4 pillars of computational thinking

**Computational thinking** is built on **four pillars**:

***Decomposition***

***pattern recognition***

***data representation and abstraction***

***Algorithms***

***Computational thinking*** is an approach to solving problems using concepts and ideas from computer science, and expressing solutions to those problems so that they can be run on a computer.

***Decomposition*** is a way of thinking about problems, algorithms, artefacts, processes and systems in terms of their parts.

The separate parts can then be understood, solved, developed and evaluated separately.

This makes complex problems easier to solve and large systems easier to design.

**Pattern recognition** in problem solving is key to determining appropriate solutions to problems and knowing how to solve certain types of problems. Recognizing a pattern, or similar characteristics helps break down the problem and also build a construct as a path for the solution.

Ever find yourself saying, '*where have I seen this before*', could be a significant step in computational thinking.

Once you have decomposed a complex problem, it helps to look for similarities or 'patterns' in each segmented part of the problem.

These patterns can help solve the larger problem more effectively. We look for things that have similarity in each order to address the problem. It may be that there are no common elements but it should still be a stage in the process.

Patterns exist between different problems and within a single problem.

***Abstraction*** is a way to make problems or systems easier to think about. It simply involves hiding detail, this removes unnecessary complexity.

An***Algorithm*** is a plan, a set of step-by-step instructions to resolve a problem. In an algorithm, each instruction is identified and the order in which they should be carried out is planned.

<https://youtu.be/qbnTZCj0ugI> - Computational Thinking: What Is It? How Is It Used? (really simple introduction that I watched)