



Computer Science – Curriculum Map

<h3>Why do we teach Computer Science?</h3> <p>We believe that teaching computer science is important for two core reasons. Firstly, it's a form of digital literacy that is of growing importance within society. Secondly it promotes analytical thinking and problem-solving skills and encourages them to explore different approaches to finding solutions. We want our students to develop computational thinking and problem-solving skills, it empowers students to understand how computers work, use digital tools effectively, and be critical thinkers in the digital age. We aim to deliver a genuine, long-lasting interest in computing.</p>	<h3>How do we teach computer Science? (Key concepts and skills)</h3> <p>Starting in Year 7, students are introduced to the basic ICT skills they will need to support other subjects across the curriculum including an introduction to the computer systems used at FHS, teams, handin, Office 365, and Satchel One. At Key Stage 3, students are introduced to programming using both block and textual programming methodologies, in accordance with the National Curriculum requirements. Alongside this, pupils are taught the importance and safe use of information technology, including protecting their online identity and privacy. The core constructs of programming are introduced when developing algorithms and the inner workings of a computer are explored when learning about internal and external components. Throughout Year 8, students will continue to build upon the skills they developed in Year 7. Year 9 is used to develop students' programming and computational thinking skills in preparation for choosing KS4 options.</p>
<h3>What do we teach in Computer science?</h3> <p>The key threads through the computing curriculum are:</p> <ul style="list-style-type: none">• computer science (programming and understanding how digital systems work),• information technology (the range of skills required to operate and manipulate specific programs, systems and content.• digital literacy (evaluating digital content and using technology safely and respectfully).	<h3>How is SUBJECT personalised for our learners?</h3> <p>The scheme of work is personalised through the choice of units. our students come from diverse cultural backgrounds. We teach 2 subjects: Computer Science and IT. Our curriculum takes this into account and teaches topics from both subjects. This also ensures that we are catering for a range of learning needs. We make sure all the software we use are free to use at home. SEN students are provided with differentiated resource's that meets their needs. At KS3 , students have a choice of task to completed that are more suitable for them. More able students have a choice of projects that can be completed which relates to topics taught at both KS3 and KS4, projects can be completed outside of lesson. At KS4 and KS5 students are encouraged to create accounts on extended learning websites such as WWw3schools.com, RePL in order to improve on their programming skills and encourage them to become self-taught coders in different programming languages.</p>



Computer Science – Curriculum Map

Year 7	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Key Content	7.1 Introduction to School systems and E-Safety	7.2 Binary: binary addition, conversion between binary and denary	7.3 Scratch coding	7.4 spreadsheets	7.5 Computer Systems	7.6 introduction to text-based programming using Microbit.
Key concept & skills	<ul style="list-style-type: none"> ➤ Appropriate use of the school systems ➤ Online safety issues ➤ Presentation software ➤ Identification of Online dangers ➤ Email composition ➤ File handling 	Binary numbers and why they are important to computer systems how to convert from binary to denary and vice versa how to add binary numbers using the associated rules how to convert binary numbers into ASCII characters how to create bitmap images using binary code.	How to structure program code using Scratch, a graphical programming language. Students will learn about the key programming constructs: <ul style="list-style-type: none"> ➤ Sequence ➤ Selection ➤ Iteration 	Introduction into Spreadsheet design. They will learn the basics of adding formulas and formatting a spreadsheet.	Fundamentals of what makes up a computer system. Students will discover the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	Students will learn about how to structure program using the BBC Microbit through a graphical programming IDE. Students will create programs and transfer them onto the physical Microbit devices. <ul style="list-style-type: none"> ➤ Developing sequencing ➤ Variables ➤ Selection ➤ Count-controlled iteration
Summative Assessment	Use presentation software for cyberbullying project	End of unit text	Practical test	Through work they have produced in class	End of unit test	End of year assessment
Builds on			Problem solving skills Year 6 programming lessons (Scratch)		7.2 scratch binary converter unit - understanding of the control structures'	
Builds towards	Year 7 - Graphics Year 8 - E-Safety Year 9 Network Threats and how to manage them	8.2 Data representation	7.6 introduction to text-based programming using Microbit. Year 8 – Python turtle , 8.3 Computational, Thinking Year 9 – 9.1 algorithms 9.2 algorithms and Programming in python	9.5 Database	8.1 Computer Hardware	Year 8 – Python turtle , 8.3 Computational, Thinking Year 9 – 9.1 algorithms 9.2 algorithms and Programming in python



Computer Science – Curriculum Map

Year 8	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Key content	8.1 – Computer Hardware	8.2 Data Representation	8.3 Computational Thinking	8.4 Python programming using turtle	8.5 Computer Crime and Cyber Security	8.6 Digital Imaging
Key concept and skills	<p>Introduction to hardware – looking at the insides of a computer, CPU, motherboards, RAM, fetch-execute-cycle and operating systems</p> <p>Understand the components that make up a computer system To explain how a processor will interpret instructions</p>	<p>Representation of data in computer systems (Binary, Hexadecimals numbers, characters, images and sound)</p> <p>Calculation of binary values Calculation of Hex values The use of ASCII text Logic Gates</p>	<p>This unit helps pupils to understand how to apply computational thinking skills to solve a range of problems, to help pupils learn how to think in a logical structured way.</p> <p>Understand the concept of abstraction and decomposition</p>	<p>Pupils move a turtle around on a screen so that it draws a shape using the fundamental concepts such as sequencing.</p> <p>Literacy Focus – Writing algorithms</p>	<p>This unit covers some of the legal safeguards regarding computer use, including overviews of the Computer Misuse Act, Data Protection Act and Copyright Law and their implications for computer use. Phishing scams and other email frauds, hacking, “data harvesting” identity theft and safe use of social media are discussed together with ways of protecting online identity and privacy. Health and Safety Law and environmental issues.</p>	<p>(Photoshop) Pixels, resolution & pixel depth Spot healing, Magnetic lasso, Clone Stamp, Camera Raw Copyright types</p> <p>Literacy Focus – Oracy discussion on graphical file</p> <p>Demonstrate basic image manipulation skills and identifying main image file types</p>
Summative Assessment	Produce a report for given computer build tasks	End of unit test	End of unit test	Project based	End of unit test	End of year assessment
Builds on	Unit 8.5 Computer systems	Interpret binary covered in unit 7.2		7.6 introduction to text-based programming using Microbit 7.5 Scratch coding	knowledge from 7.1 - Covers email scams Computer misuse, protecting personal data, copyright and health and safety	Links to copyright lesson from unit 8.5
Builds Towards			9.1 Algorithms	9.2 Algorithms and programming in python	9.3 Networks	



Computer Science – Curriculum Map

Year 9	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Key Content	9.1 computational thinking and Algorithms	9.2 Algorithms and programming in python	9.3 Networks	9.4 App development	9.5 AI and Machine Learning	iDEA Certificate
Key concepts & Skills	Decomposition Abstraction Flowchart test plan testing and design – building on Literacy Focus – Writing algorithms Be able to write flowchart based on given scenarios	Sequence instructions, write instructions to solve problems Design algorithms with repetition & Selection Literacy focus- Writing instructions of commands	WAN, LAN, WPAN, network topologies and network security measures The benefits and disadvantages associated with different networking topologies	Pupils create a mobile app, using App Lab from code.org. Research, design, develop and evaluate skills Literacy Focus – Writing & annotation of code and evaluation	What is AI? Machine learning Ethics of AI Image recognition Turing tests and chatbots	This unit will allow students to cover the iDEA online qualification completing multiple badges for points. https://idea.org.uk/ <ul style="list-style-type: none"> ➤ digital awareness ➤ safety ➤ ethics ➤ workplace ➤ Employability skills. ➤ digital creativity ➤ Digital World
Summative Assessment	Through work they have produced in class	Practical test	End of unit assessment	Through work they have produced in class	End of unit assessment	Online assessment - Multiple Choice End of year assessment
Builds on	8.3 computational thinking	Building on the skills they developed from 7.2, 7.5 and 8.2. sequencing and selection and iteration	Parts of 8.5 Computer Crime and Cyber Security	Building on the programming concepts learners and used in units 7.5,7.6,8.4 and 9.2		All prior knowledge from year 7 & 8
Builds Towards	9.2 Algorithms and programming in python	Use of programming fundamentals to construct complex programs	KS4 -Networks and Topologies	KS4- Pre-production skills /type of research	GCSE computing : Systems architecture, Logic and Languages, Programming, Ethics	2.1.1:Computational Thinking, 2.2.1: Programming Fundamentals, 1.4.1 Threats to computer systems and networks



Computer Science – Curriculum Map

Year 10	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Key Content	Design and Create algorithms	Systems architecture and storage	Additional programming techniques	Network and security		
Key concepts & Skills	2.1 Algorithms Abstraction, decomposition and algorithmic thinking 2.12 Designing creating and refining algorithms Pseudocode and flowcharts 2.2 Programming fundamentals (variables, constraints, inputs and assignments) Sequence, Selection and iteration	1.1 Systems architecture 1.2 Memory and storage 2.1.3 Searching and sorting algorithms (Binary, Linear, bubble, Merge and insertion) 2.12 Designing creating and refining algorithms Structure diagram, OCR reference language, identify common errors and trace tables 2.2.3 Additional programming techniques Open, Read, Write and close file SQL Arrays	2.2 Programming fundamentals 2.2.3 Additional programming techniques Open, Read, Write and close file SQL Arrays dedicated programming lessons	1.3 Computer networks, connections and protocols Data representations 2.2 Programming fundamentals	2.2 Programming fundamentals and additional techniques 1.4 Network security	SLR 1.6 Ethical, legal, cultural and environmental concerns programming project
Summative Assessment	Programming challenges tasks End of unit assessments	End of unit test Searching/sorting algorithms exam questions	Data representation assessment	End of unit assessment	End of unit assessment	End of year assessments
Builds on	KS3 – 8.3 Computational thinking unit, 9.2 Algorithms and programming unit.	KS3 – 8.3 Computational thinking unit, 9.2 Algorithms and programming unit	KS3 - 8.2 Data representation	KS3- 8.1 Computer Hardware		KS3 -9.3 Networks
Builds towards	Advanced programming Practical programming skills		Advanced Number Systems			



Computer Science – Curriculum Map

Year 11	Autumn 1	Autumn 2	Spring 1	Spring 2
Key Content	Ethics, System software and design	Programming languages	Revision and exams	Revision and exams
Key concepts & Skills	1.6 Ethical, legal, cultural and environmental concerns 1.5 System software 2.3 Defensive Design 2.3 Testing	2.4 Boolean logic 2.5 Programming languages and IDEs	Data representation 8 marks exam questions Boolean Logic Subroutines and parameter passing	A full review of all learning, including revisiting some areas at a greater depth to reinforce learning and push understanding.
Summative Assessment	Programming challenges	Centre assessment		Centre assessment
Builds on	Writing to file, SQL and subroutines		CS and ICT general knowledge	
Builds towards	A level programming skill			



Computer Science – Curriculum Map

Year 11-IT	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1
Key Content	R050	R050 AND R060	R060 & R070	R050	
Key concepts & Skills	R050: TA1 - Design tools R050: TA2 - Human Computer Interface in everyday life R050 TA3 - Data & Testing R050: TA4 – Cyber-security and legislation) R050: TA5 – Digital communications R050: TA6 - Internet of Everything (IoE)	R060: TA3 - Testing the spreadsheet solution (3.1 Test the user interface and technical aspects of the spreadsheet solution). Revision for the R050 to be held in January 2024.	R070: TA1 - Augmented Reality (AR) (1.1 Purpose and uses of AR, 1.2 Types of AR and user interaction, 1.3 Devices used with AR). R070: TA2 - Designing an AR model prototype (2.1 Planning and design consideration, 2.2 Design tools). R070: TA3 - Creating and AR model prototype (3.1 AR model prototype, 3.2 Triggers, 3.3 Layers/ user interaction, 3.4 Information output)	Improvement for the R070 coursework continued Revision for the R050 to be held in June 2024.	
Assessment	End of unit assessment	Centre assessment R060: Coursework internal assessment	External R050 Exam R070: Coursework internal assessment	Coursework	External R050 Resit Exam/Coursework
Builds on	9.3 App Design 7.3 spreadsheet skills	Revision/Study skills	9.3 App Design 7.3 spreadsheet skills	Revision/Study skills	
towards					



Computer Science – Curriculum Map

Year 13	Autumn 1	Autumn 2	Spring 1	Spring 2
Key content	Data Structures Implementation	Complex Algorithms	Regular Languages	Exam Prep
Key concepts & skills	<ul style="list-style-type: none"> ➤ <i>Binary Tree implementation</i> ➤ <i>Tree Rotation</i> ➤ <i>Graph Traversal</i> ➤ <i>Dijkstra</i> 	<ul style="list-style-type: none"> ➤ <i>Recursion</i> ➤ <i>Big-O Notation</i> ➤ <i>Optimisation</i> ➤ <i>A-Star</i> ➤ <i>Tour Guide</i> ➤ <i>Limits of Computation</i> 	<ul style="list-style-type: none"> ➤ <i>Finite State Machine</i> ➤ <i>Mealy Machine</i> ➤ <i>Sets</i> ➤ <i>RegEx</i> ➤ <i>Turing Machine</i> ➤ <i>Backus-Naur Form</i> ➤ <i>Reverse Polish Notation</i> 	<ul style="list-style-type: none"> ➤ <i>Skeleton Program analysis</i> ➤ <i>Suggesting Section D Tasks</i> ➤ <i>Implementing suggestions</i>
Summative Assessment	Dijkstra Implementation	Mock Paper	Regular Languages paper	
Builds on	Data Structures	Data Structures Implementation	Complex Algorithms	Entire course
Builds towards	Complex Algorithms	Regular Languages	Exam Prep	Final exam