Computing - Knowledge Organiser

Year 8- HT4 – Algorithms

Keywords:	
Computational Thinking	is the skill used to apply concepts/methods, problem solving techniques and logic thinking/reasoning applied to solve any given problem
Decomposition	Breaking down a complex problem or system into smaller, more manageable parts
Pattern Recognition	Looking for patterns amongst problems
Abstraction	Focussing on the important information only and ignoring the relevant detail and information
Algorithms	Developing a step-by-step solution to the problem or the rules to follow to solve the problem

Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions.

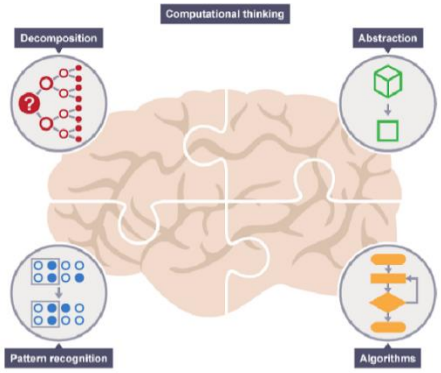
There are four key techniques (cornerstones) to computational thinking:

Decomposition - breaking down a complex problem or system into smaller, more manageable parts

Pattern Recognition – looking for similarities among and within problems

Abstraction – focusing on the important information only, ignoring irrelevant detail

Algorithms - developing a step-by-step solution to the problem, or the rules to follow to solve the problem



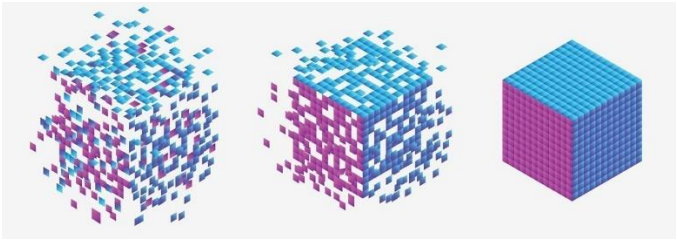
Computational thinking involves taking that complex problem and breaking it down into a series of small, more manageable problems **(decomposition)**. Each of these smaller problems can then be looked at individually, considering how similar problems have been solved previously **(pattern recognition)** and focusing only on the important details, while ignoring irrelevant information **(abstraction)**. Next, simple steps or rules to solve each of the smaller problems can be designed **(algorithms)**.

Finally, these simple steps or rules are used to **program** a computer to help solve the complex problem in the best way.

Source: BBC Bitesize, https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1

simpler. Problems are easier to solve when they share patterns, because we can use the same problem-solving solution wherever the pattern exists.

The more patterns we can find, the easier and quicker our overall task of problem solving will be.



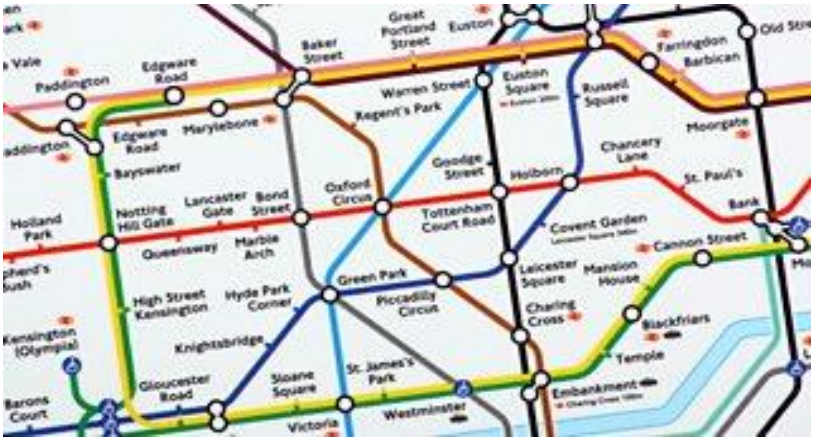
Source: Sooth Sayer Analytics, https://soothsayeranalytics.com/wp-content/uploads/2019/02/pattern-recognition-header.jpg

Computing - Knowledge Organiser

Abstraction

Abstraction involves filtering out – essentially, ignoring - the characteristics that we don't need in order to concentrate on those that we do.

An example of abstraction is the London Underground map. It details tube and rail lines and the stations that are on them. That is all that is required for a passenger to be able to plan a journey from one station to another. Other details, such as real geographical location, distance between stations, depth underground and number of platforms are not included as they are irrelevant to journey planning on the Underground.



Source: BBC Bitesize, https://www.bbc.co.uk/bitesize/guides/z4rbcj6/revision/3

Decomposition

Decomposition involves breaking down a complex problem or system into smaller parts that are more manageable and easier to understand. The smaller parts can then be examined and solved, or designed individually, as they are simpler to work with.

For example, a police officer would need to know the answer to a series of smaller problems:

- what crime was committed?
- when the crime was committed
- where the crime was committed
- what evidence there is
- if there were any witnesses
- if there have recently been any similar crimes

The complex problem of the committed crime has now been broken down into simpler problems that can be examined individually, in detail.

Source: BBC Bitesize, https://www.bbc.co.uk/bitesize/guides/zqqfyrd/revision/2

Computing - Knowledge Organiser

Flow Diagrams

A flow diagram is a diagram that shows an overview of a program. Flow diagrams normally use standard symbols to represent the different types of instruction. These symbols are used to construct the flowchart and show the step-by-step solution to the problem. Flow diagrams are sometimes known as flowcharts.

Symbol	Name	Function
	Start/End	An oval represents the start or end point
	Arrows	Lines show the relationship between different representative symbols
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Boolean Operators:

Operator Symbol	Operator Meaning
>	greater than
<	less than
=	equal to
>=	greater than or equal to
<=	less than or equal to
<>	not equal to

Computing - Knowledge Organiser

Mathematical operators:

Operator Symbol	Operator Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division

Example Flowchart: