The Arthur Terry Learning Partnership

The ATLP curriculum aims to provide children with a broad and academic programme that closely follows the National Curriculum.

Our provision is a coherent and carefully sequenced (knowledge engaged) curriculum based on the principles of cognitive science. There is a focus on the development of literacy and the application of acquired knowledge to ensure children access the curriculum at a depth to ensure a deep and enduring understanding in discrete subject areas.

The content and experiences within our curriculum are designed to accumulate and address the gaps in cultural capital of all our students in particularly the disadvantaged. Our extra-curricular offer supports our provision, with a focus within each subject thus helping to form stronger schemata for long term retention.

Curriculum Statement Overview for Computing

Purpose of Computing (taken from the NC Computing Programme of Study)

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.

The aims of Computing (taken from the NC Computing Programme of Study)

The national curriculum for computing aims to ensure that all pupils:

- 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

Computing Intent

Our aim is to develop a love of computing, to develop strong problem solving skills which can be used across the curriculum. The current skill gap at the start of KS3 is vast, so part of the aim is to create a stronger curriculum at KS2 which ensures pupils are ready to access the KS3 curriculum. Coding is developmental through the curriculum and as a result of the curriculum, pupils will be confident to articulate how to solve problems using a range of strategies. The curriculum has been developed using 2 key strands at KS2:

- Computing
- E-Safety

The programmes used for coding develop across KS2, and use a range of programmes, softwares and sites to develop children's skills, culminating in a sphero project at the end of KS2.

At Key Stage 3, there are three key areas to be covered, with progressive topics which are taught based on allocations and KS4 destinations. The three areas are:

- Computers and data
- Programming
- HTML

The knowledge and key vocabulary needed to be successful has been carefully sequenced to ensure that new content builds upon prior learning, enabling students to develop understanding and skills which are essential at GCSE and beyond.

Overview of the curriculum:

Year group	Digital Literacy	Programming
1		Key skills: • To use a beginners block coding software • To programme a physical system using simple directional language Key knowledge: • Understanding of key programming language:**algorithm, programs, debug, sequence, repeat, logic, directional language (forwards, backwards, left right)**
2	• Use technology safely and respectfully, keeping key information private • Identify where to go for help and support when they have concerns about content on the internet or other online technologies •	Key skills: • To be able to use a block based programming language • To debug a block based programming language Key Knowledge: • Understanding of key programming language: **(algorithm, programs, debug, sequence, repetition, logical reasoning)**
3		Key skills: • Controlling an external device using specific programming software (Sphero) • To be able to use key programming concepts: in out, output, decomposition Key knowledge: • How to decompose a program, and use logical reasoning to debug code and resolve errors • To use key language to label the components of a computer. **CPU, memory (ram), hard disk drive (HDD), USB port, motherboard, graphics processing unit (GPU), morning, sound card, input and output devices (keyboard, speakers, mouse, monitor, microphone)**
4		Key skills: • Controlling an external device using specific programming software (LegoWeDo) • To be able to use key programming concepts: output (text to screen), input, variable, selection. Key Knowledge: • To identify key network components and language and how they link together: **software, hardware, web (www), internet, URL, HTML, HTTP, HTTPS**

		• Understanding of programming language: **variable, selection, debug, sequence, function, repetition (loops), input, output.**
5		Key skills: • Controlling an external device using programming software (scratch- Lego) • To use a text based programming language (swift) Key knowledge: • To identify key network components and language and how they link together: **server, software, hardware, CPU, LAN, WAN, Web (www), Internet, DNS, data, router/modem, IP address, URL, HTML, HTTP, HTTPS, switch, WiFi APs, packets.** • Understanding of key programming language: **variable, selection, debug, sequence, function, logical reasoning, decomposition, abstraction, loops.**
6	• Use technology safely, respectfully and responsibly • Recognise acceptable/unacceptable behaviour • Identify a range of ways to report concerns about content and contact	Key skills: • To use a text based programming language: python • To create a text based programme using python to accomplish a specific goal • To use key programming concepts: **sequences, selection, repetition, loops** • To solve a problem using programming content • To be able top use key programming concepts output (text to screen) **input, variable, selection, loops, HTML language (tags, head, body, title, paragraphs, images, hyperlink)** • To be able to use a range of coding programmes: **scratch, kodu, swift, python, java.** • To be able to organise and manage files and storage: new folder, folder storage, cloud stgorage Key knowledge: • To learn and apply the syntax of python script that accomplishes a specific goal • To understand and apply debugging processes (error messages) • Understanding of key programming language: **algorithm, variable, selection, programs, debug, sequence, repetition, input, output, logical reasoning, decomposition, abstraction, loops.** • To identify key network components and language and how they link together: **server, software, hardware, CPU, LAN, WAN, web (www), internet, DNS, data, router, IP address, URL, HTML, HTTP, HTTPA, switch, WiFi, packets.** • History of computing and the internet. • How to approach debugging a programme

TOPICS TO BE COVERED ACROSS KS3	
COMPUTERS AND DATA	1. History of Computers 2. Hardware and Software 3. Networks and Cyber Security
	1. Data representation (Link with 2, can be taught in any order) 2. Logic (AND/OR/NOT) – truth tables
PROGRAMMING (Problem solving tasks throughout) (Embedding bugging throughout)	1. Sequencing (outputs) 2. Variable (inputs) 3. Selection 4. Iteration (for loops) 5. Iteration (while loops) (repeat until)
HTML	Flexible topic Can be taught at any point Use for creative/ enrichment opportunities