram
	2.5 Making music	Chrome Music Lab
	2.6 Programming quizzes	Scratch Jr
	KS2 Units:	Laptop/Desktop Chromebook Tablet
	3.1 Connecting computers	Paintz.app
	3.2 Stop-frame animation	iMotion (app for iOS)
	3.3 Sequencing sounds	Scratch
	3.4 Branching databases	j2data Branch and Pictogram
	3.5 Desktop publishing	Adobe Spark
	3.6 Events and actions in programs	Scratch
	4.1 The internet	Various websites
	4.2 Audio editing	Audacity
	4.3 Repetition in shapes	FMSLogo
	4.4 Data logging	Data logger
	4.5 Photo editing	Paint.net
	4.6 Repetition in games	Scratch
	5.1 Sharing information	Google slides
	5.2 Video editing	Microsoft Photos (for Microsoft Windows 10)
	5.3 Selection in physical computing	Crumble controller + starter kit + motor
	5.4 Flat-file databases	j2data Database
	5.5 Vector drawing	Google Drawings
	5.6 Selection in quizzes	Scratch
	6.1 Internet communication	Various websites
	6.2 Webpage creation	Google Sites
	6.3 Variables in games	Scratch
	6.4 Introduction to spreadsheets	Google Sheets
	6.5 3D modelling	Tinkercad
	6.6 Sensing	micro:bit and Microsoft MakeCode

Computing Concepts (Aims and purpose of NC)		**Substantive Conceptual Knowledge (Revisited over time)**
The national curriculum for computing aims to ensure that all children: • can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation • can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems • can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems • are responsible, competent, confident and creative users of information and communication technology.		KS1: • 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 purposefully to create, organise, store, manipulate and retrieve digital content • recognise common uses of information technology beyond school • 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. KS2: • 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 • use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content • select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

Substantive Concept	Revisited over time in these topics
understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions – KS1 **Concepts and skills: AL, PG, DD**	1.3 Moving a robot 1.6 Programming Animations 2.3 Robot Algorithms 2.6 programming quizzes
create and debug simple programs– KS1 **Concepts and skills: AL, PG, DD**	1.3 Moving a robot 1.6 Programming Animations 2.3 Robot Algorithms 2.6 programming quizzes
use logical reasoning to predict the behaviour of simple programs - KS1 **Concepts and skills: AL, PG, DD**	1.3 Moving a robot 1.6 Programming Animations 2.3 Robot Algorithms 2.6 programming quizzes
use technology purposefully to create, organise, store, manipulate and retrieve digital content– KS1 **Concepts and skills: CS, AL, ET, CM, DI, AL, PG, DD, NW, DI**	1.1 Technology around us 1.2 Digital painting 1.4 Grouping Data 1.5 digital writing 1.6 programming animations 2.1 Information technology around us 2.2 Digital photography 2.4 Pictograms 2.5 Making Music 2.6 programming quizzes
recognise common uses of information technology beyond school– KS1 **Concepts and skills: CS, AL, PG, DI, NW, ET, CM**	1.1 Technology around us 1.3 Moving a robot 1.4 Grouping Data 2.1 Information technology around us 2.2 Digital photography
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– KS1 **Concepts and skills: CS, AL, ET, CM, PG, DD, NW, CS, DI**	1.1 Technology around us 1.5 digital writing 1.6 programming animations 2.1 Information technology around us 2.4 Pictograms
design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts – KS2 **Concepts and skills: PG, DD, AL, NW, ET, CS**	3.3 Sequencing sounds 3.6 Events and actions in programs 4.3 Repetition in shapes 4.6 Repetition in games 5.1 Sharing Information 5.3 Selection in physical computing 5.6 Selection in quizzes 6.1 Internet Communication 6.3 Variables in games 6.6 sensing
use sequence, selection, and repetition in programs; work with variables and various forms of input and output– KS2 **Concepts and skills: NW, CS, PG, DD, AL, DI, ET, AL**	3.1 Connecting computers 3.3 Sequencing sounds 3.6 Events and actions in programs 4.3 Repetition in shapes 4.4 Data logging 4.6 Recognition in games 5.1 Sharing information 5.3 Selection in physical computing 5.6 Selection in quizzes 6.3 Variables in games 6.6 Sensing

use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs– KS2 **Concepts and skills: PG, DD, AL, CS**	3.3 Sequencing Sound 3.6 Events and actions in progress 4.3 Repetition in shapes 4.6 Repetition in games 5.3 Selection in physical computing 5.6 Selection in quizzes 6.3 Variables in games 6.6 Sensing
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– KS2 **Concepts and skills: NW, CS, SS, ET**	3.1 Connecting computers 4.1 The internet 5.1 Sharing information 6.1 Internet communication
use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content– KS2 **Concepts and skills: ET, CM, NW, SS, DD, DI**	3.5 Desktop publishing 4.1 The internet 4.2 Audio editing 4.5 Photo editing 5.2 Video editing 5.4 Flat File Databases 6.1 Internet communication 6.2 Webpage creation
select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information– KS2 **Concepts and skills: AL, CS, CM, DI, DD, ET, NW, PG, SS**	3.1 Connecting computers 3.2 Stop frame animation 3.3 Sequencing sounds 3.4 Branching databases 3.5 Desktop publishing 3.6 Events and actions in programs 4.1 The internet 4.2 Audio editing 4.3 Repetition in shapes 4.4 Data logging 4.5 Photo editing 4.6 Repetition in games 5.1 Sharing information 5.2 Video editing 5.3 Selection in physical computing 5.4 Flat file databases 5.5 Vector drawing 5.6 Selection in quizzes 6.1 Internet communication 6.2 Webpage creation 6.3 Variables in games 6.4 Introduction to spreadsheets 6.5 3D Modelling 6.6 Sensing
use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact – KS2 **Concepts and skills: NW, SS, ET, CM, DD, PG**	4.1 The internet 4.2 Audio editing 4.5 Photo editing 5.1 Sharing information 5.2 Video editing 6.2 Webpage creation 6.3 Variables in games 6.5 3D modelling