



The GCSE curriculum has been developed to instil a solid foundation in both the practical programming aspects of computer science, and the conceptual theory aspects too. Students are given ample opportunity for practical work both individually and in pairs to maximise their exposure to writing and debugging code, giving students the skills to further their coding ability through further study or individually.

Half Term 1:	Half Term 2	Half Term 3
<p>Strand A - Programming part 1: Sequence: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - Write and compile simple Python programs - Understand the programming construct of sequence - Write programs which take an input and result in an output considering data types - Use arithmetic expressions in code <p>Strand B – Computer Systems: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - The components which make up a computer and the parts of CPU's - The function and purpose of the FDE cycle - The different types and purposes of memory within a computer system <p>How this will be assessed:</p> <p>Students will take a summative assessment analysing and writing code snippets at the end of the unit and a test on computer systems.</p>	<p>Strand A - Programming part 2: Selection: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - How selection works within programming and its syntax within Python - How to interpret and create logical expressions to direct the execution of code - How to use nested selection for more complex programs <p>Strand B – Computer Systems: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - How to judge and compare computer system characteristics - The AND, OR, NOT, XOR logic gates and use them in circuits and Boolean expressions - How to interpret and write basic assembly language programs <p>How this will be assessed:</p> <p>Students will take a summative assessment analysing and writing code snippets at the end of the unit and a test on computer systems.</p>	<p>Strand A - Programming part 3: Iteration: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - Definite and in-definite iteration and how to use it in Python - What is meant by pseudocode and be able to design programs using it - How to conduct validation on inputs to programs. <p>Strand B – Algorithms part 1 + Data representation: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - Understand the terms algorithmic thinking, decomposition, and abstraction - How to construct and use trace tables to debug programs - How base 2 numbers (binary numbers) are used in computing systems <p>How this will be assessed:</p> <p>Students will take a summative assessment analysing and writing code snippets at the end of the unit and a test on data representation.</p>

<p>Half Term 4</p> <p>Strand A - Programming part 4: Subroutines: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - How to interpret and create subroutines in Python and pseudocode - The difference between functions and procedures - How scope works in Python - The best practice for structuring programs <p>Strand B – Data Representations: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - Understand how to add binary numbers together - Understand how to conduct a binary shift - How binary numbers can be used to represent signed numbers - What hexadecimal numbers are and how to convert them between binary/decimal <p>How this will be assessed:</p> <p>Students will take a summative assessment analysing and writing code snippets at the end of the unit and a test on binary and hexadecimal.</p>	<p>Half Term 5</p> <p>Strand A - Programming part 5: Strings and Lists: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - How to handle and manipulate strings within programs - The concept of the array data structure and how to use lists in Python - Use 2D lists to solve more complex challenges <p>Strand B – Data Representations: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - How text is stored in computers including ASCII and Unicode - How images are represented and how to calculate file size - How sound is represented and how to calculate file size <p>How this will be assessed:</p> <p>Students will take a summative assessment analysing and writing code snippets at the end of the unit and a test on data representation.</p>	<p>Half Term 6</p> <p>Strand A – Algorithms Part 2: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - How to conduct and compare the linear and binary search algorithms - How to conduct and compare the bubble and merge sort algorithms <p>Strand B – Data Representations: STUDENTS MUST KNOW:</p> <ul style="list-style-type: none"> - The difference between lossy and lossless compression - How to compress data using run length encoding - How to compress data using Huffman coding <p>How this will be assessed:</p> <p>Students will take a summative assessment on algorithms and data representation.</p>
<p>Embedding this knowledge can be supported at home by cross curricular experiences as well as developing computational thinking skills by use of program such as IDLE, Visual studio code. Codeacademy and code.org are also fun and challenging for anyone wanting to develop their programming skills</p>		