Immanuel College
Church of England Academy

Curriculum Map – Computer Science- Year 10

	Half term 1	Half term 2	Half term 3	Half term 4	Half term 5	Half term 6
Key focus	• Decomposition and abstraction • Truth tables • Python programming		• Binary • Data representation • Python programming		• Data storage and compression • Hardware • Software • Python programming	
Key knowledge and skills	• Understand the benefit of using decomposition and abstraction to model aspects of the real world and analyse, understand, and solve problems. • Understand the benefits of using subprograms. • Understand how to follow and write algorithms. • Understand the need for and be able to follow and write algorithms that use variables and constants and one- and two-dimensional data structures. • Understand the need for and be able to follow and write algorithms that use arithmetic operators. • Understand how to determine the correct output of an algorithm for a given set of data and use a trace table to determine what value a variable will hold at a given point in an algorithm. • Understand types of errors that can occur in programs (syntax, logic, runtime) and be able to identify and correct logic errors in algorithms. • Understand how to be able to apply logical operators (AND, OR, NOT) in truth tables with up to three inputs to solve problems. • Understand how standard algorithms (bubble sort, merge sort, linear search, binary search) work. • Develop understanding of python programming constructs, setting data types, and inputting different user inputs		• Understand that computers use binary to represent data (numbers, text, sounds, graphics) and program instructions and be able to determine the maximum number of states that can be represented by a binary pattern of a given length. • Understand how computers represent and manipulate unsigned integers and two's complement signed integers. • Understand how to convert between denary and 8-bit binary numbers (0 to 255 and –128 to +127). • Understand how to add together two positive binary patterns and apply logical and arithmetic binary shifts. • Understand how to add together two positive binary patterns and apply logical and arithmetic binary shifts. • Understand why hexadecimal notation is used and be able to convert between hexadecimal and binary. • Understand how computers encode characters using 7- bit ASCII. • Understand how bitmap images are represented in binary (pixels, resolution, colour depth). • Understand how analogue sound is represented in binary (amplitude, sample rate, bit depth, sample interval). • Understand the limitations of binary representation of data when constrained by the number of available bits. • Understand how to solve problems using python programming language. •		• Understand that data storage is measured in binary multiples (bit, nibble, byte, kibibyte, mebibyte, gibibyte, tebibyte) and be able to construct expressions to calculate file sizes and data capacity requirements. • Understand the need for data compression and methods of compressing data (lossless, lossy). • Understand the von Neumann stored program concept and the role of main memory (RAM), CPU (control unit, arithmetic logic unit, registers), clock, address bus, data bus, control bus in the fetch decode-execute cycle. • Understand the role of secondary storage and the ways in which data is stored on devices. • Understand the concept of an embedded system and what embedded systems are used for. • Understand the purpose and functionality of an operating system. • Understand the purpose and functionality of utility software. • Understand the importance of developing robust software and methods of identifying vulnerabilities. • Creating programs that use functions and procedures, include validation and other forms of error detection	
Key words/ vocabulary	Computation thinking / divide and conquer / 'Big O' notations / algorithms / Logic gates / trace tables / Pseudocode / flow diagrams / Intergrade development environment		Binary / base 2, base 10, base 16 / Two's complement / Binary sign / character sets		High level language / low level language / Von Neumann / CPU registers / scheduling / memory management / buffer /	
Assessment method	Question and answering / practice exam questions / homework / topic assessments / Mock exams					

"Perseverance produces **character,** and character, **hope**" (Romans 5:4)

Curriculum Map – Computer Science- Year 10

Immanuel College
Church of England Academy

Wider links	Mathematics
Enrichment opportunities	Code breakers
Careers links	Programmer / Ethical hacker / Software engineer / Networking consultant / Computer scientist

“**Perseverance** produces **character,** and character, **hope**” (Romans 5:4)