

Subject: Computer Science

Vision: Influence today, Innovate tomorrow!

Brief overview of topics, themes, skills or key questions for each term:

Why are we teaching a knowledge-rich curriculum; how is it different?

In a Computing knowledge-rich curriculum we want pupils to be able to understand the risks involved with the use of technology and methods that can be taken to prevent these. We also want children to understand how the actions of problem solving can be controlled through programming and the use of ready-made applications.

Computing is a subject rich with technical language and can only be understood when pupils have a firm understanding of keywords and their meaning. Keywords are explained using principles pupils are familiar with and related to technology they have experience of.

Why are we teaching this content, in this order?

The content taught is fundamental to ensuring pupils are digitally literate. We have mapped out the specific knowledge and skills which build on concepts learned from previous years, for example, Year 7 pupils learn how to decompose problems and design algorithms before applying this to a variety of programming environments and applications across the key stage. We have chosen to build knowledge of key concepts in Computing e.g e-safety and problem solving. With programming, Year 7 learn about decomposition, algorithms and the application to block based coding languages (such as Scratch) and turtle in Python. In Year 8, pupils build on this by programming for a text editor in Python and comparing the construction of web pages using both HTML and a web page software such as Serif Web plus. In year 9 the understanding of how computers process data is improved whilst investigating binary and searching & sorting algorithms using Python.

We know that for pupils to be successful at designing a program, they need to learn the skills of decomposition and abstraction, then design algorithms before building their code.

Similarly, modelling and interrogation of data is introduced in year 7 through the use of spreadsheets which is reinforced in year 8 and developed along with database in year 9

How does our curriculum match the ambition of the National Curriculum?

In line with the ethos of the National Curriculum, we have modelled our curriculum to encompass the majority of modules within our weekly hour sessions. Although varying scenarios are used for each unit to encourage creativity and independent learning, the sequence allows transferable skills such as *creating, re-using, revising and re-purposing digital artefacts for a given audience* to be embedded in subsequent units. For example, in year 8 HTML, animation and video editing skills are developed prior to web authoring.

How does the curriculum build on that from Key Stage 2?

We revisit and develop key aspects of the curriculum starting with the safe use of systems and the Internet. This trend is reiterated through the curriculum for example, using the key programming constructs introduced using a programming environment such as Kodu or Scratch to produce modular text-based codes developed in an industry related language such as Python.

By the end of Key Stage 3, what key knowledge should pupils need to remember and be able to apply in this subject?

Students will be digitally literate enabling them to safely select and use the correct tools for the purpose whether studying GCSE Computer Science, Cambridge Nationals in IT or, later on, using their skills and knowledge in the workplace.

Skills embedded include:

- Computational approach to problem solving
- How to be safe when using technology
- Modelling using simulation and spreadsheet software
- An awareness of the interrelationship between hardware and software

Image, sound and video manipulation

Key Stage 3

<u>Year</u>	<u>Autumn 1</u>	<u>Autumn 2</u>	<u>Spring 1</u>	<u>Spring 2</u>	<u>Summer 1</u>	<u>Summer 2</u>
7	7.1 Introduction: Passwords Logon	7.2 E safety: Cyberbullying Social media	7.3 Spreadsheets: Basic and advanced functions and	7.4 Cryptology & Flowol: Sequence	7.5 Scratch : Sequence-Block based programming	7.6: Introduction to text based programming:

Curriculum Plan – Computer Science

	Email SMHW Safe and effective searching Plagiarism Copyright Hardware Computer systems Networks Accessibility and usability	Netiquette Digital footprint Reporting concerns. To include; Word, PowerPoint, Publisher.	formulas Representing data Formatting	Shapes Boolean operators	(Scratch, Kodu,)	(logo, turtle, Python)
8	8.1 HTML (notepad): Input Output Processing	8.4 Animation & Video editing: File type Frame rate Transition	8.2 Web Authoring: (Serif Webplus) E-Safety information Reliability Bias Legislation Plagiarism Ethical issues	8.3 Python Chatbot: (binary, logic AND OR NOT, list) Selection Iteration Sequence	8.5 Spreadsheets: basic and advanced functions and formulae Formatting What if modelling Vlookup Macros	8.6 Database Flat file: Interrogation Data entry form Multiple criteria queries Data types Simple queries Data entry Simple reports Mail merge (letter to businesses) DPA/GDPR
9	9.1 Bitmap editing: Vector Bitmap File types Compression Resolution RGB CMYK	9.2 Coding-Data representation: Text ASCII Unicode Sounds-audacity Analogue Digital Pictures	9.3 Database relational: Key fields Entity relationship Importing data from a text file and a spreadsheet Validation Verification	9.4: Python; Searching Linear & Binary Sorting Bubble Merge Text files Array (1D & 2D) Divide & Conquer	9.5 and 9.6: Integrated Project: Planning Mind maps Gantt charts system life cycles Researching Designing Implementing Evaluating Spreadsheets - Financial Models	

Curriculum Plan – Computer Science

		Metadata Resolution Compression	Foreign key		
--	--	---------------------------------------	-------------	--	--

Key Stage 4

<u>Year</u>	<u>Autumn 1</u>	<u>Autumn 2</u>	<u>Spring 1</u>	<u>Spring 2</u>	<u>Summer 1</u>	<u>Summer 2</u>
10 <u>Computer Science</u>	Unit 1 Hardware Input & output devices Specialist devices Convergence Unit 2 Practical programming skills Fundamentals Sequence Selection Iteration	Unit 1 Systems architecture CPU Memory Secondary storage SQL Unit 2 Practical programming skills Arrays Procedures Functions Records Files	Unit 1 Software Application User interface OS Utility software Unit 2 Data representation Binary Hexadecimal Characters Images Sound Compression	Unit 1 Networks WAN LAN Wireless Client server P2P Protocols Images Sound Compression Unit 2 Algorithms Computational Thinking Searching Sorting	Unit 1 Networks Threats Preventing vulnerabilities OS Unit 2 Algorithms Flow charts Pseudocode	Problem solving
11	Unit 1	Unit 1	Unit 1	Unit 1	Practice papers	External exams

Curriculum Plan – Computer Science

<p>Computer Science</p>	<p>Database: Tables, records & fields Entities & Relationships Queries & SQL Data independence DBMS</p> <p>Unit 2 Computational thinking: Algorithms Program control flow Handling data Programming languages IDE Testing</p>	<p>Impacts of digital technology: Internet WAN LAN Wireless networking Client server P2P networks Standards, Protocols & Layers</p> <p>Unit 2 Logic & Languages Logic diagrams Truth tables Defensive design Errors & Testing Translators</p>	<p>Revision workbooks:</p> <p>Unit 2 LMC: Processor FDE cycle Assembly code</p>	<p>Revision workbooks:</p> <p>Unit 2 Revision workbooks:</p>		
<p>10 ICT</p>	<p>Theory: Project management: Systems Lifecycle SMART targets Success Criteria SWOT</p>	<p>Theory: Data Collection: Biometric Sensors Interviews Questionnaires</p>	<p>Theory: Communication Methods Audience Purpose Social demographics</p>	<p>Theory & Practical: Database skills: Tables Queries Relational Import data Data validation Reports Queries</p>	<p>Theory & Practical: Integrated presentation- Word Tables Footnotes Contents Mail merge Macros Embedding</p>	<p>Theory: Exams</p>

	<p>Practical: Powerpoint skills: Speaker notes Master slide Backgrounds Graphs & Charts Export Non linear Linking objects</p> <p>Risk management: Logical Mitigating risks Planning tools Gantt charts Mind maps Pert chart Critical Path Flow chart Task List</p>	<p>Practical: DTP skills: Master pages Booklet design Business card</p>	<p>Practical: Spreadsheet skills: Functions Linking worksheets What If Macros Importing data Graphs & Charts Security</p>	<p>Security Exporting to other application</p>	<p>data Security Watermarks Review</p>	<p>Practical: Website design: HTML CSS Forms</p>
11 ICT	R013 Practice Controlled Conditions		R013 NEA		Exam paper practice and targeted revision	External exam
	<p>R012: Revision: Planning tools System Life cycle Risk mitigation Legislation Iterative testing</p>	<p>R012: Revision: Importing data files Data entry forms Queries Reports Business cards Leaflets</p>	<p>R012: Revision:</p>	R013 Skills	Exam paper practice and targeted revision	

Key Stage 5

<u>Year</u>	<u>Autumn 1</u>	<u>Autumn 2</u>	<u>Spring 1</u>	<u>Spring 2</u>	<u>Summer 1</u>	<u>Summer 2</u>
<u>12 Computer science</u>	Unit 1 Systems & architecture: Processor components Processor performance Types of Processor Input devices Output devices Storage devices Fetch decode execute Unit 2: Data types: Binary Hexadecimal ASCII & Unicode Binary arithmetic Floating point Bitwise manipulation masks Unit 2: Programming techniques: Programming basics Selection Iteration	Unit 1 Assembly Language: FDE cycle Assembly language LMC Unit 2 : Software development Systems analysis methods Writing & following algorithms Programming paradigms Unit 2 : Programming techniques: Subroutines	Unit 1 Systems Software: OS functions Types of OS Nature of applications Programming languages Unit 2: Boolean Algebra: Logic Gates Boolean expressions Karnaugh maps Adders & D type flip flops Unit 2: Elements of Computational Thinking	Unit 1 Networks: Internet Internet communication Security & Threats HTML & CSS Javascript Search engine indexing Client server P2P Unit 2 Data types: Sound Images Compression Encryption Unit 2: Algorithms Analysis & design of algorithms Searching	Unit 1 Ethical, Cultural and Moral issues: Computers in the workplace Artificial Intelligence Automated decision making Analysis of personal information Privacy Unit 2 Problem solving: Computational thinking Writing & interpreting algorithms Finite state machines Unit 2 Object-oriented languages Pygame	Unit 3 Project documentation Unit 3 Practical programming project <ul style="list-style-type: none"> • Analysis • Design

Curriculum Plan – Computer Science

		<p>Recursion OOPS Files & exception handling</p>	<p>parameter passing by value and reference Identify the components of a solution to a problem</p>	<p>Sorting Graph traversal Optimisation</p>		
<p><u>13 Computer Science</u></p>	<p>Unit 1 Data Types: Images Sounds Compression Encryption Exchanging data: Compression & encryption Database concepts Normalisation SQL Transaction processing Unit 2 Algorithms & Complexity: Analysis & design Searching bubble sort, insertion sort Merge sort</p>	<p>Unit 1 Computational thinking: Abstraction Thinking ahead Thinking procedurally Unit 2 Data Structures: Arrays, tuples & records Queues Lists & linked lists Stacks Hash tables Graphs Trees</p>	<p>Legal & cultural issues: Legislation Ethical & cultural Privacy & censorship</p>	<p>Practice papers / Revision Revision and practice papers</p>	<p>External assessment</p>	

	Quick sort Graph traversal Optimisation Unit 3 Practical Programming Project Design	Unit 3 Practical Programming Project Development Iterative testing	Unit 3 Practical Programming Project Testing Evaluation			
--	--	---	--	--	--	--

Competitions taken part in: Astro (Computing club), Game Design (Year 7, Trust, July), FXP (Year 12/13, July), Scratchoff (Year 9, June)

External speakers in/visits out/field: Computer Science clubs Lunchtime organised by Year 12 for KS3; TTA Computer Science trip to London (KS5, November); Elvin Computer Science in Industry; Illuminate; Local businessmen

Exam syllabus followed (GCSE/A level (or equivalent) – if different for different year groups please state: OCR GCSE Computer Science (9-1) J277;

OCR A Level Computer Science H446, Cambridge National in Information Technologies J808

Any cross-curricular opportunities: ERASMUS+ participation, Contexts for skills in each area

Any further resources you wish children/parents to be directed to?

<https://www.python.org/> . <https://www.bbc.com/education/guides/zts8d2p/revision/1> (Introduction to programming). <https://scratch.mit.edu/>