Essential Skills for Algorithms & Programming: Algorithms

Grade	Standards AP.A.01	Essential Skills
K	Model daily processes and follow basic algorithms (step-by-step lists of instructions) to complete tasks.	Follow a sequence of instructions to complete a familiar task
1	Model daily processes and follow basic algorithms (step-by-step lists of instructions) to complete tasks verbally, kinesthetically, via a programming language, or using a device.	Complete an unfamiliar task as detailed by an algorithm.
2	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.	Create an algorithm by describing or programming the steps to complete a task.
3	Develop and compare multiple algorithms for the same task.	Compose (independently or collaboratively) two or more algorithms for the same task. Examine the differences among algorithms for the same task.
4	Develop, compare, and refine multiple algorithms for the same task	Modify two or more algorithms to complete the same task. Modifications may include finding possible errors (debugging), making instructions more efficient (adding loops if instructions are repeated), etc.
5	Develop, compare, and refine multiple algorithms for the same task and determine which algorithm is the most appropriate.	Evaluate two or more algorithms that complete the same task to determine which algorithm is best suited for the task at hand. Justify the choice of which algorithm is most appropriate to complete a task.

Standard: AP.A.01 Skills

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

Essential Skills for Algorithms & Programming: Variables

Grade	Standards AP.V.01	Essential Skills
K	With guidance, model the way programs store and manipulate grade-level data by using numbers or other symbols to represent information (e.g., encode or decode words using numbers, pictographs or symbols to letters, words, or direction).	Identify and interpret symbols that are used to represent information such as numbers for quantities or letters for sounds. Create and use symbols to represent information such as establishing hand signals for "I agree" or creating emoji-like symbols for feelings.
1	With guidance, model the way programs store and manipulate grade-level data by using numbers or other symbols to represent information (e.g., encode or decode words using numbers, pictographs or symbols to letters, words, or direction)	Identify and interpret symbols that are used to represent information such as mathematical operators, pictographs, Create and use symbols to represent information such as comparative quantities, repeating patterns, a series of actions, or directions.
2	Model the way programs store and manipulate grade-level data by using numbers or other symbols to represent information (e.g., encode or decode words using numbers, pictographs or symbols to letters, words, or direction).	Encode or decode messages that use representations such as arrows, pictographs, etc. when given a key.
3	Create programs that use **variables** to store and modify grade appropriate **data**.	Create a **computer program**, using **code** that is provided, in which **variables** are used to store **data**. Identify the data that is stored in a variable in a computer program that uses a variable.
4	Create programs that use variables to store and modify grade-appropriate data.	Create a computer program in which a variable is used to store data. Identify how a variable changes within a computer program that uses a variable.
5	Create programs that use variables to store and modify grade-appropriate data.	Create a computer program in which the value of a variable changes, resulting in a change in the **output** of the program. Use variables for more than one type of data (e.g., text and numbers) in a computer program.

Skills for Standard: AP.V.01 Grades K-5

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

Essential Skills for Standard: Algorithms & Programming: Control

Grade	Standards: AP.C.01	Essential Skills
K	With guidance, create a set of instructions (**programs**) to accomplish a task using a **programming language**, **device**, or **unplugged** activity, including sequencing, emphasizing the beginning, middle, and end.	Recognize the order of a sequence of instructions or occurrences as the beginning, middle and end. Create a logical sequence of instructions with guidance as needed.
1	With guidance, create programs by using creative expression or problem solving, to accomplish tasks that include sequencing and repetition. Programming languages, robot devices, or unplugged activity can serve as the means	Determine what changes will occur if there is a change in the sequence of instructions or occurrences. Identify patterns and repetition within sequences.
2	Create programs using a programming language, robot device, or unplugged activity that utilize sequencing and repetition to solve a problem or express creative ideas.	Recognize that a **computer program** is a set of instructions in a specific sequence. Create a simple computer program, including repeated sequences, to express an idea or solve a problem. Students can be supplied with the commands/code to create the program.
3	Create **programs** using a **programming language** that includes **sequences**, **loops**, **conditionals**, and **variables** to solve a problem or express an idea.	Integrate the use of a **variable** with a changing value into a **computer program**. Structure a computer program using **conditionals** (if...then...statements) and **loops** (repeated sequences).
4	Create programs using a programming language that includes sequences, loops, conditionals, and variables that utilize mathematics operations to manipulate values in order to solve a problem or express an idea.	Perform mathematical operations (addition, division, etc.) on variables in a program for a purpose such as tallying a score or keeping time (ex. If the ball crosses the line score=score+1).
5	Create programs using a programming language that includes sequences, loops, conditionals, event handlers, and variables that utilize mathematics operations to manipulate values in order to solve a problem or express an idea.	Incorporate one or more **events** that cause a set of instructions or occurrences to be executed in a computer program. Model (verbally, using a flowchart, etc.) when and how events in a computer program trigger a set of instructions (**event handlers**).

Skills for Standard: AP.C.01 Grades K-5
These annotations are a collaboration between Maryland Center for Computing Education and the Maryland State Department of Education.

Essential Skills for Algorithms & Programming: Modularity (1)

Grade	Standards: AP.M.01	Essential Skills
K	Not addressed at this level	
1	Not addressed at this level	
2	Not addressed at this level	
3	Decompose a simple problem into a precise set of sequences instructions.	Devise an **algorithm**, a set of ordered instructions, to solve a problem.
4	Decompose a large problem into smaller, manageable sub-problems to facilitate the **program** development process.	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	Decompose a large problem into smaller, manageable sub-problems and then further into sets of sequenced instructions to facilitate the program development process.	Devise **algorithms** to solve identified sub-problems Demonstrate how combinations to the solutions of sub-problems can simplify writing programs to solve complex problems.

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

Essential Skills for Algorithms & Programming: Modularity (2)

Grade	Standards: AP.M.02	Essential Skills
K	Not addressed at this level	
1	Not addressed at this level	
2	Not addressed at this level	
3	Modify, **remix**, or incorporate portions of an existing **program** into one's own work, to develop or add more advanced features (grade-level appropriate).	Select one or more features from an existing **computer program** with teacher guidance and add the feature(s) to an original program.
4	Modify, remix, or incorporate portions of an existing program into one's own work, to develop or add more advanced features (grade-level appropriate).	Add a portion of an existing computer program to an original project in order to add a new element or capability.
5	Modify, remix, or incorporate portions of an existing program into one's own work, to develop or add more advanced features (grade-level appropriate).	Increase the complexity of an original computer program by incorporating portions of existing programs and making modifications, as necessary.

Skills for Standard: Grades K-5

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

Essential Skills for Algorithms & Programming: Program Development (1)

Grade	Standard	Essential Skills
K	With guidance, create a grade level appropriate document to illustrate thoughts, ideas, or stories in a sequential manner (e.g., storyboard, story map, sequential graphic organizer).	Identify, in order, the steps to describe a process, tell a story, etc. Present ordered steps in a document with teacher assistance, as necessary.
1	Create a grade-level appropriate document to illustrate thoughts, ideas, or stories in a sequential manner (e.g., storyboard, story map, sequential graphic organizer).	Present ordered steps to describe a process, to tell a story, etc. in a document.
2	With guidance, create a grade level appropriate document to clarify the steps that will be needed to create a sequential **program** and can be used to check if the **program functionality** is correct.	Describe the ordered steps needed to create a **computer program** in a document. Explain the desired goals of a program in a document.
3	Use an **iterative process** to plan the development of a program by including others' perspectives and considering user preferences while solving simple problems.	Review and revise the plan for a **computer program** by incorporating feedback from a partner.
4	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences while solving simple problems.	Review and revise the plan for a computer program repeatedly by incorporating feedback from peers.
5	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences while solving problems.	Review and revise the plan for a computer program by repeatedly incorporating the perspective of users and others.

Skills for Standard: AP.PD.01 Grades K-5

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

Essential Skills for Algorithms & Programming: Program Development (2)

Grade	Standards: AP.PD.02	Essential Skills
K	Give attribution to ideas, solutions, and creations of others, verbally, while developing **algorithms**.	Identify ideas or items that were created by others that are used in the process of developing **algorithms**.
1	Give attribution to ideas, solutions, and creations of others, verbally, or written, while writing or developing algorithms and **programs**.	Give credit to an author, artist, etc. when using resources or artifacts they created to develop algorithms or **computer programs**
2	Give attribution to ideas, solutions, and creations of others, verbally and written, while writing and developing programs.	Give written credit to the creator (author, artist, etc.) when using ideas or artifacts of others' when writing a computer program.
3	Identify instances of **remixing**, when ideas are borrowed and treated upon, and provide attribution	Recognize and give credit when using or **remixing** the ideas and the creations of others.
4	Observe **intellectual property** rights and give appropriate attribution when creating or remixing programs	Provide attribution in an appropriate format for ideas and creations of others when used in writing **computer programs**. Determine the limitations on reusing or remixing specific items given the way they are licensed (For example determine if the artifact is copyrighted or licensed as Creative Commons.)
5	Observe intellectual property rights and give appropriate attribution when creating or remixing programs	Reflect on work produced and assess the desired restrictions to the ability of others to remix or reuse that work. Identify ways creators can restrict how others reuse and remix their work and the reasons they may do so.

Skills for Standard: AP.PD.02 Grades K-5

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

Essential Skills for Algorithms & Programming: Program Development (3)

Grade	Standard: AP.PD.03	Essential Skills
K	Identify errors in an **algorithm** that includes sequencing and repeated **procedures** using a **programming language** or **unplugged** activity. Discuss how errors in the algorithm could be corrected.	Describe how an **algorithm** or series of steps did not work as expected or desired.
1	Identify and correct errors (**debug**) in programs which include sequencing and repetition to accomplish a task, through a variety of techniques and strategies that could include an unplugged activity (e.g., changing order or sequence, following steps, trial and error).	Analyze a simple algorithm to find **bugs**. Suggest solutions to the bugs in the algorithm using a variety of strategies.
2	Develop and debug programs that include sequencing and repetition to accomplish a task, through the use of a programming language and/or unplugged activity.	Develop an algorithm for a specific purpose and identify any bugs in the algorithm. Implement a proposed fix for bugs in an algorithm and determine if the algorithm works as desired.
3	Analyze and debug an existing **program** or **algorithm** that includes sequencing, repetition, and **variables** in a programming language.	Analyze a **program** to determine if it is successful. Propose and implement a correction for a flawed portion of a program
4	Create and debug a program or algorithm that includes sequencing, repetition, and variables in a programming language	Create a program that includes sequences, loops and **variables**, and detect, and correct flaws found in the program.
5	Create, test, and debug a program that includes sequencing, repetition, and variables in a programming language to ensure it runs as intended.	Evaluate a computer program you have created (containing sequences, loops and variables) with respect to its intended outcome. Examine a computer program you have created (containing sequences, loops and variables) to determine where errors exist and make revisions in order to achieve the intended outcome.

Skills for Standard: AP.PD.03 Grades K-5

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

Essential Skills for Algorithms & Programming: Program Development (4)

Grade	Standards AP.PD.04	Essential Skills
K	Use correct terminology (e.g., first, second, etc.) in the development of an **algorithm** to solve a simple problem.	Describe the sequence of an **algorithm** using appropriate terminology
1	Use correct terminology (e.g., beginning, middle, end, etc.) and explain choices made during the development of an algorithm and/or **program** to solve a simple problem.	Justify the steps chosen when creating an algorithm
2	Use correct terminology (e.g., debug, program input/output, code, etc.) to explain the development of a program to solve a problem in an **unplugged** activity, hands-on manipulative, or **programming language.**	Describe the goals of a **computer program**. Explain the steps taken in developing a computer program using correct terminology.
3	Communicate and explain **program** development to peers and adults using comments, presentations, and demonstrations.	Using correct terminology, describe the steps taken to develop a **computer program**.
4	Communicate and explain program development to peers and adults using comments, presentations, and demonstrations.	Correlate the steps taken when developing a computer program to the final program produced.
5	Communicate and explain program development to peers and adults using comments, presentations, and demonstrations.	Summarize how choices made during **program developmen**t, including **debugging** and checking **inputs** and **outputs**, contributed to program development and the outcome achieved.

Skills for Standard: AP.PD.04 Grades K-5
These annotations are a collaboration between Maryland Center for Computing Education and the Maryland State Department of Education.