



South Carolina 4th Grade Computer Science and Digital Literacy Course Syllabus

One Year for Elementary School, 19 Hours

Course Overview and Goals

The **South Carolina 4th Grade Computer Science and Digital Literacy Course** introduces students to foundational programming concepts through **CodeHop**, a block-based programming language. Students explore digital literacy and computer science skills while engaging in lessons that integrate coding into math, science, English language arts (ELA), and social studies. This course emphasizes creativity, collaboration, and real-world connections, providing students with a strong foundation in both academic content, digital literacy, and computer science.

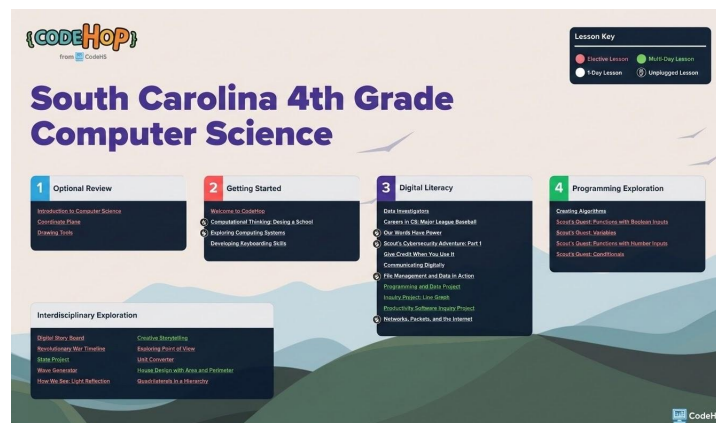
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging projects to showcase student understanding.

The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. Teachers can adapt the content to fit their schedule and instructional needs. The concepts taught in this course spiral across grade levels, ensuring that students can revisit and build upon their understanding year after year, even if all lessons are not completed within a single year. The course includes a total of **19 standards-aligned lessons** and **17 exploration lessons**, each approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week.

Programming Environment: Students will write and run programs in **CodeHop**, embedded and saved in students' accounts. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Prerequisites: There are no prerequisites for this course. It is designed to support all learners, regardless of prior computer science experience.

More Information: Browse the content of this course at <https://codehs.com/course/27368/overview>



A clickable PDF can be found at <https://codehs.com/SCRoadmap>

Course Breakdown

Unit 1: Optional Review (4 lessons)

This module reviews core programming skills, such as basic programming vocabulary, coordinate-plane movement, and custom sprite creation, to reinforce foundational coding abilities before beginning new content.

Objectives / Topics Covered	• Core computer science vocabulary • Basic programming • Using the coordinate plane for animation • Creating sprites and backdrops with drawing tools • Logging in and navigating digital learning tools
Lessons	Introduction to Computer Science • Define key computer science vocabulary and create a simple program. The Coordinate Plane • Create an open-ended animation using coordinate-plane positioning. Drawing Tools • Use drawing tools to design customized sprites and backdrops. Welcome to CodeHop! • Learn how to log in and navigate the CodeHop Playground.

Unit 2: Getting Started (3 lessons)

This module introduces foundational computer skills, including logging in, computational thinking, understanding computing systems, and developing keyboarding, to prepare students for effective technology use and early programming.

Objectives / Topics Covered	• Applying computational thinking to design tasks • Identifying computer system components and simple hardware/software issues • Practicing keyboarding with proper technique
Lessons	Computational Thinking: Design a School • Use computational thinking to plan and design a school by identifying patterns, sequencing steps, and breaking tasks down. Exploring Computing Systems • Identify computer system components and recognize basic hardware and software issues. Developing Keyboarding Skills • Practice typing paragraphs using proper finger placement.

Unit 3: Digital Literacy (16 lessons)

This module develops responsible digital citizenship by exploring data analysis, cybersecurity, cyberbullying, research and attribution, file management, productivity tools, and how networks and the Internet transmit information.

Objectives / Topics Covered	• Evaluating, analyzing, and drawing conclusions from data • Understanding digital citizenship: cyberbullying, digital footprints, online safety • Exploring cybersecurity concepts and common digital threats • Researching information and giving proper attribution • Using productivity tools (Docs, Slides, Sheets) to create and present work • Managing files and understanding digital storage • Communicating online and understanding how information travels across networks • Designing inquiry projects using visual data representations • Exploring CS-related careers and real-world applications
Lessons	Data Investigators • Evaluate data for reliability, analyze findings, and make predictions. Careers in CS: Health and Fitness • Explore how technology and coding support health and fitness, and create an animation demonstrating its use. Standing Up to Cyberbullying • Explain cyberbullying, its impact, how to act as an upstander, and how creative work online belongs to its creator. Scout's Cybersecurity Adventure: Part 1 • Learn basic cybersecurity concepts, recognize common threats, and apply practical safety strategies. Humans Vs. Machines • Compare and contrast human and computer performance on similar tasks, explain the advantages and limitations of technology, and describe computer perception. Give Credit When You Use It • Search for online information and provide proper attribution to sources. Communicating Digitally • Understand how to communicate digitally. File Management and Data in Action • Explain how data size varies and explore different ways digital files are stored. Programming and Data Project (2 classes) • Develop an investigative question, collect data, draw conclusions, and present findings in an interactive program. Inquiry Project: Line Graph (2 classes) • Follow the inquiry process and update a program to display investigation results using a line graph. Productivity Software Inquiry Project (3 classes) • Use document, spreadsheet, and presentation software to organize research and present information. Networks, Packets, and the Internet • Model how information travels through the Internet, including packet transfer and reassembly, and create a secure communication method.

Unit 4: Programming Exploration (4 lessons)

This module strengthens programming fluency through work with functions, Boolean and numeric inputs, variables, and conditionals as part of a skill-building series connected to interactive projects.

Objectives / Topics Covered	• Creating functions with Boolean and number inputs • Using variables to store and track information • Applying conditionals to control program behavior • Strengthening programming fluency through multi-part skill review
Lessons	Scout's Quest: Functions with Boolean Inputs • Create a function using a Boolean input to control actions based on password correctness. Scout's Quest: Variables • Use variables to track points within a program. Scout's Quest: Functions with Number Inputs • Create a drawing using functions that accept number inputs. Scout's Quest: Conditionals • Build a program using if/then conditionals to control behavior.

Unit 5: Interdisciplinary Exploration (13 lessons)

This module integrates programming with social studies, science, math, and literacy by using conditionals, variables, loops, events, and functions to model historical events, scientific processes, storytelling, geometry, and measurement.

Objectives / Topics Covered	• Modeling historical events and geography concepts with events and conditionals • Using variables, loops, and functions to demonstrate scientific and mathematical ideas • Building interactive storytelling projects • Creating programs that classify, convert units, or calculate measurements • Documenting thinking using comments and hierarchical classifications
Lessons	Digital Story Map • Analyze a passage to determine the main character and three important events, then use a digital storyboard to visually represent the events in the order they occurred. Revolutionary War Timeline • Create an interactive timeline using inputs, events, conditionals, and variables. • Standard: 4.2.CO State Project (2 classes) • Use events to build an interactive project about state-specific facts. Wave Generator • Use variables and loops to draw and animate wave patterns. • Standard: 4-PS4-1. How We See: Light Reflection • Use conditionals to model how light reflects off objects and enters the eye to allow vision. Creative Storytelling (2 classes) • Plan and animate an original story using events and sequences. Exploring Point of View • Identify and use first-person and third-person points of view by programming characters to retell a scene from a different perspective. • Standard: Unit Converter • Use variables and conditionals to build a unit-conversion calculator. • Standard: 4.MGSR.1.1 House Design with Area and Perimeter (2 classes)

	• Calculate and apply area and perimeter to design a house layout using functions. • Standard: 4.MGSR.1.1 Classifying Quadrilaterals in a Hierarchy • Use events to classify quadrilaterals in a hierarchy and document thinking with comments. • Standard: 4.MGSR.3.1–3.2
--	--

4th Grade Course Supplemental Materials

Resources	Description
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to their new computer science curriculum.
Warm-Up Activities	This warm-up activity slide deck provides 5-10 minute problems aligned with computer science skills to engage students at the start of class, allowing teachers to preview or review concepts with answer keys and discussion tips included in the Speaker Notes.
Program Self-Assessment (Spanish)	This is a student self-assessment tool designed to help K-6 learners reflect on their programming projects, evaluate their skills in algorithms, debugging, collaboration, and reflection, and set goals for improvement.
Peer Review Resources (Spanish)	This provides structured worksheets to facilitate student feedback during collaborative coding projects. It encourages reflection by guiding students to highlight successes, ask questions, and offer constructive feedback on their partner's work.
Lesson Reflection & Computational Thinking (Spanish)	This guides students in engaging with computational thinking concepts, preparing for discussions, reflecting on lessons, and applying their learning to real-world problem-solving.
Design-Your-Own-Lesson Programming Templates	Empower your students to explore and express their knowledge creatively with our versatile graphic organizer templates. Designed with adaptability and ease of use in mind, these interactive tools transform any subject into an engaging, hands-on learning experience.
These resources and more are found on the CodeHop Resources Page .	