Colorado 4th Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Colorado 4th Grade Computer Science Course** introduces students to foundational programming concepts through a block-based programming language. Students explore digital literacy and computer science skills while engaging in lessons that integrate coding into math, science, English language arts (ELA), and social studies. This course emphasizes creativity, collaboration, and real-world connections, providing students with a strong foundation in both academic content, digital literacy, and computer science.

Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans. Each programming lesson follows a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection.** Many digital literacy lessons contain unplugged activities, requiring printed handouts and class activities to support hands-on learning.

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.

Programming Environment: Students will write and run programs that are 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/CO_4/overview

Course Breakdown

Optional Review

This optional review unit is designed to support students who would benefit from additional practice before or after completing core lessons.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Create a simple program to demonstrate basic programming skills. ● Use the coordinate plane to position sprites.
Lessons	**Welcome to CodeHop!** ● Learn how to log in and navigate the CodeHop Playground. **Introduction to Computer Science** ● Define key computer science vocabulary and create a simple program. **Drawing Tools: Sea Creatures** ● Use CodeHop imaging tools to create and program deep sea creatures. **The Coordinate Plane** ● Create an open-ended animation using coordinate-plane positioning.

Unit 1: Getting Started (2 lessons)

In this introductory unit, students are introduced to the basics of computing by identifying key parts of a computing system and learning how to troubleshoot simple issues. They also begin developing foundational computational thinking skills by designing and sequencing steps to design a school.

Objectives / Topics Covered	● Identify parts of a computing system. ● Practice basic troubleshooting strategies for common computer problems. ● Apply computational thinking skills.
Lessons	**Computational Thinking: Design a School** ● Use computational thinking to plan and design a school by identifying patterns, sequencing steps, and breaking tasks down. **Exploring Computing Systems** ● Identify computer system components and recognize basic hardware and software issues. ● *Standards Met: CS.4.2.1a, CS.4.2.1b*

Unit 2: Getting Started: Programming (3 lessons)

In this unit, students will create basic programs while learning about events, algorithms, and loops. Students will practice debugging their programs to solve simple problems.

Objectives / Topics Covered	● Debug sequences in a program. ● Use loops to repeat actions.
Lessons	**Creating Algorithms** ● Program multiple algorithms and assess which best meets their needs. **Loops: Catch the Ball** ● Use two types of loops to create a simple game. **Debugging: Mazes** ● Decompose a program to debug and make the program run as intended.

Unit 3: Digital Literacy (10 lessons)

In this unit, students will learn what it means to be a responsible digital citizen by identifying appropriate behavior and protecting themselves online. Students will also explore digital literacy concepts such as AI, data, and networks.

Objectives / Topics Covered	● Practice safe and healthy habits online. ● Analyze data and make predictions. ● Model how networks function. ● Explain how computers store information.
Lessons	**Managing Digital Footprints** ● This lesson is coming soon! ● *Standards Met: CS.4.6.1a, CS.4.6.1b, CS.4.6.1c* **Impacts of Computing: Exploration** ● Explain how technology and culture influence each other. ● *Standard Met: CS.4.3.1b* **Online Risks & Protection** ● Demonstrate how to stay safe online by practicing secure habits. **Data Investigators** ● Evaluate data for reliability, analyze findings, and make predictions. ● *Standards Met: CS.4.1.1a, CS.4.1.1c* **Inquiry Project: Data Bar Graph (2 part lesson)** ● Follow the inquiry process and update a program to display investigation results using a bar graph. ● *Standards Met: CS.4.1.1a, CS.4.1.1b* **File Management and Data in Action** ● Explain how data size varies and explore different ways digital files are stored. ● *Standard Met: CS.4.1.1d* **Networks, Packets, and the Internet** ● Model how information travels through the Internet, including packet transfer and reassembly, and create a secure communication method. ● *Standards Met: CS.4.2.1c, CS.4.2.1d, CS.4.2.1e* **AI Chatbots (2 part lesson)** ● Create a chatbot to solve a societal problem. ● *Standard Met: CS.4.5.1a*

Unit 4: Programming Exploration (10 lessons)

In this unit, students will gain basic programming skills using a block-based programming language. Students will practice with sequences, loops, variables, and functions.

Objectives / Topics Covered	● Apply broadcast messages to enable communication between multiple sprites. ● Use loops to simplify code. ● Use conditionals to build interactive and reactive programs. ● Learn how variables store information. ● Use functions to organize code into reusable parts.
Lessons	**Broadcast Messages: Tell a Joke** ● Use broadcast messages to program two sprites to tell a joke. **Choose Your Own Path: Elements of Culture (2 part lesson)** ● Create a cultural choose-your-own-path game. ● *Standard Met: CS.4.3.1a* **Game Effects** ● Modify a game to add engaging effects. **Scout's Quest: Conditionals** ● Build a program using if/then conditionals to control behavior. **Scout's Quest: Variables** ● Use variables to track points within a program.

	Introduction to Clones ● Use clones to create an animation. **Scout's Quest: Functions with Boolean Inputs** ● Create a function using a Boolean input to control actions based on password correctness. **Designing Solutions for Accessibility (2 part lesson)** ● Redesign a game to improve accessibility for diverse users. ● *Standard Met: CS.4.3.1c*

Unit 5: Interdisciplinary Exploration (11 lessons)

This unit integrates programming with subjects like social studies, math, science, and ELA, enabling students to model real-world concepts using events, algorithms, and interactive programs.

Objectives / Topics Covered	● Use algorithms, loops, and conditionals to model concepts in social studies, math, science, and ELA.
Lessons	**Cause and Effect** ● Demonstrate cause-and-effect relationships using messages, conditionals, or events. **State Project (2 part lesson)** ● Use events to build an interactive project about state-specific facts. **How We See: Light Reflection** ● Use conditionals to model how light reflects off objects and enters the eye to allow vision. **Creative Storytelling (2 part lesson)** ● Plan and animate an original story using events and sequences. **Exploring Point of View** ● Identify and use first-person and third-person points of view by programming characters to retell a scene from a different perspective. **Unit Converter** ● Use variables and conditionals to build a unit-conversion calculator. **House Design with Area and Perimeter (2 part lesson)** ● Calculate and apply area and perimeter to design a house layout using functions. **Classifying Quadrilaterals in a Hierarchy** ● Use events to classify quadrilaterals in a hierarchy and document thinking with comments.

4th 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 Templates	Empower your students to explore and express their knowledge creatively with our versatile 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**.	