Interdisciplinary Computer Science First Grade Course Syllabus

One Year for Elementary School

Course Overview and Goals

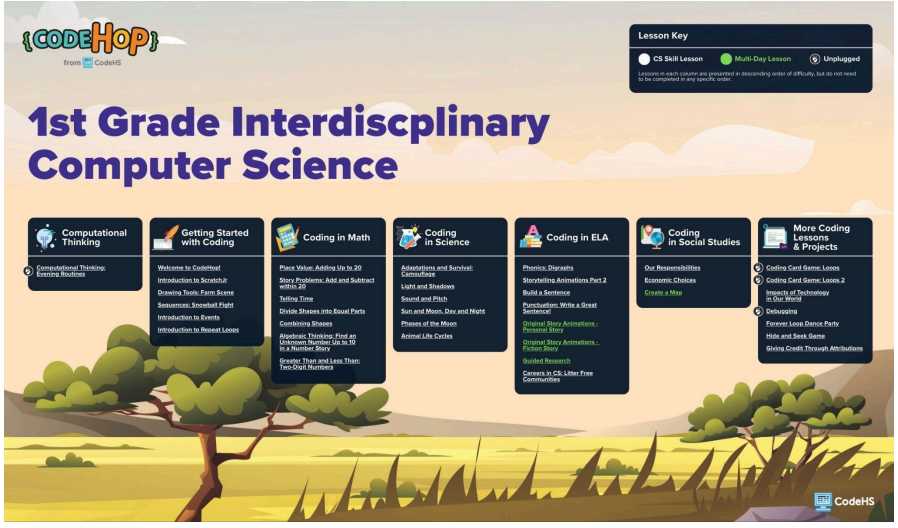
The Interdisciplinary Computer Science First Grade Course introduces students to foundational programming and computational thinking concepts through **ScratchJr**, a block-based programming language designed for early learners. Students will develop coding skills while applying them across math, science, 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/20689/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: Evening 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 learn to navigate ScratchJr. They explore key tools, create custom graphics, and use events and loops to animate characters.

Objectives / Topics Covered	● Navigating ScratchJr and CodeHop ● Creating and customizing characters and backgrounds ● Writing multiple sequences of code ● Using events and repeat loops to control actions
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: Farm Scene** ● Customize characters and backgrounds with the drawing tools. **Sequences: Snowball Fight** ● Build multiple sequences to animate a winter scene. **Introduction to Events** ● Trigger actions using different types of events. **Introduction to Repeat Loops** ● Use loops to repeat sections of code multiple times.

Coding in Math

Students connect math concepts with coding by creating interactive activities that model place value, operations, time, shapes, and number comparisons.

Objectives / Topics Covered	● Modeling place value, addition, and subtraction ● Telling and displaying time ● Partitioning and combining shapes ● Using algebraic thinking and number comparisons
Lessons	**Place Value: Adding Up to 20** ● Decompose two-digit numbers into tens and ones. **Story Problems: Add and Subtract within 20**

	● Animate a math story problem. **Telling Time** ● Create an analog clock and display times in digital and analog formats. **Divide Shapes into Equal Parts** ● Animate partitioning of shapes into equal parts. **Combining Shapes** ● Create composite shapes and animate them. **Algebraic Thinking: Find an Unknown Number Up to 10 in a Number Story** ● Visualize and solve for unknown numbers. **Greater Than and Less Than: Two-Digit Numbers** ● Compare numbers using animated programs.

Coding in Science

Students use ScratchJr to create animated models of scientific ideas, from adaptations to life cycles and astronomical patterns.

Objectives / Topics Covered	● Showing adaptations for survival ● Modeling light, sound, and pitch ● Illustrating celestial patterns like day/night and moon phases ● Depicting life cycles of animals
Lessons	**Adaptations and Survival: Camouflage** ● Animate how animals use camouflage to avoid predators or catch prey. **Light and Shadows** ● Show sources of light and how light creates shadows. **Sound and Pitch** ● Model the relationship between vibration and pitch. **Sun and Moon, Day and Night** ● Use loops to represent daily celestial movement. **Phases of the Moon** ● Use message events to animate the Moon's phases. **Animal Life Cycles** ● Show the butterfly life cycle with message events for each stage.

Coding in ELA

Students integrate literacy skills into coding by creating phonics programs, retelling stories, writing sentences, and conducting research.

Objectives / Topics Covered	● Practicing phonics with digraphs ● Retelling and creating stories ● Writing with proper punctuation and spacing ● Conducting and presenting research ● Exploring careers in computer science
Lessons	**Phonics: Digraphs** ● Create a digraph phonics activity using events and recordings. **Storytelling Animations Part 2** ● Retell a story in sequence. **Build a Sentence** ● Program sentences with audio playback. **Punctuation: Write a Great Sentence!** ● Write sentences with correct punctuation and spacing.

	Original Story Animations – Personal Story ● Animate a personal story. **Original Story Animations – Fiction Story** ● Develop and animate a fictional story. **Guided Research** ● Present research visually through animation. **Careers in CS: Litter Free Communities** ● Show how CS can solve community problems.

Coding in Social Studies (3 lessons)

Students explore responsibility, economics, and mapping skills by coding interactive projects.

Objectives / Topics Covered	● Understanding personal and civic responsibilities ● Recognizing economic decision-making ● Creating and following maps
Lessons	**Our Responsibilities** ● Animate scenarios showing responsible behavior in school and at home. **Economic Choices** ● Use message events to illustrate making choices between wants and needs. **Create a Map** ● Design a map and program a character to travel it.

More Coding Lessons & Projects (11 lessons)

Students extend their skills with loops, debugging, forever loops, games, and digital citizenship practices.

Objectives / Topics Covered	● Using loops to simplify code ● Debugging to fix errors ● Building interactive games ● Giving proper credit when reusing content
Lessons	**Coding Card Game: Loops** ● Use loops to move Scout through a maze. **Coding Card Game: Loops 2** ● Continue practicing loops with a maze challenge. **Impacts of Technology in Our World** ● Show how technology impacts daily life. **Debugging** ● Identify and fix sequence bugs. **Forever Loop Dance Party** ● Make characters repeat actions forever. **Hide and Seek Game** ● Build an interactive hide-and-seek activity. **Giving Credit Through Attributions** ● Reuse parts of a story and give proper credit.

Interdisciplinary Computer Science First Grade Course
Supplemental Materials

Resources	Description
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
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.
All of these resources and more are found on the **CodeHop Resources Page**.	