Key Stage 3 Curriculum Overview- Computer Science & ICT

Progression from Key Stage 2 and Progression through Key Stage 3:

	Autumn Term	Spring Term	Summer Term
Year 6	**Students at the end of Key Stage 2 will be able to:** • 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.		
Year 7	• E Safety • CyberSecurity	• Programming essentials in Scratch	• Computer Systems (networks)
Year 8	• Introduction to Python programming (turtle) • Logical thinking (Flowol)	• Data representation (Binary , Hex)	• Vectors
Year 9	• Introduction to Python programming (turtle) • Logical thinking (Flowol) (new curriculum model to cover skills not taught in Y8)	• Data representation (Binary , Hex) (new curriculum model to cover skills not taught in Y8)	• Vectors (new curriculum model to cover skills not taught in Y8)

By the end of Key Stage 3 a student should be able to:

- Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- Understand several key algorithms that reflect computational thinking ([for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- Use 2 or more programming languages, at least 1 of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- Understand simple boolean logic [for example, and, or and not] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
- Understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits
- Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- Create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns
- Enhance students knowdledge, skills and understanding of cultural capital through educational trips and guest speakers, bringing learning to life