

Year 12 Curriculum Map

COMPUTER SCIENCE · TOPIC OVERVIEW 2026/2027

Term	Topics (EAL)	Topics (JHA)
Autumn Term SEPT – DEC	• Introduction to the unit / procedural journey • Programming techniques: sequence, iteration and branching • Primitive data types: integer, real/float, character, string, Boolean • Variables, constants and assignment • Global and local variables • Modularity: functions and procedures; parameter passing by value and by reference • File handling (reading from and writing to files) • Strings and input validation • Using an IDE to develop and debug programs • Data structures: arrays, lists and tuples • Structure and function of the processor: ALU, control unit, registers (PC, ACC, MAR, MDR, CIR) • Buses: data, address and control • Fetch–decode–execute cycle and its effect on registers • CPU performance: clock speed, number of cores, cache • Pipelining to improve processor efficiency • Von Neumann, Harvard and contemporary architectures • CISC and RISC processors • GPUs and their uses • Multicore and parallel systems • Input, output and storage devices • Magnetic, flash and optical storage • RAM, ROM and virtual storage	• Number systems: binary representation of positive integers • Hexadecimal representation of positive integers • Converting between binary, hexadecimal and denary • Sign and magnitude; two's complement for negative integers • Addition and subtraction of binary integers • Floating point representation and normalisation in binary • Positive and negative real numbers using normalised floating point • Floating point arithmetic (addition and subtraction) • Character sets: ASCII and Unicode • Systems software: purpose and function of operating systems • Memory management: paging, segmentation, virtual memory • Interrupts, ISRs and their role in the FDE cycle • Process scheduling: round robin, FCFS, multi-level feedback queues, shortest job first, shortest remaining time • Types of OS: distributed, embedded, multi-tasking, multi-user, real-time • BIOS • Application generation; open source vs closed source software • Device drivers and virtual machines • Utility software
Spring Term JAN – EASTER	• Searching algorithms: linear and binary search • Sorting algorithms: bubble sort and insertion sort • Sorting algorithms: merge sort • Compression, encryption and hashing • Lossy vs lossless compression • Run length encoding and dictionary coding • Symmetric and asymmetric encryption • Hashing and its uses • Big O notation: constant, linear, polynomial, exponential and logarithmic complexity • Types of programming language: assembly and high-level	• Databases: relational database vs flat file • Primary, foreign and secondary keys • Entity relationship (ER) modelling and indexing • Normalisation to 3NF • SQL: interpret and modify queries • Referential integrity • Transaction processing and ACID properties • Record locking and redundancy • Methods of capturing, selecting, managing and exchanging data
Summer Term APR – JUL	• Types of programming language (continued): translators, compilers and interpreters • Networks: network hardware • LANs and WANs • Client–server and peer-to-peer networks • Packet and circuit switching • Network protocols and standards • TCP/IP stack; DNS; protocol layering • Network security: threats, firewalls, proxies and encryption • PPE revision and assessment	• Computing-related legislation: Data Protection Act 1998, Computer Misuse Act 1990, Copyright, Design and Patents Act 1988, Regulation of Investigatory Powers Act 2000 • Moral and ethical issues: computers in the workforce, automated decision making, artificial intelligence, environmental effects, censorship, surveillance, piracy • Revision across all topics • Year 12 PPE examinations

Year 13 Curriculum Map

COMPUTER SCIENCE · TOPIC OVERVIEW 2026/2027

Term	Lesson content	Homework	NEA
Autumn Term SEPT – DEC	• Web development (Topic 5): HTML structure and CSS styling • JavaScript; forms and input • Client-side and server-side validation • The internet: dynamic vs static websites • Client–server processing; search engines and PageRank • Data structures: arrays, lists, tuples, records • Stacks and queues • Linked lists • Trees • Graphs and shortest path algorithms (e.g. Dijkstra’s) • Recap Object-oriented programming: classes, encapsulation, inheritance, polymorphism • Boolean algebra: logic gate diagrams and truth tables • Karnaugh maps and simplification of Boolean expressions • De Morgan’s laws; distribution, association, commutation, double negation • D-type flip flops; half and full adders • Bitwise manipulation and masks (shifts, AND, OR, XOR)	• Ethics – computing legislation: Data Protection Act 1998, Computer Misuse Act 1990, Copyright, Design and Patents Act 1988, RIPA 2000 • Ethics – computers in the workforce • Ethics – automated decision making • Ethics – artificial intelligence • Ethics – environmental effects of computing • Ethics – censorship and the internet • Ethics – monitoring behaviour • Ethics – analysing personal information • Ethics – piracy and offensive communications	• NEA: analysis (project overview, existing systems, stakeholders) • NEA: design (decomposition diagram, interface designs, OOP class diagram, testing strategy)
Spring Term JAN – EASTER	• Databases: relational databases; flat files; primary, foreign and secondary keys • Entity relationship modelling; normalisation to 3NF; indexing • SQL: interpret and modify queries • Referential integrity; transaction processing; ACID; record locking and redundancy • Methods of capturing, selecting, managing and exchanging data • Networks: network hardware; LANs and WANs • Client–server and peer-to-peer; packet and circuit switching • Network characteristics, protocols and standards; TCP/IP stack; DNS; protocol layering • Network security: threats, firewalls, proxies and encryption • Computational methods: problem recognition and decomposition • Divide and conquer; abstraction; backtracking; data mining; heuristics; performance modelling; visualisation • Software development: waterfall lifecycle; agile; extreme programming; spiral model; RAD • Testing strategies: black box, white box, alpha and beta testing		• NEA: iterations
Summer Term APR – JUL	• Final revision across all A-Level topics • Exam technique and past paper practice • A-Level Computer Science examinations (Component 1 and Component 2)		• NEA: post-development testing and evaluation • NEA: project submission deadline (/May)