

Course/Subject: Fundamentals of Coding **Grade Level: 4**

Textbook(s) / Instructional Materials Used: Online resources through code.org

Month(s): August - October **Marking Period 1**

Unit Name

<u>Big Idea</u>	<u>CSTA K-12 Computer Science Standards</u>	<u>Essential Questions & Lesson Essential Question</u>	<u>Concepts/Objectives</u>	<u>Vocabulary</u>	<u>Competencies</u>
Sequencing Debugging Loops Nested Loops Conditionals	<p>1B-AP-08 - Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p><u>ISTE Standards</u> Computational Thinker</p> <p>Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions. Students:</p>	<p>How can building a set of directions help to complete a task?</p> <p>What strategies can be used when coding to find and fix bugs?</p> <p>In what ways do debugging strategies help in modifying an existing program to solve errors?</p> <p>What does it mean to become frustrated?</p> <p>How do I persevere through a problem and keep working when I become frustrated?</p> <p>How can I create a program using to complete images using sequential steps?</p> <p>When can a loop be used to simplify a repetitive action?</p>	<p>Algorithms Debugging Sequencing Persistence Frustration Program Programming Loop Repeat</p> <p>Students will be able to:</p> <ul style="list-style-type: none"> Reframe a sequence of steps as an encoded program. Identify and address bugs or errors in sequenced instructions. 	<p>Algorithm Bug Debugging Loop Program Repeat Programming Frustrated Persistence</p>	<p>Algorithms Debugging Sequencing Programming Loop</p>

	<p>a. formulate problem definitions suited for technology- assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.</p> <p>b. 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.</p> <p>c. break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.</p> <p>d. understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.</p>		<ul style="list-style-type: none"> • Order movement commands as sequential steps in a program. • Modify an existing program to solve errors. • Break down a long sequence of instructions into the largest repeatable sequence. • Outline steps to complete a structural engineering challenge. • Predict and discuss potential issues in structure creation. • Build a structure based on team plan. • Revise both the plan and structure until they satisfy the challenge. • Predict where a program will fail. • Modify an existing program to solve errors. 		
--	---	--	--	--	--

			<ul style="list-style-type: none"> • Reflect on the debugging process in an age-appropriate way. • Create a program to complete an image using sequential steps. • Break complex shapes into simple parts. • Identify repeated patterns in code that could be replaced with a loop. • Write instructions that use loops to repeat patterns. 		
--	--	--	--	--	--

Month(s): November - January	Marking Period 2
-------------------------------------	-------------------------

Unit Name

<u>Big Idea</u>	<u>CSTA K-12 Computer Science Standards</u>	<u>Essential Questions & Lesson Essential Question</u>	<u>Concepts/Objectives</u>	<u>Vocabulary</u>	<u>Competencies</u>
Sequencing Debugging Loops Nested Loops Conditionals	1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	When can a loop be used to simplify a repetitive action? What purpose do loops serve when writing a large set of code that repeats?	Loop Nested Loops Conditionals Personal Information Private Information Identity Theft	Loop Repeat Conditionals Condition While Loop Identity	Loop Nested Loops Conditionals

	<p>1B-AP-13 - Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> <p>1B-NI-05 - Discuss real-world cyber security problems and how personal information can be protected.</p> <p><u>ISTE Standards</u> Computational Thinker</p> <p>Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions. Students:</p> <p>a. formulate problem definitions suited for technology- assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.</p> <p>b. collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to</p>	<p>How do loops improve existing code that repeats?</p> <p>What is the difference between a loop and a nested loop?</p> <p>When is a loop, nested loop, or no loop needed when writing code?</p> <p>What role do conditionals play when writing code for a program?</p> <p>What are the circumstances when certain parts of a program should run and when they shouldn't?</p> <p>What criteria needs to be met to make something a conditional?</p> <p>How do you stay safe when you visit a website?</p> <p>What kinds of information should you keep to yourself when you use the Internet?</p>	<p>Students will be able to:</p> <ul style="list-style-type: none"> • Identify the benefits of using a loop structure instead of manual repetition. • Differentiate between commands that need to be repeated in loops and commands that should be used on their own. • Break complex tasks into smaller repeatable sections. • Recognize large repeated patterns as made from smaller repeated patterns. • Identify the benefits of using a loop structure instead of manual repetition. • Describe when a loop, nested loop, or no loop is needed. • Recognize the difference between using a loop and a nested loop. • Break apart code into 	<p>Theft Personal Information Private Information Register (online)</p>	
--	---	---	---	---	--

	<p>facilitate problem-solving and decision-making.</p> <p>c. break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.</p> <p>d. understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.</p> <p>Digital Citizen Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical. Students:</p> <p>a. cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.</p>		<p>the largest repeatable sequences using both loops and nested loops.</p> <ul style="list-style-type: none"> • Define circumstances when certain parts of a program should run and when they shouldn't. • Determine whether a conditional is met based on criteria. • Traverse a program and predict the outcome, given a set of input. • Define circumstances when certain parts of a program should run and when they shouldn't. • Determine whether a conditional is met based on criteria. • Learn about the benefits and risks of sharing information online. • Understand what type of information can put them at risk for identity theft and other scams. 		
--	---	--	---	--	--

	<p>b. engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.</p> <p>c. demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.</p> <p>d. manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.</p>				
--	---	--	--	--	--

Month(s): January - March	Marking Period 3
----------------------------------	-------------------------

Unit Name

<u>Big Idea</u>	<u>CSTA K-12 Computer Science Standards</u>	<u>Essential Questions & Lesson Essential Question</u>	<u>Concepts/Objectives</u>	<u>Vocabulary</u>	<u>Competencies</u>
Sequencing Debugging Loops Nested Loops Conditionals	<p>1B-AP-08 - Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-11 - Decompose (break down) problems</p>	<p>What role does an event play when coding a program?</p> <p>How can I use an event to help create an interactive program?</p> <p>What is a function?</p>	<p>Event Function Digital Sharing</p> <p>Students will be able to:</p>	<p>Event Function Copyright</p>	<p>Event Function</p>

	<p>into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 - Modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p><u>ISTE Standards</u> Computational Thinker</p> <p>Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions. Students:</p> <p>a. formulate problem definitions suited for technology- assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.</p> <p>b. collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways</p>	<p>How can functions make programs easier to write?</p> <p>How can predetermined functions be used to complete commonly repeated tasks?</p> <p>What is fair use?</p> <p>Why is it important to respect someone's work that they created in an online environment?</p>	<ul style="list-style-type: none"> • Create an animated, interactive game using sequence and events. • Identify actions that correlate to input events. • Locate repeating phrased inside song lyrics. • Identify sections of a song to pull into a function. • Describe how functions can make programs easier to write. • Use functions to simplify complex programs. • Use predetermined functions to complete commonly repeated tasks. • Interpret ethical sharing of copyrighted material vs. sharing that is not ethical. • Understand their own rights regarding materials that they have created. 		
--	---	---	--	--	--

	<p>to facilitate problem-solving and decision-making.</p> <p>c. break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.</p> <p>d. understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.</p>		<ul style="list-style-type: none"> • Categorize and generalize code into useful functions. • Recognize when a function could help to simplify a program. 		
--	---	--	--	--	--

Month(s): March - June	Marking Period 4
-------------------------------	-------------------------

Unit Name

<u>Big Idea</u>	<u>CSTA K-12 Computer Science Standards</u>	<u>Essential Questions & Lesson Essential Question</u>	<u>Concepts/Objectives</u>	<u>Vocabulary</u>	<u>Competencies</u>
Sequencing Debugging Loops Nested Loops Conditionals	<p>1B-AP-08 - Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 - Create programs that include sequences, events, loops, and conditionals.</p>	<p>What are some programming concepts and strategies that can be used to solve a problem?</p> <p>What is a sprite?</p> <p>How can I use an event to help create an interactive program?</p> <p>What are the steps in the design</p>	<p>Events</p> <p>Students will be able to:</p> <ul style="list-style-type: none"> • Recognize which programming concept to use to solve a given problem. 	Behavior Sprite Event Define Prepare Reflect Try	Events

	<p>1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 - Modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 - Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> <p>1B-AP-14 - Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-17 - Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>ISTE Standards</u> Computational Thinker</p>	<p>process?</p> <p>How can the design process help in creating a computer science project?</p>	<ul style="list-style-type: none"> • Describe the different ways one could solve a given problem. • Define "sprite" as a character or object on the screen that can be moved or changed. • Create a new sprite and choose its appearance. • Identify actions that correlate to input events. • Create an animated, interactive game using sequence and events. • Learn to plan in advance for an ongoing assignment. • Explain how system limitations can affect project design. • Describe how compromise can help keep a project on track and inspire creativity. • Shape ideas into reasonable goals and plans. 		
--	---	--	---	--	--

	<p>Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions. Students:</p> <ol style="list-style-type: none">a. formulate problem definitions suited for technology- assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.b. 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.c. break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.d. understand how automation works and use algorithmic		<ul style="list-style-type: none">• Recognize any potential obstacles such as time constraints or bugs.• Use a planned design as a blueprint for creation.• Overcome obstacles such as time constraints or bugs.• Articulate how the design process and how it helped shape the finished culminating project.		
--	--	--	--	--	--

	thinking to develop a sequence of steps to create and test automated solutions.				
--	---	--	--	--	--