

Computing/Computer Science Curriculum Overview

	Autumn Term		Spring Term		Summer Term	
Year 7	7a Digital Literacy Learning how to log on, store work effectively, and use Office 365, Teams, and Email. Links: 7C passwords	7b Computer Systems & Networking Understanding hardware components like input/output devices, CPU, memory, and networking basics. Links: 3.Systems storage	7c Online Safety Learning about e-safety, laws, ethics, hacking, DDoS, malware, and security measures. Links: 7a passwords, 8e fake news	7d Graphics Understanding vector and bitmap graphics, using fonts, colours, and images effectively, and learning photo editing tools. Links: 8d Binary Data	7e Spreadsheet Modelling Using spreadsheet features like cell referencing, formatting, formulas, data collection, functions, and charts. Links: CNAT IT	7f Block based Programming Introduction to programming concepts like sequencing instructions, iteration, inputs/variables, selection, concatenation, cumulative sum, and functions using Edublocks. Links: 8c text based programming, 8f algorithms
Year 8	8a Augmented Reality Introduction to AR and Blippar, including AR animations, interactions, and project planning with mindmaps and Lucidchart. Students build and test AR prototypes. Links: CNAT IT	8b Business & Finance Covers basic business concepts, entrepreneurship, market research, business planning, the marketing mix (4Ps), financial basics, stakeholder roles, and business ethics. Links: GCSE Business	8c Text Based Programming Introduction to Python programming basics: print, variables, data types, selection, iteration, lists, file handling, and functions. Links: 7f block based programming, 8f algorithms	8d Binary Data Converting between binary and denary numbers, performing binary operations, representing images/sound in binary, and understanding Boolean logic circuits. Links: 7d bitmaps	8e Experience AI Introduction to AI, how computers learn from data, addressing bias in AI, decision trees, solving problems with machine learning models, and exploring AI careers. Links: 7c fake news	8f Computational Thinking Introduction to computational thinking, including abstraction, decomposition, algorithms, flowcharts, searching algorithms, and sorting algorithms Links: 7f + 8c algorithms
Year 9	3.Systems storage This unit introduces the various types of storage systems and their applications, including embedded systems. 3.Systems architecture Covers the fundamental von Neumann architecture and the fetch-execute cycle. Links: 7b Computer Systems & Networking	3.System software Explores the purpose and functionality of operating systems, including process and peripheral management, virtual memory, file and user management, and utility system software. Links: 7b Computer Systems & Networking	2.Data representation – part 1 Focuses on how binary represents all data and instructions, number bases, two's complement representation, base number conversions, binary addition and shifts, and units. Links: 8d Binary Data	2.Data representation – part 2 Expands on data representation with lessons on character sets, bitmaps, retention, sound, and compression. Links: 8d Binary Data	4.Computer networks, protocols and layers part 1 Introduces types of networks, the internet, and wired and wireless networks. Links: 7b Computer Systems & Networking	4.Computer networks, protocols and layers part 2 Explores network speeds, common protocols, the four-layer TCP IP protocol model, and various network topologies such as star, bus, and mesh. Links: 7b Computer Systems & Networking
	6.Basic programming concepts Introduces fundamental programming concepts using Python. It covers printing, comments, input, data types such as integers and strings, basic error handling and debugging, variables, casting, constants, and basic arithmetic operators. Links: 8c Text Based Programming	6.Advanced programming concepts Building upon basic concepts, this unit delves into more advanced topics. It includes advanced operators, string manipulation, conditional statements (if, else, elif), and loops (while, for) Links: 8c Text Based Programming	6.Robust and secure programming part 1 Focused on writing robust and secure Python code, this unit covers loops (for, while), user interfaces, built-in functions (e.g., random), lists, 2D lists, and file handling (TXT and CSV). Links: 6.Advanced programming concepts	6.Robust and secure programming part 2 Continuing to enhance coding practices for robustness and security, this unit addresses code maintainability, robustness, input validation, user authentication, procedures, and parameter passing. Links: 6.Advanced programming concepts	6.Robust and secure programming part 3 Further develops skills in writing secure and maintainable Python code. It covers functions, modular code structure, retaining programming skills, and reinforcing secure coding practices Links: 8c Text Based Programming	6.Python project programming Providing practical application of Python programming skills, this unit involves building projects that integrate various concepts learnt throughout the course, fostering practical problem-solving abilities. Links: 8c Text Based Programming
Year 10	4.Network and cyber security 1 Explore the four-layer TCP/IP protocol model and various network topologies including star, bus, and mesh. They delve into network security, methods of protecting networks, and different types of malware such as viruses, worms, and trojans. Links: 7c Online Safety	4.Network and cyber security 2 Examine advanced malware like ransomware and keyloggers, as well as social engineering techniques such as pretexting, phishing, baiting, and quid pro quo. They also learn methods to protect digital systems and data. Links: 7c Online Safety	5.Ethical, legal, and environmental issues Environmental concerns, ethical and legal privacy issues including legislation, and the intersection of AI, machine learning, and robotics. Students also explore intellectual property protection in digital environments. Links: 7c Online Safety	1.Algorithms and computational logic 1 Students grasp fundamental concepts including algorithms and abstraction, algorithmic thinking, and the production of algorithms. They learn to identify inputs, processing, and outputs, as well as basic algorithmic techniques like linear and binary search. Links: 6.Basic programming concepts	1.Algorithms and computational logic 2 Building upon previous knowledge, students study visual inspection, trace tables, and more advanced search algorithms such as linear and binary search. Links: 1.Algorithms and computational logic 1	1.Algorithms and computational logic 3 Sorting algorithms like merge sort and bubble sort, alongside considerations for algorithm efficiency and evaluating a program's fitness for purpose. Students also explore truth tables for simple logic gates and diagrams. Links: 1.Algorithms and computational logic 1 + 2
	6.Basic programming concepts Revisit essential programming concepts including data types, structures, operators, variables, constants, inputs, outputs, assignments, and basic programming constructs. They also receive an introduction to subprograms. Links: 8c Text Based Programming	6.Advanced programming concepts String manipulation, randomization, math operations, time modules, the Turtle module for graphics, file handling, and procedures/functions. They also explore local and global variables in programming. Links: 8c Text Based Programming	6.Robust and secure programming Understand the importance of robust software, how to handle syntax, logic, and runtime errors, and principles of maintainability and data validation. They also explore authentication routines for secure programming. Links: 6.Advanced programming concepts	6.Classification of programming languages Learn about the characteristics of programming languages, the relationship between machine code and assembly, and the role of assemblers, compilers, and interpreters in language translation. Links: 6.Advanced programming concepts	6.Application of python skill to real world problems Students apply their Python programming skills to real-world scenarios, building programs that solve practical problems and demonstrate their understanding of Python concepts and techniques. Links: 8c Text Based Programming	
Year 11	1.Algorithms and computational logic 1 delve into algorithmic thinking, learn how to produce algorithms, and understand the process of identifying inputs, processing, and outputs. They also explore techniques like visual inspection, trace tables, and basic search algorithms including linear and binary search. Links: 6.Basic programming concepts	1.Algorithms and computational logic 2 Building upon the foundation laid in the first part, students study more advanced sorting algorithms like merge sort and bubble sort. They also examine algorithm efficiency, evaluating a program's fitness for purpose, and truth tables for simple logic gates and diagrams. Links: 1.Algorithms and computational logic prior	1. Computational thinking Revision/Exam Questions Using command words and learning skills of answering questions based on specific topics/ Covering different types of questions and various ways they could be asked. Exam techniques Links: 1.Algorithms and computational logic 1	1.Data spaced and interleaved practice /Exam Questions Using command words and learning skills of answering questions based on specific topics/ Covering different types of questions and various ways they could be asked. Exam techniques 6: Python spaced and interleaved practice /Exam Questions Learn the skills on how to use decomposition to break down python questions and implement the required code or build the required program to achieve the requirements objectives Links: 6.Advanced programming concepts	3. Computers spaced and interleaved practice /Exam Questions Using command words and learning skills of answering questions based on specific topics/ Covering different types of questions and various ways they could be asked. Exam techniques 6: Python spaced and interleaved practice /Exam Questions Learn the skills on how to use decomposition to break down python questions and implement the required code or build the required program to achieve the requirements objectives Links: 6.Advanced programming concepts	