

# **Computing/CS**

Year 10	AUTUMN		SPRING		SUMMER	
Computational Thinking	Algorithms & Programming Part 1: Sequence	Programming Part 2: Selection	Programming Part 3: Iteration	Programming Part 4: Sub-routines	Programming Part 5: Strings & lists	Algorithms Part 2
	translators. Use sequence, variables, and input in Python. Design programs using a flowchart.	programs. Work with arithmetic and logical expressions. Use selection and nested selection in Python.	loop in Python. Perform validation checks on data entry. Design programs using pseudocode.	a procedure and a function. Describe scope of variables. Use functions and procedures as part of the structured approach to programming. Test a program for robustness.	interface' (GUI). Perform string handling operations. Describe the differences between a list and an array. Manipulate a list. Work with 2D lists.	search. Explain the key algorithms for a bubble, merge, and insertion sort.
Declarative What should they know?	<ul> <li>Define the terms decomposition, abstraction and algorithmic thinking.</li> <li>Describe the difference between algorithms and computer programs.</li> <li>Algorithms are represented as written description, flowcharts and code.</li> <li>Describe what a high-level / low-level language are.</li> <li>Explain the need for translators.</li> <li>Describe characteristics of compilers &amp; interpreters</li> <li>The tools of an IDE</li> <li>What is meant by syntax and logic errors?</li> <li>Variable declaration, assignment, and initialisation</li> <li>Runtime errors</li> <li>Data types and their function</li> </ul>	<ul> <li>flowchart symbols and describe how to use them (start, end, input, output, subroutine)</li> <li>Algorithms can be translated from flowcharts – program and vice versa.</li> <li>How to generate random numbers in python</li> <li>Explain the purpose of pre-built libraries and functions.</li> <li>Describe what is meant by true random.</li> <li>Define a condition as an expression that can be evaluated as True or False</li> <li>Interpret a flow chart that includes conditions.</li> <li>Syntax relevant to and regarding selection.</li> </ul>	<ul> <li>Define iteration as a group of instructions that are repeatedly executed.</li> <li>Define a while loop.</li> <li>Define a for loop.</li> <li>State the purpose of a trace table.</li> <li>Describe the purpose of pseudocode.</li> <li>Design a program using pseudocode.</li> </ul>	<ul> <li>Describe a subroutine.</li> <li>Describe the purpose of parameters in subroutines.</li> <li>Describe how subroutines are used for decomposition.</li> <li>List the advantages of subroutines.</li> <li>Describe a procedure.</li> <li>Describe what a function does (input, process, output)</li> <li>Explain the differences between a function and a procedure.</li> <li>Describe scope of variables</li> <li>Describe the purpose of a constant in programming.</li> <li>Describe the structured approach to programming.</li> <li>List the advantages of the structured approach.</li> </ul>	<ul> <li>Define the term GUI.</li> <li>Describe the function of string operators.</li> <li>Define the term ASCII.</li> <li>Define a data structure.</li> <li>Define an array.</li> <li>Define a list.</li> <li>Describe differences between lists and arrays.</li> <li>Describe a 2D list.</li> <li>Design a program that uses 2D lists.</li> </ul>	<ul> <li>Trace tables walk through code that contains loops, selection and lists.</li> <li>Define the searching problem (finding the position of an item in a list of items.</li> <li>Describe linear search.</li> <li>Describe binary search.</li> <li>The difference between both search algorithms.</li> </ul>



	Validation checks			Describe iterative testing.		
	Flowchart symbols			Describe types of testing		
				(boundary, erroneous,		
				normal)		
				Design a program that		
				builds on skills developed		
Procedural What should they be able to do?	<ul> <li>Apply decomposition, abstraction and algorithmic thinking to solve a problem.</li> <li>Recognise scenarios where computational thinking techniques are applied.</li> <li>Analyse and create flowcharts using flowchart symbols.</li> <li>Use an IDE to write and execute a Python program.</li> <li>Perform a code walkthrough of a sequence.</li> <li>Arrange program statements in a sequence.</li> <li>Call subroutines (procedures) in a program.</li> <li>Locate and correct common syntax errors.</li> <li>Use wanables in programs.</li> <li>Use variables in programs.</li> <li>Use appropriate naming conventions.</li> <li>Obtain input from the keyboard.</li> <li>Recognise the input/output states of the input/output states of the input function.</li> </ul>	<ul> <li>Use Python documentation to inform correct use of syntax.</li> <li>Use commenting appropriately within a program.</li> <li>Use random in a program.</li> <li>Import modules into programs.</li> <li>Evaluate arithmetic expressions using rules of operator precedence.</li> <li>Use arithmetic expressions.</li> <li>Evaluate logical expressions.</li> <li>Perform code walkthroughs of selection statements.</li> <li>Evaluate logical expressions that contain Boolean operators.</li> <li>Perform code walkthroughs of nested selection statements.</li> <li>Use nested selection to control the flow of execution in programs.</li> <li>Use binary selection to control the flow of program execution.</li> </ul>	<ul> <li>Perform a code walkthrough of a while loop.</li> <li>Use variables as counters in iterative programs.</li> <li>Use a trace table to track variables in iterative programs.</li> <li>Perform a code walkthrough of a for loop.</li> <li>Use while &amp; for loops to control the flow of program execution.</li> <li>Perform a code walkthrough of iterative validation checks.</li> <li>Translate pseudocode into a program.</li> <li>Identify when to implement iterative statements.</li> <li>Create a program that uses iteration.</li> </ul>	<ul> <li>builds on skills developed</li> <li>Perform a walkthrough of procedures that accept parameters.</li> <li>Use procedures that accept parameters.</li> <li>Perform a walkthrough of function code given the arguments provided during the call.</li> <li>Use functions to return values in programs.</li> <li>Identify global and local variables in programs.</li> <li>Identify when to implement global variables.</li> <li>Use Python naming conventions for constants.</li> <li>Create a program that involves function calls.</li> <li>Use the structured approach in program that builds on skills developed.</li> <li>Use subroutines to control the flow of execution in programs</li> </ul>	<ul> <li>Use string-handling techniques.</li> <li>Traverse a string</li> <li>Access a substring.</li> <li>Perform ASCII conversions.</li> <li>Create a program that uses string-handling techniques.</li> <li>Append elements to a list.</li> <li>Traverse a list</li> <li>Use Python documentation to investigate list operations.</li> <li>Manipulate a list.</li> <li>Create a custom-built function.</li> <li>Use lists to display output on a physical computing device.</li> <li>Use randomisation to append items to a list.</li> <li>Use a 2D list</li> </ul>	<ul> <li>Perform linear search to find the position of an item in a list containing sample data.</li> <li>Perform a binary search to find the position of an item in a list.</li> <li>Interpret and analyse the code for linear search and binary search.</li> <li>Trace code for linear search with input data</li> <li>Identify factors that could influence efficiency of a linear search implementation.</li> </ul>
	<ul> <li>input function.</li> <li>Translate a flowchart into a program sequence.</li> </ul>					



Disciplinary	Interpret	Modules	• Iteration – While/for	Subroutines	String handling	Search Algorithms
1	Instructions	Flowchart	Exception handling	Functions	o Len	o Linear
Literacy	High level Languages	Kandom	Irace lables	Procedures	o isdecimai	O Binary
(Tier 3 Vocab)	Low level languages	Arithmetic expressions	Errors	Constants	o Ord	Sort Algorithms     Bubble
	Iransiators	BIDMAS	Range	Parameters	o Append	o Merge
	Compiler	<ul> <li>Arithmetic operators</li> <li>(add subtrast multiply)</li> </ul>	Validation	Arguments	o Remove	o Insertion
	• Interpreter	(add, subtract, multiply,	• Iry	Decomposition	o Index	Efficiency
		division MOD to the	• Except	Return	o Insert	Midpoint DIV 2
	Editors     Error Diagnostic	nower)	•	Global/Local Variables	о Рор	Traverse
	Ellor Diagnostic	• If		•	o Count	
	Subroutinos	• Flif			o Reverse	
		Flse			o Sort	
	<ul> <li>Sequence</li> <li>Executed</li> </ul>	Comparison Operators			o In	
		o Equal to			String operators	
	<ul> <li>Suptay</li> </ul>	<ul> <li>Not equal to</li> </ul>			Data Structure	
	Identifiers	o Less than			• List	
	Variables	o Greater than			Array	
	Declaration	o LTOET			Append	
	Initialisation	o GTOET			• 1D	
	<ul> <li>Assignment</li> </ul>	Boolean/logical			• 2D	
	<ul> <li>Naming conventions</li> </ul>	operators			•	
	Data types	Nested selection				
	Commenting					
	Randomness and using					
	modules.					
	• Arithmetic operators and					
	expressions					
	Selection (binary)					
	Comparison operators in					
	logical expressions					
	Comparison operators in					
	logical expressions					
	<ul> <li>Boolean operators and overcessions</li> </ul>					
	expressions					
	• Sumg methous					



	Computer	Computer	Computer	Data	Data	Data
	Systems	Systems	Systems	Representation	Representation	Representation
Computer Systems	Describe the role of the CPU. Explain the processes of the fetch-decode-execute cycle	Determine the role of main memory and secondary storage	Construct truth tables for three input logic circuits. Write a program using assembly language (LMC).	Explain how numbers are represented using binary digits & performing operations and conversion between number systems	Explain how text, images, and sound are represented using binary digits.	Measurements of storage and Compression
Declarative What should they know?	<ul> <li>Difference between embedded and general- purpose systems</li> <li>The role of system software as part of a computer system</li> <li>The role of the operating system and utility software</li> <li>The role and purpose of each component of the CPU in computation</li> <li>The basic components of the CPU</li> <li>The role of each part of the CPU as part of FDE Cycle</li> <li>How the FDE Cycle works by describing what happens at each stage</li> <li>The factors that impacts a CPU's performance</li> </ul>	<ul> <li>The characteristics of RAM and ROM</li> <li>The role of main memory as part of a computer system</li> <li>The purpose and role of CACHE in a computer system</li> <li>Why a computer system needs secondary storage.</li> <li>The different types of secondary storage and their functional characteristics</li> <li>The limitations of each secondary storage medium</li> <li>The definition of 'cloud storage' and its characteristics</li> </ul>	<ul> <li>The symbols and truth tables for each logic gate: AND, OR, NOT.</li> <li>Logic gates are used to carry out computation.</li> <li>Truth tables are used to show possible combinations of logic gates/circuits.</li> <li>Boolean expressions used within logic circuits.</li> <li>The difference between high-level and low-level languages</li> <li>The relationship between assembly and machine code</li> <li>Basic commands used within assembly code:         <ul> <li>INP</li> <li>OUT</li> <li>STA</li> <li>LDA</li> <li>ADD</li> <li>SUB</li> <li>BRP</li> </ul> </li> </ul>	<ul> <li>Computers use binary to represent all data and instructions.</li> <li>Binary relates to two-state electrical signals.</li> <li>The difference and digits that are used in both base-2 and base-10 number systems.</li> <li>The conversion table for binary – decimal</li> <li>The rules of binary addition up to three binary numbers</li> <li>The effect of both left and right binary shifts.</li> <li>How overflow errors can occur and what the impact is</li> <li>The characters used to represent Hexadecimal.</li> <li>Why and where hexadecimal notation is used.</li> <li>How to convert from decimal</li> </ul>	<ul> <li>The maximum number of states that can be represented by a binary pattern of a given length.</li> <li>How ASCII is used to represent characters and its limitations</li> <li>What is meant by the term 'character set'?</li> <li>The need for Unicode</li> <li>Unicode uses the same codes as ASCII up to 127.</li> <li>What a pixel is and how they relate to bitmap images</li> <li>The meaning of colour depth and resolution</li> <li>The term Metadata and examples that are applied to bitmaps</li> <li>How to calculate the file size of bitmaps</li> <li>The number of pixels and colour depth affect file sizes.</li> <li>Why analogue sound needs to be converted into binary digits.</li> <li>The concepts of sampling, sample rate, sample resolution</li> </ul>	<ul> <li>The units of data:</li> <li>Bit</li> <li>Nibble</li> <li>Byte</li> <li>Kilobyte</li> <li>Megabyte</li> <li>Gigabyte</li> <li>Terabyte</li> <li>Petabyte</li> <li>How many units of data make up the ensuing measurement of data.</li> <li>What data compression is</li> <li>Why data may be compressed and the different types of compression 'lossy' &amp; 'lossless'.</li> </ul>



Drocodural	Compare embedded and	Describe the	<ul> <li>Design a logical circuit,</li> </ul>	<ul> <li>Explain how binary relates</li> </ul>	<ul> <li>The equation/formula for calculating file size of sound files.</li> <li>The effect of that sample rate, duration and sample resolution have on the playback quality and size of sound files.</li> <li>Examples of metadata that are given to sound files</li> <li>Explain how ASCII is used to</li> </ul>	<ul> <li>Define and compare the</li> </ul>
What should they be able to do?	<ul> <li>general purpose computer systems.</li> <li>Describe the role of system software as part of a computer system.</li> <li>Explore the role of the operating system and utility software.</li> <li>Describe the basic components of the CPU.</li> <li>Describe the roles and purpose of each component of the CPU in computation.</li> <li>Explain how the fetch- decode-execute cycle works by describing what happens at each stage.</li> <li>Describe the role of each part of the CPU as part of the fetch-decode-execute cycle.</li> <li>Explore the factors that impact a CPU's performance.</li> <li>Select components to create a computer system.</li> <li>Evaluate a computer's suitability for a given task</li> </ul>	<ul> <li>characteristics of RAM and ROM</li> <li>Explain the role of main memory as part of a computer system.</li> <li>Define cache memory.</li> <li>Describe the role of cache in a computer system.</li> <li>Explain why a computer system needs secondary storage.</li> <li>State the different types of secondary storage and describe their functional characteristics.</li> <li>State how solid-state memory works and describe its characteristics.</li> <li>Apply knowledge of storage devices to compare the three mediums of storage.</li> <li>Apply the knowledge of storage devices to recommend an appropriate device.</li> </ul>	<ul> <li>combining logic gates to solve a problem.</li> <li>Construct truth tables for a three-input logic circuit</li> <li>Write a Boolean expression to describe a logical circuit.</li> <li>Describe how combinations of logic gates can perform mathematical operations.</li> <li>Determine that assembly language has a 1:1 relationship with machine code.</li> <li>Explain the basic commands in the LMC's assembly code: INP, OUT, STA, LDA, ADD, SUB, and BRP</li> <li>Design and write their own program in assembly language</li> </ul>	<ul> <li>to two-state electrical signals.</li> <li>Explain the difference between base-2 and base-10 numbers.</li> <li>Convert between binary and decimal numbers.</li> <li>Count in binary.</li> <li>Perform addition in binary.</li> <li>Perform binary shifts.</li> <li>Describe situations where binary shifts can be used.</li> <li>Explain how overflow errors can occur.</li> <li>Explain why and where hexadecimal notation is used.</li> <li>Explain how numbers are represented using hexadecimal.</li> <li>Convert decimal numbers to and from hexadecimal numbers</li> </ul>	<ul> <li>represent characters and its limitations.</li> <li>Explain what a character set is.</li> <li>Determine the maximum number of states that can be represented by a binary pattern of a given length.</li> <li>Explain the need for Unicode.</li> <li>State that Unicode uses the same codes as ASCII up to 127.</li> <li>Calculate the number of bytes needed to store a piece of text.</li> <li>Describe what a pixel is and how pixels relate to bitmap images.</li> <li>Describe colour depth and resolution.</li> <li>Define 'metadata'</li> <li>Give examples of metadata applied to a bitmap image.</li> <li>Describe how the number of bitmaps.</li> <li>Describe how the number of pixels and colour depth can affect the file size of a bitmap image, using examples.</li> </ul>	<ul> <li>terms 'bit', 'nibble', 'kilobyte', 'megabyte', 'gigabyte', 'terabyte', and 'petabyte'.</li> <li>Convert between units of measurement.</li> <li>Explain what data compression is.</li> <li>Explain why data may be compressed, and that there are different ways to compress data.</li> <li>Define 'lossy compression' and 'lossless compression'</li> </ul>



	•	Describe the limitations of secondary storage. Explain the definition of 'cloud storage' and describe the characteristics of cloud storage.			<ul> <li>Explain why analogue sound data needs to be converted into binary digits.</li> <li>Describe the concepts of sampling, sample rate, and sample resolution.</li> <li>Give examples of metadata applied to sound files.</li> <li>Calculate file size requirements of sound files.</li> <li>Describe the effect of sample rate, duration, and sample resolution on the playback quality and the size of a sound file</li> </ul>	
Disciplinary Literacy (Tier 3 Vocab)	o       General Purpose       o         o       Embedded Systems       o         o       BIOS       o         o       Memory       o         o       Storage       o         o       System Software       o         o       Operating System       o         o       Utility Software       o         o       Utility Software       o         o       Hardware       o         o       CPU       o         o       Von       Neumann         Architecture       o         o       Clock       o         o       Control Unit       o         o       MAR       o         o       MDR       o         o       ACC       o         o       FETCH       DECODE         o       EXECUTE       FDE         o       FDE       o	RAM ROM Cache Main Memory Virtual Memory Volatile Non-Volatile Secondary Storage Solid State Optical Storage Magnetic Storage Clock Cache CPU Core Channels Overhead Overclock	<ul> <li>Boolean</li> <li>AND</li> <li>OR</li> <li>NOT</li> <li>Logic Gate</li> <li>Logic Circuit</li> <li>Assembly Language</li> <li>Abstraction</li> <li>Mnemonics</li> <li>Accumulator</li> <li>Registers</li> <li>Commands:         <ul> <li>INP</li> <li>OUT</li> <li>STA</li> <li>LDA</li> <li>SUB</li> <li>BRP</li> </ul> </li> </ul>	<ul> <li>Decimal</li> <li>Denary</li> <li>Base 10</li> <li>Binary</li> <li>Base-2</li> <li>Binary Shifts</li> <li>Overflow</li> <li>Base-16</li> <li>Hexadecimal</li> <li>Integer</li> </ul>	<ul> <li>Character Set</li> <li>ASCII</li> <li>Unicode</li> <li>Character codes</li> <li>Pixel</li> <li>Bitmap</li> <li>Colour Depth</li> <li>Resolution</li> <li>Metadata</li> <li>Analogue</li> <li>Sample Rate</li> <li>Sample Rate</li> <li>Sample Resolution</li> <li>Duration</li> <li>Quality</li> </ul>	<ul> <li>Bit</li> <li>Nibble</li> <li>Byte</li> <li>Kilobyte</li> <li>Megabyte</li> <li>Gigabyte</li> <li>Terabyte</li> <li>Petabyte</li> <li>Compression</li> <li>Lossy</li> <li>Lossless</li> </ul>