Nevada Computer Science and Integrated Technology: 1st Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nevada Computer Science and Integrated Technology: 1st Grade** 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.

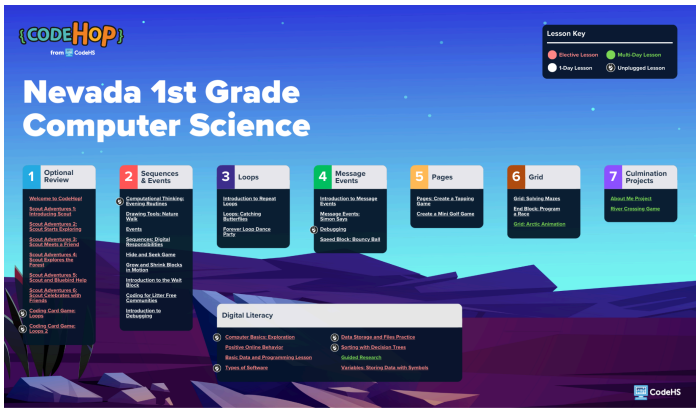
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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 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 in **CodeHopJr** that are embedded and saved in the platform. 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/26142/overview?lang=en



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

Course Breakdown

Unit 1: Optional Review (9 optional lessons)

This unit optionally introduces students to the CodeHop platform and basic programming concepts. Through the **Scout Adventures** 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 and explore the Playground ● Explore the CodeHop interface ● Build a sequence of motion blocks and loops
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 CodeHop interface and add characters. **Scout Adventures: Scout Starts Exploring** ● Add backgrounds and a page. **Scout Adventures: Scout Meets a Friend** ● Delete and modify characters. **Scout Adventures: Scout Explores the Forest** ● Explore and use motion blocks to move characters around the stage. **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 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 CodeHop 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 ● Drawing tools ● Events ● Sequences ● Additional blocks ● Careers in computer science ● Debugging strategies
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.

Unit 3: Loops (3 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	● Repeat and forever loops
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.

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	● Message events for character interaction ● 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.

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	● Create games by adding page blocks.
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 (4 lessons)

In this Grid unit, students will explore how to use the 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	● Navigating characters on a grid
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. **Grid: Arctic Animation (2 lessons)** ● In this two-part project, use the grid to create an animation.

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	● Design and program complex projects
Lessons	**About Me Project (3 lessons)** ● In this 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 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 online behavior ● Collect and visually present data using programming ● Compare and choose appropriate software ● Data storage and file management ● AI and data
Lessons	**Computer Basics: Exploration** ● Learn what a computer is, how we use it, and what to do when it doesn't work. They will be able to identify input, output, hardware, and software. **Positive Online Behavior** ● Identify appropriate and inappropriate online behavior in a program. **Basic Data and Programming** ● Collect data and create a program to present their data visually. **Types of Software** ● Compare and select appropriate software applications to complete different computing tasks. **Data Storage & File Practice** ● Recognize that computers store data as files and model how data is collected and stored. **Sorting with Decision Trees** ● Explain how AI uses data to learn and make decisions, and create a simple decision tree to sort items based on rules. **Guided Research (2 lessons)** ● In this two-part project, find information using research sources and create a program to communicate their research visually. **Variables: Storing Data with Symbols** ● Model the way programs store and manipulate data using variables.

Nevada Computer Science and Integrated Technology: 1st Grade

Course Supplemental Materials

Resources	Description
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science with CodeHop.
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.
These resources and more are found on the **Elementary Resources Page**.	