Algorithms & Programming: Algorithms Grade: 2

Standard: 2.AP.A.01
Model daily processes by creating and following **algorithms** (step-by-step lists of instructions) to complete tasks verbally, kinesthetically, via a **programming language**, or using a device.

Essential Skills
Create an **algorithm** by describing or **programming** the steps to complete a task.

Essential Questions
How can you create an **algorithm** for a common task?
Why is the order of the steps important in an algorithm?
Why is it important to be precise and accurate when creating an algorithm?

Explanation
Students should be able to follow **algorithms** for familiar tasks such as preparing simple foods and brushing their teeth and progress to following algorithms for tasks or outcomes with which they are unfamiliar. By grade 2, students should be able to compose algorithms independently. Initially, algorithms can be in the form of simple lists and flow charts. As students gain comfort with the concept they should create algorithms using pseudocode and **computer programs**.

Think of this as similar to….
When you get dressed for school you probably put your clothes on in a specific order--for example you put your shirt on, then your pants, then your socks and finally your shoes.

Implementation Examples—What would this look like in the classroom?

Title	Description	Link	Content Connection & Notes
Ruby's Algorithms	**Grade 1**--Students receive directions to complete tasks, beginning with familiar tasks. They are then given the algorithm activity map and given a starting place and an algorithm. They should follow the algorithm and determine where it should take them and notice where they end up after they follow the steps. They can give additional details, such as if they went over the bridge and/or through the river. **Grade 2**--Students create algorithms to complete familiar tasks. They then create algorithms for Ruby to visit her friends using the algorithm activity map.	Ruby's Algorithms	
Solve Problems	**Grade 2**-- Write algorithms to solve addition and/or subtraction problems within 100. Different algorithms should be written to solve different types of problems (e.g., Add to/Result unknown and Add to/Change unknown). Students should be able to choose the appropriate algorithm for a given problem as well as to explain what strategy or strategies the algorithm uses arrive at the solution. It should be clear that we are looking for students to focus on a solution path for a story problem and not how to write an algorithm for basic addition and subtraction within 100.		This lesson also aligns with **Math** 2.NBT.B.5
Ninja Maze Challenge	**Grade 2**--Students will write algorithms using motion blocks to move a character through four different maze challenges that focus on a combination of vertical and horizontal movement. Students then can create mazes that peers will solve. The lesson is written for use with Scratch Jr. but can be adapted to be done without a device.	Ninja Maze Challenge	

Standard: AP.A.01 Grade: 2

These annotations are a collaboration between Maryland Center for Computing Education and the Maryland State Department of Education.