Ansford Academy

Computing

Curriculum Booklet for 2025 - 2026

Subject Lead: Miss Worthy - Jarvis

Midsomer Norton Schools Partnership

Computing Curriculum Intent:

Computers themselves, and software yet to be developed, will revolutionise the way we learn.

Steve Jobs

At Ansford, we want every student to develop a strong understanding of how digital technology shapes the world and gain the confidence to use and apply it in purposeful ways. Computing helps students understand how systems work, how data is managed, and how digital tools can be used to solve problems, communicate ideas and develop new solutions.

Our curriculum is designed to be inclusive, engaging and accessible to all learners. It is built around the three strands of the National Curriculum: **Computer Science**, **Information Technology** and **Digital Literacy**. These form the foundation for all topics taught at **Key Stage 3.** At **Key Stage 4**, students have the opportunity to continue with Computing as one of their option subjects.

To support progression, we have identified nine key building blocks of learning that run across the curriculum:

- Algorithms
- Programming Fundamentals
- Binary and Data Representation
- Computer Systems
- Networks
- Safe and Responsible Technology Use
- Creative Technology
- Digital Tools
- Digital Skills

These blocks are revisited throughout Key Stage 3 and 4 so that students build secure knowledge and deepen their understanding over time.

Through this curriculum, students will:

- Develop technical understanding of how computers and networks function
- Learn how to design, create and test digital solutions
- Build safe and responsible habits for working online
- Improve problem-solving skills through computational thinking
- Use digital tools to communicate and present information effectively
- Think critically about how technology is used and the impact it can have

We also encourage students to extend their learning beyond the classroom. This includes taking part in national competitions such as BEBRAS, working towards digital awards like iDEA, and joining extra-curricular clubs such as Coding, Game Design and Minecraft.

Computing Curriculum Implementation:

The Computing curriculum is carefully designed so that every student, whatever their starting point, can achieve success and feel a sense of belonging in lessons.

- New concepts are introduced clearly using modelling, worked examples, and step-by-step explanations.
- We use the PRIMM method (Predict, Run, Investigate, Modify, Make) to support students in programming tasks, helping them move from guided work to independence.
- Tasks are adapted carefully: there are core expectations for all, extra support for those who need it, and opportunities for extension and challenge for those ready to go further.
- Ongoing assessment happens naturally through retrieval activities, questioning and peer feedback.
- Formal assessments blend practical projects, written responses, and online assessments to capture the full range of skills and knowledge.

Throughout every year, **digital responsibility** including online safety, cybersecurity habits, and ethical digital practice is embedded across all topics and not treated as a one-off.

Key Stage 3:

- **Year 7** focuses on strong foundations: digital skills (file management, e-safety, using a computer confidently), basic hardware knowledge, networking, and visual programming with Scratch.
- **Year 8** builds confidence and depth: students learn about cybersecurity, develop websites using HTML and CSS, edit digital graphics, design simple apps, and begin text-based programming in Python.
- **Year 9** tackles complexity: students design vector graphics, model data with spreadsheets, code projects using Micro:bits, develop algorithms, and create arcade games.

Key Stage 4:

OCR GCSE Computer Science

This course helps students understand how computer systems work and how digital solutions are designed and created.

Students will learn:

- How computers and networks store, send and manage data
- How to write, understand and improve computer programs
- The right vocabulary to explain technical processes clearly and confidently
- How to use logical thinking and problem-solving to design efficient solutions
- How to recognise and prevent common security threats in digital systems
- How to reflect on the ethical, legal and environmental impact of technology in the modern world

Computing Curriculum Impact

The success of our KS3 Computing curriculum can be seen in the skills, knowledge, independence, and confidence our students show by the end of Year 9.

By the time they finish Key Stage 3:

- Students can design and code solutions to a range of real-world problems using structured approaches.
- They can create and adapt websites, graphics, apps, spreadsheets, and other digital content confidently and independently.
- They understand how to work safely and responsibly online, and know how to protect their information and wellbeing.
- They think critically about information they find online and can assess its reliability.
- They are aware of the broader impact of technology on society, and the importance of using technology ethically.

Our students are well-prepared to succeed in **GCSE Computer Science**, **Creative iMedia**, or other technology-related courses.

They also take with them a wide range of transferable skills including logical thinking, creativity, independence, and critical analysis that will support them in any subject or career path.

Allocated Curriculum Time:

	Year 7	Year 8	Year 9	Year 10	Year 11
Lessons per fortnight	2	2	2	5/6	5/6

Year 7 Programme of Study - Foundations of Computing

Term	Curriculum Foci	Formal Assessment
1	**7.1 Digital Citizenship** The importance of usernames, passwords, and online security. Students will explore communication tools, develop online safety and critical thinking skills, understand digital footprints, learn graphic design basics, and gain proficiency in spreadsheets, word processors, and presentation software.	**Baseline Assessment** *Online* **Time:** 20 mins **Marks:** 20 marks
2	**7.2 Computer Systems** The computer model, including inputs, outputs, processing, and storage. Students will explore different types of computers, examine input and output devices, learn about internal components, gain insights into data processing with the CPU, and differentiate between memory and storage.	**Autumn Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks
3	**7.3 Think Like A Programmer** The basics of programming, focusing on sequencing, variables, and conditional statements. Students will explore logical and conditional operators, learn about iteration (particularly count-controlled), and apply problem-solving strategies to develop and debug accurate programs.	**Visual Programming I** *Practical* **Time:** 60 mins **Marks:** 36 marks
4	**7.4 Connected Worlds** The basics of networks, including types and protocols, and learn about key hardware components. Students will compare wired and wireless networks, explore the internet and World Wide Web, and review these concepts through summative assessments.	**Spring Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks
5	**7.5 Inside The Machine I** Understanding bits as the fundamental unit of data, exploring digital data units, and practising binary-decimal conversions and binary arithmetic. Students will review Alan Turing's contributions and engage in problem-solving activities related to Turing machines and their significance in computing history.	**Code Breaking** *Online* **Time:** 30 mins **Marks:** 30 marks
6	**7.6 Building Better Programs** Subroutines, condition-controlled loops, and different types of iteration. Students will explore data structures such as lists, learning to create, access, and modify them. Students will also practise problem decomposition to solve programming tasks effectively.	**Summer Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks

Year 8 Programme of Study - Building the Digital World

Term	Curriculum Foci	Formal Assessment
1	**8.1 Cyber Specialists** The importance of data safety, learn about social engineering tactics, hacking methods, and the Computer Misuse Act. Students will explore malware, network security tools, and practise recognising signs of network attacks and how to respond effectively.	**Under Attack Assessment** *Practical* **Time:** 60 mins **Marks:** 36 marks
2	**8.2 Design Laboratory** Digital graphics, pixels, resolution, and raster images. Students will explore basic and advanced Photoshop tools, learn about different image file types, and practise creating designs. Students will also analyse design briefs and evaluate their finished projects.	**Autumn Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks
3	**8.3 Building The Web** The basics of HTML and how to create web pages, display images, and control their appearance. Students will explore CSS for styling, understand SEO techniques, practise advanced search methods, and design user-friendly website navigation using HTML and CSS.	**Web Design** *Practical* **Time:** 60 mins **Marks:** 36 marks
4	**8.4 Application Creators** The basics of app design, including layout, colours, and navigation. Students will explore event-driven programming, practise debugging, capture user input, and plan and build an app prototype based on a design brief.	**Spring Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks
5	**8.5 Inside The Machine II** The basics of binary logic and operations, understanding logic gates, and exploring how computers represent images and sound using binary. Students will review binary numbers and representation and apply their knowledge through assessments and problem-solving tasks.	**Binary II Assessment** *Online* **Time:** 40 mins **Marks:** 40 marks
6	**8.6 From Blocks To Code** Students will transition from Scratch to Python, learning basic syntax, input functions, data types, and selection statements. They will explore loops (for and while) to handle repetition and design, code, and test a Python number game using key programming concepts.	**Summer Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks

Year 9 Programme of Study - Architects of Technology

Term	Curriculum Foci	Formal Assessment
1	**9.1 From Coders To Devs** Students will use sequences like lists and strings, learn to find, slice, and modify items, and explore techniques like concatenation. They will use loops to repeat actions, apply built-in functions, and understand nested sequences to organise complex data.	**Python II Assessment** *Online* **Time:** 40 mins **Marks:** 40 marks
2	**9.2 Design Like A Pro** Understanding the difference between vector and raster graphics, learning about vector formats, and exploring basic vector elements and software tools. Students will apply transformations, use layers, add colours and patterns, and explore practical applications in fields like graphic design and illustration.	**Autumn Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks
3	**9.3 Data Done Right** Spreadsheets, including the organisation of cells, rows, and columns. Students will learn about data types, cell referencing, formatting, creating charts, setting up data validation, and explore how spreadsheets are used in fields like finance, business, and science.	**Spreadsheets Assessment** *Practical* **Time:** 60 mins **Marks:** 36 marks
4	**9.4 Technology Builders** The Micro:bit's features, components, and input/output concepts in physical computing. Students will learn to set up the device, write and upload programs, use sensors and actuators, and approach problem-solving through designing, prototyping, and testing solutions.	**Spring Term Assessment** *Written Summative* **Time:** 40 mins **Marks:** 40 marks
5	**9.5 Think Like A Computer** Understanding what algorithms are and how to design them using techniques like flowcharts and pseudocode. Students will learn about searching (linear, binary) and sorting algorithms (bubble, merge, insertion) and apply algorithmic thinking techniques to solve problems and interpret pseudocode.	**Algorithms Assessment** *Online* **Time:** 20 mins **Marks:** 20 marks
6	**9.6 Code It To Play It** Students will recap how variables store information, combine them with loops, conditionals, and randomisation, and use functions and arrays. They will understand return functions and apply these concepts to create their own arcade game using these programming elements.	**End of KS3 Assessment** *Written Summative* **Time:** 50 mins **Marks:** 50 marks

Year 10 GCSE Computer Science Programme of Study
Exam Board: OCR
Exam Specification: J227

Term	Curriculum Foci	Formal Assessment
1	**1.1 Systems Architecture:** The CPU's purpose, fetch-decode-execute cycle, component functions, CPU performance factors, Von Neumann architecture, and embedded systems. **2.1 Algorithm Design:** Computational thinking principles, identifying inputs, processes, outputs, and working with structure diagrams, pseudocode, and flowcharts to solve problems.	**Systems Architecture Assessment** **Time:** 40 minutes **Marks:** 40 marks
2	**1.2 Memory & Storage:** Primary and secondary storage, RAM and ROM, binary data conversion, data storage units, and storage device types. Character sets, image and sound representation, binary shifts, and the need for compression, including lossy and lossless methods.	**Memory & Storage Assessment** **Time:** 60 minutes **Marks:** 60 marks
3	**1.3 Computer Networks:** LAN and WAN networks, performance factors, network hardware, topologies, connection modes, encryption, addressing, protocols, and the concept of network layers. **2.2 Programming Fundamentals:** Variables, constants, operators, inputs, outputs, assignments, programming constructs, arithmetic and Boolean operators, and data types including integer, real, Boolean, character, string, and casting.	**Computer Networks Assessment** **Time:** 40 minutes **Marks:** 40 marks **Programming Fundamentals Assessment** **Time:** 40 minutes **Marks:** 40 marks
4	**1.4 Network Security:** Forms of attack, including malware, social engineering, brute force, DoS, data interception, and SQL injection, along with prevention methods like firewalls, encryption, and penetration testing. **2.2 Additional Programming Techniques:** String manipulation, file handling, using records and SQL, arrays (1D and 2D), subprograms for structured code, and random number generation.	**Network Security Assessment** **Time:** 40 minutes **Marks:** 40 marks
5	**1.5 Systems Software:** The purpose and functionality of operating systems, including user interface, memory and file management, multitasking, and drivers. Utility software, encryption, defragmentation, and data compression. **2.1 Searching & Sorting Algorithms:** Searching algorithms: binary, linear and sorting algorithms: bubble, merge, insertion.	**Systems Software Assessment** **Time:** 40 minutes **Marks:** 40 marks

6	**Exam Technique:** The format and structure of GCSE Computer Science exams, key topics to be assessed, the marking scheme, and effective time management strategies. **2.1 Designing, Creating & Refining Algorithms:** Create, interpret, correct, complete, and refine algorithms, identify common errors, and use trace tables to debug programs.	**Year 10 Combined Mock** **Time:** 1 hr 30 mins **Marks:** 80 marks

Year 11 GCSE Computer Science Programme of Study
Exam Board: OCR
Exam Specification: J227

Term	Curriculum Foci	Formal Assessment
1	**2.5 Programming Languages & the IDE:** The characteristics and purposes of high-level and low-level programming languages, the role of translators, compilers, and interpreters, and explore common IDE tools such as editors, error diagnostics, and run-time environments. **1.6 Ethical, Legal, Cultural & Environmental Impacts of Technology:** The impacts of digital technology on society, including ethical, legal, cultural, environmental, and privacy issues, and explore relevant computer science legislation and software licences.	**Programming Languages & The IDE Assessment** **Time:** 40 minutes **Marks:** 40 marks **Issues Assessment** **Time:** 40 minutes **Marks:** 40 marks
2	**2.3 Producing Robust Programs:** Defensive design, input validation, maintainability, testing types, identifying syntax and logic errors, refining algorithms, and selecting suitable test data, including normal, boundary, and invalid data for effective program testing. **2.4 Boolean Logic:** Simple logic diagrams using AND, OR, and NOT operators, constructing truth tables, combining Boolean operators, and applying logical operators in truth tables to solve problems.	**Year 11 Combined Mock Paper November:** Elements of Paper 1 and Paper 2 **Time:** 1 hr 30 mins **Marks:** 80 marks
3	**Revision:** In class recap and revision of all taught topics.	**Practice Papers**
4	**Individual Personalised Revision:** Student led individual revision of topics, using mock exam feedback and action plans to target key areas of weakness.	**Year 11 Mock GCSE Exams:** **Paper 1 & Paper 2** **Time:** 1hr 30mins each paper **Marks:** 80 each paper
5	**Exam Technique:** Reading and interpreting exam questions accurately, planning and structuring clear responses, and managing time efficiently during exams, prioritising tasks appropriately. **GCSE Computing Exams take place (May)**	**Computer Science GCSE Exam Papers 1 & 2**

Revision and Support:

There are many ways in which you can support your child in the study of Computer Science such as:

- Ensuring your child has access to revision guides and up to date login details (provided by Ansford) for online platforms such as Know It All Ninja, Isaac Computer Science, and Seneca.
- Set up a dedicated space at home where your child can comfortably revise and recap topics for Computer Science without distractions. Ensure there is access to a computer or laptop with Python software installed, or Internet access for online IDEs.
- Computer Science can sometimes be challenging, especially when learning new programming concepts or algorithms. Encourage your child to persevere through difficulties and celebrate their successes along the way.
- Show genuine interest in what your child is learning. Ask them about their projects, assignments, or any challenges they may be facing. Engaging in discussions about computer science topics can help reinforce their understanding.
- Offer assistance with homework assignments or projects when needed. This could involve discussing ideas or reviewing concepts together.
- Discuss with your child how Computer Science is applied in real-world scenarios, such as in technology, business, medicine, and entertainment. Draw their attention to news articles that mention Computer Science. This will help them to foster a wider awareness of Computer Science issues.
- Support your child's curiosity by encouraging them to explore different aspects of Computer Science beyond the GCSE curriculum. This could include coding challenges, online tutorials, or exploring new programming languages.

Final GCSE Assessment Structure:

Component	Weighting (%)	Content	Proposed Examination Date
Paper 1 J227/01 1hr 30 mins	50	**Computer Systems** Central processing unit (CPU), computer memory and storage, data representation, wired and wireless networks, network topologies, system security and system software. Ethical, legal, cultural and environmental concerns associated with computer science.	June 2025
Paper 2 J227/02 1hr 30 mins	50	**Computational Thinking, Algorithms & Programming** Students apply knowledge and understanding. Computational thinking: algorithms, programming techniques, producing robust programs, computational logic and translators.	June 2025

Please see exam board websites for up to date information:

https://www.ocr.org.uk/qualifications/gcse/computer-science-j277-from-2020/ -