Interdisciplinary Computer Science
Second Grade Course Syllabus
One Year for Elementary School

Course Overview and Goals

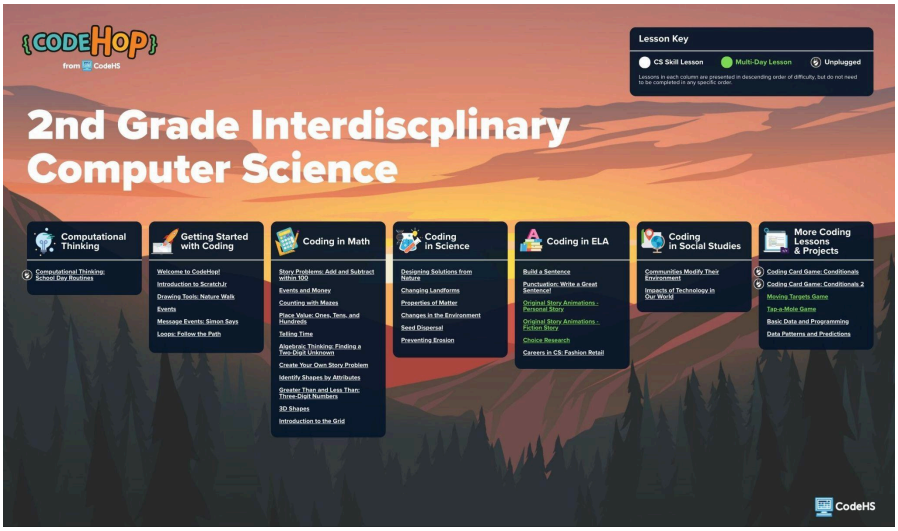
The Interdisciplinary Computer Science Second Grade Course introduces students to foundational programming and computational thinking concepts through **ScratchJr**, a block-based programming language designed for early learners. Students will develop coding skills while applying them across math, science, English language arts (ELA), and social studies. A dedicated **Computational Thinking** module reinforces practices such as decomposition, pattern recognition, abstraction, and algorithm design, helping students strengthen their problem-solving and critical thinking abilities in every subject area.

Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: Review, Guided Practice, Independent Practice, Extension, and Reflection. The course begins with a Computational Thinking module, which can be taught on its own or integrated into other subjects, followed by the Getting Started with Coding unit to establish basic programming skills. Teachers can then choose from subject-area coding lessons in any order, with topics arranged in approximate order of complexity to support flexible pacing. With **more than 36 lessons**, teachers can select activities that align with their instructional goals, while spiral review helps reinforce learning. The **"I do, we do, you do"** model supports gradual release of responsibility, building student independence and confidence.

Programming Environment: Students will write and run programs in ScratchJr within the CodeHS platform. This environment supports hands-on programming, enabling students to design, test, and debug interactive projects in a user-friendly space.

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/20690/overview



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

Course Breakdown

Computational Thinking

Students are introduced to the core practices of computational thinking. They learn how to break down tasks, recognize patterns, put steps in order, and simplify processes. These skills will serve as a foundation for problem-solving in coding and in real-world contexts.

Objectives / Topics Covered	● Identifying and recognizing patterns ● Breaking complex tasks into smaller steps ● Sequencing actions in logical order ● Simplifying processes for efficiency
Lessons	**Computational Thinking: School Day Routines** ● Identify patterns, break down tasks, and create a step-by-step sequence for daily routines.

Getting Started with Coding

Students are introduced to the CodeHop Playground and the ScratchJr interface. They learn to draw custom scenes, use events, send messages between characters, and create patterns with loops.

Objectives / Topics Covered	● Navigating ScratchJr and CodeHop ● Creating and customizing scenes ● Using multiple types of events and message events ● Recognizing and coding patterns with loops
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to ScratchJr** ● Explore the ScratchJr interface and create a simple scene by adding and moving characters. **Drawing Tools: Nature Walk** ● Use drawing tools to create and customize a nature walk scene. **Events** ● Learn what events are and use multiple event blocks. **Message Events: Simon Says** ● Communicate between characters using message events. **Loops: Follow the Path** ● Identify patterns and create looping sequences.

Coding in Math

Students use ScratchJr to visualize math concepts through interactive programs. Lessons cover operations, place value, time, algebraic thinking, geometry, and data on a grid.

Objectives / Topics Covered	● Modeling addition, subtraction, and place value ● Telling time with analog and digital clocks ● Applying algebraic thinking to solve for unknowns ● Creating and identifying 2D and 3D shapes ● Using greater than/less than comparisons ● Navigating the stage with the grid feature
Lessons	**Story Problems: Add and Subtract within 100**

	• Animate a math story problem. **Events and Money** • Show coin values with interactive events. **Counting with Mazes** • Design a maze that requires counting steps. **Place Value: Ones, Tens, and Hundreds** • Model place value with interactive events. **Telling Time** • Create clocks that show time in two formats. **Algebraic Thinking: Finding a Two-Digit Unknown** • Visualize and solve for unknowns in math stories. **Create Your Own Story Problem** • Animate original addition or subtraction problems. **Identify Shapes by Attributes** • Draw and identify shapes by their properties. **Greater Than and Less Than: Three-Digit Numbers** • Compare three-digit numbers in an interactive program. **3D Shapes** • Draw and quiz others on 3D shapes. **Introduction to the Grid** • Move characters to specific grid locations.

Coding in Science

Students explore science concepts through coding by modeling environmental adaptations, land changes, properties of matter, and more.

Objectives / Topics Covered	• Showing how nature inspires human solutions • Modeling environmental changes and erosion prevention • Grouping matter by its properties • Depicting natural processes like seed dispersal
Lessons	**Designing Solutions from Nature** • Show how humans mimic nature to solve problems. **Changing Landforms** • Animate a volcanic eruption and its effects. **Properties of Matter** • Group matter using conditionals. **Changes in the Environment** • Show causes and effects of environmental change. **Seed Dispersal** • Model seed dispersal with message events and loops. **Preventing Erosion** • Compare solutions to slow or prevent erosion.

Coding in ELA

Students use ScratchJr to develop literacy skills by building sentences, writing stories, conducting research, and exploring careers in computer science.

Objectives / Topics Covered	• Writing complete sentences with correct punctuation • Animating personal and fictional stories • Conducting and presenting research • Exploring CS careers in creative industries

Lessons	**Build a Sentence** ● Create interactive sentences with audio playback. **Punctuation: Write a Great Sentence!** ● Use loops to write correctly punctuated sentences. **Original Story Animations – Personal Story** ● Animate a personal story. **Original Story Animations – Fiction Story** ● Create and animate a fictional story. **Choice Research** ● Research a topic and present findings visually. **Careers in CS: Coding for Fashion-Retail** ● Show how coding is used in fashion design.

Coding in Social Studies

Students code projects that explore how communities interact with and change their environment, as well as the role of technology in the world.

Objectives / Topics Covered	● Modeling how communities change their environment ● Demonstrating the effects of technology on society
Lessons	**Communities Modify Their Environment** ● Animate examples of people changing their surroundings. **Impacts of Technology in Our World** ● Show positive and negative ways technology influences society.

More Coding Lessons & Projects

Students apply coding skills to create games, work with conditionals, and visualize data.

Objectives / Topics Covered	● Using conditionals in sequences ● Building interactive games ● Collecting and visualizing data ● Recognizing and predicting patterns
Lessons	**Coding Card Game: Conditionals** ● Use conditionals to move Scout through a maze. **Coding Card Game: Conditionals 2** ● Continue conditional maze challenges. **Moving Targets Game** ● Create a moving-target game with events and pages. **Tap-a-Mole Game** ● Make an interactive tapping game. **Basic Data and Programming** ● Collect and present data visually. **Data Patterns and Predictions** ● Identify patterns in data and create prediction animations.

Interdisciplinary Computer Science Second 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**.	