



# 2nd Grade Computer Science Course Syllabus

## One Year for Elementary School, 36 Hours

### Course Overview and Goals

The **2nd Grade Computer Science Course** introduces students to foundational programming concepts through **ScratchJr**, 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.

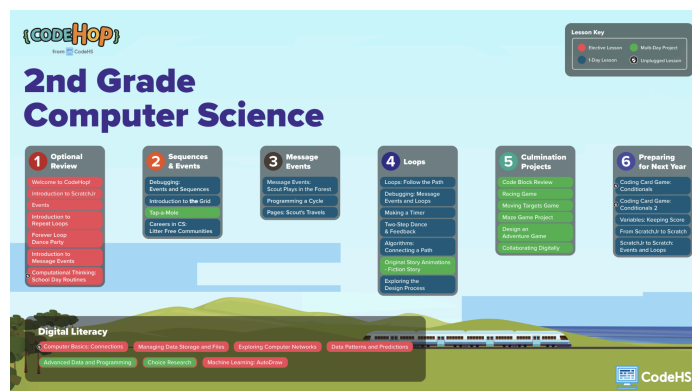
**Learning Environment:** This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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 **36 lessons**, each approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week. Optional digital literacy lessons are also available to complement the programming curriculum with non-programming computer and technology skills.

**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/20684/overview?lang=en>



A clickable PDF can be found at <https://codehs.com/K5CSRoadmaps>.

## Course Breakdown

### Unit 1: Optional Review (6 weeks)

In this optional unit, students review core concepts from earlier grades and prepare for deeper programming in ScratchJr by practicing events, loops, message interactions, and computational thinking strategies.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>Log in and navigate the CodeHop Playground and ScratchJr interface.</li><li>Create programs using events, repeat loops, and forever loops.</li><li>Coordinate character actions using message events.</li><li>Apply computational thinking to analyze and simplify real-life routines.</li></ul>
Lessons	<p><b>Welcome to CodeHop! (15 min lesson)</b></p> <ul style="list-style-type: none"><li>Log in and explore the CodeHop Playground as a quick warm-up or review activity.</li></ul> <p><b>Introduction to ScratchJr</b></p> <ul style="list-style-type: none"><li>Navigate the ScratchJr interface and create a simple animated scene with characters.</li></ul> <p><b>Events</b></p> <ul style="list-style-type: none"><li>Learn what events are and use multiple event blocks to trigger different character actions.</li></ul> <p><b>Introduction to Repeat Loops</b></p> <ul style="list-style-type: none"><li>Use repeat loops in ScratchJr to repeat actions a specific number of times.</li></ul> <p><b>Forever Loop Dance Party</b></p> <ul style="list-style-type: none"><li>Create a looping dance animation using the “repeat forever” loop for continuous movement.</li></ul> <p><b>Introduction to Message Events</b></p> <ul style="list-style-type: none"><li>Program a relay race using message events to make characters interact and respond in order.</li></ul> <p><b>Computational Thinking: School Day Routines</b></p> <ul style="list-style-type: none"><li>Use computational thinking skills to break down and sequence everyday school routines.</li></ul>

### Unit 2: Sequences and Events (5 weeks)

Students build on their understanding of sequences and events by creating interactive projects, practicing debugging, using the grid for movement, and exploring how computer science can solve real-world problems.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>Debug programs involving sequences and event blocks.</li><li>Use the ScratchJr grid to move characters with precision.</li><li>Create interactive event-driven games.</li><li>Explore real-world applications of computer science.</li></ul>
Lessons	<p><b>Debugging: Events and Sequences</b></p> <ul style="list-style-type: none"><li>Identify and fix errors in provided ScratchJr code that uses sequences and event blocks.</li></ul> <p><b>Introduction to the Grid</b></p> <ul style="list-style-type: none"><li>Use the grid feature in ScratchJr to move characters to specific locations with accuracy.</li></ul> <p><b>Tap-a-Mole Game (2-part lesson)</b></p> <ul style="list-style-type: none"><li>Create an interactive tap game that uses events to animate character reactions.</li></ul> <p><b>Careers in CS: Litter Free Communities</b></p> <ul style="list-style-type: none"><li>Learn how computer science can support community efforts and build a program to collect and sort litter.</li></ul>

### Unit 3: Message Events (3 weeks)

In this unit, students deepen their understanding of message events in ScratchJr by coordinating character actions, modeling cycles, and navigating multi-page programs with interactive storytelling.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>• Use message events to control the flow of actions between characters.</li> <li>• Model real-world processes like cycles using messages.</li> <li>• Navigate between pages in ScratchJr using message events to link scenes.</li> </ul>
Lessons	<p><b>Message Events: Scout Plays in the Forest</b></p> <ul style="list-style-type: none"> <li>• Use message events to control when and how characters perform actions in a forest-themed scene.</li> </ul> <p><b>Programming a Cycle</b></p> <ul style="list-style-type: none"> <li>• Model a repeating cycle, such as the water cycle or plant growth, using message events to show continuous progression.</li> </ul> <p><b>Pages: Scout's Travels</b></p> <ul style="list-style-type: none"> <li>• Help Scout move between pages by using message events to connect scenes and actions across multiple parts of a program.</li> </ul>

#### Unit 4: Loops (8 weeks)

Students deepen their understanding of loops by identifying patterns, debugging programs, building interactive timers and dances, and applying the design process to create algorithmic solutions in ScratchJr.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>• Identify and apply loops to repeat patterns in a program.</li> <li>• Debug programs involving message events and loops.</li> <li>• Use loops with wait and motion blocks to control timing and animation.</li> <li>• Apply algorithms and the design process to solve simple problems creatively.</li> </ul>
Lessons	<p><b>Loops: Follow the Path</b></p> <ul style="list-style-type: none"> <li>• Identify visual patterns and use loops to move characters along a path.</li> </ul> <p><b>Debugging: Message Events and Loops</b></p> <ul style="list-style-type: none"> <li>• Find and fix errors in ScratchJr programs that use both message events and loops.</li> </ul> <p><b>Making a Timer</b></p> <ul style="list-style-type: none"> <li>• Use loops, wait blocks, and turn blocks to create and compare two timers with different speeds.</li> </ul> <p><b>Two-Step Dance &amp; Feedback</b></p> <ul style="list-style-type: none"> <li>• Create a dance animation using loops and revise it based on peer feedback, giving attribution to those who contributed ideas.</li> </ul> <p><b>Algorithms: Connecting a Path</b></p> <ul style="list-style-type: none"> <li>• Develop and adjust simple algorithms that move characters based on size, shape, and starting position.</li> </ul> <p><b>Original Story Animations – Fiction Story (2-part lesson)</b></p> <ul style="list-style-type: none"> <li>• Create and animate an original fiction story using ScratchJr programming concepts.</li> </ul> <p><b>Exploring the Design Process</b></p> <ul style="list-style-type: none"> <li>• Plan, create, and improve a looping program that models a simple real-world solution using the design process.</li> </ul>

#### Unit 5: Culmination Projects (15 weeks)

Students apply everything they've learned by designing and building original games and projects that combine loops, events, sequences, messages, and pages—culminating in creative, interactive experiences.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>• Demonstrate understanding of various coding concepts.</li> <li>• Build interactive games and animations using sequences, loops, events, messages, and page navigation.</li> <li>• Apply the design process and peer feedback to improve projects.</li> <li>• Collaborate and reflect on coding through gameplay and storytelling.</li> </ul>
Lessons	<p><b>Code Block Review (2-part lesson)</b></p> <ul style="list-style-type: none"> <li>• Use a variety of coding blocks in a ScratchJr program and explain how each block functions within the sequence.</li> </ul>

	<p><b>Racing Game (2-part lesson)</b></p> <ul style="list-style-type: none"> <li>Create a racing game that uses events, loops, and messages to control character movement and interactions.</li> </ul> <p><b>Moving Targets Game (3-part lesson)</b></p> <ul style="list-style-type: none"> <li>Build a game using sequences, events, and pages where players tap moving targets as they appear.</li> </ul> <p><b>Maze Game Project (3-part lesson)</b></p> <ul style="list-style-type: none"> <li>Program an interactive maze game using events, messages, loops, and sequences, and improve it based on peer feedback.</li> </ul> <p><b>Design an Adventure Game (3-part lesson)</b></p> <ul style="list-style-type: none"> <li>Create a story-based, multi-page game that brings together learned programming skills to guide players through a branching experience.</li> </ul> <p><b>Collaborating Digitally (2-part lesson)</b></p> <ul style="list-style-type: none"> <li><i>Coming soon!</i> This lesson will focus on working together in digital environments.</li> </ul>
--	---

## Unit 6: Preparing for Next Year (5 weeks)

In this unit, students prepare to transition from ScratchJr to Scratch by reviewing conditionals, introducing variables, and exploring how events and loops function in the Scratch environment.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>Use conditionals in unplugged activities to build decision-making logic.</li> <li>Create programs with variables to simulate scorekeeping.</li> <li>Navigate the Scratch interface and transfer skills from ScratchJr.</li> <li>Build basic Scratch programs using events and loops together.</li> </ul>
Lessons	<p><b>Coding Card Game: Conditionals</b></p> <ul style="list-style-type: none"> <li>Work collaboratively to guide Scout through a maze using conditionals in a sequence of unplugged instructions.</li> </ul> <p><b>Coding Card Game: Conditionals 2</b></p> <ul style="list-style-type: none"> <li>Continue practicing conditionals with a new maze challenge that reinforces decision-making logic.</li> </ul> <p><b>Variables: Keeping Score</b></p> <ul style="list-style-type: none"> <li>Create a program that uses a variable to keep track of a score, introducing the concept of data storage in programming.</li> </ul> <p><b>From ScratchJr to Scratch</b></p> <ul style="list-style-type: none"> <li>Learn to navigate the Scratch interface and create a simple project to ease the transition from ScratchJr.</li> </ul> <p><b>ScratchJr to Scratch: Events and Loops</b></p> <ul style="list-style-type: none"> <li>Create a Scratch program that combines event triggers and loops to build a more dynamic and interactive experience.</li> </ul>

## Unit 7: Digital Literacy (10 weeks)

In this supplemental unit, students strengthen their digital literacy by exploring how computers store and share data, analyzing patterns through programming, and investigating how AI and networks function.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>Identify how computer components work together and how data is stored as files.</li> <li>Explore how networks connect devices and how message-based communication works.</li> <li>Analyze and visualize data patterns, trends, and predictions using code.</li> <li>Investigate AI technologies and apply responsible practices.</li> </ul>
Lessons	<p><b>Computer Basics: Connections</b></p> <ul style="list-style-type: none"> <li>Learn how computers work by identifying input, output, hardware, and software, and explain how these parts function together to solve problems.</li> </ul> <p><b>Managing Data Storage and Files</b></p> <ul style="list-style-type: none"> <li>Understand how computers store data as files and model the process of collecting, saving, and retrieving information.</li> </ul>

	<p><b>Exploring Computer Networks</b></p> <ul style="list-style-type: none"> <li>Describe how devices share information across networks and model message-sending and receiving using ScratchJr message blocks.</li> </ul> <p><b>Data Patterns and Predictions</b></p> <ul style="list-style-type: none"> <li>Analyze data visualizations to identify trends and create a ScratchJr program that communicates patterns and predictions using events.</li> </ul> <p><b>Advanced Data and Programming (3-part lesson)</b></p> <ul style="list-style-type: none"> <li>Develop a question, collect survey data, and present the results visually through a custom-coded program.</li> </ul> <p><b>Choice Research (2-part lesson)</b></p> <ul style="list-style-type: none"> <li>Collect and evaluate research sources to answer a question and create a program that visually communicates the findings.</li> </ul> <p><b>Machine Learning: AutoDraw</b></p> <ul style="list-style-type: none"> <li>Explore how AutoDraw uses artificial intelligence and classifiers to recognize and suggest drawings based on user input.</li> </ul>
--	--

## 2nd Grade Course Supplemental Materials

Resources	Description
<a href="#">Parent Welcome Letter (Spanish)</a>	Send this letter home to introduce families to their new computer science curriculum.
<a href="#">Warm-Up Activities</a>	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.
<a href="#">Program Self-Assessment (Spanish)</a>	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.
<a href="#">Peer Review Resources (Spanish)</a>	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.
<a href="#">Lesson Reflection &amp; Computational Thinking (Spanish)</a>	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 <a href="#">CodeHop Resources Page</a> .	