California 1st Grade
Computer Science Course Syllabus
One Year for Elementary School, 36 Hours

Course Overview and Goals

The **California 1st Grade Computer Science Course** introduces students to foundational programming concepts through **CodeHopJr**, 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.

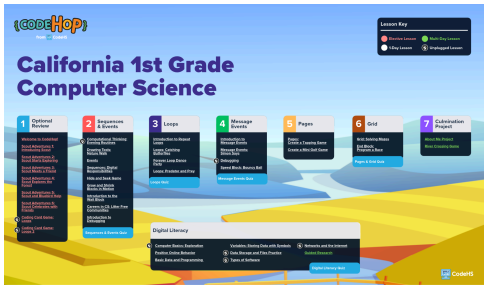
The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. Teachers can adapt the content to fit their schedule and instructional needs. The concepts taught in this course spiral across grade levels, 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 **contact hours**, with each lesson 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 in **CodeHopJr** that are embedded and saved in the CodeHop platform. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Assessments: Some units conclude with a brief two-question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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/26153/overview



A clickable PDF can be found at https://codehs.com/CA-CSRoadmaps

Course Breakdown

Unit 1: Optional (9 Lessons

This unit optionally introduces students to the platform and basic programming concepts. Through the **Scout Adventures** CodeHopJr series, students learn to create scenes, animate characters, and use events and motion blocks. Unplugged coding card games reinforce sequencing and loops in a fun, collaborative way.

Objectives / Topics Covered	• Log in to CodeHop • Explore the CodeHopJr interface • Sequences
Lessons	**Welcome to CodeHop!** • 15-minute lesson - combine with the next lesson if time allows. • Explore the CodeHop Playground and learn how to log in. **Scout Adventures: Introducing Scout** • This lesson is part of a sequential story-driven unit. Explore the CodeHopJr interface and add characters. **Scout Adventures: Scout Starts Exploring** • Add backgrounds and a page to CodeHopJr. **Scout Adventures: Scout Meets a Friend** • Delete and modify characters in CodeHopJr. **Scout Adventures: Scout Explores the Forest** • Explore and use motion blocks to move characters around the stage in CodeHopJr. **Scout Adventures: Scout and Bluebird Help** • Build a sequence of motion blocks to move characters around the stage to collect objects. **Scout Adventures: Scout Celebrates with Friends** • Create a celebration scene in CodeHopJr by adding characters, pages, backgrounds, and sequences of motion blocks with events. **Coding Card Game: Loops** • Collaboratively create a sequence of instructions with loops to move Scout through a maze. **Coding Card Game: Loops 2** • Collaboratively create a sequence of instructions with loops to move Scout through a maze.

Unit 2: Sequence & Events (9 Lessons)

In this Sequences & Events unit, students develop foundational programming skills and apply computational thinking to real-life scenarios. They'll explore sequencing through everyday routines and digital citizenship activities, then use CodeHopJr to create interactive scenes with event blocks, motion, and effects. The unit also introduces debugging and shows how coding can address community issues through a litter-sorting project.

Objectives / Topics Covered	• Computational thinking • Use events and sequences • Use motion, wait, grow, and shrink blocks • Explore careers in computer science • Identify and fix bugs
Lessons	**Computational Thinking: Evening Routines** • Identify patterns, break down tasks, sequence steps, and simplify processes in their evening routines. **Drawing Tools: Nature Walk** • Use drawing tools to create a nature walk scene. **Events**

	● Explain what an event is in programming and use multiple event blocks in a program. **Sequences: Digital Responsibilities** ● Use sequences to program characters and explain how to be responsible online. **Hide and Seek Game** ● Use the hide block to program an interactive game of hide-and-seek. **Grow and Shrink Blocks in Motion** ● Create a program using motion blocks and grow and shrink blocks to change the size of characters. **Introduction to the Wait Block** ● Use "wait" blocks to cause characters to pause in a program. **Careers in CS: Litter Free Communities** ● Explain how computer science can help solve community problems and create a program to collect and sort litter. **Introduction to Debugging** ● Find and correct bugs in sequences. **Sequences & Events Quiz (3 questions)**

Unit 3: Loops (4 Lessons)

In this unit on Loops, students will explore how to run code multiple times using repeat and forever loops. Through creative projects like a butterfly-catching game and a dance party, they'll apply loops along with blocks like "show" and "hide" to build engaging, interactive programs.

Objectives / Topics Covered	● Use repeat loops and forever loops ● Use the show and hide blocks
Lessons	**Introduction to Repeat Loops** ● Use repeat loops to run a section of code multiple times. **Loops: Catching Butterflies** ● Use "show" and "hide" blocks and loops to create a butterfly-catching game. **Forever Loop Dance Party** ● Create a sequence using a "repeat forever" loop to make characters repeat actions. **Loops: Predator and Prey** ● Use events, sequences, and loops to program two animal characters to interact with each other. **Loops Quiz (3 questions)**

Unit 4: Message Events (4 Lessons)

In this unit on Message Events, students will learn how characters can communicate in a program using message blocks. They'll apply these skills in interactive projects like a relay race and a Simon Says game, while also exploring speed control and debugging techniques to refine their programs.

Objectives / Topics Covered	● Use message events and speed blocks ● Identify and fix bugs
Lessons	**Introduction to Message Events** ● Program a relay race that uses messages to cause characters to interact. **Message Events: Simon Says** ● Use message events to make one character communicate to many characters in a program. **Debugging** ● Describe what bugs are and find and correct bugs in sequences. **Speed Block: Bouncy Ball** ● Use speed blocks and messages to program a character to move at

	different speeds. **Message Events Quiz (3 questions)**

Unit 5: Pages (2 Lessons)

In this unit on Pages, students will learn to build multi-scene projects using "go to page" blocks. They'll create interactive games—like a tapping game and a mini golf challenge—while combining page navigation with messages and loops for more complex, engaging experiences.

Objectives / Topics Covered	● Use multiple pages to create games
Lessons	**Pages: Create a Tapping Game** ● Create a game that moves from one page to the next using "go to page" blocks. **Create a Mini Golf Game** ● Use messages and loops to create a mini golf game.

Unit 6: Grid (2 Lessons)

In this Grid unit, students will explore how to use the CodeHopJr grid system to control movement and organize animations. They'll apply these skills in projects like solving mazes, animating arctic scenes, and programming races using the “end” block to create structured, goal-oriented animations.

Objectives / Topics Covered	● Use the grid when creating programs ● Use the end block
Lessons	**Grid: Solving Mazes** ● Design a maze and use the grid to program a character to move through the maze. **End Block: Program a Race** ● Complete a project to program a race and use the “end” block in an animation. **Pages & Grid Quiz (3 questions)**

Unit 7: Culmination Projects (5 Lessons)

In the Culmination Projects unit, students will apply their programming skills to design and build creative, personalized projects. They'll create an interactive "About Me" program to share their interests and a river crossing game featuring obstacles and adjustable difficulty using speed blocks.

Objectives / Topics Covered	● Culminating creative projects
Lessons	**About Me Project** (2 Lessons) ● In this three-part project, plan a program to describe their characteristics and interests. **River Crossing Game** (2 Lessons) ● In this two-part project, program obstacles in a game and change the level of difficulty using speed blocks.

Unit 8: Digital Literacy (9 Lessons)

In the Digital Literacy unit, students will build a foundational understanding of computers, software, and responsible technology use. Through interactive CodeHopJr projects, they’ll explore topics such as computer basics, online behavior, data collection, file management, research, and how AI makes decisions, developing both digital skills and critical thinking along the way.

Objectives / Topics Covered	• Identify input, output, hardware, and software • Appropriate and inappropriate online behavior • Explore software for tasks • Data storage and file management • Communicate research findings
Lessons	**Computer Basics: Exploration** • Learn what a computer is, how we use it, and what to do when it doesn't work. • Identify input, output, hardware, and software. **Positive Online Behavior** • Identify appropriate and inappropriate online behavior in a CodeHopJr program. **Basic Data and Programming** • Collect data and create a program to present their data visually. **Variables: Storing Data with Symbols** • Model the way programs store and manipulate data using variables. **Data Storage and Files Practice** • Recognize that computers store data as files and model how data is collected and stored. **Types of Software** • Compare and select appropriate software applications to complete different computing tasks. **Networks and the Internet** • Explain what a network is and how people communicate over networks and the Internet. • Model how messages are communicated using the Internet. **Guided Research** (2 Lessons) • In this two-part project, find information using research sources and create a program to communicate their research visually. **Digital Literacy Quiz (3 questions)**

California 1st Grade Computer Science Course

Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter	Introduce parents to computer science.
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	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	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 &	This guides students in engaging with computational thinking concepts,

Computational Thinking	preparing for discussions, reflecting on lessons, and applying their learning to real-world problem-solving.
All of these resources and more are found on the **Elementary Resources Page**.	