Computing: SPECIFIC LEARNING INTENTIONS

Know (blue), Understand (orange) and Do (green)

YEAR GROUP:	Year 7 – 2024-2025					
	Autumn Term 1	Autumn Term 2	Spring Term 1	Spring Term 2 / Summer Term 1		Summer Term 2
	School Network	**Make Code Arcade**	**E-Safety & interactive products**	**Binary & Security**		**Spreadsheet Modelling**
Granular learning intentions/success criteria (What will pupils know, be able to do and understand)	• Identify and listen to the rules and expectations of a computing lesson • Memorising username and password and logging on to the computer • Locating key software: email, OneNote, OneDrive • Creating folders to organise file storage • Compose an email using the correct structure **Make Code Arcade** **Introduction to programming** • Identify what a computer program is and give an example of one • Distinguish the difference between inputs and outputs • State what a sprite is • Creating a simple program that responds to inputs **Programming with variables** • Describe what a variable is • Give examples of variables • Recognise variables in a program • Create a program that utilise variables **Create a program that uses, inputs, outputs, variables and sprites** **Programming with selection** • Describe what selection is in computer programming and explain why it is required • Identify the different logical operators • Trace through programs that use selection • Create a program using selection to meet a given criteria • Discuss examples of how selection is used in real world applications	**Summative Assessment 1** **Programming with iteration** • Describe what iteration is in computer programming • Analyse code and identify why one program is more efficient than another • Explain the benefits of using loops/iteration • Use a loop to code a simple animation • Describe how a for loop works • Create a program using a for loop to meet a given criterion • Predict the outcome of a block of code that uses a while loop • Creating code using a while loop that matches a given criteria **Programming with Arrays** • Identify the difference between a variable and array • Analyse a code block that uses an array to determine the different outputs • Create a program for a magic 8 ball that selects a random response from an array **Combining programming concepts** • Complete a make code arcade skills map • Identify the necessary code blocks required to meet a given program criterion. • Combine the concepts of variable, selection and iteration to create a program to meet a criteria.	**Online Dangers** • List a series of threats associated with using technology • Select data that would be safe to include on a social media profile and justify why • Identify appropriate and safe responses when faced with online dangers • Describe what is meant by exploitation **Planning an interactive story** • Describe a key e-safety issue and explain how it can be prevented. • Use the components of a story structure (exposition, rising action, climax, falling action, and denouement) to plan an interactive story on e-safety. • Write content for an interactive story on e-safety, ensuring it is suitable for a given audience. **Create a plan using a story arc and storyboard. Keywords check** **Creating an interactive story** • Describe what interactive means • Analyse an existing interactive story and identify what works well and what needs to improve • Identify the function of a range of PowerPoint tools; shapes, animations, transitions, formatting background, inserting pictures, adding hyperlinks/buttons and looping at kiosk. • Use the tools appropriately to produce a professional looking interactive storybook • Select an appropriate colour scheme to match the theme and purpose of the story. • Program the navigation of the interactive story using a series of buttons and allow users to make decisions within the story.	**Binary and Denary Conversions** • Identify that binary (Base 2) only uses the digits 0 and 1. • Explain why computer systems use binary (e.g., made of electronic switches that are either ON or OFF). • List and understand the 8 place values in an 8-bit binary number (128, 64, 32, 16, 8, 4, 2, 1). • Define and identify what denary (Base 10) is and who uses it (e.g., humans in everyday life). • Create and use a place value table with the 8-bit binary values to convert between binary and denary. • Use the table to convert binary numbers (up to 8 bits) into denary. and to convert denary numbers (0-255) into binary (up to 8 bits). **Binary, Denary and Hexadecimal** • Recognise that hexadecimal (Base 16) uses digits 0-9 and letters A-F (where A = 10, B = 11, C = 12, D = 13, E = 14, F = 15). • Identify the users(programmers) of the hexadecimal • Describe the method use to convert between hexadecimal and binary and back • Use the 8 | 4 | 2 | 1 method to convert between binary and hexadecimal. • Convert denary numbers to binary and hexadecimal using place value. • Explain why programmers prefer hexadecimal over binary (e.g., shorter numbers, easier to read). **Binary, denary and hexadecimal conversions** **Cryptography** • Identify methods that can be used to keep computer data safe • Use historical encryption methods such as the Caesar cipher to encrypt and decrypt messages. • Apply problem-solving skills to break encoded messages • Explain why encryption is essential in protecting information. • Convert encoded messages back into readable text using different techniques (ASCII, Caesar cipher). • Encrypt messages using a range of techniques **Authentication** • Recognise that authentication is the process of verifying a user's identity to grant access to a system or data. • Identify different types of authentication methods, including biometric, physical.. • Describe how common authentication techniques work, such as passwords, fingerprints, two-factor authentication (2FA). • Categorise authentication methods into biometric (e.g., fingerprints, facial recognition), physical (e.g., keycards, security tokens), and knowledge-based (e.g., passwords, PINs, security questions). • Evaluate the effectiveness of authentication methods by considering security, convenience, and vulnerability to attacks. **Image Representation** • Identify that bitmap images are made up of pixels and that pixels are made up of binary numbers, where the binary number represents a colour. • Colour an image by converting binary and denary numbers. **Summative Assessment 2**		• Recognise what a spreadsheet is • Identify key features of spreadsheets, including cells, rows, columns, and formatting options. • Apply formatting techniques (e.g., bold, color coding, borders) to improve spreadsheet readability. • Use simple formulae (e.g., SUM) to perform basic calculations in a spreadsheet. • Apply a range of formatting techniques to enhance the presentation • Evaluate the impact of formatting and formulae • Explain the benefits of using spreadsheets • Compare the advantages of using spreadsheets versus traditional pen-and-paper methods for tasks like event planning. **Formulae and Functions** • Describe the difference between a formula and a function and how each is used in calculations. • Use basic functions such as SUM, AVERAGE, MIN, and MAX to analyse data. • Construct formulas to perform calculations on spreadsheet data. • Explain the difference between a formula and a function and determine when to use each. • Apply advanced formulas such as IF statements **Conditional Formatting** • Recognise that conditional formatting allows data to be visually represented based on specific conditions. • Apply conditional formatting rules • Explain why conditional formatting is useful for data analysis and decision-making. **Graphs and Charts** • Identify different types of graphs (e.g., bar charts, line graphs, pie charts) and their uses. • Describe how to create graphs in a spreadsheet using given data. • Construct appropriate graphs based on data sets, selecting the most effective type of graph for the given information.