Computer Science

Overview of the Curriculum

Key Stage 3 (Years 7, 8 & 9)

Key Stage 3 builds both the theory and the practical programming skills students need, going beyond what the national curriculum asks for. They learn to break problems down and think about them in computational terms, not just follow instructions.

Students look at how computer hardware and system architecture work, how networks communicate, and how data gets represented in binary and through logic. The curriculum is sequenced so that ideas build on each other rather than sitting in isolation - using tools we've built ourselves, including Turtle Lab, FlowLogic and Mars DB, students move from visual and block-based logic into text-based Python, then on to databases and web development with HTML, CSS and JavaScript.

Students arrive from primary school with wildly different levels of Computer Science experience, so the curriculum is built to give everyone a solid footing for GCSE while still pushing those who already know a fair amount further. Digital literacy, cyber security, online safety, and thinking critically about AI aren't taught as a one-off unit - they come up throughout, so students get used to using technology safely and questioning it as they go.

Computer Science GCSE - Key Stage 4 (Years 10 & 11)

At GCSE we teach the Edexcel Computer Science specification. It strikes a good balance between solid theory and hands-on application, and the on-screen practical programming exam in particular suits how we've already been teaching the subject at Key Stage 3. The papers are laid out clearly, so students aren't losing marks to confusing wording - they get a fair shot at showing what they actually know.

Python is the main language throughout, and students build up their algorithmic thinking alongside coding skills they can carry into other languages later. The course picks up directly from where Key Stage 3 leaves off, going deeper into computer architecture, networks, data structures, searching and sorting algorithms, and cyber security. There's also a strand looking at how digital technology affects society - ethically, legally, culturally and environmentally. Programming work runs right through the course rather than being bolted on at the end, so students are properly prepared for both the written and practical exams.

Key Stage 5

At A Level we follow OCR H446 Computer Science. It picks up naturally from GCSE and gets students ready for either university or a technical career, without much of a gap to bridge.

The theory side covers processor architecture, systems software, web technologies, databases, Boolean algebra, and the systems development lifecycle. Problem-solving stays central throughout, and students get used to thinking about problems in several different ways - abstractly, procedurally, logically, and in terms of concurrency.

The big piece of coursework is the NEA programming project. Students pick something they're genuinely interested in and take it through the full cycle - design, build, test, evaluate. Most stick with Python, since that's what they know, but some use the project as a chance to try something new, like Java or a C-family language.

SMSC in Computer Science

Logical reasoning, abstraction and creative problem-solving all feed into students' spiritual development. The subject gives them real chances to stop and think about what technology is doing to us - what it means to be human when so much of daily life now runs through automation and AI.

Morally, students get into cyber security, data privacy, intellectual property, digital footprints, algorithmic bias and - increasingly - how misinformation, disinformation and online hoaxes actually get made and spread. Once they've seen how a hoax is engineered, they start questioning what turns up in their own feeds, and the case for truth, integrity, legality and decent digital citizenship stops being something they're told and starts being something they've worked out for themselves.

Socially, most of this comes out of working together rather than sitting and listening - pair programming, debugging someone else's code, group projects. That collaboration opens onto bigger questions too: what happens to relationships, to public debate, to whole communities, once networked technology and online information are doing so much of the shaping.

Culturally, students look at the global reach of computing - who has access to it, who doesn't, and the people from different backgrounds who've built the field. They also think about what it actually takes to make software that works for everyone, wherever they are.

Overview of schemes of work at each key stage

Key Stage 3	Key Stage 4	Key Stage 5
Throughout Key Stage 3, students learn a variety of concepts taught through multiple topics of work. **Computational skills include:** ● Computational Thinking ● Cyber Security & eSafety ● Encryption & Decryption ● Creating Algorithms ● Binary & Hexadecimal Conversion ● Boolean Logic & Truth Tables **Practical skills:** ● Spreadsheet Modelling & Data Analysis ● Programming in Python ● Control Systems & Automation (*FlowLogic*) ● Database Creation & SQL (*Mars DB*) ● Web Development using HTML, CSS & JavaScript ● Graphics & Digital Video Production	Throughout Key Stage 4, students apply their previously acquired knowledge from Key Stage 3 to advance through a series of topics that build in complexity, fully preparing them for the Edexcel GCSE Computer Science specification. **Computational skills include:** ● Computational Thinking (decomposition, abstraction, pattern recognition, algorithmic thinking) ● Searching and Sorting Algorithms ● Encryption & Data Compression ● Creating, Designing & Debugging Algorithms ● Practical Python Programming & Problem Solving ● Binary Arithmetic, Two's Complement, Logical/Arithmetic Shifts & Hexadecimal Conversion ● Boolean Logic, Expressions & Truth Tables	Throughout Key Stage 5, students build on their prior knowledge to master the OCR A Level Computer Science specification (H446). The course is delivered co-teachingly by two specialist teachers, combining rigorous theoretical understanding, computational thinking, and advanced software development. **Computational skills include:** ● Computational Thinking (thinking abstractly, ahead, procedurally, logically, and concurrently) ● Algorithm Design, Analysis & Optimisation (Big O complexity) ● Complex Data Structures & Traversals (stacks, queues, trees, linked lists, graphs) ● Boolean Algebra & Circuit Simplification **Practical skills:** ● Advanced Programming & Paradigm Design ● Relational Database Design, Normalisation & SQL ● Software Development Lifecycle &

Computational knowledge:

- Hardware Components & Architecture
- Laying out algorithms using pseudocode/flowcharts
- Computer Networks & Cloud Services
- Misinformation, Disinformation & Online Hoaxes (*CTRL-ALT-DECEIT*)
- Machine Learning and Artificial Intelligence
- Ethical, Moral, Social and Environmental Landscape

Topics are spiralled in such a way that completing the Year 7 topics builds the foundations for Year 8 topics and so forth.

For 2026/27 the units studied are:

Year 7:

- Unit 1: Click Start (Online Safety & Digital Literacy)
- Unit 2: Computational Thinking
- Unit 3: Boolean Logic
- Unit 4: Cryptography
- Unit 5: Computer Hardware
- Unit 6: Introduction to Python Programming

Practical skills:

- Creating, Setting & Querying Relational Databases using SQL
- Fundamental & Advanced Programming Concepts (in Python using Edexcel PLS)
- File Handling (Text & CSV) & Data Structure Manipulation (2D Arrays/Lists)
- String Manipulation, Input Validation & Defensive Design
- On-Screen Practical Exam Techniques & Code Optimization

Computational knowledge:

- Hardware Components, Von Neumann Architecture & CPU Execution Cycles
- Laying out algorithms using Flowcharts & Edexcel Pseudocode
- Computer Networks, Topologies, Protocols & Internet Architecture
- Cyber Security Threats & Defensive Mitigation Techniques
- Ethical, Moral, Legal, Cultural and Environmental Landscape

System Design

- Independent Non-Exam Assessment (NEA) Programming Project

Computational knowledge:

- Processor Architecture, Systems Software & Software Engineering
- Networks, Web Technologies, Encryption & Hashing
- Data Representation, Data Types & Boolean Logic
- Computing Legislation, Moral & Ethical Implications

Year 12:
Teacher 1:

- Structure and Function of the Processor

- Types of Processor
- Input, Output and Storage
- Systems Software
- Applications Generation
- Software Development
- Types of Programming Language

Teacher 2:

- Thinking Abstractly, Ahead, Procedurally, Logically and Concurrently

Year 8:

- Unit 1: Handling Data (Spreadsheet Modelling)
- Unit 2: Control Systems (*FlowLogic*)
- Unit 3: Turtle Programming (*Turtle Lab*)
- Unit 4: Mission Control (Databases & *Mars DB*)
- Unit 5: Computer Networks
- Unit 6: Binary & Hexadecimal

Year 9:

- Unit 1: CTRL-ALT-DECEIT (Misinformation, Disinformation & Hoaxes)
- Unit 2: Creating Websites (HTML, CSS & JavaScript)
- Unit 3: Artificial Intelligence & Ethics
- Unit 4: Digital Graphics & Media
- Unit 5: GEMS (Advanced Python Chatbot Project)
- Unit 6: Digital Video Production

For 2026/27 the units studied are:

Year 10:

- Unit 1: Programming Fundamentals
- Unit 2: Algorithms & Computational Thinking
- Unit 3: Data Representation (Binary & Hexadecimal Systems)
- Unit 4: Computer Systems & Architecture
- Unit 5: Practical Programming, Boolean Logic & SQL Databases
- Unit 6: Synoptic Revision & Exam Practice

Year 11:

- Unit 1: Networks & Network Security
- Unit 2: Issues, Impacts & Cyber Security (Ethical, Legal & Environmental)
- Unit 3: Applied Problem Solving (Paper 2 On-Screen Exam Preparation)
- Unit 4: Synoptic Revision & Exam Practice
- Programming Techniques
- Computational Methods

Year 13:
Teacher 1:

- Compression, Encryption and Hashing

- Databases
- Networks
- Web Technologies
- Data Types
- Boolean Algebra
- Computing-Related Legislation

Teacher 2:

- Advanced Algorithms & Data Structures (Algorithm Analysis & Design, Efficiency & Big O Notation, Standard Sorting & Searching, Quick Sort, Dijkstra's Shortest Path, A* Algorithm, Queues, Stacks, Trees, Linked Lists, and Tree Traversals)

**Note: Both teachers support the delivery of the Non-Exam Assessment (NEA) programming project from January of Year 12 through to October of Year 13. Moral and Ethical Issues are studied independently by students during supervised study sessions and assessed in lessons.*