

Chapter	Module/Lesson	Big Ideas / Topics / Concepts	Standards (CSTA)	Learning Objectives
Chapter 1	Module 1, Lesson 1	Sequences, Problem Solving	1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.	Understand how to use sequences and loops in programs
			2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Solve a problem using a sequence of code
	Module 1, Lesson 2	Objects, Methods, Debugging, Loops	1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	Decompose a problem into subproblems
			1B-CS-03: Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	Use a troubleshooting guide to identify and fix problems
	Module 1, Lesson 3-4	Capstone Project	1A-AP-10: Develop programs with sequences or simple loops, to express an idea or address a problem.	Debug programs that contain loops
			1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Debug programs that contain syntax and logic errors
	Module 1, Lesson 5	Impacts of Computing	2-CS-03: Systematically identify and fix problems with computing devices and their components	Use a troubleshooting guide to identify and fix problems
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.			Design a game that involves sequenced commands to build obstacles, revising a loop to program a boss's movement, and customizing art assets with methods and parameters.	
		1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	Develop programs using planning tools	
		1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.	Identify relevant problems and how they are solved using computer science/various computing devices.	
Chapter 2	Module 1, Lesson 1	Sequences and Algorithms, Syntax	1A-AP-10: Develop programs with sequences or simple loops, to express an idea or address a problem.	Modify sequences to complete a goal.
			1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.	Write algorithms using valid syntax to complete defined goals.
			1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	Decompose a problem into subproblems
			1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of development.	Explain what syntax is and why it is important. Define the term algorithm.
	Module 2, Lesson 1	Debugging	1A-AP-10: Develop programs with sequences or simple loops, to express an idea or address a problem.	Understand how and why to use sequences and loops in programs
			1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.	Solve a problem using a sequence of code
			1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Identify and fix code errors and logic problems.
			1A-AP-15: Using correct terminology, describe the steps taken and choices made during the iterative process of development.	Define and differentiate between code errors and logic problems. Explain the importance of debugging and iteration in the programming process.
	Module 2, Lesson 2	Impacts of Computing	2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	Understand the impact computers have on the world
	Module 3, Lesson 1-2	Variables (Creation and Updating)	1A-AP-15: Using correct terminology, describe the steps taken and choices made during the iterative process of development.	Explain what a variable is and why variables are useful in developing programs.
			1B-AP-09: Create programs that use variables to store and modify data.	Define and modify variables to help accomplish a goal.
Module 3, Lesson 3	Networks and the Internet	2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.	Use clear variable names.	
		1B-NI-04: Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.	Students will describe how the internet works: types of networks, packets, protocols, internet privacy	
		2-NI-04: Model the role of protocols in transmitting data across networks and the Internet.	Understand how data is transmitted	

Chapter	Module/Lesson	Big Ideas / Topics / Concepts	Standards (CSTA)	Learning Objectives
	Module 4, Lesson 1-3	Conditionals and Boolean Logic	1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.	Give examples of conditional logic in everyday life.
			1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.	Understand how and why to use sequences and loops in programs
			1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.	Solve a problem using a sequence of code
			1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.	Explain Boolean logic and evaluate simple Boolean expressions.
			1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.	Explain how conditions and decision points result in different algorithm execution.
			1B-AP-10: Create programs that include sequences, events, loops, and conditionals.	Use if and if/else statements to provide varied paths based on conditional logic.
	Module 4, Lesson 4	Computing Systems	1B-CS-01: Describe how internal and external parts of computing devices function to form a system.	Identify the parts of a computer and understand their purposes
Module 5, Lesson 1	Capstone Methods		Use a defined set of methods to create simple narrative sequences. Define conditions and resulting program functionality using user input.	
	Assessment/Capstone		Students demonstrate mastery of algorithms, debugging, variables, conditionals. Students create a storytelling project that provides an opportunity to showcase their creativity.	
<b>Chapter 3</b>	Module 1, Lessons 1-2	Debugging, Data Types, Objects, Variable Arithmetic	1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.	Identify and fix code errors and logic problems.
			1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.	Differentiate between objects, methods, and arguments.
			1B-AP-09: Create programs that use variables to store and modify data.	Update the value of variables using arithmetic operations.
			2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.	Explain the difference between the basic data types.
	Module 1, Lesson 3	Computing Systems	1B-CS-02: Model how computer hardware and software work together as a system to accomplish tasks.	Understand how hardware and software work together and the results achieved
	Module 2, Lesson 1	For Loops	1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.	Simplify code using for loops and repeated actions.
			1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.	Update the values of variables within loops.
			1B-AP-10: Create programs that include sequences, events, loops, and conditionals.	Include loops in sequence with other types of code statements.
	Module 3, Lesson 1	Nesting	1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.	Create programs using complex conditionals and nested conditionals
			1B-AP-10: Create programs that include sequences, events, loops, and conditionals.	Nest conditionals within loops.
2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.			Use nested structures to accomplish a given task.	
Module 3, Lesson 2	Cybersecurity	2-IC-23: Describe tradeoffs between allowing information to be public and keeping information private and secure.	Understand and explain the pros and cons of information security and data privacy	
Module 4, Lesson 1	While Loops	1A-AP-09: Model the way programs store and manipulate data by using numbers or other symbols to represent information.	Trace updating of a variable within a while loop.	
		1B-AP-09: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	Determine whether it is best to use a for loop or a while loop for a given puzzle.	
		1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	Use while loops to condense code.	

Chapter	Module/Lesson	Big Ideas / Topics / Concepts	Standards (CSTA)	Learning Objectives	
	Module 4, Lessons 2-3	Networks & the Internet	2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.	Encrypt data to secure information	
		Data & Analysis	2-DA-07: Represent data using multiple encoding schemes.	Apply data encryption methods to protect information	
	Module 5, Lesson 1	Capstone		1A-AP-13: Give attribution when using the ideas and creations of others while developing programs.	Using defined methods and art assets, design a side scrolling game that incorporates loops.
				1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	Use planning tools to create programs, use feedback from peers to improve program
		Data & Analysis	2-DA-08: Collect data using computational tools and transform the data to make it more useful and reliable.	Collect and clean survey data.	
			1A-DA-06: Collect and present the same data in various visual formats.	Visualize the data that's collected in various formats.	
		Impacts of Computing		2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	Understand how information is gathered using technology
				1B-AP-14: Observe intellectual property rights and give appropriate attribution when creating or remixing programs.	Understand appropriate methods for crediting others' ideas and property
				1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Understand how to gather and apply feedback to improve and debug a program
				1B-AP-16: Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	Give and receive constructive and relevant feedback to peers
				1B-AP-17: Describe choices made during program development using code comments, presentations, and demonstrations.	Be able to explain why certain choices are made using evidence
				2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Understand how to gather and apply feedback to improve and debug a program
		2-AP-19: Document programs in order to make them easier to follow, test, and debug.	Use engineering design tools to plan and create programs		
<b>Chapter 4</b>	Module 1, Lessons 1-2	Review, Else/If and Compound Conditionals, Testing and Debugging	1B-AP-10: Create programs that include sequences, events, loops, and conditionals.	Review previously learned concepts and apply in new contexts	
			2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Use previously learned concepts in more complex programs	
	Module 1, Lesson 3	Networks and the Internet	2-NI-05: Explain how physical and digital security measures protect electronic information.	Understand the purpose and benefits of securing information	
	Module 2, Lesson 1	Introduction to Functions	2-AP-16: Incorporate code, media, and libraries into original programs, and give attribution.		
	Module 2, Lesson 2	Data and Analysis	2-DA-09: Refine computational models based on the data they have generated.	Revise a computational model based on simulated data to reach specific objectives.	
	Module 2, Lesson 3-4	Computing Systems	2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.		
			1B-CS-01: Describe how internal and external parts of computing devices function to form a system.		
			2-CS-02: Design projects that combine hardware and software components to collect and exchange data.		
Module 3, Lesson 1	Writing Functions	2-AP-14: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.			
		2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.			
Module 4, Lesson 1	Capstone		2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.	Sandbox Game Design using Design Thinking Process	
			2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.		
			2-AP-17: Systematically test and refine programs using a range of test cases.		
			2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.		

<b>Chapter</b>	<b>Module/Lesson</b>	<b>Big Ideas / Topics / Concepts</b>	<b>Standards (CSTA)</b>	<b>Learning Objectives</b>
			2-AP-19: Document programs in order to make them easier to follow, test, and debug.	
	Module 4, Lesson 2	Impacts of Computing: Accessibility	2-IC-21: Discuss issues of bias and accessibility in the design of existing technologies.	
	Module 4, Lesson 3	Impacts of Computing: Bias and Stereotypes	2-IC-21: Discuss issues of bias and accessibility in the design of existing technologies.	