(Summer) Learning Journey for Computing

Ad Astra

How does this unit link to prior learning?		
The Power of Instructions: This unit builds on your previous experience of following and giving clear instructions. **Logical Thinking:** You will strengthen your computational thinking by sequencing steps accurately. **Problem Solving:** You will practice checking and "debugging" instructions to make sure a task works exactly as you intended.		
What will you be learning about?		
Cracking the Code: You will learn what algorithms are and how to put instructions in the perfect order. **Solving Puzzles:** You will write algorithms to solve complex problems. **Universal Language:** You will learn to express your ideas as pseudocode (halfway between human and computer talk). **Digital Art:** You will apply these ideas in Python using Turtle graphics to create shapes, letters, and patterns.		
We will develop our learning each week by focusing on:	**Skills**	**RAG**
Lesson 1 - Algorithm Architects: What an algorithm is; sequencing and ordering instructions; writing an algorithm for a real-world problem.	**Define "algorithm"; Sequence steps: Write clear instructions for real-world problems.**	
Lesson 2 - Pseudocode Pros: Writing algorithms in structured, plain language; understanding that pseudocode does not require strict syntax.	**Identify key steps; Choose a correct order; Use plain language to solve maze / delivery challenges.**	
Lesson 3 - Turtle Power (Trinket): Set up and run Turtle code; use movement commands to draw simple lines and letters.	**Use an online IDE (Trinket); Import libraries; Use *forward / left / right* commands to draw.**	
Lesson 4 - Loop Legends: Develop drawings by improving accuracy, changing pen settings, and beginning to use repetition (loops) to reduce code.	**Change pen colours / thickness; Use loops (iteration) to make code shorter and smarter.**	
Lesson 5 & 6 - Independent Creator: Independent practice using the assessment booklet tasks; save work with clear filenames; evidence progress.	**Apply debugging strategies independently; Manage files: Reflect on progress.**	

Key vocabulary

The Logic:	Algorithm	Instruction	Sequence	Debug	Branching	Syntax	Command	Problem	Computational thinking
The Code:	Python	Turtle	Import	Library	Variable	Loop	Iteration	Trinket	

How will this help you in the future?	
KS4	**Beyond LHS**
Academic Success: This unit directly supports GCSE Computer Science topics like decomposition, abstraction, and evaluation. **Coding Foundations:** You'll build a strong base in Python, specifically in sequencing, iteration, and debugging	**Game Designer/Developer:** Use code to create movement and animations. **Robotics Engineer:** Practice giving precise instructions to machines. **Digital Artist:** Create intricate patterns and artwork using math and code. **Data Scientist:** Build confidence using Python for