Computer Science Department - Year 8

Shirley High Curriculum Map	*The Novice Computer Scientist will study the design, development and analysis of software and hardware used to solve problems in a variety of business and technical contexts, and will practise problem solving, an essential skill for life.*		
	Autumn 1 - Autumn 2	**Spring 1 - Spring 2**	**Summer 1 - Summer 2**
	Theme/Topic/Skill: Data Representation (Binary) & Logic Gates	Theme/Topic/Skill: Computational Thinking	Theme/Topic/Skill: Introduction to Python Programming
Why now?	To build upon knowledge gained in Y7 on data representation and binary arithmetic. The learners will focus on conversions between binary and other arithmetic systems such as Denary/Decimal and Hexadecimal, and the use of arithmetic operators. The learners will link the representation of binary data to a binary logic circuit. They will undertake examples where they will build Truth Tables based on simple problems and represent them using Logic Gates.	To build up on computational abstractions from Y7 and develop the concept of computational thinking and how the 4 elements (Decomposition, Abstraction, Pattern Recognition and Algorithm) work to help produce a solution to a problem. To undertake examples of Decomposition, Abstraction, Pattern Recognition and Algorithms. The learners will consider how a program can be written and the steps that come before programming.	The learners will build on their knowledge from Y7 on the Micro:bits program. This unit will enable the learners to consider planning as part of computational thinking and follow the directions set by the 4 elements to develop source codes. The learners will learn how to write programs in Python and follow procedures to carry out specific tasks, tackle challenges and solve simple problems with different data types and identify and correct syntax errors.
Fundamental Concepts	Data representation (images) Mathematical operators (addition, subtraction, multiplication) Representation of the Alphabet (ASCII) Data levels (0 and 1, True and False) Mathematical conversions (Decimal to Binary to Hex)	Identify by examples decomposition, abstraction, pattern recognition, and algorithms. Use game programs to simulate and show how the outline of real source code can be represented (Pseudocode).	Syntax of Source Code Procedures Data Types Selection Statements Arithmetic Operators Identifying Syntax Errors
Students will learn	o That computers can only understand and transfer data as binary numbers. o How to convert simple denary numbers to binary numbers. o How to convert characters (? # etc) and text into and from binary. o The three main Logic Gates, (NOT, AND, OR) and how they affect binary inputs.	o The basic concept of decomposition and being able to break down a large problem into small, simple instructions. o The basic concept of abstraction and being able to use abstraction to remove specific detail that is not relevant to a solution. o To spot patterns of repetition in sets of instructions. o What an algorithm is and how it is helpful when devising solutions to problems. o How to produce a flowchart using most of the symbols correctly in order to show a proposed solution to a problem. o What pseudocode is and be able to write some simple pseudocode to present their instructions	o To create a simple program using the Python programming language and to include a sequence of instructions, inputs from the user, variables and data types. o How to create a simple program using variables, different data types, inputs, some selection statements and arithmetic operators. o How to find and correct syntax errors in a program.
Language for Life (Key terms /Vocabulary)	o Binary o Hexadecimal o Binary addition o Truth table o Logic Gates	o Pseudocode Abstraction o Decomposition o Pattern Recognition o Algorithm	o Variables o Data Types o Inputs o Arithmetic Operators o Syntax errors o Selection
Extended writing Opportunities	Learners will be able to write a report based on research about computers.	Learners will be able to use the 4 elements of Computational Thinking to describe how a simple program can be created and implemented.	
Maths Across the Curriculum	Binary system Mathematical statements Conditional statements Boolean Logic	Formulate algorithms Mathematical thinking Problem solving	Functions Arithmetic Operators Forming equations
Links to careers/ aspirations	Assistant Computer Scientist	Programmer	Programmers Developer
Cultural Capital	Understand that numbers are represented using the binary system and that the computers use logic gates to perform some functions.	Apply appropriate techniques to break down complicated functions in various stages so that they may be used to develop programs. Ethical aspects must be considered in the development of programs.	Learners will appreciate that Python can be used to carry out simple tasks. Considering ethical issues.
Practical Application of Skills	Assessment	Assessment	Create a program applying the main components