March Gr. 1

Content Area:	**Technology**
Course(s):	
Time Period:	**March**
Length:	**4-5Weeks**
Status:	**Published**

Unit Overview

Students continue to explore coding.

Enduring Understandings

Coding is the language used to make computer programs work.

Essential Questions

What is coding?

How do you write code?

Instructional Strategies & Learning Activities

Objective: Intro to Coding - ONLINE activites code.org (Course #1) DAY 9
The student will *continue* to learn and understand basic concepts about coding creating code in a "blockly" language which writes Javascript 'under the hood'. Students will also complete "showcase" pieces for the STEM Expo.

Differentiation:
Self-paced

Assessment:
Teacher dashboard reports

Objective: Intro to Coding - ONLINE activites code.org (Course #1) DAY 10
The student will *continue* to learn and understand basic concepts about coding creating code in a "blockly" language which writes Javascript 'under the hood'. Students will also complete "showcase" pieces for the STEM Expo.

Differentiation:
Self-paced

Assessment:
Teacher dashboard reports

Objective: Intro to Coding - ONLINE activites code.org (Course #1) Day 11
The student will *continue* to learn and understand basic concepts about coding creating code in a "blockly" language which writes Javascript 'under the hood'. **Students will also complete "showcase" pieces for the STEM Expo.**

Differentiation:
Self-paced

Assessment:
Teacher dashboard reports

Objective: Intro to Coding - ONLINE activites code.org (Course #1) Day 12
The student will *continue* to learn and understand basic concepts about coding creating code in a "blockly" language which writes Javascript 'under the hood'. **Students will also complete "showcase" pieces for the STEM Expo.**

Differentiation:
Self-paced

Assessment:
Teacher dashboard reports

Integration of Career Readiness, Life Literacies and Key Skills

Students will learn about careers in computer programming while learning coding.

WRK.9.1.2.CAP	Career Awareness and Planning
WRK.9.1.2.CAP.1	Make a list of different types of jobs and describe the skills associated with each job.
TECH.9.4.2.CI	Creativity and Innovation
TECH.9.4.2.CI.1	Demonstrate openness to new ideas and perspectives (e.g., 1.1.2.CR1a, 2.1.2.EH.1, 6.1.2.CivicsCM.2).
TECH.9.4.2.CI.2	Demonstrate originality and inventiveness in work (e.g., 1.3A.2CR1a).
TECH.9.4.2.CT	Critical Thinking and Problem-solving
TECH.9.4.2.CT.1	Gather information about an issue, such as climate change, and collaboratively brainstorm ways to solve the problem (e.g., K-2-ETS1-1, 6.3.2.GeoGI.2).
TECH.9.4.2.CT.2	Identify possible approaches and resources to execute a plan (e.g., 1.2.2.CR1b, 8.2.2.ED.3).
TECH.9.4.2.CT.3	Use a variety of types of thinking to solve problems (e.g., inductive, deductive).
TECH.9.4.2.TL.1	Identify the basic features of a digital tool and explain the purpose of the tool (e.g., 8.2.2.ED.1).
	Income is received from work in different ways including regular payments, tips, commissions, and benefits.

Digital tools have a purpose.

Different types of jobs require different knowledge and skills.

Collaboration can simplify the work an individual has to do and sometimes produce a better product.

Critical thinkers must first identify a problem then develop a plan to address it to effectively solve the problem.

Brainstorming can create new, innovative ideas.

Interdisciplinary Connections

LA.RI.1.1	Ask and answer questions about key details in a text.
LA.RI.1.2	Identify the main topic and retell key details of a text.
LA.RI.1.4	Ask and answer questions to help determine or clarify the meaning of words and phrases in a text.
LA.RI.1.5	Know and use various text features (e.g., headings, tables of contents, glossaries, electronic menus, icons) to locate key facts or information in a text.
LA.RI.1.6	Distinguish between information provided by pictures or other illustrations and information provided by the words in a text.

Differentiation

- Understand that gifted students, just like all students, come to school to learn and be challenged.
- Pre-assess your students. Find out their areas of strength as well as those areas you may need to address before students move on.
- Consider grouping gifted students together for at least part of the school day.
- Plan for differentiation. Consider pre-assessments, extension activities, and compacting the curriculum.
- Use phrases like "You've shown you don't need more practice" or "You need more practice" instead of words like "qualify" or "eligible" when referring to extension work.
- Encourage high-ability students to take on challenges. Because they're often used to getting good grades, gifted students may be risk averse.

- **Definitions of Differentiation Components**:
  - Content – the specific information that is to be taught in the lesson/unit/course of instruction.
  - Process – how the student will acquire the content information.
  - Product – how the student will demonstrate understanding of the content.
  - Learning Environment – the environment where learning is taking place including physical location and/or student grouping

Differentiation occurring in this unit:

See Differentiation listed above.

Modifications & Accommodations

Refer to QSAC EXCEL SMALL SPED ACCOMMOCATIONS spreadsheet in this discipline.

Modifications and Accommodations used in this unit:

IEP and 504 accommodations will be utilized.

Benchmark Assessments

Benchmark Assessments are given periodically (e.g., at the end of every quarter or as frequently as once per month) throughout a school year to establish baseline achievement data and measure progress toward a standard or set of academic standards and goals.

Schoolwide Benchmark assessments:

Aimsweb benchmarks 3X a year

Linkit Benchmarks 3X a year

DRA

Additional Benchmarks used in this unit:

Teacher observation and checklists to show growth over time.

Formative Assessments

Assessment allows both instructor and student to monitor progress towards achieving learning objectives, and can be approached in a variety of ways. **Formative assessment** refers to tools that identify misconceptions, struggles, and learning gaps along the way and assess how to close those gaps. It includes effective tools for helping to shape learning, and can even bolster students' abilities to take ownership of their learning when they understand that the goal is to improve learning, not apply final marks (Trumbull and Lash, 2013). It can include students assessing themselves, peers, or even the instructor, through writing, quizzes, conversation, and more. In short, formative assessment occurs throughout a class or course, and seeks to improve student achievement of learning objectives through approaches that can support specific student needs (Theal and Franklin, 2010, p. 151).

Formative Assessments used in this unit:

See assessment listed above.

Summative Assessments

summative assessments evaluate student learning, knowledge, proficiency, or success at the conclusion of an instructional period, like a unit, course, or program. Summative assessments are almost always formally graded and often heavily weighted (though they do not need to be). Summative assessment can be used to great effect in conjunction and alignment with formative assessment, and instructors can consider a variety of ways to combine these approaches.

Summative assessments for this unit:

See assessment listed above.

Instructional Materials

See materials listed above.

Standards

CS.CS	Computing Systems
CS.K-2.8.1.2.AP.1	Model daily processes by creating and following algorithms to complete tasks.
CS.K-2.8.1.2.AP.2	Model the way programs store and manipulate data by using numbers or other symbols to represent information.
CS.K-2.8.1.2.AP.3	Create programs with sequences and simple loops to accomplish tasks.
CS.K-2.8.1.2.AP.4	Break down a task into a sequence of steps.
CS.K-2.8.1.2.AP.5	Describe a program’s sequence of events, goals, and expected outcomes.
CS.K-2.8.1.2.AP.6	Debug errors in an algorithm or program that includes sequences and simple loops.
CS.K-2.8.1.2.CS.1	Select and operate computing devices that perform a variety of tasks accurately and quickly based on user needs and preferences.
CS.K-2.AP	Algorithms & Programming
	Real world information can be stored and manipulated in programs as data (e.g., numbers, words, colors, images).

Individuals use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.

People work together to develop programs for a purpose, such as expressing ideas or addressing problems. The development of a program involves identifying a sequence of events, goals, and expected outcomes, and addressing errors (when necessary).

Computers follow precise sequences of steps that automate tasks.

Complex tasks can be broken down into simpler instructions, some of which can be broken down even further.

Individuals develop and follow directions as part of daily life. A sequence of steps can be expressed as an algorithm that a computer can process.