

Algorithms & Programming: Program Development (3) Grade: 3

Standard 3.AP.PD.03

Analyze and **debug** an existing **program** or **algorithm** that includes sequencing, repetition, and **variables** in a **programming language**.

Essential Skills

Analyze a **computer program** to determine if it is successful.

Propose and implement a correction for a flawed portion of a program.

Essential Questions

How do you determine if a **program** is not working as intended?

When a program is not working as intended what are strategies that can be used to determine what is wrong and how to correct it?

How do **iterative processes** (such as testing and **debugging**) contribute to successful program development?

Explanation

Students test and make necessary corrections (**debug**) to their **computer programs** and to those of others to ensure they run properly. Debugging strategies may include guessing and checking, testing small pieces of code to determine if various portions work correctly, trying each part of the program in sequence to determine where the error occurs, closely examining the algorithm, and/or soliciting assistance from peers and online resources. By fifth grade, students will create a program that includes at least one sequence, repetition (**loop**), and **variable**. They should be able to do this using a **programming language** such as Scratch, Blockly, Python, etc.

Think of this as similar to...

When you make up a tune for a musical instrument, you have to keep trying it out and correcting what does not sound right.

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

Title	Description	Link	Content Connection & Notes
Drawing Shapes with Loops	Grade 3 --Students practice creating algorithms that include loops. They determine if their algorithms achieve what they intended and debug the code as necessary if it did not.	Drawing Shapes with Loops	
Programmed to Dance	<p>Grade 3--Students watch a video showing a sequence of dance steps, then write an algorithm to “program” each other to replicate the dance. They provide the steps to another group and determine if the dance is as desired. Identify areas where the algorithm needs improvement and propose revisions.</p> <p>Grade 4--Groups will create their own dance move and write an algorithm which should include loops and variables. Follow the algorithm and make any corrections needed.</p> <p>Grade 5-- Classmates follow the algorithm of another group as written. The algorithm creators analyze the dance as done by their classmates and identify and correct any errors in the algorithm."</p>	Programmed to Dance	This is an unplugged lesson—no computing device needed.
Music and Loops	<p>Grade 3-- Students are given a program in Scratch that contains code (with comments) for short musical loops. Students are challenged to create their own musical piece by combining the different loops. Students can modify the starter program so that user input sets the value of a variable that determines how many times a loop is played; a conditional can be used to determine which loop is played based on the value of the variable. The students will determine if the musical piece plays as they wish and revise the code as necessary.</p> <p>Grade 4-- Students create a computer program code modelled on the program described above. They determine areas that are not functioning correctly and revise the code as needed.</p> <p>Grade 5--Students plan and then create a program that will play music. They analyze the program they developed and debug the code, so the program developed matches their plan.</p>	Music and Loops the write up does not include the variables, conditionals and events but can be adapted.	This lesson also aligns with CS AP.C.01 and requires students to be somewhat comfortable programming in Scratch or other programming language used.

Standard: AP.PD.03 Grade: 3

These annotations are a collaboration between [Maryland Center for Computing Education](#) and the [Maryland State Department of Education](#).