Year 8 Computer Science	**Curriculum Intent:** Computing consists of three strands: Computational Thinking, Computer Science and Digital Literacy. Within the Computer Science and Computational Thinking streams we will develop the student's knowledge and understanding of what Binary is and how Networks operate. In addition, we will examine the logic behind instructions and how they are processed. Therefore, providing opportunities for students to develop their own software using a text-based programming language.		
	Term 1 **Introduction to Programming** **AI & Machine Learning**	**Term 2** **Physical Computing & Microbits** **G-Develop**	**Term 3** **HTML & CSS** **Networking**
Topic Titles (in order of delivery)	**Algorithms & Introduction to programming** • Computational logic • Flow charts • Pseudo Code • Variables • Input, output, storage • Sequence • Selection • Iteration • Operators • Practical Programming Skills **AI & Machine Learning** • Introduction to AI and Machine Learning • Features and Classification • Training Datasets and Labelling • Algorithmic Bias and Fairness • Model Cards and Evaluation • Practical AI Project (Python/Scratch)	**Physical Computing/Microbits** • Introduction to Micro bit • Built in – Inputs and Outputs • Creating a program • Pair programming • Physical computing project • Refining and refining project **Game Development with G-Develop** • Be able to change the properties of a particle emitter object • Create instances of an object • Alter the orientation of an object • Game Objectives • Characters and Sprites • Collision Detection • Falling and Spawning • Rewards and Hazards	**HTML & CSS** • HTML Structure and Document Layout • Styling Web Pages with CSS • Hyperlinks and Navigation • Media and Layouts • Creating a Web Project **Networks** • Network Protocols • Network Hardware • Wired and Wireless Networks • Network Topologies

Key knowledge / Retrieval topics	• Algorithms are a set of instructions to solve a problem / perform a task. • How to use flowcharts to describe / define an algorithm • Know the main features of a flowchart. • A Variable is a 'named location in memory' the contents of which can change while the program is running. • How to assign values to variables • How to use variables to produce output, store input and perform calculations • Sequence is arranging instructions in a logical order. • Selection is deciding which instructions to execute based on a Boolean decision. • Iteration is the execution of a set of instruction several times based upon a Boolean decision.	**Microbits** • Describe microbit • List the micro bit's input and output devices • write a Python program for the micro: bits Design • Write programs that use the microbit's built-in input and output devices • Communicate with other devices • Write programs that use GPIO pins to generate output and receive input • Write programs that communicate with other devices by sending and receiving messages wirelessly	Network Protocols • Email servers • World Wide Web Network Hardware • Cables • Hub • Routers • Servers Wired and Wireless Networks • Bluetooth • Wifi • 3G • 4G • Wired and Wireless Devices

Understanding / Sequence of delivery	Programming techniques Variables • State what a variable is (A named location in memory that can change whilst the program is running) • Explain what a variable is (to store data while a program is running) • Perform basic mathematical or logical functions on variables Input, output, storage • Explain and show how input may be captures and assigned to a variable for use / storage within a program • Explain and show how to output text or movement on a screen • Be able to use a range of data types including: ▪ Integers ▪ Real numbers (floats) ▪ Text ▪ Boolean ▪ Lists / arrays (one dimensional) or equivalent Sequence • Understand that instructions are executed in the sequence they are written • Write programs with instructions in the correct order	• Identify subroutines • Creating a game • Changing the properties of a particle emitter object • Creating instances of an object • Altering the orientation of an object • Design sprites for a platform game • Be able to create and edit collision masks • Test and refine the playability of a game • The main character sprite falling off the screen • The main character sprite spawning back at the start • The main character reaching the end of the game • Be able to use variables to store a score • Update a score based on an event – such as a coin being collected • Be able to create hazard objects that return a character to the start of the game	• IP Addresses • TCP/IP • Packets Internet Services • Email • VoIP Network Topologies • Client Server Network Model • Peer to Peer Network Model • Bus Topology • Star Topology • Mesh • Ring • What is HTML? State that HTML stands for Hypertext Markup Language and is used to create the structure and content of a web page. • What is CSS? State that CSS stands for Cascading Style Sheets and is used to control the style, layout, and presentation of a web page. • HTML Tags: Identify and explain common tags (such as <html>, <head>, <body> for page structure, <h1> to <h6> for headings, <p> for paragraphs, and <img> for images). • HTML Attributes: Understand how attributes provide extra information about elements (such as href in a hyperlink or src and alt in an image tag). • CSS Selectors, Properties, and Values: Explain how a CSS rule works to target an element and change its appearance (e.g., changing the font color or background size).

			• Hyperlinks: Explain how the anchor tag (<a>) is used to link web pages together to make a working website. • Structure vs. Presentation: Explain the difference between writing the content (HTML) and designing the look of it (CSS).

	• Be able to identify errors in the order of a sequenced set of steps • Know the difference between rule-based programming and machine learning. • Understand how computer models extract features to classify data (text or images). • Know that training datasets must be representative to prevent bias. • Explain how bias in data leads to unfair or inaccurate predictions in real life. • Understand the role of a Model Card in documenting how an AI works and its limitations. • Design, train, and test a custom machine learning model to solve a specific problem	• Physical computing components • Design a physical computing artifact purposefully, keeping in mind the problem at hand, the needs of the audience involved, and the available resources • Decompose the functionality of a physical computing system into simpler features **Physical computing project** • Implement a physical computing project, while following, revising, and refining the project plan	• Computer Networks and Protocols • Network Hardware • Wired and Wireless Network • The Internet • Internet Services • Network Topologies
	Algorithms: • Computational logic Identify the success criteria of a problem • Flow charts		

	• Produce algorithms using flow charts • Use and be familiar with the flow chat shapes ▪ Start / Stop (End) ▪ Process ▪ Input / Output		
Assessments	Assessment on the above content	Assessment on the above content	Assessment on the above content