



Interdisciplinary Computer Science

Third Grade Course Syllabus

One Year for Elementary School

Course Overview and Goals

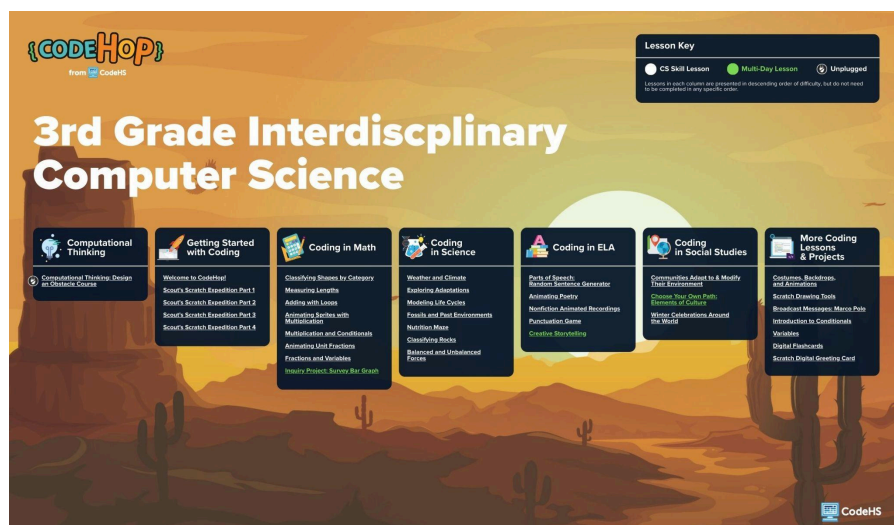
The Interdisciplinary Computer Science Third Grade Course introduces students to foundational programming and computational thinking concepts through **Scratch**, a block-based programming language designed for early learners. Students will develop coding skills while applying them across math, science, 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/20691/overview>



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

Course Breakdown

Computational Thinking

Students are introduced to computational thinking concepts through an engaging design challenge. They will identify patterns, break down complex ideas, and sequence steps to plan and create a functional obstacle course.

Objectives / Topics Covered	<ul style="list-style-type: none">• Understanding and applying computational thinking concepts• Designing a process from idea to execution• Sequencing steps and simplifying complex tasks
Lessons	Computational Thinking: Design an Obstacle Course <ul style="list-style-type: none">• Use computational thinking skills to design an obstacle course by identifying patterns, breaking down steps, and creating a step-by-step plan.

Getting Started with Coding

Students explore the CodeHop Playground and begin the sequential Scout's Scratch Expedition series. They will learn the basics of Scratch, including sprites, sequences, events, looks, and motion.

Objectives / Topics Covered	<ul style="list-style-type: none">• Navigating CodeHop and Scratch• Creating and animating sprites• Using sequences, loops, and events to tell a story
Lessons	Welcome to CodeHop! (15 minute lesson) <ul style="list-style-type: none">• Learn how to log in and use the CodeHop Playground. Scout's Scratch Expedition Part 1 <ul style="list-style-type: none">• Use basic Scratch commands to move and talk. Scout's Scratch Expedition Part 2 <ul style="list-style-type: none">• Add sprites and animate a story sequence. Scout's Scratch Expedition Part 3 <ul style="list-style-type: none">• Animate a Scout story using loops, events, and looks. Scout's Scratch Expedition Part 4 <ul style="list-style-type: none">• Continue the Scout story with motion, events, and looks..

Coding in Math

Students integrate math concepts with Scratch programming to create interactive projects that explore classification, measurement, operations, fractions, and variables.

Objectives / Topics Covered	<ul style="list-style-type: none">• Classifying and measuring shapes• Modeling operations with whole numbers and fractions• Using loops, conditionals, and variables in math projects• Creating and interpreting graphs
Lessons	Classifying Shapes by Category <ul style="list-style-type: none">• Classify quadrilaterals by properties. Measuring Lengths <ul style="list-style-type: none">• Create an interactive ruler and plot measurements. Adding with Loops <ul style="list-style-type: none">• Use loops to add multi-digit numbers. Animating Sprites with Multiplication <ul style="list-style-type: none">• Animate sprite movement using multiplication and wait blocks.

	Multiplication and Conditionals <ul style="list-style-type: none"> Review multiplication with if/then blocks. Animating Unit Fractions <ul style="list-style-type: none"> Show repeated addition of unit fractions. Fractions and Variables <ul style="list-style-type: none"> Represent fractions on a number line with variables and conditionals. Inquiry Project: Survey Bar Graph <ul style="list-style-type: none"> Follow the inquiry process and display survey results in a bar graph.
--	---

Coding in Science

Students build interactive science models to explore weather, adaptations, life cycles, nutrition, geology, and forces.

Objectives / Topics Covered	<ul style="list-style-type: none"> Using events to model scientific processes Applying conditionals to demonstrate cause and effect Creating interactive representations of life, earth, and physical sciences
Lessons	Weather and Climate <ul style="list-style-type: none"> Use data and events to model typical weather conditions. Exploring Adaptations <ul style="list-style-type: none"> Show how animal adaptations aid survival. Modeling Life Cycles <ul style="list-style-type: none"> Animate the frog life cycle with broadcast messages. Fossils and Past Environments <ul style="list-style-type: none"> Model fossil evidence with conditionals and loops. Nutrition Maze <ul style="list-style-type: none"> Create a maze game about healthy food choices. Classifying Rocks <ul style="list-style-type: none"> Sort rocks using if/then statements and the Mohs Hardness Scale. Balanced and Unbalanced Forces <ul style="list-style-type: none"> Model how forces affect movement using conditionals and variables.

Coding in ELA

Students use Scratch to enhance literacy skills by creating interactive stories, poetry animations, and grammar games.

Objectives / Topics Covered	<ul style="list-style-type: none"> Building random sentence generators Animating poetry and nonfiction Creating games to practice punctuation Developing storytelling skills
Lessons	Parts of Speech: Random Sentence Generator <ul style="list-style-type: none"> Create random sentences with nouns, verbs, adjectives, and adverbs. Animating Poetry <ul style="list-style-type: none"> Animate and narrate a poem with visual and literary meaning. Nonfiction Animated Recordings <ul style="list-style-type: none"> Create an animated reading of nonfiction text. Punctuation Game <ul style="list-style-type: none"> Build a game to add punctuation using if/then/else blocks. Creative Storytelling <ul style="list-style-type: none"> Plan and animate an original story.

Coding in Social Studies

Students use programming to model environmental adaptation, explore culture, and celebrate global traditions.

Objectives / Topics Covered	<ul style="list-style-type: none"> • Animating how communities adapt to environments • Designing interactive cultural activities • Representing global celebrations
Lessons	<p>Communities Adapt to & Modify Their Environment</p> <ul style="list-style-type: none"> • Animate community adaptation or modification. <p>Choose Your Own Path: Elements of Culture</p> <ul style="list-style-type: none"> • Build a cultural choose-your-own-path game. <p>Winter Celebrations Around the World</p> <ul style="list-style-type: none"> • Create a program about a specific winter holiday.

More Coding Lessons & Projects

Students build skills through creative projects, custom art, advanced programming concepts, and seasonal activities.

Objectives / Topics Covered	<ul style="list-style-type: none"> • Creating custom sprites and backdrops • Using broadcast messages, conditionals, and variables • Designing greeting cards and study tools
Lessons	<p>Costumes, Backdrops, and Animations</p> <ul style="list-style-type: none"> • Animate sprites with interactive backdrops. <p>Scratch Drawing Tools</p> <ul style="list-style-type: none"> • Customize sprites and backgrounds. <p>Broadcast Messages: Marco Polo</p> <ul style="list-style-type: none"> • Communicate between sprites with broadcast. <p>Loops</p> <ul style="list-style-type: none"> • Repeat instructions with loops. <p>Introduction to Conditionals</p> <ul style="list-style-type: none"> • Use if/then logic in Scratch. <p>Variables</p> <ul style="list-style-type: none"> • Create and update variables in a program. <p>Digital Flashcards</p> <ul style="list-style-type: none"> • Build flashcards for any subject using broadcasts. <p>Scratch Digital Greeting Card</p> <ul style="list-style-type: none"> • Create a greeting card with loops and events.

Interdisciplinary Computer Science Third 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.