

Unit 3: Coding and Algorithms

Content Area: **Computer Science**
Course(s):
Time Period: **Trimester 1**
Length: **10 days**
Status: **Published**

BRIEF SUMMARY OF UNIT

Students will learn the difference between an Algorithm and a computer program (coding). Students will work through learning activities where they need to give precise instructions similar to a computer program. The Scratch block coding program and others will be used by students to learn the basics of coding and to create animations, and games. Students will have the opportunity to compare the syntax of the Scratch program to others computer programming languages. A final coding project will require students to use critical thinking/problem solving/evaluating skills, and creative self-expression.

STANDARDS

MA.K-12.1	Make sense of problems and persevere in solving them.
MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
	Constructing explanations and designing solutions in 6–8 builds on K–5 experiences and progresses to include constructing explanations and designing solutions supported by multiple sources of evidence consistent with scientific knowledge, principles, and theories.
	Constructing Explanations and Designing Solutions
CS.6-8.8.1.8.AP.1	Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.
CS.6-8.8.1.8.AP.2	Create clearly named variables that represent different data types and perform operations on their values.
CS.6-8.8.1.8.AP.3	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
CS.6-8.8.1.8.AP.4	Decompose problems and sub-problems into parts to facilitate the design, implementation, and review of programs.
CS.6-8.8.1.8.AP.6	Refine a solution that meets users' needs by incorporating feedback from team members and users.
CS.6-8.8.1.8.AP.7	Design programs, incorporating existing code, media, and libraries, and give attribution.
CS.6-8.8.1.8.AP.8	Systematically test and refine programs using a range of test cases and users.
CS.6-8.8.1.8.AP.9	Document programs in order to make them easier to follow, test, and debug.
CS.6-8.8.1.8.CS.4	Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.

WRK.K-12.P.5	Utilize critical thinking to make sense of problems and persevere in solving them.
WRK.K-12.P.8	Use technology to enhance productivity increase collaboration and communicate effectively.
TECH.9.4.8.CI.2	Repurpose an existing resource in an innovative way (e.g., 8.2.8.NT.3).
TECH.9.4.8.CI.3	Examine challenges that may exist in the adoption of new ideas (e.g., 2.1.8.SSH, 6.1.8.CivicsPD.2).
TECH.9.4.8.CI.4	<p>Explore the role of creativity and innovation in career pathways and industries.</p> <p>Some digital tools are appropriate for gathering, organizing, analyzing, and presenting information, while other types of digital tools are appropriate for creating text, visualizations, models, and communicating with others.</p> <p>Gathering and evaluating knowledge and information from a variety of sources, including global perspectives, fosters creativity and innovative thinking.</p> <p>Multiple solutions often exist to solve a problem.</p>

TRANSFER

- Collaborate with others.
- Execute clear, concise directions.
- Extend their learning of computer programming languages.
- Use creative self-expression, critical thinking, and problem solving skills.

ESSENTIAL QUESTIONS

- What needs and/or problems could be fulfilled/solved with a computer program/app?
- What needs and/or problems were addressed by the development of computer programs/apps I use?

ESSENTIAL UNDERSTANDINGS

- Computer programming is a way to create solutions to problems.
- Computer programs are made up of precise instructions.
- Failure should be viewed as an opportunity for learning.
- Programming languages have specific syntax which must be followed.
- There is not always one 'right answer' when dealing with creative problem solving.

STUDENTS WILL KNOW

- A computer program is a set of instructions written in a specific computer language (i.e., Logo, Scratch, Java Script,) that tell the computer what to do.
- Create clearly named variables that represent different data types and perform operations.
- How to design and illustrate algorithms that solve complex problems.

STUDENTS WILL BE SKILLED AT

- Creative self-expression, critical thinking, and problem solving skills.
- The ability to collaborate with others.
- Using block coding and other programming languages to code animations, games, and various programs.

EVIDENCE/PERFORMANCE TASKS

- Students will take a quiz on the terms algorithm, computer program, variable, loop, and conditional statement.
- Students will replicate a demonstrated Scratch animation that contains two sprites, a background, motion, loops, looks, and sound blocks.
- Students will be able to debug coding assignments.
- Students will be able to code a Scratch game.
- Students will create a story using Scratch with a partner.
- Students will be able to use and debug other computer languages.

Assessments

- Formative: Daily assessments using examples from class notes and CodeHS.com
- Summative: Teacher-created assessments/projects and CodeHS Computer Science Projects
- Benchmark: Check for understanding benchmark assessments on CodeHS
- Alternative Assessments: Student-centered activities such as a doorbell coding project, game design projects, and other activities involving real world applications
- [Activities/Assessments Folder](#)

Core instructional materials: [Core Book List](#)

Supplemental materials: Khan Academy

LEARNING PLAN

- Discuss Algorithms - Conduct an activity that explains what an algorithm is.
- Discuss Computer Programs - show students code - show students what the code produces, i.e.,

html/CSS (webpage).

- Discuss the difference between algorithms and computer programs.
- Introduce Scratch or other Computer language give background/overview.
- Demonstrate the use of the computer language, assignment/project or learning platform.
- Circulate lab assisting when needed.
- Students should be encouraged to help each other and use the Internet for reference.

MATERIALS

- Core instructional materials: [Core Book List](#)

Supplemental materials: CodeHS

- [CodeHS.com](https://codehs.com)
- <https://scratch.mit.edu/>
- <https://codehs.com/>
- <https://code.org/>

SUGGESTED STRATEGIES FOR MODIFICATIONS

[Possible accommodations/modification for Computers - Grade 6](#)