Ansford Academy

Computing

Curriculum Booklet for 2024 - 2025

Subject Lead: Charlotte 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, the Computing curriculum has been crafted to help students learn important skills for today's digital world. We follow the National Curriculum pillars of progression: Digital Literacy, Information Technology, and Computer Science. These content areas ensure that every student feels comfortable and capable when using technology. We focus on teaching safe and smart ways to use technology, solving real-world problems with digital tools, and learning how computers work and think. Our Computing lessons mix learning ideas (declarative knowledge) with learning how to do things (procedural knowledge). We want students to be good with technology, think carefully, and be ready for what's coming next.

Digital Literacy helps students to use technology safely and sensibly. We teach them to understand how the online world works, how to behave responsibly online, and to think about the right and wrong ways to use technology.

In Information Technology, students learn to use digital tools to solve everyday problems. They practise using different software and apps to work with information effectively. This helps them become better at creating, organising, and sharing digital projects. It's all about gaining hands-on skills that improve both their technical abilities and their ability to be precise and creative in their work.

Computer Science is at the heart of what we teach. Students learn how to think logically and solve problems using computers. They practise writing code and creating step-by-step plans to solve tricky problems. This helps them develop strong computational thinking skills, bounce back from challenges, and come up with new and smart ideas.

The KS3 Computing curriculum offers an engaging program designed to build foundational skills and knowledge across a range of topics in the digital world. It builds the foundations needed to be successful at GCSE level and beyond.

We encourage students to use their creative problem-solving skills outside of computing lessons. We show them how these skills can help in subjects like Maths, Science, Design & Technology, Music, Art, and Humanities. This way, they see how computer skills are useful in many parts of school and life.

Students have the opportunity to work towards the iDEA qualification in their own time. This is set as independent learning, with the goal of completing their Bronze award at the end of Year 8, and continuing with Silver as they progress through the upper school. Students are also provided with the opportunity to take part in the BEBRAS challenge and Cyber Explorers program.

Extra-curricular activities include Minecraft Club, Computer Game Design Club and Music Production Club.

Computing Curriculum Implementation:

Computing is for all students, no matter of previous experience or perceived abilities. We make sure everyone gets the right help and challenge to do their best.

In our lessons, we plan different goals and activities to fit everyone's skills. We start with easier tasks and move to more challenging ones so that everyone can get involved no matter their prior knowledge or skill level. Students are guided to complete the tasks and activities that are at the appropriate level for them.

We teach difficult ideas step by step. We use techniques such as teacher/peer modelling, visual guides, and clear instructions. Programming is also approached through the PRIMM (predict, run, investigate, modify, make) method. This supports students who need additional help while allowing advanced learners to individually explore topics in greater depth.

Educational technology and online tools are used to help students learn in a way that suits them. They can go at their own speed and get instant feedback. There are also extra resources to help them understand more or learn more about a chosen topic. Students can nominate themselves as ‘computer scientists’, and are encouraged to support other students in the class, troubleshoot problems and offer guidance alongside the teacher.

Quick quizzes, peer reviews, and checklists are used to check how students are doing regularly. This ongoing assessment helps us see where students might need more help or further challenge. At the end of each unit, we have tests and projects to see how well students understand and use what they've learned. These assessments are designed to be accessible to all students, with clear assessment criteria and multiple ways to demonstrate learning and receive feedback.

We give feedback in different ways: we talk about it in lessons, students can give feedback to each other, teachers assess and write comments on digital assignments, and we have 1:1 conversations about individual work.

Key Stage 3

Each lesson and topic is sequenced so that it builds on previous knowledge and understanding. Due to the varying experience of Computing as a subject at primary school, a baseline test is conducted for new Year 7 students. This allows us to tailor the early stages of the curriculum to particular areas of need within a cohort.

Key Stage 4

At Key Stage 4, students follow the OCR GCSE Computer Science curriculum, which enables them to progress to post-16 courses.

Allocated Curriculum Time:

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

Year 7 Programme of Study

Term	Curriculum Foci	Formal Assessment
1	**7.1 Digital Literacy** 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 Hardware** 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 Visual Programming I** 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 Networks** 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 Binary 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 Visual Programming II** 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.	**Year 7 Exam** *Written Summative* **Time:** 40 mins **Marks:** 40 marks

Year 8 Programme of Study

Term	Curriculum Foci	Formal Assessment
1	**8.1 Cyber Security** 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 Raster Graphics** 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.	**Digital Graphics** *Practical* **Time:** 1hr 30 mins **Marks:** 36 marks
3	**8.3 Web Design** 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 Mobile App Development** 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.	**App Development** *Practical* **Time:** 60 mins **Marks:** 36 marks
5	**8.5 Binary 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 Python I** 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.	**Year 8 Exam** *Written Summative* **Time:** 40 mins **Marks:** 40 marks

Year 9 Programme of Study

Term	Curriculum Foci	Formal Assessment
1	**9.1 Python II** 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 Vector Graphics** 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.	**Vector Graphic Design Task** *Practical* **Time:** 1hr 30 mins **Marks:** 36 marks
3	**9.3 Spreadsheets** 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 Physical Computing** 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.	**The Micro:bit Project** *Practical* **Time:** 60 mins **Marks:** 36 marks
5	**9.5 Algorithms** 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 Minecraft GameCode** 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.	**Year 9 Exam** *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:** 40 minutes **Marks:** 40 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
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/ -