

Y9 Long Term Plan

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Unit	Computer Basics	Data representation in binary	Computational Logic	Computational Thinking and Algorithm	Intro to Programming (Variable, Data Type and IF Function)	Intro to Programming (Menu, Loops, List and File)
Driving Question	How the main component inside the computer works?	How do numbers, characters, image and sound into represented in binary?	Why do computers use binary? Understand logic gates and their relationship to electrical circuits	Why do app developer need to plan before creating application?	What are software application made up of?	How to create a robust program
STEM Partnership / Career link	University of Manchester	University of Manchester Sharp Project SANS Institute	MOSI STEM Ambassadors	University of Manchester MOSI CAS	University of Manchester Digital Advantages CAS	University of Manchester Digital Advantages CAS
Topics covered	<p>Topic:</p> <ul style="list-style-type: none"> Input, output, storage and communication devices Inside a Computer system Characteristics of CPU Main Memory <p>Key aims:</p> <ul style="list-style-type: none"> Be able to identify and describe key components that makes up computer system Understands the von Neumann architecture <p>Wows:</p> <ul style="list-style-type: none"> Build a computer system for a particular purpose 	<p>Key aims:</p> <ul style="list-style-type: none"> Be able to convert numbers to 8-bit binary Be able to convert binary Hexadecimal Be able to add two 8-bits binary digits together Be able to convert images, characters and sound to binary <p>Wows:</p> <ul style="list-style-type: none"> Be able to represents anything into binary 	<p>Key aims:</p> <ul style="list-style-type: none"> Understand how to draw circuit diagram for AND, OR and NOT Gate Be able to create a truth table for each gate Be able to combine more than one gate and write the Boolean expression for the gates. Be able to create a truth table for a logic circuit combining two or more gates together <p>Wows:</p> <ul style="list-style-type: none"> Using online logic Labs to create logic circuit 	<p>Topic: Introduction to Algorithm</p> <p>Key aims:</p> <ul style="list-style-type: none"> Understands what Computational Thinking is Understand what is Algorithm Know what abstraction is and how it is used in computational thinking Know what decomposition is and how it is used in computational thinking Be able to read and create Flowchart Be able to read, correct and produced Pseudocode <p>Wows:</p> <p>Use Computational Thinking to creativity to understand and change the world.</p>	<p>Topic: Introduction to Programming</p> <p>Key aims:</p> <ul style="list-style-type: none"> Be able to create a program that allow Input and Output Use Variable and Data type in a program Use IF Function in a program <p>Wows:</p> <ul style="list-style-type: none"> Create a simple Artificial Intelligence 	<p>Topic: Python Programming</p> <p>Key aims:</p> <ul style="list-style-type: none"> Use IF and ELIF Menu in a program Iteration in program Use List or Arrays in a Program Be able to read and write to a file. <p>Wows:</p> <p>Create a simple Apps</p>
National Curriculum Coverage	Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	Understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits	Understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]	Understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem Create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability	Use logical reasoning to compare the utility of alternative algorithms for the same problem Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions	Use logical reasoning to compare the utility of alternative algorithms for the same problem Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
Big idea development	Different hardware components are connected together to perform multiple task.	Everything in the computer is represented in binary (0s and 1s)	The digital computer uses an arrangement of tiny electronic switches (transistors) called logic gates connected together.	How can ROBUST programs be created that are defensive against hacks and attacks? How can this be implemented in the design stage of the program development?	How can ROBUST programs be created that are defensive against hacks and attacks? How can this be implemented in the design stage of the program development?	How can ROBUST programs be created that are defensive against hacks and attacks? How can this be implemented in the design stage of the program development?
ICT Working Focus	Understanding Hardware and Software Digital Workbook	Digital Workbook	Digital Workbook Online simulators (Logic Lab)	Digital Workbook Online simulators (Draw.IO) Flowgorithm	Calculating means Percentages Understanding python programming syntaxes	Calculating means Percentages Understanding python programming syntaxes
Literacy and communication focus	Paragraphs Evaluating and reflecting on the work completed	Remembering Evaluating and reflecting on the work completed	Comparison such AND, OR and NOT Evaluating and reflecting on the work completed	Planning an argument? Evaluating and reflecting on the work completed	Evaluating and reflecting on the work completed	Evaluating and reflecting on the work completed Spelling correctly on code comments
Numeracy focus	LMC instruction set (Numeric Values) Researching different prices of hardware components	Converting binary to number (Division, Multiplication, Addition, Subtractions)	Working with Truth Tables	n/a	Logical reasoning when creating simple programs	Creating simple Maths Quiz program
SMSC development	Cost of hardware Effect of Changing Environmental Factors on disposing hardware.	Working with each other Respecting peers	Working with each other Respecting peers	Working with each other Respecting peers	Creating a simple artificial intelligence Ethical and Moral consideration when creating simple application	Social and cost simple application software Creating a simple artificial intelligence Ethical and Moral consideration when creating simple application