Year 10 Computer Science Sequence

Year	Content Taught	Reference	Essential Knowledge	Assessment	Rationale
Year 10 Computer Science					
HT1	In this half term students will study a topic focused on: **Computational Thinking, Programming, and Algorithms.** Algorithms Programming fundamentals	Algorithms, Computational thinking, Designing, creating and refining algorithms, Searching and sorting algorithms, Programming fundamentals	Computational Thinking, Programming, and Algorithms. ▪ Principles of computational thinking: 　o Abstraction 　o Decomposition 　o Algorithmic thinking ▪ Identify the inputs, processes, and outputs for a problem ▪ Structure diagrams ▪ Create, interpret, correct, complete, and refine algorithms using: 　o Pseudocode 　o Flowcharts 　o Reference language/high-level programming language ▪ Identify common errors ▪ Trace tables ▪ Standard searching algorithms: 　o Binary search 　o Linear search ▪ Standard sorting algorithms: 　o Bubble sort 　o Merge sort 　o Insertion sort ▪ The use of variables, constants, operators, inputs, outputs and assignments	**Formative** Students will complete retrieval exercises each lesson to review and recall knowledge from previous lessons and apply this knowledge to alternate scenarios to deepen understanding. **Summative** Students will complete a Teams based KO Test to summarise content. Within this half term students will develop essential knowledge in lessons and 'bring it all together', by implementing it into a challenging Python programming creation task. Students will apply algorithmic thinking to create a Python based solution to the task set.	**Computer Science Year 10** This Essential Knowledge also builds on knowledge gained in KS3. Computer Science is engaging and practical, encouraging creativity and problem solving. It encourages students to develop their understanding and application of the core concepts in computer science. Students also analyse problems in computational terms and devise creative solutions

			▪ The use of the three basic programming constructs used to
			▪ control the flow of a program:
			   ○ Sequence
			   ○ Selection
			   ○ Iteration (count- and condition-controlled loops)
			▪ The common arithmetic operators
			The common Boolean operators AND, OR and NOT
				Students will complete an in-class assessment containing the following questions:
				
				'Perform a search on the following data…'
				
				'Perform a bubble sort on the following set of data'
				
				'What are the three main programming constructs'
					by designing, writing, testing and evaluating programs.
					
					This includes all units relating to Algorithms, programming techniques, Producing Robust Programs, Computational

HT2	In this half term students will study a topic focused on:
	
	Computational Thinking, Programming, and Algorithms
	Programming fundamentals
		Data types, Additional programming techniques	**Data types**
			▪ The use of data types:
			   ○ Integer
			   ○ Real
			   ○ Boolean
			   ○ Character and string
			   ○ Casting
			Additional programming techniques
			▪ The use of basic string manipulation
			▪ The use of basic file handling operations:
			   ○ Open
			   ○ Read
			   ○ Write
			   ○ Close
			The use of records to store data
				Formative
				Students will complete retrieval exercises each lesson to review and recall knowledge from previous lessons and apply this knowledge to alternate scenarios to deepen understanding.
				
				Summative
				Students will complete a Teams based KO Test to summarise content.
				
				Within this half term students will develop essential knowledge in lessons and 'bring it all together', by implementing it into a challenging Python programming creation task. Students will apply algorithmic
					Logic, Data Representation.
					This is taught at the start of Year 10 as it introduces the students to the 4 main concepts of Computer Science. It teachers them the theory and essential knowledge of programming in order to build them up to being able to complete their programming project in Year 11.
					
					This will equip the students with the

				thinking to create a Python based solution to the task set.	essential knowledge to
HT3	In this half term students will study a topic focused on:	Programming fundamentals Producing robust programs, Producing robust programs, Defensive design, Testing	▪ The use of SQL to search for data ▪ The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) ▪ How to use sub programs (functions and procedures) to produce structured code ▪ Random number generation	**Formative** Students will complete retrieval exercises each lesson to review and recall knowledge from previous lessons and apply this knowledge to alternate scenarios to deepen understanding.	progress to specific ICT & Computer Science KS5 courses or employment that is computer oriented.

	Computational Thinking, Programming, and Algorithms		**Producing robust programs**
			Defensive design
			▪ Defensive design considerations:
			   o Anticipating misuse
			   o Authentication
			▪ Input validation
			▪ Maintainability:
			   o Use of sub programs
			   o Naming conventions
			   o Indentation
			   o Commenting
			Testing
			▪ The purpose of testing
			▪ Types of testing:
			   o Iterative
			   o Final/terminal
			▪ Identify syntax and logic errors
			▪ Selecting and using suitable test data:
			   o Normal
			   o Boundary
			   o Invalid/Erroneous
			Refining algorithms
				Summative
				Students will complete a Teams based KO Test to summarise content.
				
				Within this half term students will develop essential knowledge in lessons and 'bring it all together', by implementing it into a challenging Python programming creation task. Students will apply algorithmic thinking to create a Python based solution to the task set.
				
				Students will complete an in-class assessment containing the following questions:
				
				'Explain, using examples from the program, **two** ways Finn can improve the maintainability of the program'
					Sequence and Progression:
					
					Databases in Yr10 is linked to Systems Security and Computer Storage in Computer Science in Yr11.
					
					Computational Logic in Yr10 is linked to Computer memory and Computer Storage in Computer Science in Yr11.
					
					Producing Robust Programs in Yr10 is linked to Legal and Ethical Issues and Systems Security in

				'Describe **two** examples of defensive design that should be considered when developing this program.'

'Identify syntax and logic errors in the following Python program' | Computer Science in Yr11.

Algorithms and Programming Techniques in Yr10 is linked to Systems Architecture in Computer Science in Yr11. |
| **HT4** | In this half term students will study a topic focused on:

Computational Thinking, Programming, and Algorithms | Boolean logic, Programming languages and Integrated Development Environments Languages. The Integrated Development Environment (IDE) | **Boolean logic**
- Simple logic diagrams using the operators AND, OR and NOT
- Truth tables
- Combining Boolean operators using AND, OR and NOT
- Applying logical operators in truth tables to solve problems

Programming languages and Integrated Development Environments
 Languages
- Characteristics and purpose of different levels of programming language:
  - High-level languages
  - Low-level languages
- The purpose of translators
- The characteristics of a compiler and an interpreter
The Integrated Development Environment (IDE)
- Common tools and facilities available in an Integrated Development Environment (IDE):
  - Editors
  - Error diagnostics
  - Run-time environment | **Formative**
Students will complete retrieval exercises each lesson to review and recall knowledge from previous lessons and apply this knowledge to alternate scenarios to deepen understanding.

Summative
Students will complete a Teams based KO Test to summarise content.

Within this half term students will develop essential knowledge in lessons and 'bring it all together', by answering the following questions: "What is meant by a high-level language?"

"Describe the common features of an IDE" | |

			Translators		
HT5	In this half term students will study a topic focused on: **Computer systems:** Systems architecture, Memory and storage	Systems architecture, Architecture of the CPU, CPU performance, Embedded systems Memory and storage, Primary storage (Memory)Secondary storage	**Systems architecture** **Architecture of the CPU** ▪ The purpose of the CPU: 　○ The fetch-execute cycle ▪ Common CPU components and their function: 　○ ALU (Arithmetic Logic Unit) 　○ CU (Control Unit) 　○ Cache 　○ Registers ▪ Von Neumann architecture: 　○ MAR (Memory Address Register) 　○ MDR (Memory Data Register) 　○ Program Counter 　○ Accumulator **CPU performance** ▪ How common characteristics of CPUs affect their performance: 　○ Clock speed 　○ Cache size 　○ Number of cores **Embedded systems** ▪ The purpose and characteristics of embedded systems ▪ Examples of embedded systems **Memory and storage** **Primary storage (Memory)** ▪ The need for primary storage ▪ The difference between RAM and ROM ▪ The purpose of ROM in a computer system ▪ The purpose of RAM in a computer system	**Formative** Students will complete retrieval exercises each lesson to review and recall knowledge from previous lessons and apply this knowledge to alternate scenarios to deepen understanding. **Summative** Students will complete a Teams based KO Test to summarise content. Within this half term students will develop essential knowledge in lessons and 'bring it all together', by answering the following questions: "What affects the performance of a CPU? "What is the relationship between main memory and the CPU?" "What are the common characteristics of a magnetic storage device?"	

			▪ Virtual memory
			Secondary storage
			▪ The need for secondary storage
			▪ Common types of storage:
			   o Optical
			   o Magnetic
			   o Solid state
			▪ Suitable storage devices and storage media for a given application
			▪ The advantages and disadvantages of different storage devices and storage media relating to these characteristics:
			   o Capacity
			   o Speed
			   o Portability
			   o Durability
			   o Reliability
			   o Cost
			Units
			▪ The units of data storage:
			   o Bit
			   o Nibble
			   o Byte
			   o Kilobyte
			   o Megabyte
			   o Gigabyte
			   o Terabyte
			   o Petabyte
			▪ How data needs to be converted into a binary format to be
			▪ processed by a computer
			Data capacity and calculation of data capacity requirements
				'Describe the relationship between the CPU and virtual memory'	

HT6	In this half term students will	Data storage, Numbers Characters, Images	**Data storage** Numbers	**Formative** Students will complete retrieval exercises each lesson to review	

study a topic focused on:
 	
Computer systems
Memory and storage
Computer networks, connections and protocols
	Sound, Compression, Computer networks, connections and protocols, Networks and topologies, Wired and wireless networks, protocols and layers	■ How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa
		■ How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur
		■ How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa
		■ How to convert binary integers to their hexadecimal equivalents and vice versa
		■ Binary shifts
		Characters
		■ The use of binary codes to represent characters
		■ The term 'character set'
		■ The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.:
		   ○ ASCII
		   ○ Unicode
		Images
		■ How an image is represented as a series of pixels, represented in binary
		■ Metadata
		■ The effect of colour depth and resolution on:
		   ○ The quality of the image
		   ○ The size of an image file
		Sound
		■ How sound can be sampled and stored in digital form
		■ The effect of sample rate, duration and bit depth on:
			and recall knowledge from previous lessons and apply this knowledge to alternate scenarios to deepen understanding.
			
			Summative
			Students will complete a Teams based KO Test to summarise content.
			
			Within this half term students will develop essential knowledge in lessons and 'bring it all together', by answering the following questions:
			
			"Define what is meant by a character set"
			
			"Which type of compression is the most appropriate for compressing a text file? Explain your answer"
			
			"Draw a star network with 3 laptops, a printer and a server. Clearly label the diagram."
			
			"Explain why a star network may be appropriate for use in a school"	

| | | | | o The playback quality
o The size of a sound file
Compression
▪ The need for compression
▪ Types of compression:
    o Lossy
    o Lossless

Computer networks, connections and protocols Networks and topologies
▪ Types of network:
    o LAN (Local Area Network)
    o WAN (Wide Area Network)
▪ Factors that affect the performance of networks
▪ The different roles of computers in a client-server and a peer-to peer network
▪ The hardware needed to connect stand-alone computers into a Local Area Network:
    o Wireless access points
    o Routers
    o Switches
    o NIC (Network Interface Controller/Card)
    o Transmission media
▪ The Internet as a worldwide collection of computer networks:
    o DNS (Domain Name Server)
    o Hosting
    o The Cloud
    o Web servers and clients
▪ Star and Mesh network topologies
Wired and wireless networks, protocols and layers
▪ Modes of connection:
    o Wired
        ▪ Ethernet | "Describe two advantages and two disadvantages of using the cloud" | |

			○ Wireless ▪ Wi-Fi ▪ Bluetooth ▪ Encryption ▪ IP addressing and MAC addressing ▪ Standards ▪ Common protocols including: ▪ TCP/IP (Transmission Control Protocol/Internet Protocol) ▪ HTTP (Hyper Text Transfer Protocol) ▪ HTTPS (Hyper Text Transfer Protocol Secure) ▪ FTP (File Transfer Protocol) ▪ POP (Post Office Protocol) ▪ IMAP (Internet Message Access Protocol) ▪ SMTP (Simple Mail Transfer Protocol) The concept of layers		