



Subject:

Curriculum vision

The Computer Science curriculum at Avonbourne Academy is designed to develop students into confident, capable and responsible users and creators of technology. Students are introduced to the essential digital literacy skills needed to work effectively in a modern school environment, before progressing into computational thinking, programming, computer systems, data representation, networks, cybersecurity and the wider impact of digital technologies.

Across Key Stage 3, students build a secure foundation in how computers work, how data is represented, how algorithms are designed, and how programming constructs are used to solve problems. The curriculum develops students' ability to break down complex problems, apply logical reasoning, create digital products, and understand the ethical, legal, cultural and environmental implications of technology. Students experience both creative and technical aspects of computing, including block-based programming, Python, HTML, graphic design, computer hardware, binary logic, networks and digital media.

At Key Stage 4, students build on this foundation through GCSE Computer Science. They develop a deeper understanding of computer systems, including CPU architecture, memory and storage, networks, network security, systems software and the impact of technology. Alongside this, students develop programming knowledge through sequencing, selection, iteration, data types, Boolean operators, string manipulation, random number generation, algorithms, defensive design, testing and programming languages. This allows students to become more independent problem-solvers who can design, write, test and refine programs.

At Key Stage 5, students extend their knowledge further through advanced study of computer science theory, programming and problem-solving. They explore more complex topics such as computational thinking, data structures, algorithms, Boolean algebra, computer architecture, networks, databases, systems software and the consequences of technological development. Students also develop independence through larger programming tasks and project-based work, where they are required to analyse problems, design solutions, implement code, test thoroughly and evaluate their outcomes. This prepares students for higher education, apprenticeships and careers in computing, digital technologies, engineering, cyber security and related fields.

The curriculum is sequenced so that knowledge and skills develop over time. Students begin by learning how to use digital systems safely and effectively, then move towards increasingly complex programming, theoretical and problem-solving concepts. This prepares students for success at GCSE and gives them the confidence, vocabulary and computational understanding needed for further study, employment and life in a digital world.



Avonbourne Boys' & Girls' Academies

The best in everyone™

Part of United Learning

Curriculum Overview



United Sixth Form

The best in everyone™

Part of United Learning

Term 1

Year	Autumn 1	Why this? Why now?	Autumn 2	Why this? Why now?
Year 7	Introduction to Computer Systems- Navigating Teams- Sending Emails- Folder Structures	This unit ensures students can confidently access, organise and submit work using school systems. It builds essential digital literacy and prepares students to work independently across the curriculum.	Block-Based Programming- Programming Movement- Creating Variables- Arithmetic and Logic Operators	Students begin programming in a visual environment so they can focus on core logic before syntax. This introduces movement, variables and operators in an accessible way.
Year 8	Computational Thinking- Decomposition- Abstraction- Pattern Recognition	Students develop problem-solving strategies that underpin all later programming and theory work. This builds on Year 7 programming by helping students approach problems logically.	Data Representation- Binary Number System- Character Sets- Images	Students learn how computers store and process data. This follows computational thinking because students now apply logic to binary, text and image representation.
Year 9	Python Search and Sort Algorithms- Algorithms and Linear Search- Binary Search- Bubble Sort	Students now apply their Python knowledge to more formal algorithmic problems. Search and sort algorithms strengthen problem-solving and prepare students for GCSE-style algorithmic thinking.	Python Search and Sort Algorithms- Merge Sort- Insertion Sort- Programming an Algorithm	This builds directly on the previous unit by increasing complexity. Students compare algorithms and program solutions, developing efficiency, accuracy and refinement.
Year 10	1.1 Systems Architecture- The purpose of the CPU- CPU Components and Performance- Von Neumann Architecture Programming Fundamentals- Sequence- Selection- Iteration	Students begin GCSE with core theory and core programming constructs. This creates a strong foundation for both papers and links computer architecture with how instructions are processed.	1.2 Memory and Storage- Binary / Hexadecimal / Denary Conversions- Representing Images, Sound and Text Programming- Data Types- Boolean Operators	Students build on KS3 data representation and apply this knowledge to GCSE-level conversions and data storage. Programming work develops their ability to use appropriate data types and logic.
Year 11	2.1 Algorithms- Search and Sort Algorithms- Writing Algorithms	Students revisit and deepen algorithmic understanding from Year 9 and Year 10. This supports exam readiness by strengthening pseudocode, flow diagrams and problem-solving skills.	1.5 Systems Software- Operating Systems- Utility Software	Students complete key Paper 1 theory by studying the role of operating systems and utility software. This links back to hardware and systems architecture from Year 10.
Year 12				
Year 13	1.3.3 Networks	Students begin Year 13 by developing their understanding of networks, building on prior knowledge of computer systems and communication technologies. This supports their ability to explain how data is transmitted, how networks are structured, and how network technologies are used in real-world computing environments.	1.4 Data Structures and Boolean Algebra	Students then move on to data structures and Boolean algebra, developing a deeper understanding of how data can be organised, processed and manipulated. This strengthens both theoretical understanding and problem-solving skills needed for more advanced computational thinking.



Term 2

Year	Spring 1	Why this? Why now?	Spring 2	Why this? Why now?
Year 7	Scratch to Python- Python Variables- Movement using Turtle- Data Types and User Input	Students now transition from block-based to text-based programming. Turtle allows students to apply Python syntax visually while introducing variables, data types and user input.	Computer Hardware and Components- What is a "Computer"?- Types of Hardware- Computer Components	After learning basic programming, students explore the physical systems that allow programs to run. This develops understanding of input, output and internal components.
Year 8	Text-Based Python Game- Sequence- Selection- Iteration	Students develop their Python programming by applying the three key constructs in a creative text-based game. This builds confidence in structured programming.	Text-Based Python Game- Random Number Generation- Creating a Program- Refining a Program	Students extend their programs using randomisation, then create, test and refine a working product. This encourages debugging, independence and iterative development.
Year 9	Binary Logic- AND, OR, NOT Gates- Trace Tables- Building Logic Circuits	Students move from software algorithms to the logic that underpins computer systems. This develops abstract thinking and prepares students for GCSE Boolean logic.	Networks and Protocols- LAN, WAN and Topologies- Common Protocols- Data Transmission and Packet Switching	Students study how devices communicate and how data moves across networks. This builds on systems knowledge and prepares students for GCSE networks and security.
Year 10	1.2 Memory and Storage- Compression- RAM, ROM and Secondary Storage Programming- String Manipulation- Random Number Generation	Students continue memory and storage by exploring how data is stored, compressed and accessed. Programming lessons build on data types and Boolean logic by introducing string handling and randomisation.	1.3 Computer Networks, Connections and Protocols- LAN, WAN and Topologies- Network Hardware- Network Protocols and Encryption	Students build on KS3 networks and Year 10 systems knowledge. They develop a secure understanding of how networks operate and how protocols support communication.
Year 11	2.3 Producing Robust Programs- Defensive Design Considerations- Maintainability- Testing and Test Data	Students refine programming knowledge by focusing on reliability, testing and maintainability. This prepares them for programming exam questions and supports accurate, robust solutions.	2.4 Boolean Logic- Logic Diagrams- Truth Tables- Logical Operators and Circuits 2.5 Programming Languages and IDEs- Compilers, Interpreters and Translators	Students connect hardware logic with programming logic, then complete key theory on translators and IDEs. This consolidates Paper 1 and Paper 2 knowledge before revision.
Year 12				
Year 13	1.4 Data Structures and Boolean Algebra 2.2 Computational Methods	Students continue developing their understanding of data structures and Boolean algebra before moving into computational methods. This sequencing allows students to apply logical reasoning and structured problem-solving to increasingly complex computational problems.	2.3 Algorithms	Students focus on algorithms, applying computational thinking to design, analyse and evaluate solutions. This builds directly on data structures and computational methods, helping students prepare for advanced exam questions and problem-solving tasks.



Term 3

Year	Summer 1	Why this? Why now?	Summer 2	Why this? Why now?
Year 7	Computer Hardware and Components- How the CPU works- RAM, ROM and Storage- Building a computer	Students deepen their understanding of internal components by studying the CPU, memory and storage. The unit culminates in building or modelling a computer to apply knowledge practically.	Online Safety- Staying Safe Online- Proper use of the Internet- Computer Related Laws	Students finish Year 7 by considering how to use technology safely, legally and responsibly. This reinforces good digital citizenship before they progress into more independent digital work in Year 8.
Year 8	Graphic Design- Interpreting a Client Brief- Working with Layers- Image Editing	Students apply digital creativity and problem-solving to a design context. This develops confidence using software tools and introduces client briefs and visual communication.	HTML and Web Development- Introduction to HTML Tags- Formatting Tags- Adding Images and Links	Students learn another text-based language and create structured webpages. This broadens their programming experience and links creative design with web development.
Year 9	Impact of Digital Media and Technologies- Digital Laws- Ethical and Moral Issues- Environmental Impacts	Students consider the wider consequences of technology and digital systems. This develops critical thinking and prepares them for GCSE ethical, legal, cultural and environmental topics.	Impact of Digital Media and Technologies- Impact of Digital Media- Media Planning and Design- Media Creation and Evaluation	Students apply their understanding of digital media by planning, creating and evaluating a digital product. This gives students a practical endpoint to KS3 and supports progression into GCSE Computer Science or Creative Media.
Year 10	1.6 Ethical, Legal, Cultural and Environmental Impacts- Digital Laws- Privacy Issues- Open-source vs Proprietary Software 1.4 Network Security- Forms of Attacks	Students now examine how technology affects individuals and society, while also beginning network security. This links technical knowledge with real-world implications and risks.	1.4 Network Security- Threats to Networks- Preventing Vulnerabilities 2.1 Algorithms- Computational Thinking- Structure Diagrams- Pseudocode and Diagrams	Students complete the network security sequence and then return to algorithmic thinking. This prepares them for Year 11 programming, pseudocode and exam-style problem solving.
Year 11	Exam Practice and Revision	Students consolidate knowledge across both GCSE papers, revisit weaker areas and practise exam technique using past-paper style questions.	Exam Practice and Revision	Students complete final revision, focusing on retrieval practice, programming skills, command words, exam timing and applying knowledge accurately under exam conditions.
Year 12				
Year 13	2.3 Algorithms Exam Practice and Revision	Students complete their study of algorithms before moving into targeted exam practice and revision. This allows students to consolidate the most demanding problem-solving content while identifying and addressing gaps in knowledge.	Exam Practice and Revision	Students complete final preparation for their examinations through retrieval practice, exam-style questions, programming practice and targeted revision of key theory. This supports confidence, accuracy and independence before the final assessments.

Wider reading

Extracurricular Opportunities (competitions, associations and clubs)

Students are encouraged to take part in opportunities that extend their computing knowledge beyond the classroom. In Year 9, students have the opportunity to visit Bletchley Park, allowing them to explore the history of computing, codebreaking and the development of early computer systems. Within school, students can also take part in Coding and Robotics, Web Design and 3D Design enrichment activities. These opportunities allow students to apply problem-solving, programming and digital creativity in wider contexts.

Revision Guides

Useful links:

- CGP GCSE Computer Science OCR Revision Guides:

<https://www.cgpbooks.co.uk/secondary-books/gcse/computer-science>

- Craig'n'Dave YouTube Channel: <https://www.youtube.com/@craigndave>

- Craig'n'Dave OCR GCSE J277 Playlist:

<https://www.youtube.com/playlist?list=PLCiOXwirraUAEhj4TUjMxYm4593B2dUPF>

Academic Reading

KS3 level reading:

- How Computers Work by Ron White
- Can You Solve My Problems? by Alex Bellos
- Artificial Intelligence: A Ladybird Expert Book by Michael Wooldridge
- The Code Book by Simon Singh

KS4 level reading:

- Code: The Hidden Language of Computer Hardware and Software by Charles Petzold
- The Pattern on the Stone by W. Daniel Hillis
- Once Upon an Algorithm by Martin Erwig
- Hello World: How to be Human in the Age of the Machine by Hannah Fry

KS5 level reading:

- Computer Science: An Overview by J. Glenn Brookshear and Dennis Brylow
- Algorithms to Live By by Brian Christian and Tom Griffiths
- The New Turing Omnibus by A. K. Dewdney
- Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig