South Carolina 5th Grade Computer Science and Digital Literacy Course Syllabus

One Year for Elementary School, 25 Hours

Course Overview and Goals

The **South Carolina 5th Grade Computer Science and Digital Literacy Course** introduces students to foundational programming concepts through **Scratch**, a block-based programming language. Students will develop computational thinking and problem-solving skills while learning to create interactive projects, animations, and games. This course emphasizes creativity and collaboration, providing students with a solid base in computer science concepts and digital literacy.

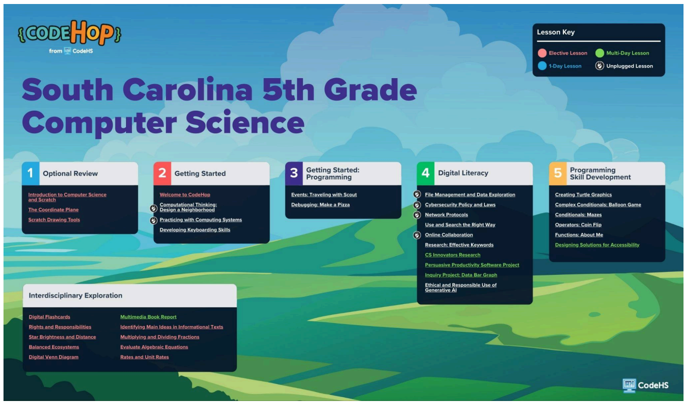
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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 **25 standards-aligned lessons** and **11 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 **Scratch** 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/27369/overview



{CODEHOP} from CodeHS

Lesson Key
● Elective Lesson ● Multi-Day Lesson
● 1-Day Lesson Unplugged Lesson

South Carolina 5th Grade Computer Science

1 Optional Review
- Introduction to Computer Science and Scratch
- The Coordinate Plane
- Scratch Drawing Tools

2 Getting Started
- Welcome to CodeHop
- Computational Thinking: Design a Neighborhood
- Practicing with Computing Systems
- Developing Keyboarding Skills

3 Getting Started: Programming
- Events: Traveling with Scout
- Debugging: Make a Pizza

4 Digital Literacy
- File Management and Data Exploration
- Cybersecurity Policy and Laws
- Network Protocols
- Use and Search the Right Way
- Online Collaboration
- Research: Effective Keywords
- CS Innovators Research
- Persuasive Productivity Software Project
- Inquiry Project: Data Bar Graph
- Ethical and Responsible Use of Generative AI

5 Programming Skill Development
- Creating Turtle Graphics
- Complex Conditionals: Balloon Game
- Conditionals: Mazes
- Operators: Coin Flip
- Functions: About Me
- Designing Solutions for Accessibility

Interdisciplinary Exploration
- Digital Flashcards
- Rights and Responsibilities
- Star Brightness and Distance
- Balanced Ecosystems
- Digital Venn Diagram
- Multimedia Book Report
- Identifying Main Ideas in Informational Texts
- Multiplying and Dividing Fractions
- Evaluate Algebraic Equations
- Rates and Unit Rates

CodeHS

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

Course Breakdown

Unit 1: Optional Review (3 optional lessons)

This module reinforces foundational Scratch skills—such as basic programming vocabulary, coordinate-plane animation, and custom sprite creation—to prepare students for new coding challenges.

Objectives / Topics Covered	● Understanding basic computer science vocabulary ● Creating simple Scratch programs ● Using the coordinate plane for animation ● Designing custom sprites and backdrops with drawing tools
Lessons	**Introduction to Computer Science and Scratch** ● Define key computer science vocabulary and create a simple Scratch program. **The Coordinate Plane** ● Create an open-ended Scratch animation using coordinate-plane movement. **Scratch Drawing Tools** ● Design customized sprites and backdrops using Scratch's drawing tools.

Unit 2: Getting Started (4 lessons)

This module builds core digital skills by introducing logging in, computational thinking, computer system basics, and keyboarding to support confident technology use and early programming success.

Objectives / Topics Covered	● Logging in and navigating CodeHop ● Applying computational thinking to design tasks ● Identifying computer system components and basic hardware/software issues ● Practicing keyboarding with proper technique
Lessons	**Welcome to CodeHop! (optional)** ● Learn how to log in and navigate the CodeHop Playground. **Computational Thinking: Design a Neighborhood** ● Use computational thinking skills to plan and design a neighborhood. **Practicing with Computing Systems** ● Identify computer components and troubleshoot basic hardware and software issues. **Developing Keyboarding Skills** ● Practice typing paragraphs using proper finger placement on the keyboard.

Unit 3: Getting Started: Programming (2 lessons)

This module introduces foundational programming concepts by teaching students how to use events and debugging strategies to create and correct Scratch programs.

Objectives / Topics Covered	● Using events to trigger actions in programs ● Debugging by decomposing sequences and correcting program behavior
Lessons	**Events: Traveling with Scout** ● Use events in Scratch to trigger character actions. **Debugging: Make a Pizza** ● Decompose a program to find and fix bugs so the sequence runs correctly.

Unit 4: Digital Literacy (13 lessons)

This module develops strong digital citizenship through lessons on data storage, cybersecurity, research skills, online collaboration, credible searching, productivity tools, inquiry projects, and responsible AI use.

Objectives / Topics Covered	● Understanding data size, storage, and file organization ● Exploring cybersecurity laws, policies, and common threats ● Explaining online collaboration and responsible communication ● Conducting effective online searches and evaluating source credibility ● Applying research skills to create informational programs ● Using productivity tools (Docs, Sheets, Slides) to organize and present information ● Modeling data using inquiry-based projects and visual graphs ● Understanding ethical and responsible use of generative AI
Lessons	**File Management and Data Exploration** ● Explain how digital data varies in size and explore where different types of files are stored. **Cybersecurity Policies and Laws** ● Research school technology policies and examine a cybersecurity law from your state. **Networks and Protocols** ● Explore how network protocols enable data transfer and compare WiFi, wired, and cellular networks. **Online Collaboration** ● Explain how collaboration can occur both online and offline, and how diverse perspectives improve a project. **Use and Search the Right Way** ● Search online for information and give proper credit to sources. **Research: Effective Keywords** ● Evaluate keyword choices in search engines and assess the credibility and relevance of sources. **CS Innovators Research Project (2 classes)** ● Research a computer science innovator and summarize key information in an informational program. ● Standard: SC/CCSS: 3.W.2 **Persuasive Productivity Software Project (2 classes)** ● Use word processing, spreadsheet, and presentation tools to organize and present persuasive information. ● Standard: ELA.5.C.1.1 **Inquiry Project: Data Bar Graph (2 classes)** ● Follow the inquiry process and update a program to display results using a bar graph. **Ethical and Responsible Use of Generative AI** ● Describe the benefits and drawbacks of generative AI and help create a class AI Code of Conduct.

Unit 5: Programming Skill Development (7 lessons)

This module deepens programming proficiency by exploring loops, conditionals, operators, functions, and design thinking while applying these skills to solve problems and improve program usability.

Objectives / Topics Covered	● Creating graphics with loops and pen tools ● Using conditionals, including if/then and if/then/else ● Applying variables and operators in interactive programs ● Building functions with inputs ● Redesigning programs using design thinking for accessibility
Lessons	**Creating Turtle Graphics** ● Use the pen tool and loops in Scratch to create turtle-style graphics. **Complex Conditionals: Balloon Game**

	● Explain and apply “if/then/else” conditionals in an interactive game. **Conditionals: Mazes** ● Create a maze program that uses conditionals to guide actions. **Operators: Coin Flip** ● Build a coin-flipping simulation using variables and operators. **Functions: About Me** ● Create and use a function with input to design an interactive “About Me” program. **Designing Solutions for Accessibility (2 classes)** ● Use the design thinking process to redesign a game to make it more accessible for diverse users.

Unit 6: Interdisciplinary Exploration (11 lessons)

This module integrates programming with science, math, literacy, and social studies by using variables, events, operators, simulations, and interactive designs to model real-world concepts and academic content.

Objectives / Topics Covered	● Creating digital flashcards using broadcast messages ● Modeling civic concepts like rights and responsibilities with variables and events ● Showing scientific ideas (stars, ecosystems, light reflection) using conditionals and simulations ● Producing multimedia academic projects ● Modeling math concepts such as fractions, equations, and unit rates ● Classifying geometric shapes and documenting code with comments
Lessons	**Digital Flashcards** ● Use broadcast messages to build digital flashcards for any subject area. ● Standard: **Rights and Responsibilities** ● Create a voting simulation that uses variables and events to model civic concepts. ● Standard: **Star Brightness and Distance** ● Demonstrate how distance affects star brightness using conditionals. ● Standard: 5-ESS1-1 **Balanced Ecosystems** ● Build an ecosystem simulation and adjust elements to show how populations change. ● Standard: 5-ESS3-1. **Digital Venn Diagram** ● Program an interactive Venn diagram to compare two things. **Multimedia Book Report (2 classes)** ● Create a program that communicates key information about a book. ● Standard: SL.5.5 **Identifying Main Ideas in Informational Texts** ● Identify the main idea of an informational text and explain 2–3 supporting details that show how the author develops that idea. ● Standard: ELA.AOR.2.2 **Multiplying and Dividing Fractions** ● Model multiplying and dividing fractions through a program that solves word problems. ● Standard: 5.PAFR.2.1, 5.PAFR.2.2 **Evaluate Algebraic Equations** ● Create a program using operators and conditionals to check and solve real-world algebraic equations. ● Standard: 5.PAFR.3.1–3.3 **Rates and Unit Rates** ● Design a unit rate converter using variables and operators. ● Standard: 5.NR.3.1

5th 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 Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch 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**.	