Computing: SPECIFIC LEARNING INTENTIONS

Know (blue), Understand (orange) and Do (green) | Learning Check (Purple)

YEAR GROUP:	Year 8 – 2024-2025					
	Autumn Term 1	Autumn Term 2	Spring Term 1	Spring Term 2	Summer Term 1	Summer Term 2
	Small Basic Programming	Small Basic Programming	Computer Systems – networks	Computer Systems	Web Design	Web Development
Granular learning intentions/success criteria **(What will pupils know, be able to do and understand)**	**Introduction to programming** • Define what a computer program is and explain its purpose. • Compare block-based programming (e.g., Scratch, MakeCode) to text-based programming (e.g., Small Basic, Python). • describe what an Integrated Development Environment (IDE) is and its purpose in programming. • Identify and explore the key features of the Small Basic IDE (editor, run button, error messages, etc.). • Open, write, and run a simple Small Basic program that uses Textwindow.writeline() **Programming with inputs and outputs** • Define what input and output (I/O) mean in computing. • Identify real-world examples of inputs and outputs in everyday technology. • Create a variable using small basic • Compare TextWindow.Write and TextWindow.WriteLine and TextWindow. Read() • Use TextWindow.Read() and TextWindow.Writeline() • Explain the purpose of a variable within a computer program. • Concatenate (combine) text and variable values to form meaningful output.	**Programming with Iteration** • Define iteration and explain its purpose in programming. • Describe the benefits of using loops instead of repeating code manually. • Predict the outcome of a block of code that uses iteration. • Calculate the number of times a FOR loop will repeat • Write a FOR loop to repeat Turtle movements. • Use a FOR loop to draw shapes more efficiently (e.g., squares, polygons). • Create a program that uses a while loop to repeat the entry of a value. • Compare FOR loops vs. WHILE loops and determine when to use each. **Summative Assessment 1** **Combining programming techniques to create a program for a criteria** • Identify key requirements in a problem • Plan a program by breaking down the problem into smaller steps • Select the best type of iteration (FOR or WHILE loops) to repeat tasks efficiently. • Develop the code in parts • Test/run the code during development.	**Computer Networks** • identify and describe various ways to communicate over the internet, including email, instant messaging, video calls, and social media. • Compare the advantages and disadvantages of different communication methods. • Define what a computer network is and explain its purpose. • Identify the different types of networks, including Local Area Networks (LANs), Wide Area Networks (WANs), and the internet. • Recognise essential network components • Describe the differences between wired and wireless networks. • Identify the advantages and disadvantages of each type of network in terms of speed, security, reliability, and accessibility. • Evaluate which type of network is best suited for different situations **How to make the most of the WWW** • Follow a structured approach to researching a topic effectively. • Identify reliable sources and organise information into a report • identify potential concerns when researching online, such as misinformation, bias, and outdated data. • Evaluate sources for credibility, accuracy, and reliability. • Explain how misleading information can impact	**Malicious Code** • Define what malicious code is and explain how it is designed to harm, disrupt, or exploit computer systems. • Identify different types of malware, including viruses, ransomware, Trojans, and worms. • Describe the different types of malicious code and the impact that they have on computers • Analyse real-world examples of malware attacks and their consequences • Identify strategies to prevent malicious code. **Sway Presentation** • Recall and consolidate knowledge of computer systems by reviewing and presenting key concepts learned throughout the unit • Develop digital literacy skills by using Microsoft Sway as a tool for creating interactive presentations. • Enhance presentation and communication skills by structuring information clearly and effectively in Sway. • Compare different types of presentation software, identifying the advantages and disadvantages of using Sway over traditional tools like PowerPoint. • Apply creativity and multimedia elements by incorporating text, images, videos, and animations.	**Mind maps for design** • Identify what a mind map is and explain how it helps to plan a website. • Describe how a well-structured mind map improves website navigation and user experience. • Explain how organizing content in a mind map helps create a clear website layout. • Create a mind map to organise key website sections (e.g., homepage, about, contact). • Categorise content ideas into branches and subtopics to improve website planning. • Categorise design ideas into branches and subtopics to improve website planning . **Mood boards for design** • Define what a mood board is and explain its purpose in web design. • Describe how mood boards help communicate visual ideas, branding, and website aesthetics. • Identify key elements of a mood board, including color schemes, typography, images, and layout inspiration. • Research and collect relevant design inspiration for a website project • Create a mood board that visually represents the theme, style, and branding of a website to be developed **Web Design Portfolio**	**Creating a website** • Identify and use a range of basic HTML commands to structure a website. • Explain the purpose of CSS in web design and how it is used to style webpages. • Explain the importance of considering the target audience and designing for a given purpose when creating a website. • Use mind maps, mood boards, and visualisation diagrams to plan and guide the website development process • Build a functional website using HTML and CSS, incorporating tables and styling for structure. • Use a colour picker tool to choose a gradient before applying it in CSS. • Change the background colour of a webpage using CSS. • Style links to change colour when hovered over. • Apply a gradient effect to improve website appearance. **Peer feedback on website portfolio** • Re-purpose digital artifacts (e.g., images, text, multimedia) to suit a new theme or target audience. • Apply accessible design principles, such as good colour contrast and alternative text for images • Test and evaluate the website, ensuring usability, functionality, and design effectiveness. • Check that all links work correctly.
		Computer systems		Web Development		

	• Design a simple interactive program that takes multiple user inputs and produces meaningful output. **Programming with selection** • Define selection and explain how it allows a program to make decisions. • Identify real-world examples of selection (e.g., traffic lights, login systems). • Identify the operators for less than, greater than and equal to (<,> and =) • Explain the difference between IF, ELSEIF, and ELSE statements. • Write programs that use IF...ELSE to make decisions based on user input. • Use IF...ELSEIF...ELSE to handle multiple conditions. • Predict the output of selection-based programs by tracing through code. • Modify and debug selection statements to ensure correct functionality. **Create a quiz or decision-making program that uses, inputs, outputs, variables and IF...ELSEIF...ELSE.** **Programming with the turtle** • Identify key Turtle commands (e.g., Turtle.Move, Turtle.Turn). • Write simple programs using the Turtle object to draw basic lines • Compare internal and external angles • Use key operations to draw a square, rectangle, and triangle step by step.	**Hardware** • Define what a computer system is. • Identify different types of computer systems (e.g., desktops, laptops, tablets, embedded systems • Define what hardware is and explain its purpose. • Identify the main components of a computer system (CPU, RAM, storage, motherboard, input/output devices) • Describe the function of key hardware components • Identify the difference between input, output and storage • Categorise components into, input, output or storage. **Software** • Define what software is and explain its purpose. • Differentiate between system software and application software • Define what an operating system (OS) is and why it is needed. • Identify common operating systems (Windows, macOS, Linux, Android, iOS). • Describe the key functions of an OS, including • Define what utility software is and explain its purpose. • Explain how utility software helps maintain and improve computer performance. **Building a computer** • Identify the intended purpose of the computer (e.g., office work, gaming, video editing, programming). • Select appropriate hardware based on performance needs • Locate where each of the components are placed on the motherboard • Visually recognise the key internal components • Recognise that the operating system must be installed before any other software • Identify a range of key utility software and the role they play.	understanding and decision-making. • Define the internet as a global network connecting computers. • Explain the World Wide Web (WWW) as a system of websites that run on the internet. • Apply the CARS method (Credibility, Accuracy, Reliability, Support) to assess online sources. • Create report on fake news, misinformation or information bias **Social Engineering** • Define what social engineering is and explain how it manipulates human psychology to gain unauthorized access to information. • Identify different types of social engineering attacks, including phishing, pharming, shoulder surfing and blagging • Describe how phishing works and recognize the common signs of a phishing attack. • Analyse real-world examples of social engineering attacks and their impact on individuals and organisations. • Identify the risks associated with social engineering, including identity theft, financial loss, and data breaches. • Explain how to protect against social engineering attacks by verifying sources, avoiding suspicious links, education and using strong passwords. **keywords quiz**	**Introduction to web development** • identify and list different uses of websites, including e-commerce, information sharing, social networking, entertainment, and education. • Evaluate two different websites by analysing their design, functionality, user experience, and purpose. Comparing a good design against a bad design. • Explain the role of HTML in web development by explaining what it is used for and how it structures web pages. • Explain the importance of web browsers in displaying HTML content and how they interpret and render websites. • Analyse what makes a website successful **HTML Basics** • identify and describe the purpose of common HTML elements, including <head>, <body>, <b>, <i>, <p>, and <h1>. • Demonstrate how to create a basic web page using the common elements. • Identify what navigation is and use the navigation element • Recognise the element for hyperlinks and create a series of external hyperlinks in HTML. • Write and demonstrate the code used to insert an image (<img> tag) • Explain and demonstrate the use of iframes (<iframe> tag) by embedding content from another webpage, describing how it functions.	**Visualisation diagrams for design** • Define what a visualisation diagram is and explain its purpose in web design. • Identify key elements of a visualisation diagram, such as images, colors, layout, and annotations. • Explain how visualisation diagrams help plan the look and feel of a website. • Create a visualisation diagram to represent the design of 3 webpages. • Justify design choices based on branding, target audience, and accessibility considerations. **Summative Assessment 2**	• Ensure the website is easy to read and navigate. • Ask a peer to test your website and provide feedback.