Curriculum Map: Year 12 Computer Science

Topic	Key Knowledge *What will all students KNOW by the end of the topic?*	Key Skills *What key skills will be learnt/developed by the end of the topic? What will all students be able to DO by the end of the topic?*	Assessment Opportunities *What are the key pieces of assessment? How will students be assessed?*
Characteristics of contemporary processors, input, output and storage devices	**Students will be able to know and understand:** Structure and function of the processor The Arithmetic and Logic Unit; Control Unit and Registers; Buses; how this relates to assembly language programs The Fetch-Decode-Execute Cycle The factors affecting the performance of the CPU The use of pipelining in a processor to improve efficiency	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA.	**Students will be assessed by:** Exam questions End of topic / end of section tests
Software and software development	**Students will be able to know and understand:** The need for, function and purpose of operating systems Memory management: Interrupts, Scheduling Distributed, embedded, multi-tasking, multi-user and Real Time operating systems BIOS Device drivers Virtual machines	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA.	**Students will be assessed by:** Exam questions End of topic / end of section tests

Curriculum Map: Year 12 Computer Science

	Applications generation, Utilities, Open vs. closed source Translators: Interpreters, compilers and assemblers Linkers and loaders and use of libraries Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development Procedural languages, Assembly language. Modes of addressing memory Object-oriented languages with an understanding of classes, objects, methods, attributes, encapsulation, inheritance and polymorphism		
Exchanging Data	**Students will be able to know and understand:** Compression, Encryption and Hashing: Lossy vs Lossless compression, RLE and dictionary coding for lossless compression, Symmetric and asymmetric encryption, Different uses of hashing Relational database concepts: flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing Methods of capturing, selecting, managing and exchanging data Normalisation to 3NF SQL – Interpret and modify Referential integrity Transaction processing: ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA. Write HTML, CSS and JavaScript. Create a 3NF relational database. Solve PageRank algorithms	**Students will be assessed by:** Exam questions End of topic / end of section tests

Curriculum Map: Year 12 Computer Science

	Characteristics of networks and the importance of protocols and standards The internet structure The TCP/IP stack, DNS, Protocol layering LANs and WANs Packet and circuit switching Network security and threats, use of firewalls, proxies and encryption Network hardware Client–server and peer-to-peer HTML, CSS and JavaScript Search engine indexing, PageRank algorithm Server and client-side processing		
Data types, data structures and algorithms	**Students will be able to know and understand:** Primitive data types, integer, real/floating point, character, string and Boolean Represent positive integers in binary Use of sign and magnitude and two's complement to represent negative numbers in binary Represent positive integers in hexadecimal Representation and normalisation of floating point numbers in binary Floating point arithmetic, positive and negative numbers, addition and subtraction How character sets (ASCII and UNICODE) are used to represent text Arrays (of up to 3 dimensions), records, lists, tuples	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA. Addition and subtraction of binary integers Convert positive integers between binary, hexadecimal and denary Carry out Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR	**Students will be assessed by:** Exam questions End of topic / end of section tests

Curriculum Map: Year 12 Computer Science

	Define problems using Boolean logic Manipulate Boolean expressions, including the use of statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation Using logic gate diagrams and truth tables The logic associated with D type flip flops, half and full adders	Create, traverse, add data to and remove data from: linked-list, graphs, stack, queue, tree, binary search tree, hash table Karnaugh maps to simplify Boolean expressions	
Legal, moral, cultural and ethical issues	**Students will be able to know and understand:** Computing related legislation • The Data Protection Act 1998 • The Computer Misuse Act 1990 • The Copyright Design and Patents Act 1988 • The Regulation of Investigatory Powers Act 2000 Moral and ethical issues The individual moral, social, ethical and cultural opportunities and risks of digital technology: • Computers in the workforce • Automated decision making. • Artificial intelligence • Environmental effects. • Censorship and the Internet. • Monitor behaviour • Analyse personal information	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA.	**Students will be assessed by:** Exam questions End of topic / end of section tests

Curriculum Map: Year 12 Computer Science

	• Piracy and offensive communications • Layout, colour paradigms and character sets		
NEA 3.1	Further programming practice will be completed over the course culminating in the creation of a computer programming project for the NEA. Students will analyse the problem, create a design including the development and testing of the solution.		