

IT / Computing @ Caedmon

National curriculum for IT/Computing: Purpose of study

A high-quality computing education equips students to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which students are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, students are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that students become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

Aims of the National curriculum for IT/Computing

The national curriculum for computing aims to ensure that all students:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.

Aims of our Caedmon curriculum for IT/Computing

Computers and ICT are a part of everybody's lives at work and in the home. Therefore, we aim to develop all students' digital literacy and computing skills to ensure they can function and thrive in an increasingly digitalised world.

Foundation learning – what the National Curriculum expects students to have studied in IT/Computing by the end of KS3

Pupils will be taught to:

- design, use and evaluate computational models that help understand the behaviour of real-world problems and physical systems
- understand key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- use programming languages to solve a variety of computational problems; make appropriate use of data for example Python, SQL and, HTML
- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems

- understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits
- undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns

Year 7 IT / Computing @ Caedmon

Our aim in Year 7 is to ensure students know how to be safe online. Introduce students to the core concepts of computer hardware and software. Develop an understanding of algorithms and a basic introduction to coding. Build on existing knowledge and understanding of software that helps students to produce good quality work across all subjects and to use the internet to find accurate and relevant information.

	Topics, themes and skills covered	Assessment
Autumn 1	General Organisation - File mgt, passwords, email, VLE, Google Docs, Homework DL - Office Documents & The Cloud DL - Web Research Unit - Local Heritage Webquest - web searches C - Matrix1 3.2 - 3.6 Data representation & binary	Yacapaca Baseline Assessment - ICT & Computing
Autumn 2	C - Matrix1 Computational Thinking 1.1 - 1.5 (see links below BBC Computation Thinking Learner Guides & Assessment) Algorithms, psuedocode, flowcharts, control software e.g. Logo DCT - Matrix1 5.5 -5.6 E-Safety Unit - booklet	
Spring 1	PC Basics - Input, output, storage, intro to internet C - Matrix1 4.1 - 4.6 Programming with BBC Microbit Python	Mid-year computing assessment
Spring 2	C - Matrix1 4.1 - 4.6 Programming with BBC Microbit Python DL - Word Processing DL & DC - DTP & Photoshop	
Summer 1	DL - Data Handling - intro to databases introduction & superheroes	
Summer 2	DL - Modelling Basic data entry, formatting, basic calculations, graph & questions. DL & DC - DTP & Flash Animation	End of year assessment - ICT & Computing

Year 8 IT / Computing @ Caedmon

Our aim in Year 8 is to develop student's knowledge and understanding of computer hardware and software that they were introduced to in Year 7. Students explore more complex algorithms and develop their coding skills using a programming language. Introduce more advanced features of software and searching techniques that allow students to produce great work across all subjects. Develop a more in-depth understanding of how to be safe online.

	Topics, themes and skills covered	Assessment
Autumn 1	General Organisation - File mgt, passwords, email, VLE, GoogleDocs, Homework DL - Matrix2 5.4 -5.6 Web Research Unit - advanced searches, reliability & bias DC - Adobe Premier Pro Film Editing	Yacapaca Baseline Assessment - ICT & Computing
Autumn 2	DL - Matrix2 6.6 Modelling Formatting, formulae, statistical functions, charts & questions. DCT - E-safety - Your Digital world	
Spring 1	DL - Data Handling - databases intermediate C - Matrix1 3.1, 5.1- 5.4 How computers and the Internet work	Mid-year computing assessment
Spring 2	C - Matrix2 4.1 - 4.6 Programming with Python DL - Wordprocessing	
Summer 1	C - Matrix2 6.1 - 6.5 HTML & Webpage design DCT - Matrix3 Esafety	
Summer 2	Computer Hardware - CPU, RAM, ROM, Storage C - Website Design Software	End of year assessment - ICT & Computing

Year 9 IT / Computing @ Caedmon

Our aim in Year 9 is to introduce students to the core concepts studied at GCSE and give them a realistic experience of what they will be doing in their GCSE. Students investigate in more depth computer hardware and software. Develop more independent coding skills and apply algorithmic thinking to their coding solutions. Students continue to develop core skills in other software and advanced searching techniques to help them produce great work that will be applied across all subjects.

	Topics, themes and skills covered	Assessment
Autumn 1	General Organisation - File mgt, passwords, email, VLE, GoogleDocs, Homework Searching the web & asset management inc. legal considerations (DL) ICT Literacy - email, WP,SS, Cloud storage & software C - Matrix3 Computer Hardware Pseudocode & algorithms (C) Python Programming Introduction (C) Booklet	Yacapaca Baseline Assessment - Computing
Autumn 2	C - Matrix3 Computer Hardware Psuedo Code & algorithms (C) Python Programming Introduction (C) Booklet DL - Spreadsheets DL - Databases	
Spring 1	C - Matrix3 Web Development HTML & CSS (C) DCT - Matrix3 Esafety	Mid-year computing assessment
Spring 2	C - Matrix3 How the internet works - networks & network hardware (C) DC - Sound Project using Audacity	
Summer 1	Legal & ethical computing issues (DCT) DC - Flash Animation	
Summer 2	C - Matrix3 Computer hardware & software DC - Web Graphics C - Code Academy SQL/PHP & Data Manipulation (C)	End of year assessment - Computing

Year 10 IT / Computing @ Caedmon

Our aim in Year 10 is to begin the formal learning required Unit 1 of the GCSE Computer Science qualification. Students will be taught the functions of core computer hardware and software, how networks work and how they are kept secure. Students also will be taught the fundamental ethical, legal, cultural and environmental impacts of digital technology

Topics, themes and skills covered	Assessment
1.1 Systems Architecture 1.1.1 Architecture of the CPU 1.1.2 CPU Performance 1.1.3 Embedded systems	Unit assessment
1.2 Memory and storage 1.2.1 Primary storage (Memory) 1.2.2 Secondary storage 1.2.3 Units 1.2.4 Data storage 1.2.5 Compression	Unit assessment
1.3 Computers networks, connections and protocols 1.3.1 Networks and topologies 1.3.2 Wired and wireless networks, protocols and layers	Unit assessment Trial Examination of content covered to date
1.4 Network security 1.4.1 Threats to computer systems and networks 1.4.2 Identifying and preventing vulnerabilities	Unit assessment
1.5 Systems software 1.5.1 Operating systems 1.5.2 Utility software	Unit assessment
1.6 Ethical, legal, cultural and environmental impacts of digital technology 1.6.1 Ethical, legal, cultural and environmental impact	Unit assessment Trial Examination of whole unit content covered

Year 11 IT / Computing @ Caedmon

Our aim in Year 11 is to begin the formal learning required for Unit 2 of the GCSE Computer Science qualification. Students will be taught about algorithms and computational thinking including how algorithms can be expressed. Programming fundamentals is delivered in a coding lesson per week over the two year course to build up knowledge and confidence in Python coding. Students will be taught how robust programs are created and tested. Logical thinking and boolean logic is covered and how coding languages and development interfaces help users create programs.

Topics, themes and skills covered	Assessment
2.1 Algorithms 2.1.1 Computational thinking 2.1.2 Designing, creating and refining algorithms 2.1.3 Searching and sorting algorithms	Unit assessment
2.2 Programming fundamentals 2.2.1 Programming fundamentals (developed in a weekly coding lesson over 2 years) 2.2.2 Data types 2.2.3 Additional programming techniques	Unit assessment
2.3 Producing robust programs 2.3.1 Defensive design 2.3.2 Testing	Unit assessment Trial Examination of content covered to date
2.4 Boolean logic 2.4.1 Boolean logic	Unit assessment
2.5 Programming languages and Integrated Development Environments 2.5.1 Languages 2.5.2 The Integrated Development Environment (IDE)	Trial Examination of whole unit content covered Unit assessment