

**KS3 Curriculum**

	1. design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
	2. 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
	3. use 2 or more programming languages, at least one 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
	4. 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]
	5. understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
	6. 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
	7. 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
	8. create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
	9. 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

## KS4 Curriculum

	All pupils should be able to:
	1. develop their capability, creativity and knowledge in computer science, digital media and information technology
	2. develop and apply their analytic, problem-solving, design, and computational thinking skills
	3. understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns

## Allocated hours

Working on the basis that there are 7 weeks on average per half term

Year	Hours a week	Hours every 2 weeks	Lessons per half term	Total number lessons a year	Total number lessons previous year (2019-20)
7	1	2	7	42	42
8	1	2	7	42	42
9	1.5	3	10	63	84
10	2.5	5	18	105	84
11	2.5	5	18	105	84
12	4.5	9	32	189	168
13	4.5	9	32	189	210

Curriculum Overview – KS3 and KS4

				Computer Science		ICT	
	Y7	Y8	Y9	Y10	Y11	Y10	Y11
Autumn 1	Digital Literacy + E-Safety 	E-Safety 	FDE Cycle 	Data representation 	Software 	Analysing digital graphics 	Camera shots/storyboards 
			Computer Crime and Cyber Security 	AI and Robotics 	Testing and Translators 	Client requirements and work plans 	Analysing a script 
Autumn 2	HTML and Web design 	Binary – Text, sound and images 	Python –Next steps 	Computational Logic 	Algorithms 	Creating a digital graphic 	Exploring digital games 
				Python – Problem solving 	Computational thinking 	Legislation and plagiarism 	Plan the structure of a game 
Spring 1	Programming with Scratch 	Band manager EPK 	Animation and storyboards 	Algorithms – Flow charts and Pseudo code 	Databases - SQL 	R082: Creating digital graphics 	Preparing a game proposal 
				Python – functions, procedures and modules 	Law and Ethics 	Exporting and Reviewing digital graphics 	R091: Designing a game concept 
Spring 2	Algorithms and Minecraft Education 	Programming with Python 	Sorting Searching algorithms 	OCR Programming Task 	Threats to a Computer system 	R082: Creating digital graphics 	Reviewing a digital game proposal
			HTML + CSS 		HTML Java and CSS 	Understanding the uses of multi-media products	R091: Designing a game concept
Summer 1	Spreadsheet modelling 	Databases 	Databases – Virtual Quest 	Hardware + LMC 	Revision and exams	Plan and create multimedia products 	Revision and exams
				Networking 			
Summer 2	Communications – Virtual tour 	Logic Theory + Hardware 	App invention 	Theory Revision - Mocks 	Review interactive multimedia products	R087 - interactive multimedia products	
				Networking 			

Curriculum Overview – KS5

	Computer Science Year 12		Computer Science Year 13	
	Teacher 1	Teacher 2	Teacher 1	Teacher 2
Autumn 1	Computational thinking	Problem solving	Computer hardware	Computer hardware
		Programming Techniques	Computational thinking	Computer Software and Programming paradigms
Autumn 2	Operating Systems		Programming Project – Analysis Software Methodologies	Compression and Encryption Algorithms Hashing algorithms
		Computer Hardware	Operating systems	Databases and Sorting and searching algorithms
Spring 1	Software Methodologies		Networking	Data structures
	User Interfaces, HTML and Java	Software and translators	HTML, CSS and Java	Web technologies
Spring 2	Networking	Algorithms	Legislation and Ethical Computing Programming Project work	Data representation and Boolean logic
		Data and databases		Walking talking mocks
Summer 1	Legislation and Ethical Computing	Data representation and Boolean logic	Revision and exams	Revision and exams
	Revision and exams	Revision and exams		
Summer 2	Introduction to Practical Programming Project	Practical Programming Project		
		Assembly code and Memory addressing		