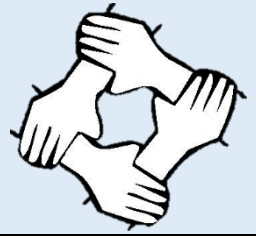
Year 3 Term 3

What does it mean to be strong?

COMPUTING



National Curriculum Links:
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Essential Prior Learning:
Know that programs work by following clear, step-by-step instructions. Know that to find and fix mistakes, you need to check each step carefully. Know that we can predict what will happen before running algorithms using logical thinking. Design, write, implement, and debug more complex algorithms. Use logical reasoning to predict what will happen before running algorithms.

Progression in Skill:
Know that decomposition means breaking complex tasks into smaller, more manageable steps.

Know that debugging uses logical steps to find and fix problems in programs. Know that repetition can be used to reduce repeating code. Know that programs can use various forms of input (e.g. clicking buttons, dragging blocks, pressing keys, entering data, or responding to sensors like sound, light, or tilt). Design, write, implement, and debug increasingly complex computer algorithms that accomplish specific goals by decomposing tasks into smaller steps and using repetition to reduce repeated code. Incorporate different types of input into programs (e.g., clicking buttons, dragging blocks, pressing keys, entering data, or responding to sensor input like sound, light, or tilt).

Long-term Memory Knowledge:
Understand that big tasks can be broken into smaller steps to make them easier. Know how to find and fix problems by checking things carefully. Understand that repeating instructions can make programs simpler. Know that programs can respond to different actions or information. Know how to create and improve instructions to make something work. Know how to include different ways for a program to get information.

Key Vocabulary	
input	information or actions that a program uses, like pressing a key or clicking a button
algorithm	a set of steps or instructions to solve a problem or make something happen
debugging	finding and fixing mistakes in a program
decomposition	breaking a big task into smaller, easier parts
program	a set of instructions that a computer follows to do a task
repetition	doing something again and again, often using a loop in programming

Progression in Resources:
J2e: J2code, micro:bit

Relevance	
Now	Know how to break big tasks into smaller steps to make them easier and know how to create and improve instructions to make something work. Know that repeating instructions can make programs simpler and know that programs can respond to different actions or information (input).
Future	Children have enough understanding of computer systems to identify why programs they are familiar with are not working as they would expect and correct this.
Aspiration	Develop programs that can be used in the wider worlds of industry, gaming, etc.