Computing Year 9	**Curriculum intent:** The year 9 curriculum builds on the achievements of year 8 and consolidates that understanding through the use of interleaving the key concepts listed below and tasks to scaffold the students' learning experiences. As in year 7 and 8, new concepts and techniques are acquired through a combination of teacher led tasks, class discussions and practical activities. These are designed to increase the student understanding of the concepts, allowing them to apply their new found understanding in a variety of different scenarios related to the concepts being taught in an engaging way. This approach not only allows for co-ordinated efforts of students to be encouraged but also makes use of numerous open ended tasks that allow all students to stretch their understanding of these concepts as far as they are able.

Topic	Computational Thinking	Databases	MicroBit - Robotics	Gamemaker	Spreadsheets	Animation
Interleaving	Key knowledge from previously studied topics	Key knowledge from previously studied topics	Key knowledge from previously studied topics	Key knowledge from previously studied topics	Key knowledge from previously studied topics	Key knowledge from previously studied topics
Knowledge	Students will study computer architecture and use of binary, input and output covered in previous learning and the Fetch-Decode-Execute cycle through practical activities. Binary to decimal conversion and how text characters are represented using the ASCII code.	Students will demonstrate knowledge of the hardware and software components that make up computer systems, and how they communicate with one another and with other systems.	Students will study and extend their learning of MicroBit programming from year 7 and 8 to include understanding and applying these concepts they have leant to the topic of robotics.	Students will study, design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.	Students will study, design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.	Students will undertake a creative project that involves selecting, using, and combining multiple applications achieve challenging goals, including collecting and analysing data and meeting the needs of known users.
Understanding	Students will be able to: 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; be able to convert between binary and decimal, and perform simple binary arithmetic	Students will be able to: create a database table using several fields with different data types. State the purpose of a primary key in a database. Query the database using more than one criterion to find answers to user queries. Create a basic report with suitable headings. Create the relationship between two linked tables	Students will be able to: demonstrate understanding of the topic through the creation of a number of practical projects both individually and in groups covering topics such as LEDS, music, radio, sensors and Bluetooth.	Students will be able to: 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. 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 and arrays); design and develop modular programs that use procedures or functions	Students will be able to: explain what is meant by a financial model. Explain the advantages of naming cells in a spreadsheet model. Format, construct and manipulate a simple spreadsheet model using formulae. Use conditional functions in calculations. Use conditional formatting. Create a macro & assign it to a button. Customise a chart to present information.	Students will be able to: create a simple animation using simple drawing and frame-by-frame techniques. Explain how frame rate and speed affect the smoothness of the animation. Use multiple layers. Use tweening and frame-by-frame techniques. Add sound effects
Skills	Computational Thinking / Problem Solving / Analysis, Evaluation and Implementation / Technical Vocabulary	Computational Thinking / Problem Solving / Analysis, Evaluation and Implementation / Technical Vocabulary	Computational Thinking / Problem Solving / Analysis, Evaluation and Implementation / Technical Vocabulary	Computational Thinking / Problem Solving / Analysis, Evaluation and Implementation / Technical Vocabulary	Computational Thinking / Problem Solving / Analysis, Evaluation and Implementation / Technical Vocabulary	Computational Thinking / Problem Solving / Analysis, Evaluation and Implementation / Technical Vocabulary
Assessment	End of Unit test	Pupils will create a database suitable to solve a given problem. They will also answer questions about databases and complete a self-assessment.	Pupils will write and run a program and submit the code and screenshots of the program running in an Assessment Portfolio.	Pupils will create a game that can be assessed and answer questions on object oriented programming and include screenshots of the code they used and details of their testing.	Pupils will create an Assessment Portfolio showing their final spreadsheet. They will also answer questions on spreadsheet modelling and complete a self-assessment.	Pupils will create a short animation which will be evidenced in an Assessment Portfolio. They will also answer questions about their animations and complete a self-assessment.