



KEY STAGE 5 – YEAR 12 – A-Level Computer Science CURRICULUM MAP

Autumn Term		Spring Term		Summer Term	
Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Key Themes	Key Themes	Key Themes	Key Themes	Key Themes	Key Themes
<p>1.1.1 Structure and function of the processor</p> <p>1.1.2 Types of processor</p> <p>1.1.3 Input, output and storage</p> <p>1.2.1 Systems Software</p> <p>1.2.2 Applications Generation</p> <p>1.2.3 Software Development</p> <p>NEA - research existing projects</p>	<p>1.2.4 Types of Programming Language</p> <p>NEA - Research project ideas</p>	<p>1.3.1 Compression, Encryption and Hashing</p> <p>1.3.2 Databases</p> <p>1.3.3 Networks</p> <p>1.3.4 Web Technologies</p> <p>NEA - 3.1. Analysis of the problem (10 marks)</p>	<p>1.4.1 Data Types</p> <p>NEA - 3.2 Design of the solution (15 marks)</p> <p>NEA - 3.3 Developing the solution (25 marks)</p>	<p>1.4.2 Data Structures</p> <p>1.4.3 Boolean Algebra</p> <p>NEA - 3.3 Developing the solution (25 marks)</p>	<p>1.5.1 Computing related legislation</p> <p>1.5.2 Moral and ethical Issues</p> <p>2.1 – Algorithms</p> <p>NEA - 3.3 Developing the solution (25 marks)</p>
Assessment / Composite Tasks	Assessment / Composite Tasks	Assessment / Composite Tasks	Assessment / Composite Tasks	Assessment / Composite Tasks	Assessment / Composite Tasks
1.1 End of Unit Test	1.2 End of Unit Test	1.3 End of Unit Test	1.4.1 End of Unit Test	1.4.2 & 1.4.3 End of Unit Test	1.5 End of Unit Test



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

SCHEME OF WORK

Autumn Half Term One: Key Theme – 1.1 The characteristics of contemporary processors, input, output and storage devices					
	Intent (weekly outline)	Implementation (T and L Pedagogy/components used)	Impact	Powerful Knowledge (keywords and terminology)	Personal Development Links
Week 1	1.1.1 Structure and function of the processor	Direct Instruction Interactive Learning Practical Activities Collaborative Learning Formative Assessment Flipped Classroom Key Terms and Definitions Diagrams and Visual Aids Simulations and Software Tools Hands-On Activities Assessment Tools Project-Based Learning	The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: how this relates to assembly language programs. The Fetch-Decode-Execute Cycle; including its effects on registers. The factors affecting the performance of the CPU: clock speed, number of cores, cache. The use of pipelining in a processor to improve efficiency. (e) Von Neumann, Harvard and contemporary processor architecture.	Characteristics of Contemporary Processors CPU (Central Processing Unit) Clock Speed Cores (Single-core, Multi-core) Cache (L1, L2, L3) Pipelining Parallel Processing Instruction Set RISC (Reduced Instruction Set Computer) CISC (Complex Instruction Set Computer) ALU (Arithmetic Logic Unit) Control Unit Registers Bus (Data Bus, Address Bus, Control Bus) Fetch-Decode-Execute Cycle Von Neumann Architecture Harvard Architecture Transistors Input Devices	Improving focus and memory retention techniques. Learning time management and productivity techniques. Developing effective communication skills. Applying the cycle of learning new skills, understanding them, and putting them into practice.
Week 2	1.1.2 Types of processor		(a) The differences between and uses of CISC and RISC processors. (b) GPUs and their uses (including those not related to		



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

			graphics). (c) Multicore and Parallel systems.	Keyboard, Mouse, Scanner, Microphone, Camera, Touchscreen, Barcode Reader,	
Week 3	1.1.3 Input, output and storage		(a) How different input, output and storage devices can be applied to the solution of different problems. (b) The uses of magnetic, flash and optical storage devices. (c) RAM and ROM. (d) Virtual storage.	Biometric, Sensors, Game Controller, Graphic Tablet, Output Devices Monitor, Printer, Speakers, Projector, Headphones, VR Headset, Plotter, LED Display Storage Devices	
Week 4	1.1 The characteristics of contemporary processors, input, output and storage devices recap and End of Unit test		1.1 The characteristics of contemporary processors, input, output and storage devices recap	Primary Storage (Main Memory) RAM (Random Access Memory) ROM (Read-Only Memory) Cache Memory Secondary Storage Hard Disk Drive (HDD) Solid State Drive (SSD) Optical Discs (CD, DVD, Blu-ray) USB Flash Drives SD Cards External Hard Drives Network Attached Storage (NAS) Tertiary Storage Cloud Storage Tape Drives Virtual Memory Storage Capacity Data Access Speed Non-volatile Memory Volatile Memory General Concepts	



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

				I/O (Input/Output) Operations Device Drivers Peripheral Devices Bandwidth Latency
Week 5	1.2.1 Systems Software		(a) The need for, function and purpose of operating systems. (b) Memory Management (paging, segmentation and virtual memory). (c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle. (d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time. (e) Distributed, embedded, multi-tasking, multi-user and Real Time operating systems. (f) BIOS. (g) Device drivers. (h) Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate code or running an operating system within another.	Types of Software System Software Operating Systems (OS) Utility Programs Device Drivers Application Software Productivity Software (e.g., Word Processors, Spreadsheets) Database Management Systems (DBMS) Graphics Software Communication Software (e.g., Email Clients) Web Browsers Games Development Software Integrated Development Environments (IDEs) Compilers Interpreters Debuggers
Week 6	1.2.2 Applications Generation		(a) The nature of applications, justifying suitable applications for a specific purpose. (b) Utilities.	



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

			<p>(c) Open source vs closed source.</p> <p>(d) Translators: Interpreters, compilers and assemblers.</p> <p>(e) Stages of compilation (lexical analysis, syntax analysis, code generation and optimisation).</p> <p>(f) Linkers and loaders and use of libraries.</p>		
Week 7	1.2.3 Software Development		<p>(a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development.</p> <p>(b) The relative merits and drawbacks of different methodologies and when they might be used.</p> <p>(c) Writing and following algorithms.</p>	<p>Software Development Methodologies</p> <p>Waterfall Model</p> <p>Agile Development</p> <p>Scrum</p> <p>Extreme Programming (XP)</p> <p>Spiral Model</p> <p>Incremental Development</p> <p>Prototyping</p> <p>Software Development Process</p> <p>Requirements Analysis</p> <p>System Design</p> <p>Implementation (Coding)</p> <p>Testing</p> <p>Unit Testing</p> <p>Integration Testing</p> <p>System Testing</p> <p>Acceptance Testing</p> <p>Deployment</p> <p>Maintenance</p>	



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

	Intent (weekly outline)	Implementation (T and L Pedagogy/components used)	Impact	Powerful Knowledge (keywords and terminology)	Personal Development Links
Week 1	1.2.4 Types of Programming Language	Direct Instruction Interactive Learning Practical Activities Collaborative Learning Formative Assessment Flipped Classroom Key Terms and Definitions Diagrams and Visual Aids Simulations and Software Tools Hands-On Activities Assessment Tools Project-Based Learning	(a) Need for and characteristics of a variety of programming paradigms. (b) Procedural languages.	Low-Level Languages Machine Code Assembly Language High-Level Languages Procedural Languages Object-Oriented Languages Functional Languages Translators Compiler Interpreter Assembler Programming Paradigms Imperative Programming Syntax Memory Management Concurrency	Improving focus and memory retention techniques. Learning time management and productivity techniques. Developing effective communication skills. Applying the cycle of learning new skills, understanding them, and putting them into practice.
Week 2	1.2.4 Types of Programming Language Assembly language		(c) Assembly language (including following and writing simple programs with the Little Man Computer instruction set). See appendix 5d.		
Week 3	1.2.4 Types of Programming Language Modes of addressing		(d) Modes of addressing memory (immediate, direct, indirect and indexed).		



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

Week 4	1.2.4 Types of Programming Language Classes, objects, methods, attributes		(e) Object-oriented languages (see appendix 5d for pseudocode style) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism.		
Week 5	1.2.4 Types of Programming Language				
Week 6	1.2.4 Types of Programming Language Inheritance				
Week 7	Introduction to Unit 3 - NEA -		Look at existing project Finalise your project idea.		

Spring Half Term One: Key Theme – 1.3 Exchanging data & NEA - 3.1. Analysis of the problem (10 marks)					
	Intent (weekly outline)	Implementation (T and L Pedagogy/components used)	Impact	Powerful Knowledge (keywords and terminology)	Personal Development Links
Week 1	1.3.1 Compression, Encryption and Hashing	Direct Instruction Interactive Learning Practical Activities Collaborative Learning Formative Assessment	a) Lossy vs Lossless compression. (b) Run length encoding and dictionary coding for lossless compression.	Lossy Compression Lossless Compression Compression Algorithms Run-Length Encoding (RLE)	Improving focus and memory retention techniques.



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

		<p>Flipped Classroom Key Terms and Definitions Diagrams and Visual Aids Simulations and Software Tools Hands-On Activities Assessment Tools Project-Based Learning</p>	<p>(c) Symmetric and asymmetric encryption. (d) Different uses of hashing.</p>	<p>JPEG (Joint Photographic Experts Group) PNG (Portable Network Graphics) MP3 (MPEG-1 Audio Layer III) ZIP Files Encryption Symmetric Encryption Asymmetric Encryption Encryption Algorithms</p>	<p>Learning time management and productivity techniques.</p> <p>Developing effective communication skills.</p> <p>Applying the cycle of learning new skills, understanding them, and putting them into practice.</p>
Week 2	1.3.2 Databases		<p>(a) Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing. See appendix 5f. (b) Methods of capturing, selecting, managing and exchanging data. (c) Normalisation to 3NF. (d) SQL – Interpret and modify. See appendix 5d. (e) Referential integrity. (f) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking</p>	<p>Normalization First Normal Form (1NF) Second Normal Form (2NF) Third Normal Form (3NF) SQL (Structured Query Language) SQL Commands SELECT INSERT UPDATE DELETE CREATE ALTER DROP Join Inner Join Left Join</p>	



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

			and redundancy.	Right Join Full Join	
Week 3	1.3.3 Networks		<p>(a) Characteristics of networks and the importance of protocols and standards.</p> <p>(b) The internet structure:</p> <ul style="list-style-type: none"> • The TCP/IP Stack. • DNS • Protocol layering. • LANs and WANs. • Packet and circuit switching. <p>(c) Network security and threats, use of firewalls, proxies and encryption.</p> <p>(d) Network hardware.</p> <p>(e) Client-server and peer to peer</p>	Network LAN (Local Area Network) WAN (Wide Area Network) MAN (Metropolitan Area Network) VPN (Virtual Private Network) Internet Intranet Topology Bus Topology Star Topology Ring Topology Mesh Topology Network Hardware Router Switch Hub Bridge Modem Access Point (AP) Network Interface Card (NIC) Firewall	
Week 4	1.3.4 Web Technologies		<p>(a) HTML, CSS and JavaScript. See appendix 5d.</p> <p>(b) Search engine indexing.</p> <p>(c) PageRank algorithm.</p>	Web Page Website Web Server Client-Server Model Web Browser HTTP (Hypertext Transfer Protocol) HTTPS (Hypertext Transfer Protocol Secure)	
Week 5	1.3.4 Web Technologies		<p>(d) Server and client side processing.</p>		



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

				URL (Uniform Resource Locator) Domain Name Hosting HTML (Hypertext Markup Language) CSS (Cascading Style Sheets) JavaScript Libraries	
Week 6	3.1. Analysis of the problem (10 marks)		Problem identification Stakeholders Research the problem Specify the proposed solution	Problem identification Stakeholders Research the problem Specify the proposed solution	

Spring Half Term Two: Key Theme – 1.4 Data types, data structures and algorithm & NEA - 3.2 Design of the solution (15 marks)					
	Intent (weekly outline)	Implementation (T and L Pedagogy/components used)	Impact	Powerful Knowledge (keywords and terminology)	Personal Development Links
Week 1 -2	1.4.1 Data Types	Direct Instruction Interactive Learning Practical Activities Collaborative Learning Formative Assessment Flipped Classroom Key Terms and Definitions Diagrams and Visual Aids Simulations and Software Tools Hands-On Activities	(a) Primitive data types, integer, real/floating point, character, string and Boolean. (b) Represent positive integers in binary. (c) Use of sign and magnitude and two's complement to represent negative numbers in binary. (d) Addition and subtraction of binary integers.	Algorithm Boolean Binary Hexadecimal Two's Complement Recursion Abstraction Stack Queue Operating System	Improving focus and memory retention techniques. Learning time management and productivity techniques.



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

		Assessment Tools Project-Based Learning Interactive Websites Quizzes	(e) Represent positive integers in hexadecimal. (f) Convert positive integers between binary hexadecimal and denary.		Developing effective communication skills. Applying the cycle of learning new skills, understanding them, and putting them into practice.
Week 3-4	1.4.1 Data Types		(g) Representation and normalisation of floating point numbers in binary. (h) Floating point arithmetic, positive and negative numbers, addition and subtraction. (i) Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR. (j) How character sets (ASCII and UNICODE) are used to represent text.	Floating Point Normalisation Exponent Mantissa Floating Point Arithmetic Sign Bit Precision Underflow Overflow Bitwise Manipulation Bit Masking Logical Shift Arithmetic Shift AND OR XOR ASCII Unicode Character Encoding Code Point	Personal development in computer science involves mastering complex concepts like floating point arithmetic, bitwise manipulation, and character encoding, which enhance problem-solving skills. By continuously engaging with these topics, students develop critical thinking,



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

					resilience, and a deeper understanding of how computer systems work at a fundamental level.
Week 5-6	NEA - 3.2 Design of the solution (15 marks)				

Summer Half Term One: Key Theme – 1.5 Legal, moral, cultural and ethical issues & NEA - 3.3 Developing the solution (25 marks)					
Summer Half Term One: Key Theme – 1.5 – Systems software	Summer Half Term One: Key Theme – 1.5 – Systems software	Implementation (T and L Pedagogy/components used)	Impact	Powerful Knowledge (keywords and terminology)	Personal Development Links
Week 1 -2	1.4.2 Data Structure		(a) Arrays (of up to 3 dimensions), records, lists, tuples. (b) The following structures to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table.	Array Record List Tuple Linked List Graph Stack Queue Binary Search Tree Hash Table	Personal development in computer science involves understanding and applying data structures such as arrays, linked lists, and hash tables, which enhance logical thinking and problem-solving abilities



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

					essential for tackling complex programming challenges.
Week 3	1.4.2 Data Structure		(c) How to create, traverse, add data to and remove data from the data structures mentioned above. (NB this can be either using arrays and procedural programming or an object-oriented approach).	Data Structures Traversal Array Linked List Stack Queue Binary Search Tree Hash Table Insertion Deletion Boolean Logic Karnaugh Maps Boolean Algebra De Morgan's Laws Distribution Association Commutation Double Negation Logic Gate Truth Table	Personal development in computer science involves gaining proficiency in manipulating data structures, using Boolean logic, and simplifying Boolean expressions, which enhances both technical and analytical skills required to solve complex computational problems.
Week 4-5	1.4.3 Boolean Algebra		(a) Define problems using Boolean logic. See appendix 5d. (b) Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions. (c) Use the following rules to derive or simplify	Boolean Logic Boolean Expression Karnaugh Map Simplification De Morgan's Laws Distribution Association Commutation Double Negation	Personal development in computer science involves mastering Boolean logic and algebra, which sharpens



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

			<p>statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.</p> <p>(d) Using logic gate diagrams and truth tables. See appendix 5d.</p> <p>(e) The logic associated with D type flip flops, half and full adders.</p>	<p>Logic Gate AND Gate OR Gate NOT Gate NAND Gate NOR Gate XOR Gate Truth Table Flip-Flop D-Type Flip-Flop Adder (Half and Full)</p>	<p>problem- solving skills and enhances the ability to simplify complex expressions and design efficient digital systems using logic gates and flip-flops.</p>
Week 6	NEA - 3.3 Developing the solution (25 marks)		✓		

Summer Half Term Two: Key Theme – 2.1 – Algorithms & 3.3 Developing the solution (25 marks)					
Summer Half Term One: Key Theme – 1.5 – Systems software	Summer Half Term One: Key Theme – 1.5 – Systems software	Implementation (T and L Pedagogy/components used)		Powerful Knowledge (keywords and terminology)	Personal Development Links
Week 1	1.5.1 Computing related legislation	<p>Legal, moral, cultural, and ethical issues in technology involve ensuring compliance with laws, protecting privacy, and considering the impact of technology on different communities. These issues also address the responsible use of technology, including fairness, transparency, and accountability in</p>	<p>(a) The Data Protection Act 1998. (b) The Computer Misuse Act 1990. (c) The Copyright Design and Patents Act 1988. ✓ (d) The Regulation of Investigatory Powers Act 2000.</p>	<p>Data Protection Act Computer Misuse Act Copyright Patents Intellectual Property Cybersecurity Privacy Digital Rights Investigatory Powers Legal Compliance</p>	<p>Personal development in computer science involves understanding key legal frameworks such as the Data Protection Act, Computer Misuse Act, Copyright</p>



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

		areas like AI and data management.			Design and Patents Act, and the Regulation of Investigatory Powers Act, which are essential for ensuring ethical and legal compliance when handling data and technology.
Week 2	1.5.2 Moral and ethical Issues		<p>The individual moral, social, ethical and cultural opportunities and risks of digital technology:</p> <ul style="list-style-type: none"> • Computers in the workforce. • Automated decision making. • Artificial intelligence. • Environmental effects. • Censorship and the Internet. • Monitor behaviour. • Analyse personal information. • Piracy and offensive communications. • Layout, colour paradigms and character sets. 	<p>Workforce Automation Artificial Intelligence Ethical Implications Censorship Data Privacy Behaviour Monitoring Personal Data Analysis Cybersecurity Digital Piracy Online Communication Ethics</p>	
Week 3	<p>2.1.1 Computational thinking</p> <p>NEA - 3.3 Developing the solution (25 marks)</p>	<p>Teaching Strategies Case Studies Debates Role-Playing Guest Speakers</p>	<p>✓ Understanding of these principles and how they are used to define and refine problems</p>	<p>Abstraction Decomposition Algorithmic Thinking Pattern Recognition Generalization</p>	<p>Improving focus and memory retention techniques.</p>



KEY STAGE 5 – YEAR 12 – A-Level Computer Science

Week 3	2.1.1 Computational thinking NEA - 3.3 Developing the solution (25 marks)	Flipped Classroom Project-Based Learning (PBL) Group Discussions Think-Pair-Share Interactive Lectures Problem-Based Learning		Flowcharts Pseudocode Variables Loops	Learning time management and productivity techniques.
Week 3	2.1.2 Designing, creating and refining algorithms NEA - 3.3 Developing the solution (25 marks)	Learning Activities Research Projects Critical Analysis Essays Collaborative Group Work Peer Review Multimedia Presentations Mind Mapping Digital Tools and Resources Educational Videos Interactive Websites Quizzes Assessment Methods Formative Assessments Summative Assessments Peer Assessments Self-Assessments Rubrics and Checklists Multiple-Choice Questions Short Answer Questions	✓ Produce simple diagrams to show: <ul style="list-style-type: none"> • The structure of a problem • Subsections and their links to other subsections 	Algorithms Flowcharts Pseudocode Step-by-step Instructions Control Structures Sequencing Selection Iteration Loops Conditionals Decision Making Efficiency Optimization Refinement Debugging Testing Trace Tables Inputs Outputs Problem Solving	Developing effective communication skills. Applying the cycle of learning new skills, understanding them, and putting them into practice.
Week 4	2.1.2 Designing, creating and refining algorithms		✓ Complete, write or refine an algorithm using the techniques listed ✓ Identify syntax/logic errors in code and suggest fixes ✓ Create and use trace tables to follow an algorithm ✓ Use of nesting for selection and iteration		
Week 5	2.1.3 Searching and sorting algorithms NEA - 3.3 Developing the solution (25 marks)			✓ Understand the main steps of the algorithm and the segments of code in it ✓ Understand any pre-requisites of an algorithm ✓ Apply the algorithm to a data set ✓ Identify an algorithm if given the code, pseudocode or Exam ✓ Reference Language for it	Linear Search Binary Search Bubble Sort Insertion Sort Merge Sort Quick Sort Efficiency Comparisons Swaps Divide and Conquer Recursion Iterative Sorted List Unsorted List
Week 6	2.1.3 Searching and sorting algorithms NEA - 3.3 Developing the solution (25 marks)				



KEY STAGE 5 – YEAR 12 – A-Level Computer Science