Long Term Plan Year 12 Computer Science

Half Term	Unit Title	Key Knowledge/Content to learn and retain	Essential Skills to acquire (subject & generic)	Link to intent and ethos	Anticipated misconceptions	Links to previous KS	Link to future KS	Opportunity for stretch and high prior attainers	SMSC & British Values	Cultural Capital	Career Link
One	1.1. The characteristics of contemporary processors, input and storage devices 1.1.1 Structure and function of the processor 1.1.2 Types Of processors 1.1.3 Input, output, and storage	Learners are required to know components of a processor and how they work together to fetch-decode-execute data and instructions from main memory. This leads to an understanding of processor performance, different types of processors and processor architecture. Learners are required to learn primary and secondary storage, different storage media and the technologies that underpin them.	Drawing and labelling processor architecture diagrams. Extended writing Develop a line of enquiry based on observation	Learners will develop knowledge and understanding of technical content in this unit through computational approaches by using essential skills they will be able to logically think about how a CPU processes data, how data is stored in different memory and storage components whilst concurrently thinking how all these components function together.	Learners need to understand the purpose and function or registers within the processor, including the PC, accumulator, MAR, MDR and CIR; registers hold/store data that change as the program executes. CISC processors have a complex instruction set. RISK processors have a reduced instruction set. RAM is volatile/ ROM is non-volatile	Learners are expected to develop an understanding of this topic such as programming, Binary, and networking. Learners will benefit from knowing how memory and storage work when beginning the practical elements of the course such as how algorithms are created, stored, and executed by the CPU. This in turn leads to a deeper understanding that data is stored and communicated in a computer using the Binary number system. In networking learners will benefit from knowing computer components and their functionality before understanding how they are networked together.	This topic provides learners with the knowledge and skills that can be applied in the workplace or to continue to study more complex topics/modules at university.	Learners are encouraged to learn how the CPU fetches, decodes and executes through simulation software 'The Little Man Computer'. This enables learners to develop an understanding of how the CPU works visually and at a much deeper level whilst introducing learners to assembly programming language. Learners are encouraged to further deepen their knowledge and understanding through reading and evaluating higher level material such as; Nanotechnology and computer science; Trends and advances.	Learners will have opportunities to think and discuss how memory has evolved in computer systems over time. Influential people involved from 16th century to present day. Computer component manufacturers leading in processor architecture. How government agencies (GCHQ, Police, NHS, and Council) store data, what kind of data, is this moral? The laws surrounding data; private and public data and copyright. Impacts on storage space and how these impacts on the environment and society.	Learners will have the opportunity to think and discuss the topic on 'Big Data'; issues relating to how enormous amounts of data is stored which in turn relates to mainframe computers and quantum computers. Further concerning issues are discussed such as misinformation (Information warfare), and social media and the impacts this has on society	https://www.gchq-careers.co.uk/ GCHQ Apprenticeships https://www.mi5.gov.uk/careers https://www.sis.gov.uk/explore-careers.html https://www.yhrocu.org.uk/vacancies/ https://nationalcrimeagency.gov.uk/careers/vacancies https://www.gov.uk/apply-apprenticeship University of Hull Computer Science courses University of York Top Cyber Security Universities in the UK
Two	1.2 Software and software development 1.2.1 Systems Software 1.2.2 Applications Generation 1.2.3 Software Development 1.2.4 Types of programming language	Learners are required to know the need and purpose of operating systems which leads to an understanding of systems software in more detail such as how a program is read and processed. Different software development methodologies are introduced and the need for and characteristics of programming paradigms, assembly language and Object-Oriented Programming language.	Drawing abstract diagrams to visualise processes and software development methodologies Extended writing Develop a line of enquiry based on observation Evaluate impacts	Learners will develop knowledge and understanding of technical content in this unit through computational approaches by using essential skills they will be able to logically think about how different types of software development are more suited to certain types of projects.	Some advantages overlap into other software methodologies. Systems software is often confused with applications software; these are two different categories that software falls into.	Learners should know how to classify software and give examples of each type of software. This leads to a deeper understanding of new content studied at A Level such as software development and programming paradigms.	This topic provides learners with the knowledge and skills that can be applied in the workplace or to continue to study more complex topics/modules at university. Computer Science A Level requires the learner to research, design, develop, and test a software solution; a software development project is an expected component studied in the third year as part of a Computer Science degree.	Learners are encouraged to learn another type of programming language to provide them with a broader range of knowledge and skills which in turn gives learners more choice in their software development project in year 13.	Learners will have opportunities to think and discuss how different types of software is used - is it free? Implications of software development; how will it impact users/society? The laws surrounding software development and how these impacts on the user and society.	Learners will have the opportunity to think and discuss the wider impacts of software development from across the world.	Software Engineer at GCHQ Project Manager at GCHQ

Three	1.3 How data is connected between different systems 1.3.1 Compression, Encryption, and hashing 1.3.2 Databases 1.3.3 Networks 1.3.4. Web Technologies	Learners are required to know different methods of compression which leads to an understanding of databases and SQL (Structured Query Language). Learners are required to know characteristics of networks, network security, and web technologies.	Drawing and labelling abstract diagrams for compression, relational databases, and network topologies Identify relationships between entities in a database Identify the process of normalising a database to third normal form Interpret and modify SQL (Structured Query Language) statements Identify the structure of the Internet	Learners will develop knowledge and understanding of technical content in this unit through computational approaches by using essential skills they will be able to logically think about how compression and encryption is linked to databases, networks, and web technologies.	Compression reduces a file size whereas encryption turns readable data into an unreadable format. There are several terms of hashing: hashing tables and hashing algorithms. Topologies are the logical design of a network whereas network types are either local or wide area networks. The Internet and the WWW are completely two different technologies; the Internet is the physical hardware needed to transmit and retrieve data whereas the WWW is the software that uses the Internet.	Learners should know the purpose of compression and identify compression methods. The purpose of encryption and methods should already be known The basics of what a database is along with simple SQL statements to create and modify database tables. Networks, topologies, protocols and layers are known which provides a sound basis for further understanding of complex processes within networking and web technologies	Learners should have required the knowledge and skills in this in-depth unit to apply them in the workplace or to progress further in their studies at university. Typically, this unit is one the popular career pathways that learners choose to study at university.	Learners are encouraged to learn different types of compression and encryption methods. Learners will have the opportunities to create a relational database and to use SQL statements to create and modify a database. Opportunities for learners to use CISCO packet tracer software to simulate networks. Learners will use their knowledge and understanding to create a simple website and how a database is connected using MySQL.	Learners will have opportunities to explore legislation surrounding the Internet in the UK and globally. -Internet services in other countries – why do certain countries limit access to the internet and what impact does this have on society; is this moral? -What impact does the Internet have on society regarding social media?	Learners will have the opportunity to think and discuss issues surrounding the Internet: - Who owns the Internet? -Origins of the Internet -Difference between the Internet and the World Wide Web -Sir Tim Berners Lee founder of WWW -Future trends of the Internet and IoT	Mathematician or Cryptographer at GCHQ IT Service Manager at GCHQ Network Manager SQL Developer Web Developer
Four	1.4. How data is represented and stored within different data structures. 1.4.1 Data Types 1.4.3 Boolean Algebra	Learners are required to know how a computer stores data using Binary and Hexadecimal number systems; Converting between denary, Binary and Hexadecimal number systems and how a computer stores negative numbers. which leads to an understanding of Boolean algebra; AND, OR, NOT, XOR, NAND, NOR De Morgan's Law.	Represent positive and negative numbers in Binary Addition and subtraction of binary numbers Representation and normalisation of floating-point numbers in binary Bitwise manipulation ; Binary logic, multiplication, and division Draw logic circuits to a given expression Use Karnaugh maps to simplify Boolean expressions	Learners will develop knowledge and understanding of technical content in this unit through computational approaches by using essential skills they will be able to logically think about why different numbers of systems are used in computing and the logical operations a computer performs.	Hexadecimal uses the decimal number system, 0-9 and six letters, A-F to represent data. Hexadecimal is used to identify a physical computer on a network known as a MAC address. Hexadecimal is used to represent RGB values in web development and assembly languages. Binary is a number system that only represents two states but is used with different types of logical gates to build complex circuitry.	Learners should know how to perform number conversions which leads to a deeper understanding of how a computer calculates negative numbers.	Learners should have required the knowledge and skills in this unit to apply them in the workplace or to progress further in their studies at university. This unit will enable learners of the key mathematical underpinnings of computer science to aid learners in problem solving and programming. .	Learners will have opportunities to use logic gate software to simulate logic gate circuitry to solve complex problems and to simplify Boolean expressions.	Learners will have opportunities to explore how law enforcement agencies use hexadecimal (MAC Address) values to trace a computer that has been used in a crime. How do hackers spoof their location to avoid detection?	Learners will have the opportunity to discuss the works of George Boole, a self-taught English mathematician, philosopher and logician of whom is credited with laying the foundation of the information age and Boolean algebra.	
Five	1.4.2 Data Structures	Learners are required to know and understand essential data structures and how to traverse data for a given structure; arrays, linked-list, graph(directed/ undirected), stack, queue, tree, binary search tree, and hash table	Drawing and labelling data structure diagrams. create, traverse, add data to and remove data from a given data structure Extended writing Compare and contrast the efficiency and performance of a given data structure to make a reasoned and sound argument	Learners will develop knowledge and understanding of technical content in this unit through computational approaches by using essential skills they will be able to logically think about how different data structures work and how they can be used together in storing data or retrieving data.	Tree traversal algorithms fall into two categories: Depth-First Traversal and Breadth-First Traversal. Undirected graphs have edges that have no direction which indicates a two-way relationship. A stack is a LIFO (Last in first out) data structure A queue is a FIFO (first in first out) data structure)	Learners should have knowledge of common computer science algorithms studied at KS4; acquired skills are used to learn data structures at A Level. For each data structure learners will have acquired the skills from KS4 to know how a data structure works logically, and how to analyse and explain the code.	Learners should have required the knowledge and skills in this unit to apply them in the workplace or to progress further in their studies at university. This unit will enable learners to apply their knowledge through practical coding in Algorithms and data structure modules at university.	Learners will have opportunities to learn the theory of data structures, to analyse them and to explore the pseudocode and actual code using a chosen programming language and then to use them in larger programs.	Ethical discussions on open and closed source code and the impacts this has in the industry. Ethical discussions on Ethical hacking by governments and well-known hacking groups.	Learners will have the opportunity to discuss the origins of programming: - GCHQ and Bletchley Park -The history of programming from machine code to low level code, to high level code - First computer programmer Ada Lovelace, daughter of Lord Byron the Poet.	

Six	2.3 Algorithms **NEA; Content of non exam assessment programming project (component 3)** **3.1. Analysis of the problem** 3.1.1 Problem identification 3.1.2 Stakeholders 3.1.3 Research the problem 3.1.4 Specify the proposed solution	Learners are required to know and understand essential algorithms for computer science. Bubble sort, Insertion sort, merge sort, quick sort, Dijkstra's shortest path, A* algorithm, Binary search, and Linear search. Learners are expected to analyse, develop, evaluate, and document a program written in a suitable programming language. Topic 3.1 is the first phase of the NEA; Analysis Learners are required to document evidence in a report to describe and justify the features that make the problem solvable by computational methods.	Drawing and labelling diagrams for common algorithms Analysis and design of algorithms for a given data set create, traverse, add data to and remove data from a given data structure Extended writing Identify and determine the efficiency of different algorithms. Big O notation (constant, linear, polynomial, exponential and logarithmic complexity) Research a problem and solutions to similar problems to identify and justify suitable approaches to a solution. Identify and describe end users and stakeholders for the intended solution Specify and justify the solution requirements	Learners will develop knowledge and understanding of technical content in this unit through computational approaches by using essential skills they will be able to logically think about common computer science algorithms and their uses.	Learners are required to identify algorithms by analysing and reading its code. Bubble sort uses an unordered array of items by comparing each item with the next one and swapping them if they are out of order; when analysing the code, a temporary variable is used to process the swaps made. There is more than one way to write the Quick sort; Tony Hoare's method (uses pointers) and Nico Lomutos method (uses pivots)	Learners should have knowledge of common computer science algorithms which prepares them to learn new content at A Level such as the efficiency of algorithms known as Big O notation.	Learners should have required the knowledge and skills in this unit to apply them in the workplace or to progress further in their studies at university. This unit will enable learners to apply their knowledge through practical coding in Algorithms and data structures modules at university.	Learners will have opportunities to learn the theory of computer science algorithms, to analyse them and to explore the pseudocode and actual code using a chosen programming language.	Ethical discussions on open and closed source code and the impacts this has in the industry. Ethical discussions on Ethical hacking by governments and well-known hacking groups.	Learners will have the opportunity to discuss the origins of programming: - GCHQ and Bletchley Park -The history of programming from machine code to low level code, to high level code - First computer programmer Ada Lovelace, daughter of Lord Byron the Poet. Discuss the origins of each individual algorithm e.g.. The quick sort algorithm was created to sort Russian words before looking them up in an English-Russian dictionary.	