

## Computing – Programming Computational Thinking

**What I should already know:** I can articulate what decomposition is. I can decompose a game to predict the algorithms used to create it. I know that there are different levels of abstraction. I can explain what an algorithm is. I can follow an algorithm. I can create a clear and precise algorithm. I know that programs execute by following precise instructions. I can incorporate loops within algorithms. I can use decomposition to explain the parts of a laptop computer. I can use decomposition to explore the code behind an animation. I can use repetition in programs. I can use logical reasoning to explain how simple algorithms work. I can explain the purpose of an algorithm. I can form algorithms independently.

### Enquiry Questions

- What are the four key strands that make up computational thinking?
- What is decomposition and how can we apply it to solve problems?
- What do pattern recognition and abstraction mean?
- How can we create an algorithm and what can it be used for?
- Can we combine computational thinking skills to solve a problem?

### Key Vocabulary

**Abstraction** – Identifying the important detail and ignoring irrelevant information.  
**Algorithm design** – Creating a formula or set of instructions to solve the problem.  
**Code (computer)** – A set of instructions written in programming language, to tell a computer what to do.  
**Code blocks** – A visual representation for a section of code that performs a certain job. They can be snapped together to build a program.  
**Computational thinking** – A method of tackling a complex problem, to devise a solution which both computers and humans can understand.  
**Computer** – Electronic machines that accept and process information to produce an output, and then store the results.  
**Decompose** – To break something down into smaller chunks.  
**Pattern recognition** – Identifying similarities and recurrences in data.  
**Problem** – A matter or situation that needs to be resolved.  
**Sequence** – A set order or pattern for something to follow.

### Computing Skills

Pupils will:

#### Computational Thinking

- Use decomposition to solve a problem by finding out what code was used.
- Use decomposition to understand the purpose of a script of code.
- Identify patterns through unplugged activities.
- Use past experiences to help solve new problems.
- Use abstraction to identify the important parts when completing both plugged and unplugged activities.

#### Programming

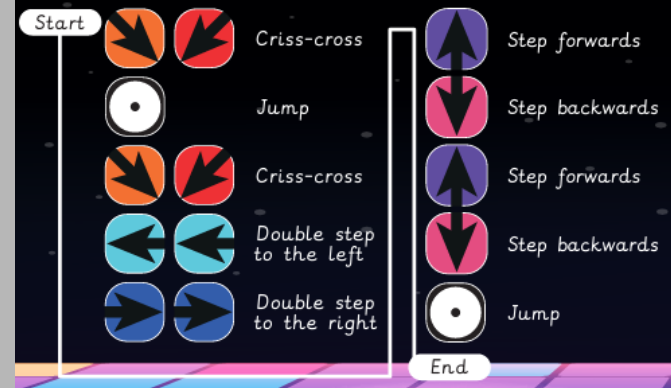
- Create algorithms for a specific purpose.
- Use abstraction and pattern recognition to modify code.
- Know that combining computational thinking skills (sequence, abstraction, decomposition etc.) can help you to solve a problem.
- Understand that pattern recognition means identifying patterns to help them work out how the code works.
- Understand that algorithms can be used for a number of purposes e.g. animation, games design etc.

**Useful Images**

Data without any identification, order or sequence.



Sequence of dance moves:



Decomposition:



Pattern recognition:



**Links to other curriculum areas:** Maths – identifying patterns and sequences