A-level OCR Computer Science (2026-2028) Curriculum Plan

#	Topic	Content	Programming tasks
YEAR 12 — AUTUMN TERM (SEPT – DEC)			
1	**Programming techniques**	Sequence, iteration and branching; using an IDE to develop and debug programs	**Journey:** Procedural Journey (Student Grade Tracker) — Keyword Matchup; Steps 1–3: selection, iteration, subprograms **Homework:** FizzBuzz (modulo, if/elif/else); Fruit Machine (random symbols, win/loss rules, running balance)
2	**Data types, variables, constants & modularity**	Primitive data types; variables, constants and assignment; global and local variables; functions and procedures; parameters by value and by reference; arrays, lists and tuples	**Journey:** Procedural Journey — Steps 4–5: 1D lists, 2D lists **Homework:** Treasure Hunt (5×5 grid); Noughts & Crosses; Four in a Row (6×7 grid); Battle Boats (6×6 grids); Satellite Data (2D list column extraction); Text Adventure (nested dictionaries)
3	**File handling, strings & input validation**	Reading from and writing to files; string handling; validating input	**Journey:** Procedural Journey — Steps 6–8: strings, input validation, files; quizzes and final assessment **Homework:** Hangman (string masking); Music Quiz (CSV file, first-letter clues); Fundraiser (validated activity list) Nought and Crosses, Text Adventure, Satellite.
4	**Computer architecture**	ALU, control unit and registers; buses; fetch–decode–execute cycle; CPU performance; pipelining; Von Neumann and Harvard; CISC and RISC; GPUs; multicore; storage devices; RAM, ROM and virtual storage	—
5	**Little Man Computer (LMC)**	Assembly language programming with LMC	
6	**Number systems**	Binary and hexadecimal; sign and magnitude, two's complement; binary arithmetic; normalised floating point; floating point addition and subtraction; bitwise manipulation and masks (shifts, AND, OR, XOR); ASCII and Unicode	—
7	**Systems software**	OS purpose and types; memory management (paging, segmentation and virtual memory); interrupts and ISRs; scheduling (round robin, FCFS, multi-level feedback queues, SJF, SRT); BIOS; device drivers and virtual machines; the nature of applications and justifying suitable applications for a purpose; utility, open and closed source software	—
YEAR 12 — SPRING TERM (JAN – EASTER)			
8	**Searching & sorting algorithms**	Linear and binary search; bubble, insertion, merge and quick sort; Big O notation (constant, linear, polynomial, exponential and logarithmic) and comparing algorithms for time and space	**Assignments:** Linear Search (O(n)); Binary Search (O(log n)); Bubble Sort (swap flag, $O(n^2)$); Insertion Sort (compare with bubble sort)
9	**Compression, encryption & hashing**	Lossy vs lossless; run length encoding and dictionary coding; symmetric and asymmetric encryption; hashing and its uses	**Journey:** Codebreaking Journey — Caesar cipher, brute force, frequency analysis, Vigenère cipher **Homework:** Password Checker (A-Level) — complexity scoring, common-password check, hashlib storage
10	**Programming languages & translators**	Programming paradigms and procedural languages; assembly vs high-level languages; modes of addressing memory (immediate, direct, indirect and indexed); compilers, interpreters and assemblers; stages of compilation (lexical analysis, syntax analysis, code generation and optimisation); linkers, loaders and libraries	—
11	**Databases**	Relational vs flat file; primary, foreign and secondary keys; ER modelling and indexing; normalisation to 3NF	—
12	**SQL, integrity & transactions**	Interpret and modify SQL queries; referential integrity; transaction processing and ACID; record locking and redundancy; capturing and exchanging data	**Journey:** SQL Journey — SELECT/FROM, WHERE, ORDER By, JOIN, GROUP BY
YEAR 12 — SUMMER TERM (APR – JUL)			
13	**Computational thinking**	Thinking abstractly, ahead, procedurally, logically and concurrently	**Assignments:** Pot Luck Board Game (decomposition, abstraction); Hare & Tortoise Race (simulation); API Practice (Weather) (JSON parsing); Fitness App (API integration, decomposition)
14	**Networks**	Network hardware; LANs and WANs; client–server and peer-to-peer; packet and circuit switching	—
15	**Protocols & network security**	TCP/IP stack and protocol layering; DNS; standards; threats, firewalls, proxies and encryption	—

#	Topic	Content	Programming tasks
16	**Boolean algebra**	Logic gates and truth tables; Karnaugh maps; De Morgan's laws, distribution, association, commutation and double negation; D-type flip-flops; half and full adders	—
17	**Object-oriented programming**	Classes and objects; encapsulation; inheritance; polymorphism	**Journeys:** OOP Cars Journey; VP OOP Journey (Virtual Pet) **Assignments:** Pet Class (getters/setters); Dog Inheritance (super()); Pet & Human Age (lookup table); House Road (inheritance vs aggregation) **Homework:** Bank Account; Bicycle; Farm Animals (polymorphism, aggregation); Monster; Pokémon Battle
YEAR 13 — AUTUMN TERM (SEPT – DEC)			
18	**Web development**	HTML structure and CSS styling; JavaScript, forms and input; client-side and server-side processing and validation; dynamic vs static websites; search engine indexing and PageRank	**Journey:** Web Journey (HTML / CSS / JS) — page skeleton, CSS box model, flexbox and grid, JavaScript events, DOM and form validation
19	**Advanced data structures**	Records; arrays of up to 3 dimensions; stacks and queues; linked lists; trees and binary search trees, including depth-first (post-order) and breadth-first traversal; graphs (directed and undirected); hash tables; shortest path algorithms (Dijkstra's and A*)	**Assignments:** Stack (LIFO); Linear Queue (FIFO); Circular Queue (modulo wrap-around); Reverse Word (stack application), Linked Lists and Tree Traversals
20	**Computing legislation**	Data Protection Act 1998; Computer Misuse Act 1990; Copyright, Design and Patents Act 1988; RIPA 2000	—
21	**Moral & ethical issues**	Artificial intelligence and automated decision making; censorship; monitoring and surveillance; environmental effects; computers in the workforce; personal information; piracy and offensive communications; layout, colour paradigms and character sets	—
22	**NEA: analysis & design**	Project overview, existing systems and stakeholders; decomposition diagram, interface designs, OOP class diagram and testing strategy	NEA project work
YEAR 13 — SPRING TERM (JAN – EASTER)			
23	**Computational methods**	Problem recognition and decomposition; divide and conquer; recursion compared with iteration; trace tables; abstraction; backtracking; data mining; heuristics; performance modelling; visualisation	**Assignments:** Factorial (recursion, call stack); 4 Queens; Magic Square; Maze Solver; Rat in the Maze (recursive backtracking) **Journey:** Data & Visualisation Journey — pandas CSV loading, line/bar/scatter charts, subplot dashboard (data visualisation)
24	**Software development**	Waterfall lifecycle; agile and extreme programming; spiral model; RAD; relative merits and drawbacks of each methodology; writing and following algorithms	—
25	**Testing strategies**	Black box, white box, alpha and beta testing	—
26	**NEA: iterations & evaluation**	Development iterations; post-development testing and evaluation	NEA project work
YEAR 13 — SUMMER TERM (APR – JUL)			
27	**NEA: project submission**	Final submission, May deadline	NEA project work
28	**Final revision**	Revision across all A-Level topics	—
29	**Exam technique & past papers**	Past paper practice; A-Level examinations (Component 1 and Component 2)	—