6.AP.1 Algorithms and Programming

> *The student will apply computational thinking to identify patterns, make use of decomposition to break down problems or processes into sub-components, and design algorithms. (a) Identify patterns and repeated steps in an algorithm, problem, or process. (b) Decompose an algorithm, problem, or process into sub-components. (c) Abstract relevant information to identify essential details. (d) Design algorithms using abstraction to accomplish a task or express a computational process.*

Understanding the Standard

"Computational thinking" involves re-imagining problems, tasks, or ideas in such a way that they are compatible with computing technologies. For example, a student might write down a sequence of instructions (an algorithm) and notice that there's a lot of repetition (practicing pattern recognition). They could isolate the part of the instructions that repeats (decomposition), give that section a name (abstraction), and simplify the instructions by referring to the name of the section rather than writing the same sequence of instructions over and over again. Students practice these computational thinking skills when they write code that includes procedures.

Term	Definition
pattern recognition	Finding instances of repetition, especially in the context of algorithms designed around a task
decomposition	Breaking down a complex process into simple steps
algorithm	A sequence of instructions
abstraction	Grouping related things based on important attributes those things share

Prerequisite Knowledge

In order to develop the skills described in this standard, students must be able to write simple algorithms (sequences of instructions) with or without a programming language. These algorithms, when written with code, should include conditional logic and variables (see lower grade levels for details).

Integration Opportunities

Visual Arts 6.5b Design algorithms to show how to care for materials in the art room.

Health 6.3.z Work collaboratively to identify an environmental health or safety issue and develop a plan to address this issue. Use abstraction to determine and delegate important tasks efficiently.

Math 6.PFA.4a Design an algorithm to create an efficient daily schedule by decomposing activities into time blocks and identifying patterns of time use, and then represent time constraints with a linear inequality in one variable on a number line to ensure all activities fit within their available time.

Summary of a Lesson

For an "unplugged" activity, have students write down instructions for navigating a task or problem (e.g., navigating a maze, making lunch, weeding a garden, cleaning a room, drawing a shape). Make sure you choose a task or problem that will involve repeated steps. Have students write down instructions for completing the task (create an algorithm) and identify which sections repeat (pattern recognition). Then, have them give a "name" to the repeated section (decomposition, abstraction) and re-write their instructions referencing the named sequence, replacing the repetitive instructions with something like "perform the [name] instructions again".

For a "plugged" activity, teach students how to organize code into procedures, functions, or methods (depending on your chosen programming language). There are many ways to use procedures depending on the context, so reference examples in curricula or in the programming environment for details about how to implement these kinds of programs.