All Saints' Catholic High School

Luceat lux Vestra

Subject: Computer Science GCSE	**Year:** 10

10	**Unit 1- Programming part 1: Sequence**	**Unit 1- Programming part 2: Selection**	**Unit 1- Programming part 3: Iteration**	**Unit 4 – Computer systems**	**Unit 5 – programming Part 4**	**Unit 6 – Data Representation**	**Unit 7 - Algorithms**
Aim of Unit	Understand the key programming concept of sequence. Therefore, this is a unit to establish the base of programming. This unit is designed to establish a good foundation to determine how a computer uses 1s & 0s to make actions happen. Therefore, we need to determine the need for translators. There is the key concept of sequence which leads to variables, and input. The language	Understand the key programming concept of selection by using randomisation in programs. Work with arithmetic and logical expressions. Use selection and nested selection in Python.	Understand the key programming concept of Iteration by using a loop in Python. Perform validation checks on data entry. Design programs using pseudocode.	Understand 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).	Understand the differences between a procedure and a function. Describe the scope of variables. Use functions and procedures as part of the structured approach to programming. Test a program for robustness. Define the term 'graphical user interface' (GUI). Perform string	Explain how numbers, text, images, and sound are represented using binary digits. Perform operations on binary digits. Convert between units of measurement.	Define the terms 'decomposition', 'abstraction', and 'algorithmic thinking'. Use trace tables. Describe a linear and binary search. Explain the key algorithms for a bubble, merge, and insertion sort.

	chosen is Python. Enable pupils to design programs using a flowchart and pseudocode.				handling operations. Describe the differences between a list and an array. Manipulate a list. Work with 2D lists.		
Composite Knowledge *(a task that requires several building blocks or components)*	In this unit students demonstrate their knowledge and skills of programming techniques. Students will develop and refine a complete solution, using sequence that meets the requirements of a substantial problem. They will analyse problems in computational terms to make reasoned judgements about the solutions. These thinking and programming skills that are extremely attractive in the modern workplace	In this unit students demonstrate their knowledge and skills of programming techniques. Students will develop and refine a complete solution using selection, that meets the requirements of a substantial problem. They will analyse problems in computational terms to make reasoned judgements about the solutions.	In this unit students demonstrate their knowledge and skills of programming techniques. Students will develop and refine a complete solution using Iteration, that meets the requirements of a substantial problem. They will analyse problems in computational terms to make reasoned judgements about the solutions.	In this unit, students will gain an understanding and knowledge of how computer systems work. Starting with the building blocks of the microprocessor and logic gates. Students will discover how a computer system works and executes instructions.	This extensive programming unit takes students from being novices in unit 1,2 & 3 to having the confidence to tackle more advanced programming challenges with the introduction of procedures and functions.	The focus is on how data can be represented in many different forms. You can see this happening throughout time, for example, in the use of cave paintings and clay tablets, through to the use of Morse code. Data and instructions in a computer are formed using a series of 1s and 0s. In this unit, students will discover how numbers, letters, images, and sound are represented with	The focus of this unit is on searching and sorting algorithms, though other topics are covered, such as computational thinking, flow charts, and tracing algorithms. Students will have opportunities to analyse, interpret, modify, and implement a range of algorithms.

						1s and 0s. They will also learn about the factors that impact on the quality of those representations, such as bit depth. Finally, students will be introduced to the concept of compression and discover how to perform run length encoding and Huffman coding as forms of lossless compression	
Component Knowledge *(the building blocks that together, when known, allow successful performance of a complex task)*	Compare how humans and computers interpret instructions. Explain the differences between high- and low-level programming languages. Describe why translators are necessary. List the differences, benefits, and drawbacks of using a	Define a condition as an expression that can be evaluated to either True or False. Identify flowchart symbols and describe how to use them (decision) Identify that selection uses conditions to control the flow of execution.	Define iteration as a group of instructions that are repeatedly executed. Modify a program to incorporate a while loop. Use a trace table to walkthrough code that uses a while loop. Use a trace table to detect and correct errors in	Compare embedded and general-purpose computer systems. Describe the role of system software as part of a computer system. Explore the role of the operating system and utility software. Describe the	Describe a subroutine Describe the purpose of parameters in subroutines Use procedures that accept arguments through parameters. Describe how subroutines are used for decomposition.	Give examples of the use of representation. Explain that computers use binary to represent all data and instructions. Explain how binary relates to two-state electrical signals. Give examples of the	Define the terms decomposition, abstraction, and algorithmic thinking. Recognise scenarios where each of these computational thinking techniques is applied. Apply decomposition, abstraction, and

	compiler, or an interpreter. Describe the tools an IDE provides (editors, error diagnostics, runtime environment, translators) Use subroutines in programs. Define a sequence as instructions performed in order, with each executed in turn. Predict the outcome of a sequence and modify it. Interpret error messages and define error types and identify them in programs (logic, syntax). Use meaningful identifiers. Determine the need for variables. Distinguish between declaration, initialisation, and assignment of variables.	Walkthrough code that includes selection (if, elif, else).	programs. Define a 'for' loop. Walk through code that uses a 'for' loop. Modify a program that uses a 'for' loop. Compare a 'while' loop and a for loop.	basic components of the CPU. Describe the roles and purpose of each component of the CPU in computation. Explain how the fetch-decode-execute cycle works by describing what happens at each stage. Describe the role of each part of the CPU as part of the fetch-decode-execute cycle. Describe the characteristics of RAM and ROM. Explain the role of main memory as part of a computer system. Define cache memory. Describe the role of cache in a computer	List the advantages of subroutines. Explain the difference between a function and a procedure. Use trace tables to investigate functions. Use functions to return values in programs. Describe the scope of variables. Describe how parameters can reduce the need for global variables. Identify when to use global variables. Describe a constant.	use of representation. Explain that computers use binary to represent all data and instructions. Explain how binary relates to two-state electrical signals. Perform binary shifts. Describe situations where binary shifts can be used. Explain how overflow errors can occur. Explain how underflow occurs. Explain why and where hexadecimal notation is used. Explain how numbers are represented using hexadecimal notation. Convert decimal	algorithmic thinking to help solve a problem. Describe the difference between algorithms and computer programs. Identify algorithms that are defined as written descriptions, flow charts, and code. Analyse and create flow charts using the flow chart symbols. Use a trace table to walk through code that contains a while loop, a for loop, and a list of items. Use a trace table to detect and correct errors in a program. Identify why computers often need to search

	Demonstrate appropriate use of naming conventions.			system. Explain why a computer system needs secondary storage. State the different types of secondary storage and describe their functional characteristics. State how solid-state memory works and describe its characteristics.		numbers to and from hexadecimal numbers. Explain how ASCII is used to represent characters, and its limitations. Explain what a character set is. Describe how character codes are commonly grouped and run in sequence within encoding tables. Describe what a pixel is and how pixels relate to bitmap images. Describe colour depth and resolution. Define 'metadata'. Calculate the file size of bitmaps. Describe how the number of pixels and colour depth can affect the	data. Describe how linear search is used for finding the position of an item in a list of items. Perform a linear search to find the position of an item in a list. Describe how binary search is used for finding the position of an item in a list of items. Perform a binary search to find the position of an item in a list. Identify scenarios when a binary search can and cannot be carried out. Identify why computers often need to sort data. Traverse a list of items, swapping the items that are out of order.

						file size of a bitmap image, using examples. Calculate file size requirements of sound files. Describe the effect of sample rate, duration, and sample resolution on the playback quality and the size of a sound file. Give examples of metadata applied to sound files. Define the terms ‘bit’, ‘nibble’, ‘kilobyte’, ‘megabyte’, ‘gigabyte’, ‘terabyte’, and ‘petabyte’. Compare these terms and convert between units of measurement. Explain what	Perform a bubble sort to order a list containing sample data. Merge two ordered lists of items into a new ordered list Describe how merge sort is used for ordering a list of items. Perform a merge sort to order a list containing sample data. Insert an item into an ordered list of items. Describe how insertion sort is used for ordering a list of items. Perform an insertion sort to order a list containing sample data.

						data compression is. Explain why data may be compressed, and that there are different ways to compress data. Define 'lossy compression' and 'lossless compression'.	
Rationale *(why?):* Links to prior & future learning	This creates thinking and programming skills that are extremely attractive in the modern workplace. It builds on the programming concepts from KS2 & KS3: sequence, selection and iteration. The langauge is a Text one ie Python (used in KS3). However, they have experience of a visual language i.e. Scratch in KS2. The concepts are the same no not matter	This creates thinking and programming skills that are extremely attractive in the modern workplace. It builds on the programming concepts from KS2 & KS3: sequence, selection and iteration. The langauge is a Text one ie Python (used in KS3). However, they have experience of a visual language i.e. scratch in KS2.	This creates thinking and programming skills that are extremely attractive in the modern workplace. It builds on the programming concepts from KS2 & KS3: sequence, selection and iteration. The langauge is a Text one ie Python (used in KS3). However, they	Builds on the ideas from KS2 & 3 that a computer is input, process and then output. It can be any device if this concept is adhered to. It is the bedrock for A level computing and for a wider grasp of how the technology is pervasive in society today.	This creates thinking and programming skills that are extremely attractive in the modern workplace. It builds on the programming concepts from KS2 & KS3: sequence, selection and iteration. The langauge is a Text one ie Python (used in KS3). However,	Uses calculation skills and number bases from KS2 & KS3. Introduces them to different counting systems and helps develop their appreciation of number work. They will learn to handle extremely large number, like trillions and calculate in them. This is important because	Helps to develop their capability, creativity, and knowledge in computing and apply their analytic, problem-solving, design, and computational thinking skills. This is vital for further study at S35 where problem solving and learning from mistakes is vital for success.

	what language and therefore, the skills are transferable. These skills and concepts extend into KS5 and employment and apprenticeships.	The concepts are the same no not matter what language and therefore, the skills are transferable. These skills and concpets extend into KS5 and employment and apprenticeships.	have experience of a visual language i.e. scratch in KS2. The concepts are the same no matter what language and therefore, the skills are transferable. These skills and concpets extend into KS5 and employment and apprenticeships.		they have experience of a visual language i.e. Scratch in KS2. The concepts are the same no not matter what language and therefore, the skills are transferable. These skills and concepts extend into KS5 and employment and apprenticeships.	everything they do in society is based around these large numbers. For example, downloading data and data usage for mobile phones etc.	
Assessment Task	A series of individual exam questions on each building block of the unit with a final summative assessment to be used at the end of the unit.	A series of individual exam questions on each building block of the unit with a final summative assessment to be used at the end of the unit.	A series of individual exam questions on each building block of the unit with a final summative assessment to be used at the end of the unit.	A series of individual exam questions on each building block of the unit with a final summative assessment to be used at the end of the unit.	A series of individual exam questions on each building block of the unit with a final summative assessment to be used at the end of the unit.	A series of individual exam questions on each building block of the unit with a final summative assessment to be used at the end of the unit.	A series of individual exam questions on each building block of the unit with a final summative assessment to be used at the end of the unit.
Enrichment	Sign up to a python programming course with an industry recognised qualification.	Sign up to a python programming course with an industry	Sign up to a python programming course with an industry	Building a computer from the component parts. They will develop	Sign up to a python programming course with an industry	Visit to Across to see their immersive, interactive video wall.	Use raspberry Pi to control a robot.

		recognised qualification.	recognised qualification.	employable skills.	recognised qualification.		