Computing Department Curriculum Overview

Curriculum Overview

The Computing department at Bentley Wood school aims to equip students with the skills to participate in a rapidly-changing world through challenging and engaging topics. Students will develop an understanding and application in the fundamental principles of Computer Science by having the opportunity to design algorithms, write programs, investigate and experiment with a range of technologies and produce professional digital products.

Students study Computer Science to help them think in a more logical way and become better at making decisions and solve problems. Students learn about how the different parts of a computer work together and why they work like that. In addition, they develop skills in programming systems and start to understand how computers communicate via networks. They then look at how important Technology is in today's society and the impact and issues that can arise from using computer systems and how to improve them.

Computing skills are a major factor in enabling students to be confident, creative and independent learners and it is our intention that students have every opportunity available to allow them to achieve this. The Computing department aims to ensure that all pupils:

• 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.

In Computing we are dedicated to ensuring our students leave with the skills to fully embrace a future of rapidly advancing computer technology.

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 7	**Introduction to the school network** • Logging into the Bentley Wood systems • MS Teams and OneNote • MS Outlook and email • The safe use of computing • Presenting and collaborating	**E-Safety and digital literacy** • Cyberbullying and • The safe use of Social media • Malware and how to prevent it • Passwords and security	**Python turtle** • Introduction to Python turtle and Edublocks • Turtle and iteration • User input and data types • Variables and assignment • Functions and sub routines	**Computer Systems** • What is a computer • What's inside a computer? • How it all works • The CPU	**Fundamentals of computer science** • Sequencing • Events and event handling • Loops • Conditionals • Functions • Variables • For loops	**Coding with Kodu** • Building your world • Movement and scoring • Creating your first game • Additional game features • Game launch!

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 8	**Web design** • How are websites made? • HTML the basics • Images and hyperlinks • Creating webpages using DreamWeaver • Creating webpages using online tools • CSS and styling • Linking pages together • Evaluating websites and creating structure charts	**Cyber crime** • Knowing what to trust online • Email scams • Hacking • E-Safety • Encryption	**Micro:bit** • Introduction and sequencing • From blocks to text coding • Responding to input using selection • Touchy subject • Making a point	**Python programming** • The basics of Python, inputs and outputs • Planning algorithms using flow charts • Variables • Using the time function • Data types • Decision making in Python • Selection with multiple outcomes • Making a chatbot	**Computing theory** • Input and output devices • Parts of a computer • Counting with binary • Binary addition • Storage • Convergence	**Searching and sorting algorithms** • Searching and sorting in the real world • Bubble sort and insertion sort • Linear search and binary search
	Appshed • Intro to AppShed • Icons and app structure	**Physical computing with Arduino** • Electricity basics • Ohm's law	**2.2.1 - Programming fundamentals** • Variable constants and outputs	**2.2.1 - Programming fundamentals** • Selection and conditionals	**2.1.3 - Searching and sorting algorithms** • Linear search • Binary search	**2.5 - Programming languages and IDEs** • High and low level languages

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 9	• Main menu, maps and gallery • Publishing your app • Making your own app	• Working with LEDs - Traffic signals • Using the dimmer switch • Holiday lights • Sports robot • Using servos - Windshield wipers • Musical keyboard • Light wave radar	• Inputs and outputs in Python • Arithmetic and logical operators **2.2.2 - Common data types, arithmetic operators and boolean operators** • Integers • Boolean • Characters and strings • Casting	• Iteration (while loops) • Iteration (for loops) **2.1.2 - Designing, creating and refining algorithms** • Designing algorithms using flowcharts • Designing algorithms using pseudocode • Interpreting algorithms	• Bubble sort • Merge sort • Insertion sort Revision for end of year exams	• Assembly language and the little man computer • Translators • Compilers and interpreters • IDEs **1.1 - systems architecture** • Architecture of the CPU • CPU performance
Year 10	**2.1.1 - Computational thinking** • Abstraction • Decomposition • Algorithmic thinking • Computational thinking **1.2 - Memory and storage** • Primary storage (RAM & ROM) • Secondary storage types and characteristics • Secondary storage – choosing suitable devices • Units and calculating storage	**2.3.1 - Defensive design** • Code maintenance • Validation, authentication and anticipating misuse • Implementing defensive design **1.2 - Memory and storage** • Binary and denary • Hexadecimal • Binary arithmetic • Characters • Images • Sounds	**2.3.2 – Testing** • Identifying syntax and logic errors • Selecting suitable test data **1.3 - Computer networks, connections and protocols** • The internet and the world wide web • Local area networks • Wireless networking • Client server and P2P networks • Standards protocols and layers	**2.2.3 - Additional programming techniques** • String manipulation • File handling • SQL **1.4 – Network security** • Network threats • Preventing vulnerabilities	**2.2.3 - Additional programming techniques** • Arrays • 2 dimensional arrays • Procedures and function • Random number generation **1.5 - Systems software** • Operating systems • Utility systems software	**Programming project** • Analysis and design • Developing longer programs • Testing the solution • Evaluation **Revision for end of Year 10 exams**

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 11	**2.4 - Boolean logic** • Truth tables • Logic gates • Logic diagrams	**1.6 - Ethical legal, cultural and environmental impacts of digital technology** • Ethics and privacy issues • Legal and cultural issues • Environmental issues • Impacts of technology on wider society • Open source vs proprietary software	**Mock follow up & Revision** • Recap of topics 2.1, 2.2, 2.3 • Recap of topics 1.1, 1.2 and 1.3	**Revision** • Recap of topics 2.4 and 2.5 • Recap of topics 1.4, 1.5 and 1.6	**Revision and preparation for final exams**	

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 12	**1.2.3 Introduction to programming** • Procedural programming techniques • File handling • Assembly language • Programming practice **2.2.1 Programming techniques** • Programming basics • Selection • Iteration • Subroutines **1.1.1 Structure and function of the processor** • Processor components • Processor performance **1.1.2 Types of processor** • CISC vs RISC • Von Neumann vs Harvard **1.1.3 Input, output and storage** • Input devices • Output devices • Storage devices	**1.4.1 Data types** • Data types, binary and hexadecimal • ASCII and Unicode • Binary arithmetic • Floating point binary **1.4.2 Data structures** • Arrays, tuples and records • Queues • Stacks • Linked lists **1.2.1 Operating systems** • Functions of an OS • Types of OS **1.2.2 Applications generation** • The nature of applications • Utilities • Open source vs closed source **1.5 Legal, moral, ethical and cultural issues** • Computer related legislation • Ethical, moral and cultural issues	**1.4.3 Boolean algebra** • Logic gates and truth tables • Karnaugh maps **2.1 Computational thinking** • Thinking abstractly • Thinking ahead • Thinking procedurally • Thinking logically **1.3.1 Databases** • Database concepts • Methods of capturing data • 1.4.2 Data structures •	**2.2.2 Software development** • Systems analysis methods • Writing and following algorithms **2.3 Algorithms** • Analysis and design of algorithms • Searching algorithms • Bubble sort and insertion sort • Further algorithms **1.3.2 Networks** • Structure of the internet • Internet communication • Client server and peer to peer **1.3.3 Web technologies** • HTML • CSS • Javascript	**Revision and exam preparation**	**Programming project – choose project, work through tutorials and write analysis** **Year 12 recap + Yr 13 content for unit 1.1 & 1.2** **1.1 Structure and function of the processor** • Pipelining • GPUs 1.2 Systems software • Stages of compilation • Linkers loaders and libraries • Software development methodologies • Modes of addressing memory • Programming paradigms • Object Oriented programming

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
		• Privacy and censorship				
Year 13	**1.4.1 Data types** • Floating point arithmetic • Bitwise manipulation and masks **1.4.2 Data Structures** • Linked lists • Graphs • Stacks and queues • Trees • Hash tables **1.4.3 Boolean Algebra** • Simplifying statements in Boolean Algebra	**2.2.1 Programming techniques** • Recursion • The use of object oriented techniques **2.2.2 Computational methods** • Problem recognition • Problem decomposition • Divide and conquer • Use of abstraction • Visualisation to solve problems • Data mining • Heuristics • Performance modelling • Pipelining	**1.3.2 Databases** • Normalisation to 3NF • SQL interpret and modify • Transaction processing **1.3.3 Networks** • Network security and threats • Network hardware • Client server and peer to peer **1.3.4 Web technologies** • Search engine indexing	**2.3 Algorithms** • Algorithm execution time and space complexity • Big O notation • Merge sort • Quick sort • Dijkstra's shortest path algorithm • A* Algorithm **2.1 Computational thinking** • The nature, benefits and drawbacks of caching • Thinking concurrently	**Revision**	

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
		1.3.1 Compression encryption and hashing • Run length encoding and dictionary coding • Symmetric and asymmetric encryption • Hashing **Revise for mocks**	• PageRank algorithm • Server and client side processing			