Computing: we aim to ...

"Everybody should learn to program a computer, because it teaches you how to think."
— Steve Jobs

YEAR 13

- I will analyse, design, develop, test, evaluate, and document a program written in a suitable language. I will apply principles of computational thinking to a practical problem. I will apply appropriate principles from an agile development approach.
- I can describe a range of data types including performing basic operations on data. I understand data structure, and can use Boolean logic and expressions.
- I will understand and evaluate computing related legislation, and explore a wide range of moral and ethical issues around computing.
- I will be able to describe a variety of programming techniques and tools, as well as the features that make a problem computationally solvable.

YEAR 12

- I can describe and compare different types of processor, register, storage, software, development techniques, and types of programming language
- I will explain compression, encryption, and hashing, and describe features of the technology used to build connected applications.
- I will be familiar with the elements of computational thinking and their application to problem solving and programming
- I will be able to examine, analyse, and design algorithms for a give situation, as well as describing the efficiency of algorithms and knowing standard algorithms.

YEAR 11

- I will undertake a practical programming task in a high-level text based programming language (usually Python), showing that I can design, write, test, and refine.
- I will describe and apply the principles of defensive design and testing of code. I will explain different types of programming language, and use Boolean logic.
- I will formalise my understanding of operating system software and its functions. I will discuss and evaluate the impacts of digital technology on wider society, and relevant legislation.

YEAR 10

- I will learn about the architecture of computer systems, including the operation of the CPU, the purpose of the different types of memory and storage, and how computers store data.
- I will be able to explain how computer networks operate, including describing the different technologies and protocols which underpin them, as well as how networks are secured.
- I will explore more advanced algorithms, how they are described in English and in code/pseudocode, and apply the principles of computational thinking to the problems these algorithms solve

YEAR 9

- I can discuss the latest developments in the field of computer science
- I can apply computing skills and knowledge to a range of business applications
- I will apply my coding skills to create a simple game from scratch using text-based code
- I can produce code for an embedded system (micro:bit) and explain how the code operates

YEAR 8

- I can describe the development of video game systems over time
- I can create sprites and graphics to use in my own video game
- I can use a block based coding system to create a functional video game
- I can evaluate my game and suggest improvements for subsequent versions

YEAR 7

- I can write some code, which runs as expected.
- I have created a spreadsheet. I have explored the hardware and software in common computer systems.
- I can use a flowchart to represent a series of steps that a system will carry out.
- I can log on to the school computer system and Google Classroom. I know how to stay safe online.

Wider Concepts

Coding:
Understanding how to teach computers to be useful.

Computers:
Understanding how computers operate and communicate.

The Online World:
Understanding the personal and business uses of computers, the hazards of the online world, and the latest developments in Computer Science.

Key Terms and Concepts

Computational Thinking
Boolean Logic
Pseudocode
Abstraction
Selection
Iteration
Variables and Data Types
Algorithms
Input and Output
Flowchart
Data Structures
Functions and Procedures
Debugging
Binary Representation
CPU
Memory (RAM and ROM)
Operating System
Networks
Internet
Data Security
Encryption
Cybersecurity
Online Safety

Object-Oriented Programming (OOP)
Recursion
Dynamic Data Structures
Complexity Analysis (Big O Notation)
Databases
SQL
Regular Expressions
Artificial Intelligence (AI)
Machine Learning Algorithms
Data Warehousing
Compiler Design
Operating System Principles
Turing Machine
Digital Logic and Gates
Low-Level Programming (Assembly Language)
Cryptographic Protocols
Concurrency
Parallel Computing
Ethical and Legal Aspects of Computing
System Architecture
Microarchitecture