



Colorado 1st Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Colorado 1st 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_1/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	<ul style="list-style-type: none">Log in and navigate the Playground.Explore and interact with the CodeHop environment.Use motion blocks and sequencing to animate characters.
Lessons	<p>Welcome to CodeHop! (15 minute Lesson)</p> <ul style="list-style-type: none">Learn how to log in and explore the Playground; a brief introduction that can be used on its own or before starting a full lesson. <p>Scout Adventures 1: Introducing Scout</p> <ul style="list-style-type: none">Explore the CodeHop interface and practice adding characters to the stage. <p>Scout Adventures 2: Scout Starts Exploring</p> <ul style="list-style-type: none">Add backgrounds and create a new page in CodeHop to build out a story. <p>Scout Adventures 3: Scout Meets a Friend</p> <ul style="list-style-type: none">Delete and modify characters, building confidence in editing elements in a project. <p>Scout Adventures 4: Scout Explores the Forest</p> <ul style="list-style-type: none">Use motion blocks to move characters across the stage and begin creating animations. <p>Scout Adventures 5: Scout and Bluebird Help</p> <ul style="list-style-type: none">Build a sequence of motion blocks to guide characters through a task, such as collecting items. <p>Scout Adventures 6: Scout Celebrates with Friends</p> <ul style="list-style-type: none">Create a celebration scene by combining characters, pages, backgrounds, and events using motion blocks.

Unit 1: Getting Started (2 lessons)

In this unit, students will explore the basics of computing by learning the functions of computers and their components and apply foundational computational thinking skills like pattern recognition, sequencing, and task decomposition to real-life routines.

Objectives / Topics Covered	<ul style="list-style-type: none">Recognize the basic parts of a computer.Practice computational thinking in everyday routines.
Lessons	<p>Computational Thinking: Evening Routines</p> <ul style="list-style-type: none">Use patterns, sequencing, and task breakdown to map out an evening routine.<i>Standards Met: CS.1.1.1a, CS.1.1.2a</i> <p>Computer Basics: Exploration</p> <ul style="list-style-type: none">Learn what a computer is, how it works, and the roles of hardware, software, input, and output.<i>Standard Met: CS.1.2.1a</i>

Unit 2: Getting Started: Programming (4 lessons)

In this unit, students will create basic programs while learning about events and sequences. Students will practice debugging their programs to solve simple problems and will become familiar with drawing tools.

Objectives / Topics Covered	<ul style="list-style-type: none">Using events and sequencesApplying digital responsibility conceptsDebug problems in a program.Creating scenes with drawing tools
-----------------------------	---

Lessons	<p>Events</p> <ul style="list-style-type: none"> Introduce events and use multiple event blocks to trigger actions in a program. <p>Sequences: Digital Responsibilities</p> <ul style="list-style-type: none"> Use sequences to program characters while explaining responsible online behavior. <i>Standards Met: CS.1.3.1b, CS.1.6.1a, CS.1.6.1b</i> <p>Introduction to Debugging</p> <ul style="list-style-type: none"> Find and correct bugs in sequences. <i>Standards Met: CS.1.1.1c, CS.1.1.2a</i> <p>Drawing Tools: Nature Walk</p> <ul style="list-style-type: none"> Use drawing tools in CodeHop to create a nature-themed scene.
---------	--

Unit 3: Digital Literacy (9 lessons)

In this unit, students will learn what it means to be a responsible digital citizen by identifying appropriate behavior and following laws. Students will also explore digital literacy concepts such as computer networks, data and research, and types of software.

Objectives / Topics Covered	<ul style="list-style-type: none"> Behave respectfully online. Model how computers store information. Collect and represent data. Model how networks function. Compare software tools.
Lessons	<p>Positive Online Behavior</p> <ul style="list-style-type: none"> Identify appropriate vs. inappropriate online behavior by analyzing a CodeHop program. <i>Standards Met: CS.1.6.1a, CS.1.6.1b</i> <p>Exploring Responsible Technology Use</p> <ul style="list-style-type: none"> Explain ways to use technology safely and responsibly. <i>Standards Met: CS.1.6.1a, CS.1.6.1b</i> <p>Giving Credit Through Attributions</p> <ul style="list-style-type: none"> Reuse parts of an existing program and give credit to the original creator. <i>Standard Met: CS.1.3.1c</i> <p>Data Storage and Files Practice</p> <ul style="list-style-type: none"> Understand data storage and model how computers store information in files. <i>Standard Met: CS.1.1.3a</i> <p>Basic Data and Programming</p> <ul style="list-style-type: none"> Collect data and present it visually through a CodeHop program. <i>Standards Met: CS.1.1.3b, CS.1.1.3c</i> <p>Guided Research (2 classes)</p> <ul style="list-style-type: none"> Use research sources to find information and create a program that visually communicates findings. <i>Standards Met: CS.1.1.3b, CS.1.3.1c</i> <p>Networks and the Internet</p> <ul style="list-style-type: none"> Learn what networks are and model how Internet messages are sent and received. <i>Standards Met: CS.1.2.1b, CS.1.2.1c</i> <p>Types of Software</p> <ul style="list-style-type: none"> Compare and select appropriate software applications to complete tasks. <i>Standard Met: CS.1.3.1a</i>

Unit 4: Programming Exploration (12 lessons)

In this unit, students will gain basic programming skills using a block-based programming language. Students will practice with events, sequences, loops, and the design thinking process.

Objectives / Topics	<ul style="list-style-type: none"> Explore different types of programming blocks to control character behavior. Use loops to repeat action in a program.
---------------------	--

Covered	<ul style="list-style-type: none"> Use the design process to solve a problem.
Lessons	<p>Grow and Shrink Blocks in Motion</p> <ul style="list-style-type: none"> Create animations using motion blocks combined with grow/shrink effects. <p>Introduction to the Wait Block</p> <ul style="list-style-type: none"> Use wait blocks to add pauses and timing to character actions. <p>Practicing the Design Process</p> <ul style="list-style-type: none"> Plan, create, and revise a programming project to solve a real-world problem. <i>Standards Met: CS.1.1.1c, CS.1.3.1b</i> <p>Introduction to Repeat Loops</p> <ul style="list-style-type: none"> Use loops to repeat actions multiple times within a program. <p>Debugging</p> <ul style="list-style-type: none"> Describe what bugs are and correct bugs in sequences. <i>Standards Met: CS.1.1.1b, CS.1.1.1c</i> <p>Introduction to Message Events</p> <ul style="list-style-type: none"> Program character interactions using message events in a relay-style activity. <p>Pages: Create a Tapping Game</p> <ul style="list-style-type: none"> Create a game that moves to multiple pages using “go to page” blocks. <p>Grid: Solving Mazes</p> <ul style="list-style-type: none"> Design a maze and use the grid to move a character. <p>Create a Mini Golf Game</p> <ul style="list-style-type: none"> Build an interactive mini-golf game using messages and loops. <p>Variables: Storing Data with Symbols</p> <ul style="list-style-type: none"> Model how programs store and manipulate data. <i>Standard Met: CS.1.1.3d</i> <p>River Crossing Game (2 part lesson)</p> <ul style="list-style-type: none"> Program obstacles and adjust difficulty using speed blocks

Unit 5: Interdisciplinary Exploration (9 lessons)

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

Objectives / Topics Covered	<ul style="list-style-type: none"> Use events and sequences to model concepts.
Lessons	<p>Economic Choices</p> <ul style="list-style-type: none"> Use messages to model wants, needs, and simple economic decisions. <p>Light and Shadows</p> <ul style="list-style-type: none"> Create a program that demonstrates light sources and the effects of shadows. <p>Phases of the Moon</p> <ul style="list-style-type: none"> Model the phases of the Moon using message events and character motions. <p>Phonics: Digraphs</p> <ul style="list-style-type: none"> Create a phonics activity using events and recorded audio for digraph sounds. <p>Original Story Animations – Personal Story (2 part lesson)</p> <ul style="list-style-type: none"> Animate an original story and explain the steps used to build the program. <p>Patterns and Music</p> <ul style="list-style-type: none"> Create musical patterns using sequences and loops. <p>Greater Than and Less Than: Two-Digit Numbers</p> <ul style="list-style-type: none"> Program a comparison tool that shows whether one number is greater, less, or equal to another. <p>Telling Time</p> <ul style="list-style-type: none"> Create an analog clock and program it to display digital and analog time.

1st Grade Course Supplemental Materials

Resources	Description
<u>Parent Welcome Letter (Spanish)</u>	Send this letter home to introduce families to their new computer science curriculum.
<u>Warm-Up Activities</u>	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.
<u>Program Self-Assessment (Spanish)</u>	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.
<u>Peer Review Resources (Spanish)</u>	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.
<u>Lesson Reflection & Computational Thinking (Spanish)</u>	This guides students in engaging with computational thinking concepts, preparing for discussions, reflecting on lessons, and applying their learning to real-world problem-solving.
These resources and more are found on the <u>CodeHop Resources Page</u> .	