Computing Year 7	**Curriculum intent:** The year 7 curriculum will consolidate and build on the key themes studied at KS2. Students will study aspects of Computational Thinking, Problem Solving, Analysis, Evaluation and Implementation as well as developing their Technical Vocabulary. Opportunities to revisit key concepts through retrieval practice have been built into the curriculum as well as spaced learning and consolidation exercises in order to maximise retention of key knowledge. Knowledge and skills are acquired through the use of carefully planned practical activities with an emphasis on an investigative approach and teamwork. Key skills will be developed with repeated practice. Students will develop understanding of key concepts and will be given the opportunity to demonstrate this in a range of different contexts.					
Topic	**Intro to Computing**	**Kodu**	**MicroBit**	**Hardware**	**Spreadsheets**	**Graphics**
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 a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns.	Students will learn how to undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users. They will make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions, design, use and evaluate computational abstractions that model the state and behaviour of real world problems and physical systems.	Students will learn how to use two or more programming languages, one of which is textual, to solve a variety of computational problems. As part of this they will learn how to make appropriate use of data structures; design and develop modular programs that use procedures and functions	Students will study 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 learn how to design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.	Students will study and understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits. They will learn how to create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability.
Understanding	Students will be able to: use basic file management techniques to create folders, save, copy, move, rename and delete files and folders and make backup copies of files. List some of the dangers and drawbacks of social networking sites. List some possible responses to cyberbullying Describe guidelines for keeping their identity secure on the Internet. Describe what is meant by identity theft and how to minimise the risks of this. Identify a probable phishing email and deal with it appropriately. Describe why the information they find may not be accurate.	Students will be able to: identify what the terms program, navigate, object and world mean in computer games design. Explain that a computer program requires a precise series of instructions to operate. Explain the difference between cloning and creatable techniques and give the advantages of each in terms of ease of program maintenance. Link knowledge and understanding to independently create or modify a game, adding extra depth and complexity by using a range of more advanced game techniques such as power ups, timers etc.	Students will be able to: correctly use different variable types (e.g. integer and floating point), assignment statements, and arithmetic operators. Distinguish between syntax and logic errors and be able to find and correct both types of error. Write an error-free, well-documented program involving selection and iteration. Test and debug their programs, and correct both syntax and logic errors. Make allowances in their programs for user input errors.	Students will be able to: draw a block diagram showing CPU, input, output and storage devices. Name different types of permanent storage device. Explain what RAM and ROM are used for. Show how numbers and text can be represented in binary. Perform simple binary arithmetic. State strengths and weaknesses of different storage devices.	Students will be able to: format a simple spreadsheet model. Use simple formulae and functions. Explain the advantages of naming cells in a spreadsheet model. Format, construct and manipulate a simple spreadsheet model using formulae. Create a macro and assign it to a button on the spreadsheet. Customise a chart to present information effectively. Evaluate the effectiveness of a computer model.	Students will be able to: describe the characteristics of bitmap and vector graphics, state the advantages of each and give examples of situations in which each would be appropriate. Use white space effectively. Use layers in the creation of an artwork. Use the advanced facilities of a graphics package, for example to manipulate, cut out, and alter images.
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	Pupils will sit a multiple choice test as their final assessment.	Pupils will write and run a program and submit the code and screenshots of the program running in an Assessment Portfolio.	Pupils will write and run a program and submit the code and screenshots of the program running in an Assessment Portfolio.	Pupils will sit a multiple choice test as their final assessment.	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 an Assessment Portfolio showing their final spreadsheet. They will also answer questions on spreadsheet modelling and complete a self-assessment.