Long-term planning

Computer Science - Year 10

Year 10	Autumn term 1	Autumn term 2	Spring term 1	Spring term 2	Summer term 1	Summer term 2
	Students will know that	**Students will know that**	**Students will know that**	**Students will know that**	**Students will know that**	**Students will know that**
	An algorithm is a set of instructions to solve a problem and can be represented in many ways. The flowchart symbols: start/end, process, input/output, decision and sub are used to create a diagrammatic representation of an algorithm. A brute force algorithm is a simple, comprehensive search strategy that systematically explores every option until a problem's answer is discovered. E.g. bubble sort and linear search Divide and conquer algorithms solve problems by dividing the main problem into subproblems, solving them individually and then merging them to find solution to the original problem. E.g. merge sort and binary search. The efficiency of linear search algorithms can be improved by stopping the search ones the value of the item in the current index position is more than the target value. A variable is a value that can be stored and changed throughout a program.	A binary number can be signed or unsigned: • Unsigned - used to represent positive binary numbers • Signed - used to represent both positive and negative binary numbers Binary subtraction can be achieved by adding the twos complement of the second number to the first number e.g. x – y = x + y Binary numbers are multiplied and divided through a process called shifting. There are two types of binary shift - arithmetic and logical. Arithmetic shifts pad the MSB meaning that the MSB does not shift but all other values do. Overflow occurs when the result of the calculation is too large to fit into the allocated number of bits. Hexadecimal (base 16) is a shorthand for binary. American Standard Code for Information Interchange is a	Compression is a technique for reducing file size and is sometimes necessary or desirable e.g. during data transmission. A lossy compression algorithm results in data being permanently lost that cannot be restored. Whereas a lossless compression algorithm allows the original file to be exactly reconstructed from the compressed data as no data is lost during this compression. A general purpose computer is a computer that is designed to be able to carry out many different tasks e.g. desktop/laptop. All general purpose computers follow the same basic model. Data is input, processed, then either output or sent to secondary storage. It is held in main memory while it is being processed. The CPU receives instructions and data from an input or memory. Input is from an input device such as a keyboard, mouse, camera or scanner. Output is to an	The CPU sequentially fetches, decodes and executes instruction stored in memory. The control unit sends a read signal to the RAM once it has fetched an instruction. Also decodes the instructions to send commands to the other components that need to be involved in executing the instruction. The control unit controls hardware. Registers are small memory cells within the CPU which is fast, they are used for temporary storage of data and results from the ALU are stored here. The arithmetic logic unit completes all calculations (maths/logic). This happens at the decode/execute phase. Buses transmit data/signals from one component to another. • Control Bus sends signals such as the read signal to the RAM. Sends commands e.g. commanding the ALU what to do with the data. (Bi-directional) • Data Bus carries the data from memory and transports around the	Magnetic devices use magnetic fields to magnetise tiny individual sections of a metal spinning disk. Each tiny section represents one bit. Magnetised represents a binary '1', demagnetised represents a binary '0'. Optical devices use a laser to scan the surface of the disc. Digital data is stored along the track by etching pits onto the surfaces of the disc. When the laser shines on the disc surface, land reflects the light back '1', whereas pits scatter the laser beam '0'. Solid state devices have no moving parts and are more durable. They have floating gate transistors that keep their charge even when switched off, at first all transistors are charged (1) but when save begins current is blocked to some transistors (0). An embedded system differs from a general-purpose computer as it is a small computer that forms part of a larger system, device or machine. Its purpose is to control the device and to allow a user to interact with it.	Logical operators, AND, OR and NOT have different rules that must be followed. • **AND** – both inputs have to be ON (1) in order to create an ON (1) state. • **OR** – either input needs to be ON (1) in order to create an ON (1) state. • **NOT** – the output is the exact opposite of the input. E.g. (1) becomes (0) / (0) becomes (1)

	Whereas a constant is a value that does not change during program execution. A one-dimensional data structure is a list where each item has a specific position, accessed by an index. The index will start at the value 0. A two-dimensional array is a data structure that organises elements in rows and columns, forming a grid-like structure. It's like a table and is used to store and access multiple related values efficiently. An array is a sequence of items with the same (homogeneous) data type and a record is a sequence of items with different (heterogeneous) data types. Items that make up record are fields.	standard for encoding characters (letters, numbers, symbols) where each character is assigned its own unique value. It uses 7-bits to represent each value. A pixel is the smallest element of a bit-mapped image, and the size of an image is expressed as width*height in pixels. The higher the resolution, the more pixels per inch (ppi) and the better quality image. Colour depth is the number of bits used to represent the colour of a pixel. The more bits the greater number of colours that can be represented in an image. The amplitude of a sound wave determines the sound's loudness – the higher the amplitude the louder the sound. A sample is a measure of amplitude at a point in time and the sample rate is the number of samples taken per second (measured in hertz). The time between the samples is the sample interval. Relational operators compare two values and give a result of True or False. Iteration is used to repeatedly execute the same block of code on each element of a data structure e.g. a string, an	output device, such as a monitor, printer or speaker. Von Neumann architecture is the design upon which many general purpose computers are based, using the stored program concept. The syntax of opening comma-separated value text files for reading, appending or writing to. Appending to a text file is used when you want to add to what is already in the text file, whereas writing to a text file is used when you want to overwrite what is already in the text file. Functions return values, whereas procedures do not return values. Information can be passed into functions as arguments; this is through parameters. Arguments are specified after the function name, inside the parentheses. E.g. def price (pItem): You can add as many arguments as you want, just separate them with a comma. While loops test the condition at the beginning of the loop. If the condition is met, the code within the loop is executed before the program loops back to test the condition again. While loops are also known as repetition or condition-controlled loops.	CPU to various components. (Bi-directional) • Address Bus carries memory addresses from the processor to other components such as RAM and input/output devices. (Unidirectional) The clock synchronises the actions of the CPU, with each tick of the clock triggering an operation (measured in hertz). A higher clock speed is desirable. Main memory (memory internal to the computer) consists of RAM, ROM and cache. ROM is non-volatile and RAM and cache are volatile. Main memory is limited in size. Using the cache allows for faster processing as the processor does not have to wait for the data and instructions to be fetched from the random-access memory (RAM). Secondary storage provides long-term, non-volatile storage for programs and data. A pre-existing library is reusable code that can be used throughout a program. The code has been made public through reusable modules or functions. Using pre-existing libraries saves programmers time by using	In an embedded system, a microcontroller acts as the "brain," managing specific tasks and functions. It's a compact integrated circuit containing a processor, memory, and input/output peripherals, all on a single chip. The purpose of an operating system is to control the computer's hardware and software. The main roles of the OS are: file management, process management, peripheral management, user management and to provide a user interface. Only one process can have exclusive use of the CPU and the OS uses scheduling algorithms to prioritise processes, they are held in a queue whilst waiting. The OS shares main memory between processes, allocated each one its own section of RAM and uses a paging algorithm to swap processes between RAM and virtual memory. Utility software is a collection of tools that help maintain the functionality of a computer. Robust software is software that is free from vulnerabilities, is efficient and follows good programming practice.	

		array, a numeric range or an open file, until every element has been processed.		working code that has already been tested. Some examples of pre-existing libraries are: • Random • Math • Turtle The turtle library is a pre-built library of code which allows the creation of images and shapes using a 'pen'. The random library is a library of code which allows users to make use of 'random' in their programs e.g. random choice or random numbers.	A code review can be done by a piece of software or by a more senior programmer. The purpose is not to spot errors but to ensure programs are robust. Code written in a high-level language need translating into a low-level language, using a compiler or interpreter, in order for the computer to be able to understand.	
	Students will know how	**Students will know how**	**Students will know how**	**Students will know how**	**Students will know how**	**Students will know how**
	To follow and write algorithms represented as flowcharts, as written descriptions and in program code. To apply a bubble sort and merge sort algorithm to a list of items. To apply a binary search and linear search algorithm to a list of items. To compare and contrast a bubble sort with a merge sort and a linear search with a binary search. To locate and fix syntax errors in code. To use indexing in python to access items in one-dimensional and two-dimensional lists.	To determine the maximum number of unique states that can be represented by a binary pattern of a given length (2n). To convert both unsigned and signed denary integers into binary and vice versa. To represent signed integers in two's complement, using the MSB as a negative value. To complete binary addition of 2 8-bit binary numbers. To use the hexadecimal table to complete hexadecimal conversions. To interpretate a sound graph, knowing the y axis is amplitude and the x axis is time.	To write code to merge two sorted lists into one sorted list using while loops. To identify when it is appropriate to use lossy or lossless compression. To describe how all of the hardware components work together in the Von Neumann architecture. To use decomposition and abstraction to help understand problems and design effective solutions. To be able to write programs that use pre-existing (built-in library) and user-devised subprograms (procedures and functions). To be able to write subprograms that take parameters and ones that	To decompose literacy dependent problems to create working solutions. To create and complete simple diagrams of the fetch-decode-execute cycle. To explain why a higher clock speed is desirable. To convert flowcharts and written descriptions into python programs. To use the programming language subset (PLS) document to assist with problem solving activities. To use the turtle library to create movement programs on a canvas which are based on X, Y coordinates. To use the turtle library to show creativity.	To recognise which secondary storage device is more suitable than another for a particular purpose taking into account: cost, capacity, speed of access, portability, durability, reliability. To describe the purpose of utility software such as file repair, backup software, data compression, defragmentation and anti-malware. Including how it works and why it is needed. To describe the advantages and disadvantages of low-level languages and high-level languages. To identify tasks which low-level languages are used. E.g. writing control software for embedded systems with limited processing power and memory.	To apply logical operators (AND, OR, NOT) in truth tables with up to three inputs to solve problems. To draw logic circuit diagrams. To apply the use of logical operators in python programs to practically display this skill by combining multiple expressions that produce a single true or false outcome. To recall knowledge throughout year 10 to apply to exam style questions. To answer exam style coding questions 1-4. To reflect on progress made in year 10.

		To explain the factors that affect the fidelity of the digital representation. To calculate the file size of an image (width*height*colour depth). To calculate the file size of an audio recording (sample rate*bit depth*recording length in seconds) To identify the input, processing and output of an algorithm. To identify different types of errors (syntax/logic/runtime) and correct these errors. To use a loop to iterate through a data structure to perform a linear search.	return one or more values to the calling code. To be able to describe the benefits of using sub-programs when coding.	To incorporate selection, iteration and repetitio into turtle graphics. To use the random library to create solutions to problems that randomly select items from arrays and random numbers within a given range.	To compare a compiler vs an interpreter for translation. To select and justify which translator should be used for a given purpose. To answer exam style coding questions 1-3.	To identify gaps in their own knowledge to target revision to these areas. To apply revision strategies to Computer Science for both Paper 1 and Paper 2.
	Vocabulary and the concepts they link to	**Vocabulary and the concepts they link to**	**Vocabulary and the concepts they link to**	**Vocabulary and the concepts they link to**	**Vocabulary and the concepts they link to**	**Vocabulary and the concepts they link to**
	Algorithms, Decomposition, Abstraction, Subprograms, Flowchart, Pseudocode, Variable, Data Structure, Integer, String, Float, Boolean, Constant, One-Dimensional Array, Two-Dimensional Array, String Slicing, String Concatenation, Record, Field, Efficiency, Bubble Sort, Merge Sort, Linear Search, Binary Search, Brute Force, Divide and Conquer, Index, Homogeneous, Heterogeneous, Data Types, String, Integer, Float, Boolean, Syntax,	Binary, Denary, Hexadecimal, Base 2, Base 10, Base 16, Unsigned integer, signed integer, Twos complement, Logical Shift, Arithmetic shift, Overflow, American Standard Code for Information Interchange (ASCII), Pixel, Bitmap, Resolution, Colour Depth, Metadata, Amplitude, Sample Frequency, Sample Rate, Sample Interval, Bit Depth, Relational Operators, Iteration, Data Structure, Most Significant Bit, Fidelity, Logic, RunTime,	Compression, lossy, lossless, hardware, Von Neumann architecture, Stored program concept, Input, Process, Output, Processor, Data, Instructions, Central Processing Unit, Repetition, Condition-Controlled Loops, Secondary Storage, Main Memory, Functions, Procedures, Parameters, Parentheses, Built-in library, User-defined subprogram,	Control Unit, Control Bus, Address Bus, Data Bus, Registers, Arithmetic Logic Unit, Clock, Cache, Cores, Random Access Memory, Read Only Memory, Main Memory, Fetch, Decode, Execute, Hardware, Bi-directional, Unidirectional, Hertz, Volatile, Non-Volatile, Selection, Nested Selection	Secondary Storage, Magnetic Storage, Optical Storage, Solid State Drive, Fetch-Decode-Execute, High Level Language, Low Level Language, Capacity, Portability, Durability, Reliability, Operating system, Peripheral, User interface, Command line interface, Graphical user interface, Transistors, Embedded Systems, Utility Software, Robust Software, Vulnerabilities, Defragmentation, Anti-Malware, Compiler, Interpreter, Translation, Electromagnets, Assembly Language, Assembler	Logic Gates, Truth tables, AND, OR, NOT, Logical operators,

	Assessment	**Assessment**	**Assessment**	**Assessment**	**Assessment**	**Assessment**
	Algorithms Assessment Low stakes quizzes	Python Quiz Low stakes quizzes	Binary Assessment Low stakes quizzes	Python Assessment (Q1-3) Low stakes quizzes	Computers Assessment Low stakes quizzes	End of year mock assessment (Paper 1 and Paper 2) Low stakes quizzes
	Diversity & development of cultural capital	**Diversity & development of cultural capital**	**Diversity & development of cultural capital**	**Diversity & development of cultural capital**	**Diversity & development of cultural capital**	**Diversity & development of cultural capital**
	Spiritual – use of imagination and creativity and reflective of their experiences Social – class/group discussions	Spiritual – use of imagination and creativity and reflective of their experiences Social – class/group discussions	Spiritual – use of imagination and creativity and reflective of their experiences Social – class/group discussions	Spiritual – use of imagination and creativity and reflective of their experiences Social – class/group discussions	Spiritual – use of imagination and creativity and reflective of their experiences Moral – Accessibility of user interfaces Social – class/group discussions	Spiritual – use of imagination and creativity and reflective of their experiences Social – class/group discussions
	Cross-curricular opportunities and enrichment	**Cross-curricular opportunities and enrichment**	**Cross-curricular opportunities and enrichment**	**Cross-curricular opportunities and enrichment**	**Cross-curricular opportunities and enrichment**	**Cross-curricular opportunities and enrichment**
	National Coding Week	Festive coding challenges Maths – Computing related Math calculations	Safer Internet Day	National Careers Week Amazon Virtual Tour Easter coding challenges Maths – Plotting coordinates		Electronics – Logic Gates in circuits