Interdisciplinary Computer Science Kindergarten Course Syllabus

One Year for Elementary School

Course Overview and Goals

The Interdisciplinary Computer Science Kindergarten course introduces students to foundational programming and computational thinking through a block-based programming environment designed for early learners. Students apply coding concepts across mathematics, science, English language arts (ELA), and social studies while developing problem-solving and critical thinking skills.

Learning Environment: This teacher-led course includes ready-to-use lesson plans that follow a consistent instructional format: Review, Guided Practice, Independent Practice, Extension, and Reflection. The course begins with the Computational Thinking module, followed by the Getting Started with Coding unit. After completing these introductory units, teachers may teach lessons from the mathematics, science, English language arts (ELA), social studies, and More Coding sections in the order that best supports their instructional goals. Lessons within each section progress in approximate order of complexity, and spiral review opportunities help reinforce previously learned concepts.

Programming Environment: Students write and run programs within the CodeHop platform using a block-based programming environment.

Assessments: Some units conclude with a brief two-question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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/20688/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
Lessons	**Computational Thinking: Morning Routines** ● Identify patterns, break down tasks, and create a step-by-step sequence for daily routines. **Computational Thinking: Morning Routines Quiz (2 questions)**

Getting Started with Coding

Students become familiar with the CodeHop Playground and the coding environment through a story-driven series called Scout Adventures. They learn to add characters, use motion blocks, and create interactive sequences.

Objectives / Topics Covered	● Navigating the programming environment ● Adding, modifying, and deleting characters ● Using motion blocks to move sprites ● Creating sequences of commands ● Adding backgrounds, pages, and events
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Scout Adventures 1: Introducing Scout** ● Explore the interface and add characters. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a new page. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters. **Scout Adventures 4: Scout Explores the Forest** ● Use motion blocks to move characters. **Scout Adventures 5: Scout and Bluebird Help** ● Build sequences to collect objects. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene using multiple pages, characters, and events. **Getting Started with Coding Quiz (2 questions)**

Coding in Math

Students apply coding skills to math concepts such as number decomposition, place value, patterns, and shape creation. They use events, loops, and sequences to create interactive math activities.

Objectives / Topics Covered	● Visualizing addition, subtraction, and place value ● Programming interactive math story problems ● Creating and combining shapes ● Comparing numbers and building patterns with music

Lessons	**Decompose Numbers Up to 10** ● Use events to show number decomposition. **Finding an Unknown in a Word Problem** ● Program number stories with missing values. **Story Problems: Add and Subtract within 10** ● Animate addition or subtraction problems. **Creating Shapes** ● Program shapes to respond to user interaction and combine into complex shapes. **Greater Than and Less Than: Single-Digit Numbers** ● Create a comparison program. **Place Value: Ones and Tens** ● Animate counting by ones, fives, and tens. **Patterns and Music** ● Use sequences and loops to create musical patterns.

Coding in Science

Students explore science topics through coding, modeling real-world phenomena such as weather, motion, and the needs of living things.

Objectives / Topics Covered	● Classifying living and nonliving things ● Modeling weather, seasons, and survival needs ● Comparing organisms and types of motion ● Showing plant needs using events and messages
Lessons	**Living and Nonliving** ● Classify and animate living vs. nonliving. **Weather and Seasons** ● Animate seasonal weather changes. **How Living Things Survive** ● Illustrate survival needs. **Comparing Organisms** ● Group animals by characteristics. **Types of Motion** ● Program animations for different motion types. **Needs of Plants** ● Model plant needs using events and messages.

Coding in ELA

Students strengthen literacy skills by coding interactive phonics activities, building sentences, retelling stories, and conducting research.

Objectives / Topics Covered	● Creating phonics and reading comprehension games ● Building and reading sentences aloud ● Animating story retellings and original stories ● Using research to create visual programs ● Connecting CS to careers
Lessons	**Phonics: Letter Sounds** ● Create a phonics game using the “on tap” event and "record audio" block. **Draw and Tell** ● Create a short scene in CodeHopJr and write/dictate a sentence about the scene they created. **Build a Sentence**

	● Create an interactive program that uses events to write sentences and then read them aloud. **Storytelling Animations** ● Create a program that retells the details of a story in the correct order. **Original Story Animations – Personal Story** ● Design a program to animate an original story and describe their program development process. **Introduction to Research** ● Find information using research sources and create a program to communicate their research visually. **Careers in CS: Spacecraft Design** ● Explain advantages of using technology in different careers and create a space program to animate an astronaut and spacecraft.

Coding in Social Studies

Students use coding to represent social studies concepts such as roles in the community and spatial awareness.

Objectives / Topics Covered	● Identifying authority figures and their roles ● Using cardinal directions and maps ● Understanding impacts of computing in daily life
Lessons	**Who Keeps Us Safe?** ● Create a program that describes the role of an authority figure. **Cardinal Directions** ● Use events to create an interactive compass rose. **Impacts of Computing: Daily Life** ● Identify examples of computing devices in their daily lives and explain how technology has changed the way people live, work, and play.

More Coding Lessons & Projects

Students extend their coding skills with games, special effects, and debugging activities. They explore loops, size changes, speed control, and more.

Objectives / Topics Covered	● Using loops, grow/shrink, and speed blocks ● Debugging code and adjusting game difficulty ● Creating animations with message events
Lessons	**Coding Card Game: Sequences** ● Work together to create a sequence of instructions to move Scout through a maze. **Coding Card Game: Sequences 2** ● Work together to create a sequence of instructions to move Scout through a maze. **Loops** ● Create a program using loops and explain how loops are used to repeat code. **Introduction to Grow and Shrink Blocks** ● Create a program that uses "grow" and "shrink" blocks to change the size of characters. **Introduction to Speed Blocks** ● Create a program that uses different speed blocks to animate characters. **Introduction to Debugging** ● Find and correct bugs in sequences. **Chicken Crossing Game** ● Program an interactive player character and adjust the difficulty of play in a game. **Garden Project** ● Use message events, grow, shrink, hide, and show blocks to animate seeds growing in a garden.

Interdisciplinary Computer Science Kindergarten 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**.	