

Curriculum Map Year 7: Computer Science

	Topic	Key Knowledge <i>What will students know by the end of this topic?</i>	Key Skills <i>What skills will students have developed by the end of this topic?</i>	Assessment Opportunities <i>How will student progress be measured? What are the key assessment pieces?</i>
Half Term 1	1. Systems and Safety	- Computer room rules & AUP: Rules and safe guidelines for using school computers. - Passwords: Why keeping passwords private matters and what makes a password strong. - Data protection: What personal details are and why they must be kept safe. - Updates & backups: Why software needs updates and why copying data prevents loss. - Online scams: Knowing what phishing, smishing, and vishing mean, and spotting fake emails. - Getting help: Where and how to report digital safety or tech problems.	- Logging on: Getting onto school computers and software like Typing Club on your own. - Managing files: Creating and sorting folders in OneDrive and opening OneNote. - School apps: Using EduLink to find homework and using Microsoft Teams correctly. - Spotting threats: Finding suspicious messages and reporting them safely.	Self-assessment Quizzes End of unit test with student reflection and actions
Half Term 2	Bebras Challenge	- Bebras rules: The layout and rules of the Bebras competition tasks. - Thinking patterns: Basic ideas of breaking down problems, spotting patterns, and designing steps. - Problem solving: Different ways to think through tricky logic puzzles. - Coding links: How puzzle solving connects to writing computer code.	- Logic analysis: Breaking big, unfamiliar problems into smaller pieces. - Pattern spotting: Finding similarities in puzzles and ignoring unneeded details. - Step-by-step thinking: Creating clear steps or rules to solve challenges fast. - Resilience: Staying confident and trying again when tackling hard puzzles.	Challenge scored remotely Certificates
Half Term 3	2. Scratch	- Algorithms: What an algorithm is and how it forms the base of a program. - Coding blocks: The three core building blocks: sequence, selection (choices), and iteration (loops). - Variables & maths: Using variables to store info, plus how random numbers and math blocks work. - Joining text: How the 'join' block glues words or data together. - Game flow: How programs read user answers and use messages (broadcasts) to start or end a game.	- Changing graphics: Picking backdrops and drawing or editing characters (sprites). - Using code blocks: Putting sequences, loops, and choices (if-statements) into code. - Using math & variables: Making and updating variables, random numbers, and math equations in a game. - Making it interactive: Taking user input to check answers and using broadcasts to run the game. - Fixing bugs: Testing code to find and fix mistakes on your own.	Self-assessment Quizzes End of unit test with student reflection and actions
	3. Theory	Inputs & outputs: Identifying and sorting devices like keyboards (input) and screens (output).	Finding hardware: Spotting and naming internal parts like the CPU and RAM.	Self-assessment Quizzes

	Topic	Key Knowledge <i>What will students know by the end of this topic?</i>	Key Skills <i>What skills will students have developed by the end of this topic?</i>	Assessment Opportunities <i>How will student progress be measured? What are the key assessment pieces?</i>
Half Term 4		- Inside the computer: What the CPU, RAM, ROM, motherboard, and fan do. - Storage: Why extra storage matters and the difference between two types of hard drives. - Extra cards: Two types of plugin cards (like graphics or sound) and why we use them. - Software types: What an operating system does, three examples of utility tools, and common apps.	- Sorting devices: Separating hardware into inputs and outputs correctly. - Comparing parts: Explaining differences between memory, storage, and plugin cards. - Sorting software: Telling the difference between system software and everyday apps.	End of unit test with student reflection and actions
Half Term 5	4. App Design	- Breaking down problems: Splitting a large app idea into smaller, doable tasks. - App setup: How to build apps using multiple screens, buttons, and text styles. - Triggers (Events): How user actions like clicks make code run. - If-statements: How selection (choices) controls what an app does. - Project review: How checking success criteria and reflecting helps plan next steps.	- Planning apps: Mapping out app features and parts before typing any code. - Designing layouts: Making new screens and adding, placing, and styling buttons or boxes. - Coding triggers: Writing code that responds to clicks and uses 'if' rules for choices. - Finding errors: Using different testing methods to track down and fix code bugs. - Reviewing work: Writing clear goals, checking progress, and planning what to do next.	Self-assessment Quizzes End of unit test with student reflection and actions
Half Term 6	5. Animation	- Animation basics: How frame-by-frame drawings, frame speeds, and timelines work. - Onion skinning: How seeing faint copies of past drawings helps align the next frame. - Interactive frames: How code stops a drawing from moving on or reacts to clicks. - Organising assets: Using named frames, timeline layers, and reusable clips. - Saving files: How to export finished animation projects to share with others.	- Drawing animations: Making a short video by drawing frame after frame. - Smooth movement: Creating automated shape changes (tweens) to move objects easily. - Layering work: Organizing files with multiple layers, named sections, and animation clips. - Coding interactions: Adding scripts to pause animations or make buttons clickable. - Creating projects: Mixing drawing, motion tools, and code to make a unique animation. - Exporting work: Saving finished projects as video or interactive files.	Self-assessment Verbal feedback End of unit test with student reflection and actions