4th Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **4th Grade Computer Science Course** introduces students to foundational programming concepts through **Scratch**, a block-based programming language. Students will develop computational thinking and problem-solving skills while learning to create interactive projects, animations, and games. This course emphasizes creativity and collaboration, providing students with a solid base in computer science concepts and digital literacy.

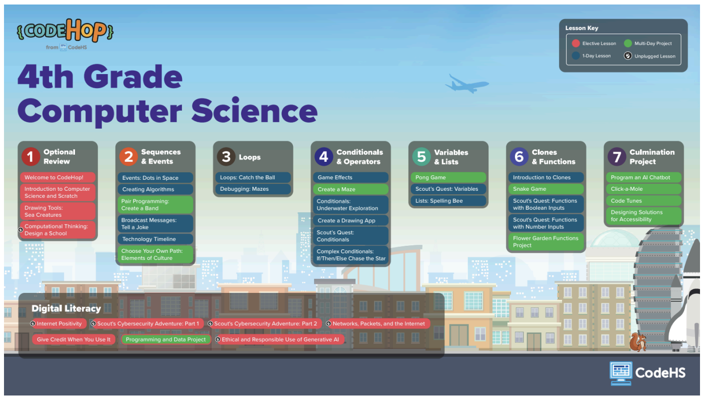
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging projects to showcase student understanding.

The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. Teachers can adapt the content to fit their schedule and instructional needs. The concepts taught in this course spiral across grade levels, ensuring that students can revisit and build upon their understanding year after year, even if all lessons are not completed within a single year. The course includes a total of **36 lessons**, each approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week. Optional digital literacy lessons are also available to complement the programming curriculum with non-programming computer and technology skills.

Programming Environment: Students will write and run programs in **Scratch** embedded and saved in students' accounts. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

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/20686/overview?lang=en


{CODE HOP} from CodeHS
4th Grade Computer Science
Lesson Key
Elective Lesson
Multi-Day Project
1-Day Lesson
Unplugged Lesson
1 Optional Review
Welcome to CodeHop!
Introduction to Computer Science and Scratch
Drawing Tools: Sea Creatures
Computational Thinking: Design a School
2 Sequences & Events
Events: Dots in Space
Creating Algorithms
Pair Programming: Create a Band
Broadcast Messages: Tell a Joke
Technology Timeline
Choose Your Own Path: Elements of Culture
3 Loops
Loops: Catch the Ball
Debugging: Mazes
4 Conditionals & Operators
Game Effects
Create a Maze
Conditionals: Underwater Exploration
Create a Drawing App
Scout's Quest: Conditionals
Complex Conditionals: If/Then/Else Chase the Star
5 Variables & Lists
Pong Game
Scout's Quest: Variables
Lists: Spelling Bee
6 Clones & Functions
Introduction to Clones
Snake Game
Scout's Quest: Functions with Boolean Inputs
Scout's Quest: Functions with Number Inputs
Flower Garden Functions Project
7 Culmination Project
Program an AI Chatbot
Click-a-Mole
Code Tunes
Designing Solutions for Accessibility
Digital Literacy
Internet Positivity
Scout's Cybersecurity Adventure: Part 1
Scout's Cybersecurity Adventure: Part 2
Networks, Packets, and the Internet
Give Credit When You Use It
Programming and Data Project
Ethical and Responsible Use of Generative AI
CodeHS

A clickable PDF can be found at https://codehs.com/K5CSRoadmaps.

Course Breakdown

Unit 1: Optional Review (3 weeks)

In this unit, students review core concepts and build foundational skills in Scratch programming, creative design, and computational thinking through engaging activities and guided projects.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Define basic computer science terms and build simple programs in Scratch. ● Use image editing tools in Scratch to customize sprites and backdrops. ● Apply computational thinking to solve open-ended design challenges.
Lessons	**Welcome to CodeHop! (15 minutes lesson)** ● Log in and explore the CodeHop Playground as an introductory activity or refresher before full lessons. **Introduction to Computer Science and Scratch** ● Learn key computer science vocabulary and use Scratch to create a simple introductory program. **Drawing Tools: Sea Creatures** ● Explore all of Scratch's image editing tools to design and animate custom sea creatures. **Computational Thinking: Design a School** ● Use computational thinking strategies to plan and design a functional, creative school layout.

Unit 2: Sequences and Events (8 weeks)

Students expand their understanding of sequences and events in Scratch by programming interactive stories, coordinating sprite communication, building timelines, and exploring cultural connections.

Objectives / Topics Covered	● Use multiple types of event blocks to trigger sprite actions. ● Design and compare algorithms to solve problems efficiently. ● Apply broadcast messages to coordinate sprite interactions. ● Create culturally themed interactive experiences and visual timelines.
Lessons	**Events: Dot in Space** ● Create a program using different event blocks (e.g., when green flag clicked, key pressed, sprite clicked) to animate a space-themed scene. **Creating Algorithms** ● Program and compare multiple algorithms to determine which solution is most effective for completing a task. **Pair Programming: Create a Band (2-part lesson)** ● Collaborate with a partner to design and code a virtual band using keyboard inputs and pair programming strategies. **Broadcast Messages: Tell a Joke** ● Use broadcast messages in Scratch to animate a two-sprite knock-knock joke with coordinated timing. **Technology Timeline** ● Build an interactive timeline that showcases key developments in music player technology and explains its influence on culture. **Choose Your Own Path: Elements of Culture (2-part lesson)** ● Create a branching, interactive story game that explores cultural elements such as food, clothing, language, and traditions.

Unit 3: Loops (2 weeks)

In this unit, students strengthen their understanding of loops by building a simple game using multiple loop types and practicing debugging through decomposing Scratch programs.

Objectives / Topics Covered	● Use repeat and forever loops to control game behavior and sprite actions. ● Decompose programs to identify and fix logical or structural bugs. ● Apply looping logic to build simple, functional gameplay experiences.
Lessons	**Loops: Catch the Ball** ● Use both repeat and forever loops to program a game where players try to catch a falling ball with a moving sprite. **Debugging: Mazes** ● Debug a Scratch maze program by breaking it down into parts and correcting errors related to events, movement, or loops.

Unit 4: Conditionals and Operators (7 weeks)

In this unit, students explore conditionals and operators by creating interactive programs and games that use if/then and if/then/else logic, while incorporating user input, visual effects, and creative design.

Objectives / Topics Covered	● Use conditionals and comparison operators to control program logic. ● Create interactive experiences using keyboard and mouse inputs. ● Apply feedback to improve game design and visual engagement. ● Design original programs that include branching behaviors and layered interactions.
Lessons	**Game Effects** ● Modify an existing game to add visual effects and enhancements, then update it based on peer feedback. **Create a Maze (2-part lesson)** ● Design a maze in Scratch and program Scout to navigate through it using directional logic and conditionals. **Conditionals: Underwater Exploration** ● Use if/then statements to create an interactive underwater-themed Scratch program. **Create a Drawing App** ● Build a custom drawing app using keyboard and mouse inputs, loops, and conditionals to control drawing behaviors. **Scout's Quest: Conditionals** ● Complete the final part of the Scout's Quest series by creating a program using if/then conditional logic. **Complex Conditionals: Chase the Star** ● Use if/then/else conditionals to control sprite behavior in a game where the player must catch a moving star.

Unit 5: Variables and Lists (4 weeks)

In this unit, students apply variables and lists to track data and build interactive games in Scratch, while combining conditionals and scorekeeping to enhance gameplay experiences.

Objectives / Topics Covered	● Create and update variables to store and display game data. ● Use conditionals alongside variables for interactive control. ● Build games that use lists to store and manage multiple items. ● Combine variables, lists, and logic to personalize and expand game functionality.
Lessons	**Pong Game (2-part lesson)** ● Build a Scratch pong-style game using variables to track and display player scores. **Scout's Quest: Variables** ● Create and use a variable to track points in a program as part of the Scout's Quest review series. **Lists: Spelling Bee**

	• Use lists in Scratch to build a spelling bee game that tracks and verifies words during gameplay.

Unit 6: Clones and Functions (7 weeks)
In this unit, students explore advanced Scratch concepts by using clones and functions to create reusable code, dynamic animations, and interactive games with varied inputs and outputs.

Objectives / Topics Covered	• Create and manage clones to duplicate sprites and automate repetition. • Use variables and clones to build interactive games with dynamic behavior. • Design and apply functions with Boolean and numeric inputs. • Structure code efficiently using custom blocks to reduce repetition and improve clarity.
Lessons	**Introduction to Clones** • Create an animation using clones and reflect on the limitations of clone behavior in Scratch. **Snake Game (2-part lesson)** • Build a classic snake game using variables to track score and clones to extend the snake's body. **Scout's Quest: Functions with Boolean Inputs** • Create a function with a Boolean input that controls behavior based on conditions like password correctness. **Scout's Quest: Functions with Number Inputs** • Use a function with number inputs to generate drawings, customizing features like size or repetition based on input. **Flower Garden Functions Project (2-part lesson)** • Design a program that uses a function to draw multiple flowers, emphasizing reuse and efficiency in coding patterns.

Unit 7: Culmination Projects (8 weeks)
Students synthesize their programming knowledge to build creative and impactful projects, including games, chatbots, and accessible design solutions using conditionals, variables, lists, and the design thinking process.

Objectives / Topics Covered	• Use lists and conditionals to create responsive, interactive experiences. • Design games and tools that use variables, booleans, and events. • Apply the design thinking process to improve accessibility and usability. • Demonstrate real-world application of computer science through creative problem-solving.
Lessons	**Program an AI Chatbot (2-part lesson)** • Use lists in Scratch to store questions and responses, building a basic chatbot that simulates conversation. **Click-a-Mole (2-part lesson)** • Create a Whack-a-Mole style game using conditionals, events, variables, and booleans to track gameplay and user interaction. **Code Tunes (2-part lesson)** • Use various programming skills to create a custom music player. **Designing Solutions for Accessibility (2-part lesson)** • Apply the design thinking process to identify and solve accessibility challenges by redesigning a Scratch game to be more inclusive and user-friendly.

Unit 8: Digital Literacy (8 weeks)
In this supplemental unit, students explore key digital literacy topics such as cybersecurity, internet safety, responsible online behavior, and ethical data and AI use, while also practicing how to give credit and communicate effectively online.

Objectives / Topics Covered	● Understand how to stay safe online through strong habits and cybersecurity awareness. ● Promote positive digital behavior and ethical interactions with others and with technology. ● Explain how the Internet works, including data transmission using packets. ● Practice responsible use of generative AI, data collection, and attribution.
Lessons	**Internet Positivity** ● Learn how actions online can spread kindness and create a personal code of conduct for responsible digital behavior. **Scout's Cybersecurity Adventure: Part 1** ● Understand cybersecurity basics, recognize common online threats, and identify ways to stay safe on the Internet. **Scout's Cybersecurity Adventure: Part 2** ● Practice safe browsing habits and explore tools and technologies that help protect personal information. **Networks, Packets, and the Internet** ● Explain how information is transmitted across the Internet and simulate sending and reassembling packets through a classroom communication model. **Giving Credit Through Attributions** ● Learn the importance of attribution when creating or remixing digital content and practice giving proper credit in Scratch or other projects. **Programming and Data Project (2-part lesson)** ● Develop a question, collect and analyze data, and create a Scratch program to visually present findings and conclusions. **Ethical and Responsible Use of Generative AI** ● Explore the pros and cons of generative AI and contribute to a class Code of Conduct for using AI tools in ethical and responsible ways.

4th - 5th 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**.	