4 TECH 2020 - Algorithms and Programming

Content Area: **Generic Content Area**
Course(s): **Technology Education 3**
Time Period: **Marking Period 3**
Length: **16 days**
Status: **Published**

Established Goals/Standards

Please choose the appropriate Goals/Standards from the Standards tab above.

CS.3-5.8.1.5.AP.1	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
CS.3-5.8.1.5.AP.2	Create programs that use clearly named variables to store and modify data.
CS.3-5.8.1.5.AP.3	Create programs that include sequences, events, loops, and conditionals.
CS.3-5.8.1.5.AP.4	Break down problems into smaller, manageable sub-problems to facilitate program development.
CS.3-5.8.1.5.AP.5	Modify, remix, or incorporate pieces of existing programs into one's own work to add additional features or create a new program.
CS.3-5.8.1.5.AP.6	Develop programs using an iterative process, implement the program design, and test the program to ensure it works as intended.
	Individuals develop programs using an iterative process involving design, implementation, testing, and review.
	Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that already exist.
	A variety of control structures are used to change the flow of program execution (e.g., sequences, events, loops, conditionals).
	Programming languages provide variables, which are used to store and modify data.
	Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific use than others.

Essential Questions

- How is the flow of a program changed while running?
- What is an algorithm?
- What is the best process to develop a program?

Essential Understandings

- Designing a program follows an iterative process of implementation and testing with redesign and refinement.
- Programs can be broken down into smaller sections that perform a specific task.

Content

Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
Students select and use digital tools to plan and manage a design process that considers design constraints and calculated risks
Students develop, test and refine prototypes as part of a cyclical design process.
Students exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems
Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
Students collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.
Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

Assessment

Performance based assessment

Accommodations and Modifications

Accommodations and Modifications according to student IEP, 504, I&RS goals, and/or gifted status.

Resources

Please add your Resources by clicking on the Lists tab above.

- code.org
- Scratch
- Teacher developed presentations