



South Carolina 2nd Grade Computer Science and Digital Literacy Course Syllabus

One Year for Elementary School, 19 Hours

Course Overview and Goals

The **South Carolina 2nd Grade Computer Science and Digital Literacy Course** introduces students to foundational programming concepts through **ScratchJr**, 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 **ScratchJr** 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/27366/overview>



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

Course Breakdown

Unit 1: Optional Review (5 optional lessons)

This module reinforces foundational ScratchJr skills—including navigation, events, loops, and message interactions—to prepare students for more advanced programming.

Objectives / Topics Covered	• Navigating the ScratchJr interface • Using events to trigger actions • Applying repeat and forever loops • Sending and receiving messages between characters
Lessons	Introduction to ScratchJr • Navigate the ScratchJr interface and create a scene using characters. Events • Explain events in programming and use event blocks to trigger actions. Introduction to Repeat Loops • Use repeat loops to run a section of code multiple times. Forever Loop Dance Party • Create a sequence with a “repeat forever” loop to make characters continuously repeat actions. Introduction to Message Events • Program a relay-race animation using message blocks to coordinate character interactions.

Unit 2: Getting Started (4 lessons)

This module introduces essential computer skills such as logging in, computational thinking, computer components, and keyboarding to support early success with technology and programming.

Objectives / Topics Covered	• Logging in and navigating the CodeHop Playground • Sequencing and decomposing routines using computational thinking • Understanding hardware, software, input, output, and computer connections • Typing simple words and sentences
Lessons	Welcome to CodeHop! (optional) • Log in and navigate the CodeHop Playground. Computational Thinking: School Day Routines • Identify patterns, sequence steps, and break down tasks using evening routines as an example. Computer Basics: Connections • Learn what a computer is, how it works, and how hardware, software, input, and output work together. Keyboarding Basics • Type simple words and sentences using basic keyboard functions.

Unit 3: Digital Literacy (15 lessons)

This module develops students' digital literacy by teaching positive online behavior, the impact of technology, productivity tools (word processing, spreadsheets, presentations), Internet communication, research skills, and basic data concepts.

Objectives / Topics Covered	● Safe and responsible digital behavior, including strong passwords and digital citizenship ● Digital collaboration and communication through networks and online tools ● Using productivity software such as word processing, presentations, and spreadsheets ● Researching information, evaluating sources, and interpreting data patterns ● Understanding data storage, file management, and how AI tools use data ● Exploring how technology and computer science connect to real-world careers
Lessons	Password Protectors ● Learn the importance of usernames and passwords and strategies to keep login information safe. Responsible Digital Citizens ● Understand digital footprints, cyberbullying, and appropriate ways to report online concerns. Collaborating Digitally (2 classes) ● Work with partners digitally to create a program about classroom experiences. Choice Research (2 classes) ● Gather information from sources and present findings visually. Exploring Computer Networks ● Describe how networks connect devices and model sending and receiving information using message blocks. Exploring Word Processing ● Use word processing tools to create a document about a favorite season. Exploring Presentations ● Create a presentation using basic slide-editing features to share personal interests. Exploring Spreadsheets ● Use spreadsheet software to review and present data on class pets and favorite colors. Data Patterns and Predictions ● Identify patterns and trends in data visualizations and create a program to communicate predictions. Managing Data Storage and Files ● Recognize how computers store information as files and model basic data storage. How AI Helps Us ● Observe how AI assistants answer questions and learn how AI supports people in different tasks. What Can AI Do? ● Identify tools that use AI, understand how AI learns from data, and compare human tasks vs. AI tasks. Careers in CS: Fashion-Retail ● Explore how coding influences fashion design and create a program to design and animate a character.

Unit 4: Programming Exploration (8 lessons)

This module strengthens students' programming abilities through grid-based movement, sequencing, events, debugging, multi-page projects, and game creation informed by peer feedback.

Objectives / Topics Covered	● Using the grid to move characters precisely ● Creating programs and revising them based on peer feedback ● Combining sequences, events, and pages to build games ● Designing multi-page interactive stories or games
Lessons	Introduction to the Grid ● Use the ScratchJr grid to move characters to specific locations. Two-Step Dance & Feedback ● Create a program, revise it using peer feedback, and credit helpers appropriately. Moving Targets Game (3 classes) ● Build a moving-target game using sequences, events, and multiple pages. Design an Adventure Game (3 classes) ● Create a story-based, multi-page adventure game using accumulated programming skills.

Unit 5: Interdisciplinary Exploration (10 lessons)

This module connects programming to academic subjects by using events, loops, conditionals, and animations to model concepts in geography, science, literacy, math, and financial understanding.

Objectives / Topics Covered	● Using events to model geography concepts like cardinal directions ● Applying conditionals to sort or classify scientific properties ● Using loops and messages to simulate natural processes ● Comparing erosion solutions through interactive models ● Creating literacy-based animations and sentence-building tools ● Modeling math concepts such as place value, equal parts, and coin values
Lessons	Cardinal Directions ● Use events to create an interactive compass rose. ● Standard: 2.G.1 Properties of Matter ● Use conditionals to program a robot that classifies matter based on its properties. ● Standard: 2-PS1-1 Seed Dispersal ● Model seed dispersal using message events and loops to show how animals help spread seeds. ● Standard: 2-LS2-2. Preventing Erosion ● Create a program comparing different solutions that prevent wind or water erosion. ● Standard: 2-ESS3-1. Build a Sentence ● Make an interactive program that uses events to build and read sentences aloud. ● Standard: 2.L.2 Storytelling Animations ● Retell a story in order using an animated program. ● Standard: 2.W.3 Punctuation: Write a Great Sentence! ● Use sequences and loops to write sentences with correct punctuation and spacing. Place Value: Ones, Tens, and Hundreds ● Program an activity that connects digit placement to number value using events. ● Standard: 2.NBT.1, 2.NBT.2 Divide Shapes into Equal Parts ● Animate circles and rectangles partitioned into equal parts.

	- Standard: 2.G.3 Events and Money - Create a program that identifies the value of a set of coins. - Standard: 2.MD.8
--	---

2nd 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.
These resources and more are found on the CodeHop Resources Page .	