Algorithms & Programming: Modularity (1) Grade Band: 3-5

Grade	Standards: AP.M.01
3	Decompose a simple problem into a precise set of sequences instructions.
4	Decompose a large problem into smaller, manageable sub-problems to facilitate the **program** development process.
5	Decompose a large problem into smaller, manageable sub-problems and then further into sets of sequenced instructions to facilitate the program development process.

Grade	Essential Skills
3	Devise an **algorithm**, a set of ordered instructions, to solve a problem.
4	Break a complex problem (including a **programming** challenge) into smaller sub-problems. Identify reasons for developing a number of programs to solve smaller sub-problems, rather than developing one program to address a larger, more complex problem.
5	Devise **algorithms** to solve identified sub-problems Demonstrate how combinations to the solutions of sub-problems can simplify writing programs to solve complex problems.

Explanation
Students will be able to break down, or **decompose**, complex problems into smaller problems that are easier to solve and manage. They can then express the solutions to those smaller problems as a series of instructions or an **algorithm**. Students should understand how to combine the algorithms addressing the smaller problems in order to solve the more complex problem, and how that process facilitates program development. Additionally, students should demonstrate an understanding that modifying solutions or finding errors in instructions is easier when addressing smaller subproblems than large complex problems and that decomposition enables different people to work on different parts of the same project simultaneously.

Think of this as similar to….
When you make a holiday meal, you may start by preparing the main course. Someone else may set the table. Yet another person makes the dessert. You combine the parts to create the whole meal.

Essential Questions
What is the advantage of **decomposing** problems?
How does decomposing a problem into sub-problems help when writing a complex **computer program**?

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

Grade(s)	Title	Description	Link	Content Connection & Notes
3	**Determine Area by Decomposition**	**Grade 3**-- This a sequence of three lessons. Day 1 covers defining the computer science vocabulary term "decomposition", defining area, and relating the two. Day 2 covers decomposing a rectilinear figure into cubic squares. In Day 3, students apply their learning to determine the area of letters represented on a grid.	Determine Area by Decomposition	This lesson also aligns with **Math** 3.MD.C.5 and 3.MD.C.6
3	**Nested Loops in Maze**	**Grade 3**--Students will use the puzzles in this CS Fundamentals lesson to decompose (break down) a long code into smaller sections of code. As they play with the Bee and Plants vs Zombies students further decompose the code into smaller sections and examine them for patterns.	Nested Loops in Maze	
3	**Measuring Jack's Path**	**Grade 3**--Students mark the start and stop point of Jack (a Code and Go mouse or other floor robot) when it moves forward by pushing the forward arrow one time. Students measure this distance to the nearest quarter inch. Using this measurement, students create a maze Jack can travel through. Once maze is complete, students create a code for Jack to complete each section of the maze to get to the end. Students test their code for each section of the maze, and have a peer test the code to ensure it works well. Students add on to their code as the robots successfully navigates each section of the maze. Students then measure the distance Jack travelled from start to end of the maze while executing all the code. The class comes together and records the distance Jack traveled for their particular maze (nearest quarter inch), and records data on a line plot.		This lesson also aligns with **Math** 3.MD.B.4; floor robots are used.
4	**The Great Number Hunt**	**Grade 4--**Students are asked to find the fastest way to guess a number from a sorted list. They will find that if they decompose the list repeatedly it becomes easier to find the number. Students should be able to list the steps that they use and explain why each is important.	The Great Number Hunt	

Grade(s)	Title	Description	Link	Content Connection & Notes
4-5	**Abstraction Unplugged**	**Grade 4**--Students will be presented with a Scratch project that they will decompose with their partners without having access to its code and without access to a computer. They will propose the smaller chunks of programming that would be most helpful in programming the project. **Grade 5**--Students propose an arrangement to assemble the smaller chunks of programming in order to create the whole Scratch program. Write the algorithms the sub programs (smaller chunks) in Scratch and determine if they work as desired.	Abstraction Unplugged	
4-5	**Functions in Minecraft**	**Grade 4**--Students find reusable patterns in code when they name that chunk of code they have created a function. That function can accomplish a portion of what the puzzle asks them to do. **Grade 5**--Students create functions and use them repeatedly in their program to tell the computer to run that chunk of code in different environments. They should be able to describe their solution to the larger puzzle in terms of the functions they have created. (This would be done in the Free Play portion of the lesson.)	Functions in Minecraft	
5	**Energy**	**Grade 5--**In order to plan a model to demonstrate that energy in animals' food was once energy from the sun, the students will decompose the problem into how animals get energy (from food) what is the food (other animals and ultimately plants) and where plants get energy (from the sun). Students then plan their model and plan how their program (in Scratch or another programming language) will demonstrate the different transfers of energy. Once the plan is complete, students create each part of program and adjust each so the parts work together.	Energy	This lesson also aligns with **NGSS** 5-PS3-1

Standard: AP.M.01 Grade Band: 3-5
These annotations are a collaboration between Maryland Center for Computing Education and the Maryland State Department of Education.