

Introduction to key computing systems, principles and algorithmic thinking					
	Learning Cycle	Key Concepts and Themes	Vocabulary		
HT1	Systems Architecture	<ul style="list-style-type: none"> Architecture of the CPU CPU Performance Memory Secondary Storage 	Fetch-execute, CPU, ALU CU, cache, registers, Von Neumann architecture, MAR, MDR, Program Counter, Accumulator, Clock speed, cache size, cores, embedded systems, memory address, Primary storage, RAM, ROM, virtual memory, volatile, non-volatile, Secondary storage, optical, magnetic, solid state, drive, disk, hard disk, floppy disk, tape drive, Blu-ray, DVD, CD, capacity, speed, portability, durability, reliability, cost, storage device, storage media	<ul style="list-style-type: none"> Systems architecture assessment Programming challenges 	
	PROGRESS CHECK 1				
	HT2	Algorithms	<ul style="list-style-type: none"> Computational thinking Searching algorithms Sorting algorithms Developing algorithms using flowcharts Developing algorithms using pseudocode Interpret, correct and complete algorithms 	Computational thinking, abstraction, decomposition, algorithmic thinking, inputs, processes, outputs, structure diagrams, pseudocode, flowcharts, reference language, Trace tables, syntax error, logical error, algorithm, decision, terminal, sub program, process, Binary search, linear search, bubble sort, merge sort, insertion sort, Variables, constants, operators, assignments, sequence, selection, iteration, Boolean operators, arithmetic operators, modulus, quotient, exponentiation	<ul style="list-style-type: none"> Paper 1 cumulative assessment (Systems architecture + Data Representation) <ul style="list-style-type: none"> Algorithms assessment
	HT3	Data Representation	<ul style="list-style-type: none"> Units and binary numbers Binary arithmetic and hexadecimal Characters Images Sound Compression 	Bit, nibble, byte, kilo, mega, giga, tera, peta, binary, Bit depth, sample rate, colour depth, pixel, bit per character, Binary shift, shift left, shift right, most significant bit, least significant bit, Character set, ASCII, Unicode, metadata, hertz, compression, lossy, lossless	
	PROGRESS CHECK 2				
	HT4	Networks	<ul style="list-style-type: none"> The internet and wide area networks Local area networks Network Topologies 	LAN, WAN, bandwidth, latency, Wireless access points, routers, switches, NIC, Transmission media, DNS, Hosting, The Cloud, Web servers and clients, star network, mesh network, topology, IP address, web server, file server	<ul style="list-style-type: none"> Paper 1 cumulative assessment Paper 2 cumulative assessment Year 10 Mock exams
HT5	Networks	<ul style="list-style-type: none"> Wireless networking Client server and P2P networks Standards, protocols and layers 	Wired network, wireless network, Ethernet, Wi-Fi, Bluetooth, encryption, IP addressing, MAC addressing, TCP/IP, FTP, POP, I, IMAP, SMTP, layers, IPv4, IPv6.		
PROGRESS CHECK 3					
HT6	Programming (Ongoing throughout the year)	<ul style="list-style-type: none"> Programming Fundamentals Sequence and selection Iteration Arrays Procedures and functions Records and files 	Variables, constants, operators, inputs, outputs, assignment, sequence, selection, iteration, Arithmetic operators, Boolean operators, AND, OR, NOT, ==, !=, <=, >, >=, +, -, *, /, MOD, DIV, ^, exponentiation, Data types, integer, real, Boolean, character, string, casting, string manipulation, File handling, open, read, write, close, records, Arrays, one-dimensional array, two-dimensional array, Sub program/subroutine, functions, procedures, random numbers, concatenation, slicing		

Skill Development	<ul style="list-style-type: none"> Problem Solving - Take a systematic approach to problem solving including the use of decomposition and abstraction, and make use of conventions including pseudo code and flowcharts Programming - Design, write, test and refine programs, using one or more high-level programming language with a textual program definition, either to a specification or to solve a problem Algorithmic Thinking - Use abstraction effectively to model selected aspects of the external world in a program and to appropriately structure programs into modular parts with clear, well documented interfaces Apply computing-related mathematics such as hexadecimal conversions, binary conversions, binary addition and shifts.
--------------------------	--