Nevada Computer Science and Integrated Technology: 4th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nevada Computer Science and Integrated Technology: 4th Grade Course** introduces students to foundational programming concepts through CodeHop, 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.

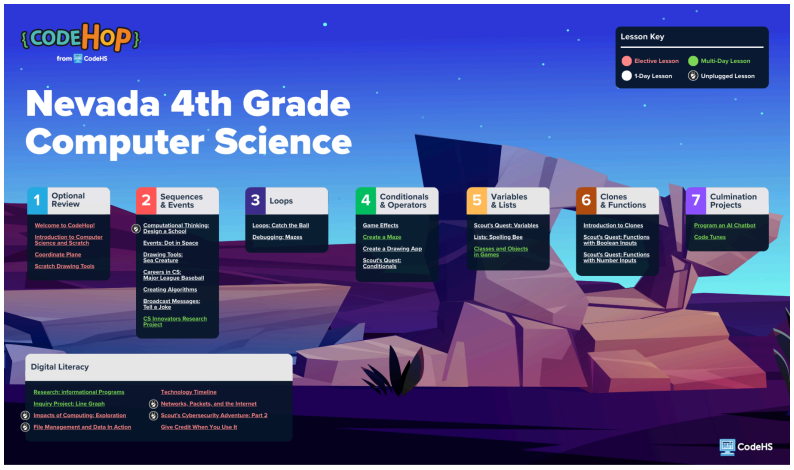
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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 the **CodeHop interface** 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/26147/overview?lang=en



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

Course Breakdown

Unit 1: Optional (4 optional lessons)

Students are optionally introduced to the platform and can build foundational skills by exploring basic computer science vocabulary and tools. They'll create simple programs, design custom sprites and backdrops, and use the coordinate plane to animate creative scenes.

Objectives / Topics Covered	● Logging in and navigating the platform ● Define key computer science vocabulary ● Create a simple program ● Use the coordinate plane to create an open-ended animation ● Customize sprites and backdrops using drawing tools
Lessons	**Welcome to CodeHop!** ● 15-minute lesson - combine with the next lesson if time allows. ● Explore the Playground and learn how to log in. **Introduction to Computer Science** ● Define important computer science vocabulary and create a simple program. **The Coordinate Plane** ● Create an open-ended animation using the coordinate plane. **Drawing Tools** ● Create customized sprites and backdrops using the drawing tools.

Unit 2: Sequences & Events (8 lessons)

In this Sequences & Events unit, students will explore how sequences, events, and algorithms drive program behavior and communication between sprites. They'll build interactive projects using event blocks and broadcast messages, investigate real-world applications of coding in sports and innovation, and practice abstracting information to create timelines and informational programs.

Objectives / Topics Covered	● Create a program using multiple types of event blocks ● Explain how coding is used in sports and create a timeline program ● Program and compare multiple algorithms ● Use broadcast messages to program a knock-knock joke between sprites ● Research a computer science innovator and create an informational program based on abstracted facts
Lessons	**Computational Thinking: Design a School** ● Use computational thinking to design a school. **Events: Dot in Space** ● Create a program using multiple types of event blocks. **Drawing Tools: Sea Creatures** ● Use all of the image editing tools to create and program deep-sea creatures. **Careers in CS: Major League Baseball** ● Explain how coding can be used in sports, and abstract events from an article to retell important events in a timeline program. **Creating Algorithms** ● Program multiple algorithms and assess which one best meets their needs. **Broadcast Messages: Tell a Joke** ● Use broadcast messages to program two sprites to tell a knock-knock joke. **CS Innovators Research Project (2 Lessons)** ● In this two-part project, research an innovator and abstract facts from an article to use as an informational program.

Unit 3: Loops (2 lessons)

In this Loops unit, students will strengthen their understanding of repetition in programming by using different types of loops to build interactive projects. They will also practice decomposing and debugging code to ensure their programs run as intended.

Objectives / Topics Covered	● Use two types of loops to create a simple game ● Decompose and debug a program to make it function correctly
Lessons	**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 4: Conditionals & Operators (5 lessons)

In this unit, students will enhance their programs by using conditional statements and interactive inputs. Through projects like a maze game, a drawing app, and feedback-driven game updates, students will apply if/then logic and user controls to create dynamic, engaging experiences.

Objectives / Topics Covered	● Modify a game by adding effects and incorporating peer feedback ● Draw a maze and program a character to navigate it ● Create a drawing app using keyboard/mouse inputs, loops, and conditionals ● Create a program using if/then conditionals
Lessons	**Game Effects (2 Lessons)** ● Modify a game to add engaging effects and make updates to their game based on peer feedback. **Create a Maze** ● Draw a maze backdrop and program Scout to navigate through the maze. **Create a Drawing App** ● Create a drawing app by programming keyboard and mouse inputs, loops, and conditional statements. **Scout's Quest: Conditionals** ● Create a program using if/then conditionals. *Part 4 of 4 in Scout's Quest skill review series.*

Unit 5: Variables & Lists (4 lessons)

In this unit, students will create games like a spelling bee and a randomized object game, applying key concepts such as tracking points, storing data, and using object properties to enhance gameplay.

Objectives / Topics Covered	● Create and use variables to track points in a program ● Use lists to build a spelling bee game ● Create an interactive game using classes, objects, and randomizers
Lessons	**Scout's Quest: Variables** ● Create and use variables to track points in a program. *Part 2 of 4 in Scout's Quest skill review series.* **Lists: Spelling Bee** ● Use lists to create a spelling bee game. **Classes and Objects in Games (2 Lessons)** ● Create an interactive game and use randomizers to change the characteristics of objects.

Unit 6: Clones & Functions (3 weeks)

In this Clones & Functions unit, students will explore how to create reusable and dynamic code using cloning and functions. They'll build animations with clones and develop programs using functions with boolean and number inputs to control behavior and create interactive outputs.

Objectives / Topics Covered	● Create an animation using clones and explore program limitations ● Create a function with a boolean input to perform actions based on a condition ● Use functions with number inputs to create a drawing
Lessons	**Introduction to Clones** ● Create an animation using clones and investigate the limitations of their program. **Scout's Quest: Functions with Boolean Inputs** ● Create a function including a Boolean input to perform different actions based on whether a password is correct. *Part 1 of 4 in Scout's Quest skill review series.* **Scout's Quest: Functions with Number Inputs** ● Create a drawing using functions with number inputs. *Part 3 of 4 in Scout's Quest skill review series.*

Unit 7: Culmination Projects (4 lessons)

In this unit, students will apply advanced programming concepts to build interactive, personalized applications. Through projects like designing an AI chatbot and creating a custom music player, they'll use lists, variables, operators, and conditionals to develop programs that respond to user input and showcase creativity.

Objectives / Topics Covered	● Use lists to create an AI chatbot that stores prompts and responses ● Use variables, operators, and conditionals to build a custom music player
Lessons	**Program an AI Chatbot (2 Lessons)** ● Two-part project, use lists to create a chatbot to store prompts, responses, and answer questions. **Code Tunes (2 Lessons)** ● Two-part project, use variables, operators, and conditionals to create their own custom music player.

Unit 8: Digital Literacy (10 lessons)

In this Digital Literacy unit, students will explore how to responsibly navigate, communicate, and create in the digital world. They'll conduct research, visualize data, examine the evolution of technology, practice safe online habits, and learn to give proper credit—all while building meaningful digital projects like PSAs, timelines, and secure communication models.

Objectives / Topics Covered	● Create a PSA by summarizing key ideas from research on healthy sleep habits

	● Follow the inquiry process and display investigation results using a line graph ● Promote internet positivity and develop a code of conduct for responsible online behavior ● Create an interactive timeline to show the evolution of music player technology ● Explain how music player technology has influenced cultural practices ● Describe how information is sent over the internet using networks and packets ● Model and implement a secure communication system ● Practice online safety habits and understand cybersecurity tools and concepts
Lessons	**Research: Informational Programs (2 Lessons)** ● Examine the information from different resources and creatively communicate the main ideas in those sources by creating a Public Service Announcement (PSA) on healthy sleep habits. **Inquiry Project: Line Graph (2 Lessons)** ● Follow the inquiry process and modify a program to display the results of their investigation. **Impacts of Computing: Exploration** ● Explain how technology and culture influence each other and create a project showing a past technology, its present version, and an improvement idea. **File Management and Data In Action** ● Explain that different types of digital data take up different amounts of space and can be stored in different ways. **Technology Timeline** ● Create an interactive timeline to illustrate the key developments in music player technology. ● Explain how music player technology has influenced cultural practices. **Networks, Packets, and the Internet** ● Explain how information is communicated through the Internet. ● Model how communication is broken into smaller pieces, transmitted as packets, and reassembled at the destination. ● Design and implement a secure communication method within the classroom. **Scout's Cybersecurity Adventure: Part 2** ● Demonstrate how to stay safe online by practicing secure habits and understanding the tools and technologies that protect your information. **Give Credit When You Use It** ● Search for information to answer questions online and provide proper attribution to sources.

Nevada Computer Science and Integrated Technology: 4th 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.
Design-Your-Own-Lesson Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch 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 **Elementary Resources Page**.	