Algorithms & Programming: Program Development (4) Grade: K

Standard K.AP.PD.04
Use correct terminology (e.g., first, second, etc.) in the development of an **algorithm** to solve a simple problem.

Essential Skills
Describe the sequence of an **algorithm** using appropriate terminology.

Essential Questions
How can you describe the steps in an **algorithm** or **computer program**?
Why is using correct terminology important when describing an algorithm or the process used to create a computer program?

Explanation
Students should be able to accurately describe the sequence of steps in an **algorithm**. In first grade, students should be able to explain why they chose the steps in their algorithm and by second grade, students should be able to talk or write about the goals and expected outcomes of the **computer programs** they create. These explanations can be in coding journals, discussions with a teacher, class presentations, or blogs.

Think of this as similar to….
Describe and explain the steps of brushing your teeth.

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

Title	Description	Link	Content Connection & Notes
Happy Maps	**Grade K**-- Students are given a maze and a character “flurb” and work in teams to get the "flurb" to the fruit. Students will create precise instructions as they work to translate instructions into the symbols provided. Students should be able to identify the beginning, middle and end of the directions that they give the flurb. **Grade 1**--Students should experiment with changing the order of the instructions to see if the flurb still reaches the fruit. They should explain their algorithm in terms of the order of the directions and be able to justify their choice of algorithms and explain the changes they make.	Happy Maps	This lesson also aligns with **CS** K.AP.V.01 and AP.C.01; similar activities can be done with floor robots see Code and Go Introduction
The Snowy Day	**Grade K**--Students retell The Snowy Day story using a robot to move from event to event. Students first describe the story in sequential terms (first, second, etc.). They then use a recording sheet to plan their code to move the robot to each story event in order. They should describe their code in sequential terms as well. **Grade 1**--Students explain why they chose the sequence they used to have the robot reach the different events in the story. **Grade 2**--Students describe their program development in terms of input (pushing the buttons) and output (the robot moving). The routes should be more complex (add some indication of squares where the robots cannot go) and students should also use the word “debug" to describe finding and correcting errors in their code. Any story can be used for this activity; an historical event or a scientific process can be used as well.	The Snowy Day	This lesson also aligns with **ELA** K.W.2. This lesson uses robots but can be adapted to be unplugged. Two similar lessons are Beginning Middle End Project and Retelling Nursery Rhymes.

Title	Description	Link	Content Connection & Notes
Dancing Alone	**Grade K**--Students use Scratch Jr. to create a silly dance for Scratch Cat using motion blocks. Students are introduced to creating sequences of code in Scratch Jr. Students should be able to identify the blocks at the beginning, middle and end of the program they create and in the dance the Scratch Cat does. **Grade 1**-- Students should identify how the order of the motion blocks determines the order of the dance and explain what they wanted the cat to do. Students should predict how changing the order of the blocks will change the dance, explain why the reasons for the changes and the reasons they make the changes in the code that they do. and test their predictions. **Grade 2**--Students should explain the algorithm(s) they use and how they are designed in terms of input (the code), output (the dance), debugging, etc.	Dancing Alone	This lesson also aligns with **CS** AP.V.01, AP.C.01, AP.PD.01, and AP.PD.03 and is similar to Getting Loopy

Standard: AP.PD.04 Grade: K

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