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

Course Overview and Goals

The Interdisciplinary Computer Science Fourth Grade Course introduces students to foundational programming and computational thinking concepts through **Scratch**, 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 Scratch 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/20693/overview

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

Course Breakdown

Computational Thinking

Students begin by applying computational thinking practices to solve a design challenge. They will learn to identify patterns, break down complex problems into smaller steps, and create organized plans to bring their ideas to life.

Objectives / Topics Covered	● Applying computational thinking to design challenges ● Sequencing steps to complete a project ● Identifying and simplifying complex processes
Lessons	**Computational Thinking: Design a School** ● Use computational thinking skills to design a school by identifying patterns, breaking down steps, and sequencing instructions.

Getting Started with Coding

Students learn to navigate the CodeHop Playground and begin programming in Scratch. They will explore events, loops, and other foundational coding skills.

Objectives / Topics Covered	● Navigating the CodeHop platform ● Using events and loops in Scratch programs ● Understanding basic computer science vocabulary
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to Computer Science and Scratch** ● Define computer science terms and create a simple Scratch program. **Events: Dot in Space** ● Create a program using multiple event blocks. **Loops** ● Learn how loops repeat instructions and use them in Scratch.

Coding in Math

Students integrate math concepts with coding by creating interactive programs that explore geometry, factors, division, multiplication, fractions, and data visualization.

Objectives / Topics Covered	● Classifying shapes by properties ● Modeling math operations with code ● Using variables, operators, and functions in math-based projects ● Designing visual data displays
Lessons	**Classifying Shapes Using Lines and Angles** ● Categorize shapes and add code comments. **Animating Sprites with Factors** ● Use factors, loops, and waits to animate sprites. **Division and Conditionals** ● Build an interactive division game using conditionals. **Multi-digit Multiplication and Conditionals** ● Review multiplication with if/then logic. **Multiplying Fractions by Whole Numbers** ● Demonstrate fraction multiplication with animations.

	Programming and Data Project ● Collect and present data in an interactive program. **Naming Numbers Game** ● Use variables and operators to match number forms. **House Design with Area and Perimeter (2 part lesson)** ● Design a floor plan using functions, area, and perimeter. **Inquiry Project: Data Bar Graph** ● Display investigative results in a bar graph.

Coding in Science

Students use Scratch to explore and model science concepts such as heat, senses, cycles, sound, cells, light, astronomy, and waves.

Objectives / Topics Covered	● Modeling natural processes through animation ● Using variables, events, and conditionals to explain science topics ● Creating interactive science visualizations
Lessons	**Exploring Heat** ● Show how sunlight affects objects on Earth. **Sensing and Responding to the Environment** ● Model animal sensory responses with events. **Exploring the Water Cycle** ● Animate the water cycle using broadcast messages. **Sound Frequency & Amplitude** ● Use variables to adjust sound and vibration strength. **Plant and Animal Cells** ● Create interactive diagrams of cell structures. **How We See: Light Reflection** ● Use conditionals to model light reflection into the eye. **Star Brightness and Distance** ● Show how distance affects a star's brightness. **Wave Generator** ● Draw wave patterns with variables and loops.

Coding in ELA

Students create interactive literacy activities, games, and stories, applying programming concepts to strengthen reading, writing, and grammar skills.

Objectives / Topics Covered	● Using conditionals to create interactive quizzes ● Animating poetry, nonfiction, and creative writing ● Incorporating lists and events into ELA projects
Lessons	**Grammar Quiz Game** ● Build a grammar review quiz using conditionals. **Creative Storytelling** ● Plan and animate an original story. **Animating Poetry** ● Animate and narrate poetry with visual meaning. **Nonfiction Animated Recordings** ● Create an animated nonfiction reading. **Mad Libs Project** ● Build a Mad Libs game using lists. **Careers in CS: Health and Fitness**

	● Show how technology supports health and fitness.

Coding in Social Studies

Students explore civics, state research, and computer science history through interactive coding projects.

Objectives / Topics Covered	● Demonstrating rights and responsibilities ● Researching and presenting historical or cultural topics ● Creating interactive informational programs
Lessons	**Rights and Responsibilities** ● Create a voting simulation with variables and events. **State Project** ● Present state facts in an interactive format. **CS Innovators Research Project** ● Research and present information about a computer science innovator.

More Coding Lessons & Projects

Students expand their coding skills with conditionals, variables, digital tools, and game design projects.

Objectives / Topics Covered	● Applying conditionals, including complex logic, to interactive programs ● Using variables and broadcast messages ● Designing and debugging games and digital media projects
Lessons	**Scout's Quest: Conditionals** ● Use if/then logic in a program. **Conditionals: Color Sense** ● Apply conditionals to detect and respond to colors. **Complex Conditionals: Balloon Game** ● Use if/then/else logic to enhance gameplay. **Scout's Quest: Variables** ● Track points with variables. **Digital Flashcards** ● Build subject-specific flashcards using broadcasts. **Scratch Digital Greeting Card** ● Create an animated greeting card. **Platform Game Design** ● Build and debug a platform game with player movement and game condition checks.

Utah Interdisciplinary Computer Science Fourth 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.
Design-Your-Own-Lesson Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch graphic organizer templates. Designed with adaptability and ease of use in mind, these interactive tools transform any subject into an engaging, hands-on learning experience.
These resources and more are found on the **CodeHop Resources Page**.	