Curriculum overview: Computer Science 2026/27

Key Stage 2

No prior learning is assumed due to variation of teaching amongst primary schools. Desirable prior learning is listed below:

Digital Literacy

- Evaluate sources of online content.
- Recognise common uses of technology.
- Understand how to seek help online.
- Recognising unacceptable behaviour and the reasons for age restrictions.

Information technology

- Creating, opening, saving files within a folder structure.
- Basic proficiency in the use of office software.
- Use the internet to search for and retrieve information.
- Be able to use a range of software to present data and information in a suitable format.

Computer Science

- Key programming concepts including sequence, selection, iteration, and inputs / outputs.
- Design, write and debug programs to achieve a specific goal.
- Writing of simple algorithms.
- Understand the concept of networking and communication to provide.

Key skills/content requirements at GCSE

Theoretical Computing

Students will follow the OCR J277 Computer Science course. Students will develop an understanding of the following key areas within computer science:

1.1 – Systems architecture
1.2 – Memory and storage
1.3 – Computer networks, connections and protocols
1.4 – Network security
1.5 – Systems software
1.6 – Ethical, legal, cultural and environmental impacts of digital technology
2.1 – Algorithms
2.2 – Programming fundamentals
2.3 – Producing robust programs
2.4 – Boolean logic
2.5 – Programming languages and Integrated Development Environments

It is essential that students have a strong grasp on the key vocabulary associated with the subject allowing them to describe and explain the theory behind how a computer works. Students must also have the knowledge of how to solve a problem and develop a solution.

Students will need to understand a range of processes which occur within computers, in order to explain how a computer ultimately responds to user input and guarantees security.

Programming

- Write algorithms to solve given problems.
- Visually represent algorithms using flowcharts.
- Write subroutines to make flowcharts more manageable.
- Use Python to program solutions to given problems.
- Order code so that Python runs in sequence.
- Store values in variables so they can be manipulated.
- Take user input and store the values in variables.
- Manipulate values using mathematical operators.
- Use selection to decide what code to run based on a condition.
- Make decisions using comparative operators.
- Use iteration to repeat sections of code.
- Work out when to use condition-controlled (while) or count-controlled (for) iteration.
- Write functions to better organise programs.
- Pass parameters between functions to make code more efficient.
- Use inbuilt functions contained within libraries.

Curriculum Overview

Theoretical Computing – Throughout study students will develop their understanding of the technology underlying the modern day systems which we rely on and utilise in our everyday lives. Students will be introduced to concepts surrounding computer memory and storage, data representation, wired and wireless networking, and the ethical, legal, and cultural considerations associated with Computer Science.

Programming – Students will develop their ability to recognise a problem, extract important information, design a solution to a problem, implement using code, test and evaluate fully. Students will be introduced to concepts including variables, sequencing, selection, iteration, file handling, arithmetic and Boolean operators, random number generation, and the use of subroutines. Languages that will be covered in depth during GCSE include Python and SQL.

	Term 1	Term 2	Term 3	Portable knowledge	Key terms
Year 6	**6.1 E-Safety and Computing Introduction** • Purpose and features of strong passwords. • Create folder structures to allow for structured and orderly storage of files. • Communicate via email using CC, BCC, attachments, importance, flags. • Access, complete, and upload working using Teams as a platform • Describe the difference between personal and private information. • Understand the risks that exist online how to reduce and respond to them. • To explain what cyber bullying is, its effects and how to report it. • Identify reliable sources of support for online issues	**6.2 Word Processing Software** • Explain the purpose of various formatting tools. • Use headings to format a document with consistency and create contents tables. • Make use of headers/footers appropriately. • Separate content using bullet points and numbering where appropriate. • Use keyboard shortcuts to optimise working. • Use images and tables to enhance appearance and readability of documents. • Effectively use formatting tools for large documents in preparation for future schoolwork and career	**6.3 – Presentation Software** • Explain the purpose of various formatting tools. • Use animations, transitions, and triggers effectively to aid a presentation. • Use hyperlinks to navigate through a presentation. • Incorporation of multimedia into a presentation. • Use of the slide master to create a consistent look and feel to a presentation. **6.4 – Micro:bits** • Using block-based programs with a physical computing device • Use of iteration to repeat sections of code • Use of selection to divert flow of a program • Using variables to manage data	Use of passwords and how to create strong passwords. File and folder access and manipulation. Use of Office software to create documents; understanding of how to best format documents for a given purpose. Risks faced online, risk reduction strategies and actions to be taken if in danger. Use of email and MS Teams, including attachments, highly useful for contacting teachers and sending work. Concepts of variables, sequence, selection, iteration in programming	File Folder Email Email address Attachment Carbon Copy Blind Carbon Copy Username Password Private information Public information Inappropriate content Presentation Animation Transition Trigger Formatting Text Font Image Content Sequence Selection Iteration Variable

Year 7	**7.1 Spreadsheet Modelling** • Identify a cell's location using cell referencing. • Identify a group of cells using a range. • Format and structure data as appropriate in a spreadsheet. • Use formulae to perform calculations. • Use conditional formatting to format certain elements based on a condition. • Sort and search for data within a spreadsheet. • Use graphs and charts to display data in a user-friendly visual format.	**7.2 Hardware and Software** • Understanding the difference between input, output and storage peripherals. • Identify the purpose of hardware devices that exist within a computer system. • Explain how certain devices can be used by users with certain needs to assist with their day-to-day lives. • Explain why binary numbers are used within computers • Convert between binary and denary system numbers • Categorise software based on its purpose (applications vs systems software)	**7.3 Algorithms 1** • Explain what an algorithm is, and what we mean by efficiency of algorithms • Explain how linear and binary search can be used to find data. • Explain how bubble sort operates when sorting data. • To explain the purpose of sequence, selection, and iteration in algorithm design & programming • Build flowcharts which visually represent real life algorithms • To create block-based programs which use selection and iteration.	Use of Excel to manipulate data, create formulae, sort and search and format data. Understand how assistive technologies can be beneficial for themselves or peers at school. What an algorithm is and how it can be designed using a flowchart. Explaining how algorithms can be used to sort and search data. Basic programming concepts from block-based language (input, variable, output, selection, iteration).	Spreadsheet Cell Cell reference Range Formula Hardware Peripheral Assistive technology Binary Denary Algorithm Flowchart Sequence Selection Iteration Input Process Output Variable

Year 8	**8.1 Online Safety and Cyber-Security** • Builds on unit 6.1 • Recognise the safety related risks when communicating online. • Explain how to practice respectful communication when online. • Recognise how to distinguish between real and fake news using a range of sources. • Explain the differences between types of malware. • Describe threats other than malware that could compromise a computer system. • Explain the different methods of protecting a computer system from malware and other threats. • Explain the role and purpose of encryption. • Perform the Caesar cipher to encrypt and decrypt data.	**8.2 Representing Data** • Explain why binary numbers are used by computers • Convert between binary and denary representation of numbers • To perform binary addition on 2 8-bit binary numbers. • Explain the need for hexadecimal numbers. • Convert between hexadecimal, binary and denary numbers. • Explain how images & sound are stored in a computer system and the factors that affect quality and file size. • Explains the purpose of character sets in a computer system. • Compare and contrast ASCII and UNICODE and explain why the latter is used in modern computer systems.	**8.3 Python 1** • Create programs to solve problems in a text-based language. • Use variables and inputs to take user entry and store it for later use in a program. • Output data to the shell using print. • Describe the difference between data types (Integer, Float, String). • Identify the correct data type to be used for a variable.	Staying safe online. Knowledge of malware and how to protect a computer system. Knowledge of best practice to protect a computer system from threats (eg. password strength). Binary conversion techniques that translate to most other forms of data representation. Representation of image, sound and characters within a computer system. Problem solving skills within text-based programming.	Malware Virus Worm Trojan Hacker Phishing Hacking Firewall Anti-malware Encryption Decryption Encryption key Cipher Binary Denary Bit Byte Hexadecimal ASCII Unicode Pixels Resolution Bit depth Compression Lossy Lossless Data type Comparative operator Comparative Operator Arithmetic operator Syntax

Year 9	**9.1 Cyber-Security** • Understand the safety risks when communicating online and how to respectfully and safely use online platforms • Distinguish between news sources of varying legitimacy • Explain the differences between types of malware. • Describe threats other than malware that could compromise a computer system. • Explain different methods of protecting a system from malware and other threats. • Explain the role and purpose of encryption. • Perform the Caesar and keyword ciphers to encrypt and decrypt data. • Compare and contrast the Caesar and keyword ciphers. • Explain the features that make a good keyword for encryption purposes.	**9.2 Representing Data** • Understand and explain why binary numbers are used within computers • Explain the need for the use of hexadecimal numbers • Be able to convert between the three number systems of binary, denary, and hexadecimal • Perform basic addition on 8-bit binary numbers • Perform binary shifts on 8-bit binary numbers • Explain the effects of a binary shift • Understand when an overflow error occurs, and what the effect is. • Explain how characters can be represented within computer systems • Understand the need for character sets, and compare and contrast examples of sets – ASCII and Unicode • Explain how bitmap images are stored as binary within a computer system • Understand the differences between analogue and digital data • Explain how sound can be converted to be stored in a digital form within a computer • Explain the impacts on quality and file size of image and sound files	**9.3 Python 2** • Use an IDE to be able to write, test, and debug programs. • Create programs to solve a given problem with a text-based language. • Understand the need for variables, inputs, and outputs in a program. • Describe the difference between common data types within programming. • Identify the most suitable data type for a given problem context. • Understand the principles of sequence, selection, and iteration within programming. • Use arithmetic and comparative operators within programs. • Understanding that different languages exist and the reasons for different forms of program expression.	Respectfully and safely communicating online. Knowledge of threats to, and how to protect, a computer system. Binary conversion techniques that translate to most other forms of data representation. Representation of images, sounds, and characters within a computer system. Understanding of the need for different data representation systems in a global context. Logical thinking and problem-solving in text-based programming design.	Malware Virus Worm Trojan Hacker Phishing Hacking Firewall Anti-malware Encryption Decryption Encryption key Cipher Binary Denary Bit Byte Hexadecimal ASCII Unicode Pixels Resolution Bit depth Compression Lossy Lossless Data type Comparative operator Comparative Operator Arithmetic operator Syntax

Year 10	**10.1 Representing Data** • Explain the purpose of binary numbers. • Name the units of measurement between bit and Petabyte • Calculate data capacity requirements within a system. • Convert numbers between binary, denary and hexadecimal representation. • Explain the purpose and use of number systems. • Perform binary addition. • Explain the purpose of and perform binary shifts. • Explain the purpose of a character set • Compare and contrast ASCII & Unicode • Explain how bitmap images are stored in a computer system. • Explain what metadata is • Explain how the quality and size of image and sound files is affected by a range of factors. • Explain the need for compression • Explain the differences between lossy and lossless compression. **10.2 Algorithms & Programming 1** • Explain the principles of computational thinking • Design, create and refine algorithms using pseudocode, flowcharts • Complete trace tables to show the values of variables when an algorithm is executed. • Perform binary & linear search. • Perform Bubble, Merge & Insertion sort.	**10.2 Algorithms & Programming 1 (Cont.)** • Compare the efficiency of sorting & searching algorithms • Explain the purpose of & use variables, constants, operators, inputs, outputs and assignments in Python. • Explain the three constructs of programming used to control program flow. • Name and perform common comparison and arithmetic operations. • Explain the need for data types and their features (Integer, real, Boolean, character, string) **10.3 – Systems Architecture** • Explain the difference between primary memory and secondary storage. • Explain the differences between RAM and ROM. • Explain the purpose and operation of virtual memory. • Explain the characteristics of CPUs and their effect on performance (Clock speed, Cache size, Number of cores) • Explain the purpose of the CPU and the FDE cycle. • Explain the purpose and function of the ALU, CU, Cache and Registers • Explain features of Von-Neumann architecture devices. • Explain the purpose of embedded systems and name examples. • Create simple logic diagrams using AND, OR and NOT • Create truth tables for given logic diagrams. • Combine operators to create logic circuits.	**10.4 – Programming 2** • Perform string manipulation • Perform file handling in Python. • Create both 1D and 2D arrays to solve a given problem • Explain the purpose of subroutines, distinguishing between procedures and functions. • Explain how random numbers can be generated in Python. • Create subroutines in Python. • Explain why programs should be designed with defensive considerations and maintainability considerations. • Explain the purpose of validation methods and implement in Python. • Explain common conventions for improving program maintainability. • Explain the purpose of testing and the various forms which exist. • Select and use suitable testing data. • Explain what SQL is • Explain how SQL can be used to interrogate data. • Build and execute SQL statements to interrogate data. **10.5 – Extended Programming Challenge** Students apply their prior learning from 10.2 and 10.4 to complete an extended programming challenge. Students will: - Analyse a given problem - Design a solution - Develop the solution - Test the solution - Evaluate the solution.	The use of binary and its purpose within a computer system. Conversion between binary, hexadecimal and denary numbers. Units of measurement within computing and how conversions are performed. Purpose of character sets and the different types. Storage of sound, images in a computer system and the role that compression plays on affecting file size. Principles of computational thinking and problem solving using abstraction, decomposition, sequence, selection and iteration. Development of flowcharts and pseudocode to model algorithms. The use of constants and variables within programming and their purpose. Comparative and arithmetic operators used in computer science. The role of memory and storage within a computer system. The purpose and use of logic gates when building circuits and writing programs. The impact of processor characteristics on the performance of a device.	Binary Denary Hexadecimal Bit Byte Kilobyte Megabyte Gigabyte Petabyte Binary shift Character set ASCII Unicode Bitmap Metadata Compression Lossy Lossless Flowchart Process Decision Input Terminator Trace table Binary search Linear search Bubble sort Insertion sort Merge sort Variable Constant Sequence Selection Iteration Data type Integer String Float Boolean RAM ROM Virtual memory Magnetic Optical Solid State Clock speed Cores Cache F-D-E cycle

				The role of CPU components within the FDE cycle. Manipulation of strings to achieve a given outcome. Manipulation of lists to store, modify and interrogate data in an efficient manner. The use of subroutines to improve code efficiency. Implementation in Python. Types of testing data and why they should be used to test a program effectively. Features of defensive design and maintainability when writing programs. The role of SQL and how it is used to interrogate data.	Arithmetic logic unit Cache Register Von Neumann Embedded system AND/OR/NOT Logic gates Truth table File handling Array Subroutine Parameter Function Procedure Library Defensive design Maintainability Validation Syntax error Logic error SQL Query SELECT FROM WHERE

Year 11	**11.1 – Networking & Security** • Explain the characteristics of LAN and WAN networks. • Explain factors which affect the performance of a network. • The roles of computers in CS and P2P networks. • Explain the role of hardware required to create a LAN network. • Explain the concepts of DNS, Hosting, the cloud and webservers. • Compare star and mesh topologies. • Name modes of connection and explain their use within networks. • Explain the purpose of addressing • Name common networking standards • Explain the purpose of common network protocols. • Explain the concept of layering. • Explain the various forms of threat to computer security • Explain methods of identifying and preventing network vulnerabilities. **11.2 – Software** • Explain the purpose and functionality of operating systems (UI, Memory management, Multi-tasking, Peripheral management, Drivers, User management, File management) • Explain the difference between the purpose and function of applications and utility software. • Explain the role and purpose of encryption, defragmentation, and compression software. • Explain the purpose and features of high- and low-level languages • Explain the purpose of translator software • Explain the characteristics of compilers and interpreters. • Explain the tools and facilities which are available in an IDE. • Explain the difference between open-source and proprietary software licensing, and off-the-shelf and custom written software.	**11.3 – The Impact of Computing** • Explain the impact of technology on wider society. • Evaluate the impact of computing in relation to ethical, legal, cultural, environmental and privacy related issued. • Explain the purpose of key legislation in the computer science field: • - Data Protection Act (2018) • - Computer Misuse Act (1990) • - Copyright, Designs and Patents Act (1988) **11.4 Revision Plan** Topics to be determined as appropriate. This will be guided by the mock assessments and departmental data collected throughout the duration of the course.	**11.4 Revision Plan (Cont.)** Topics to be determined as appropriate. This will be guided by the mock assessments and departmental data collected throughout the duration of the course.	Key networking hardware and their role within LAN and WAN networks. Classifications of networks and associated characteristics. Concepts of client-server and peer to peer networks, their strengths and drawbacks. The role of a topology and its impact on network design and performance. Why standards are used in networking and their relevance with regards to network design. Types of software and their classification. The purpose, roles and features of operating systems. The role of utility software in maintaining a computer system. The purpose of translator software and the different types, comparing their strengths and weaknesses. The impacts of computing on wider society, in terms of ethical, legal, cultural, environmental and privacy implications. Key legislation within the computing field and how this impacts upon practice in the workplace.	LAN WAN Latency Packet Client Server Peer Router Modem Switch Hub WAP NIC DNS Host Cloud Topology Star Mesh Full mesh Partial mesh Ethernet Wi-Fi Protocol Layering Security Malware Hacker Brute force Antivirus Firewall Software User interface GUI Command line Application software Utility software Encryption Defragmentation Translator Compiler Interpreter IDE Open-source Proprietary Custom-written Off-the-shelf Ethical Legal

	• Recommend a type of license for a given scenario.				Cultural Environmental Privacy Data Protection Act GDPR Computer Misuse Act Copyright

GCSE external assessment:

Computer Science uses the GCSE 1-9 grading system, where 9 is the best grade. All examinations are terminal (at the end of Year 11). The assessments are comprised of the following components:

- Paper 1: Computer Systems. The paper lasts for 1 and a half hours and is worth 50% of the GCSE grade, 80 marks are available on this paper.
- Paper 2: Computational thinking, algorithms, and programming. The paper lasts for 1 and a half hours and is worth 50% of the GCSE grade, 80 marks are available on this paper.

SMSC in computer science

Spiritual development in computer science

Students are continually reflecting on their own lives and the lives of others as they look at various Computing case studies. Students debate and formulate their own set of values and beliefs through case studies as they share their own experiences. Computing is an area of rapid development and change, this provides students with the opportunity to reflect upon this progress and potential new technologies which will be developed in time.

Moral development in computer science

Within computing, it is important to consider many areas of the human impact technology has. Society is not only becoming more reliant on technology, but the increasing rate in which computers are updated causes substantial waste, as well as increased carbon footprint in line with their increased production. Students will investigate the use of social-networking and cyber bullying, whilst learning about the legal implications of immoral acts undertaken online. Students will consider where boundaries should lie and the impact of computing on the environment.

Social development in computer science

Computing can also help all students to express themselves clearly and to communicate. As students progress through their learning, they will consider more complex social needs and are encouraged to research and work to find appropriate solutions to issues that may affect particular groups within society.

Cultural development in computer science

With the increased use of social media sites, people are becoming more culturally aware due to the diversity of content posted online for all the world to see. Computational thinking encourages problem solving and thinking about how to solve an issue from another perspective – a valuable transferable skill that translates to many aspects of life. Students will consider the positive and negative effects of computing upon various groups of people.