Nevada Computer Science and Integrated Technology: Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nevada Computer Science and Integrated Technology: Kindergarten** 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.

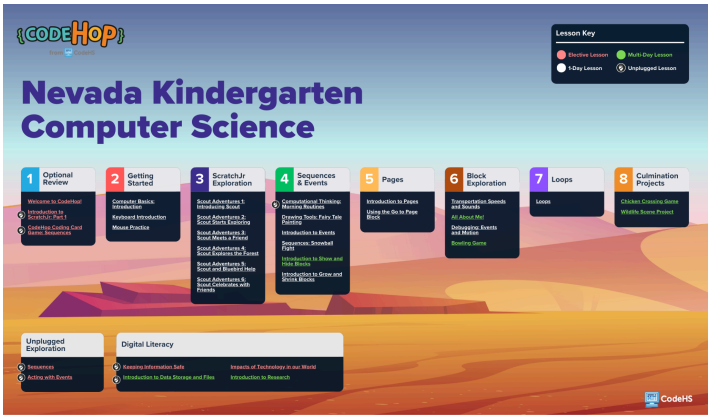
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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 30 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 **CodeHopJr** 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/26140/overview?lang=en



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

Course Breakdown

Unit 1: Optional Introduction Activities (2 optional lessons)

This unit provides students with an introduction to programming concepts through engaging, hands-on activities. This unit serves as a flexible starting point, teaching students about what CodeHop is and allowing students to familiarize themselves with the basics of coding in a creative and interactive manner.

Objectives / Topics Covered	● Log in and explore the CodeHop Playground ● Create and sequence basic programs using CodeHop ● Build a sequence of instructions
Lessons	**Welcome to CodeHop** ● 15-minute lesson, which can be combined with the next lesson if time allows. ● Explore the CodeHop Playground and learn how to log in. **Unplugged Introduction to Programming: 1** ● Create and sequence unplugged programs using CodeHop. **Coding Card Game: Sequences** ● Work together to create a sequence of instructions to move Scout through a maze.

Unit 2: Getting Started (3 lessons)

Students will explore the fundamentals of how computers work, practice identifying key components like hardware and software, and build essential mouse and keyboard skills through interactive activities.

Objectives / Topics Covered	● Understand what a computer is, how it's used, and basic troubleshooting steps ● Identify input and output devices, as well as hardware and software components ● Practice mouse and keyboarding skills
Lessons	**Computer Basics: Introduction** ● Learn what a computer is, how we use it, and what to do when it doesn't work. Identify input, output, hardware, and software. **Keyboard Introduction** ● Demonstrate mouse skills by dragging and clicking with the mouse in multiple games. **Mouse Practice** ● Use the letters, numbers, and basic functions of the keyboard effectively.

Unit 3: Unplugged Exploration (2 lessons)

In this unplugged exploration, students will learn to create step-by-step sequences by choreographing a dance and understand how events can trigger actions through interactive role-playing.

Objectives / Topics Covered	● Algorithmic thinking with step-by-step sequences of instructions ● Events in programming
Lessons	**Sequences** ● Create a sequence of step-by-step instructions for a dance. **Acting with Events** ● Act out how an event can trigger an action.

Unit 4: Programming Introduction (6 lessons)

In this sequential, story-driven CodeHop exploration, students will progressively build their coding skills by navigating the CodeHop interface, adding and modifying characters and backgrounds, using motion blocks to animate characters, responding to events, and culminating in the creation of a celebratory scene that showcases their understanding of sequencing and storytelling.

Objectives / Topics Covered	● The CodeHop interface ● Add, delete, and modify characters in a project ● Backgrounds and pages ● Build a sequence of motion blocks
Lessons	**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.

Unit 5: Sequences & Events (7 lessons)

In this six-week unit on Sequences & Events, students will develop computational thinking by sequencing everyday tasks, create digital artwork using painting tools, and build interactive programs in CodeHop using events, as well as show/hide and grow/shrink blocks to control character behavior and animation.

Objectives / Topics Covered	● Computational thinking ● Build familiarity with drawing tools ● Build a program that responds to different types of events with sequences of instructions ● Additional blocks
Lessons	**Computational Thinking: Morning Routines** ● Identify patterns, break down tasks, sequence steps, and simplify processes in their morning routines. **Drawing Tools: Fairy Tale Painting** ● Use painting tools to create a fairy-tale scene. **Introduction to Events** ● Create a program using different types of events. **Sequences: Snowball Fight** ● Create a program using multiple sequences. **Introduction to Show and Hide Blocks (2 lessons)** ● In this two-part project, use “show” and “hide” blocks in a sequence to make characters appear and disappear. **Introduction to Grow and Shrink Blocks** ● Create a program that uses “grow” and “shrink” blocks to change the size of characters.

Unit 6: Pages (2 lessons)

Students will learn how to create multi-page programs and use the “go to page” block to navigate between pages, enhancing the structure and interactivity of their projects.

Objectives / Topics Covered	● Use multiple pages ● Use the “go to page” block
Lessons	**Introduction to Pages** ● Create a program with multiple pages. **Using the Go to Page Block** ● Create a program including a “go to page” block to switch from page to page in an activity.

Unit 7: Block Exploration (6 lessons)

Students will deepen their CodeHop skills by using sound, speed, and say blocks to animate transportation scenes, creating personalized "All About Me" projects, practicing debugging of event and motion sequences, and applying their knowledge in a two-part interactive bowling game project.

Objectives / Topics Covered	● Use “say,” “sound,” and speed blocks ● Find and fix bugs in sequences ● Apply coding skills to build more complex projects
Lessons	**Transportation Speeds and Sounds** ● Use “say” or “sound” blocks and speed blocks to program modes of transportation. **All About Me! (2 lessons)** ● In this two-part project, create a program that tells information about their favorites. **Debugging: Events and Motion** ● Find and correct bugs in sequences. **Bowling Game (2 lessons)** ● In this two-part project, apply their coding skills to create an interactive bowling game.

Unit 8: Loops (1 lesson)

Students will learn how to use loops to repeat actions efficiently and explain how looping simplifies repetitive coding tasks.

Objectives / Topics Covered	● Loops in programming
Lessons	**Loops** ● Create a program using loops and explain how loops are used to repeat code.

Unit 9: Culmination Projects (4 lessons)

Students will apply their coding skills to program an interactive Chicken Crossing game with adjustable difficulty and create a dynamic wildlife scene using events, sequences, and loops.

Objectives / Topics Covered	• Culminating creative projects
Lessons	**Chicken Crossing Game (2 lessons)** • In this two-part project, program an interactive player character and adjust the difficulty of play in a game. **Wildlife Scene Project (2 lessons)** • In this two-part project, create a wildlife scene using events, sequences, and loops.

Unit 10: Digital Literacy (5 lessons)
Students will explore essential tech concepts by learning how to gather and present information through research, identify private vs. personal data to stay safe online, understand how data is stored and organized through file management, and create programs that illustrate the impacts of technology on our world.

Objectives / Topics Covered	• What data is, and how it can be collected and used • Finding information using research sources • Displaying data visually • Private vs personal information • Impacts of technology
Lessons	**Introduction to Research Project (2 lessons)** • In this two-part project, find information using research sources and create a program to communicate their research visually. **Keeping Information Safe** • Identify private and personal information. **Introduction to Data Storage and Files** • Recognize that computers store data as files and model how data is collected and stored. **Impacts of Technology in our World** • Create a program to demonstrate how technology impacts our world.

Nevada Computer Science and Integrated Technology: Kindergarten

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**.	