Nevada Computer Science and Integrated Technology: 2nd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nevada Computer Science and Integrated Technology: 2nd Grade** introduces students to foundational programming concepts through **CodeHopJr interface**, 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.

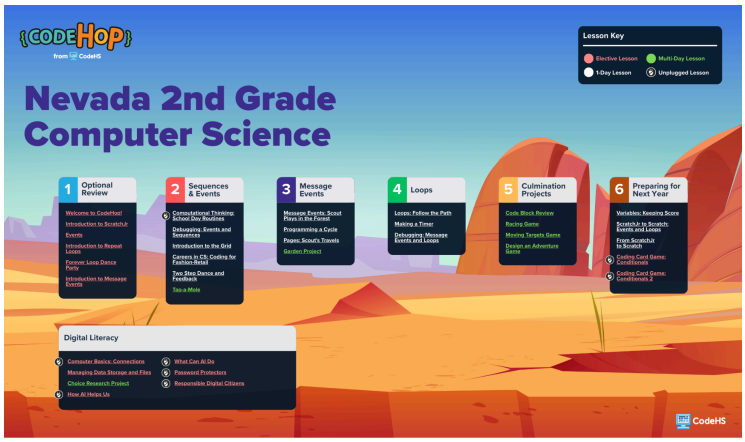
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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 **CodeHopJr interface** 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/26143/overview?lang=en



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

Course Breakdown

Unit 1: Optional (6 optional lessons)

In this Optional Review unit, students can revisit key computer science concepts to reinforce their foundational skills. Lessons include logging into CodeHop, exploring the interface, and reviewing essential programming ideas like events, repeat loops, and message events through engaging activities such as a dance party and a relay race. These lessons are ideal for extra practice or as a refresher before moving forward.

Objectives / Topics Covered	● Log in to CodeHop and explore the Playground ● Navigate the interface to create a scene with characters ● Explain what an event is and use multiple event blocks in a program ● Use repeat loops to run a section of code multiple times ● Create a sequence using a “repeat forever” loop for repeated actions ● Program character interactions using message events in a relay race
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. **Introduction to Programming** ● Navigate the interface to create a scene with characters. **Events** ● Explain what an event is in programming and use multiple event blocks in a program. **Introduction to Repeat Loops** ● Use repeat loops to run a section of code multiple times. **Forever Loop Dance Party** ● Create a sequence using a “repeat forever” loop to make characters repeat actions. **Introduction to Message Events** ● Program a relay race that uses messages to cause characters to interact.

Unit 2: Sequence & Events (7 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, then use the CodeHop interface to create interactive scenes with event blocks, motion, attributions, and effects. The unit also explores debugging.

Objectives / Topics Covered	● Use computational thinking to identify patterns and sequence steps in school day routines ● Find and fix errors in code involving events and sequences ● Use the grid to move characters to specific locations on the stage ● Explore how coding is used in fashion design and animate a fashion-themed character ● Create and revise a program based on peer feedback, including giving attribution ● Design an interactive Tap-a-Mole game using events
Lessons	**Computational Thinking: School Day Routines** ● Use computational thinking concepts to identify patterns, break down tasks, sequence steps, and simplify processes in their school day routines. **Debugging: Events and Sequences** ● Find and fix errors in the provided code.

	Introduction to the Grid ● Use the grid feature to move characters to a specific location on the stage. **Careers in CS: Coding for Fashion-Retail** ● Explain how coding helps create and improve fashion designs, and create a program to design and animate a fashion character. **Two-Step Dance & Feedback** ● Create a program and revise it based on peer feedback, and give attribution to a peer who helped improve their work. **Tap-a-Mole Game (2 lessons)** ● Create an interactive game using events.

Unit 3: Message Events (5 lessons)

In this Message Events unit, students will learn how to control the flow of a program by using message events to coordinate actions between characters and scenes. Through projects like modeling a cycle, animating garden growth, and guiding Scout's travels across pages, students will combine events with visual effects and page transitions to build dynamic, interactive programs.

Objectives / Topics Covered	● Use message events to control the flow of a program ● Model a cycle using message events ● Use messages to help a character travel between pages ● Animate seed growth in a garden using message events, grow, shrink, hide, and show blocks
Lessons	**Message Events: Scout Plays in the Forest** ● Use message events to control the flow of a program. **Programming a Cycle** ● Use message events to model a cycle. **Pages: Scout's Travels** ● Use messages to help Scout travel between pages in a program. **Garden Project (2 lessons)** ● Use message events, grow, shrink, hide, and show blocks to animate seeds growing in a garden.

Unit 4: Loops (3 lessons)

In this Loops unit, students will deepen their understanding of repetition in programming by identifying patterns and using loops to control actions. They'll create programs like path-following and timers, and strengthen their problem-solving skills by debugging code that uses message events and loops.

Objectives / Topics Covered	● Identify patterns and create a program using loops ● Use loops, wait blocks, and turn blocks to create and compare timers with different speeds ● Debug errors involving message events and loops in a given program
Lessons	**Loops: Follow the Path** ● Identify patterns and create a program using loops. **Making a Timer** ● Use loops, wait blocks, and turn blocks to create and compare two timers with different speeds. **Debugging: Message Events and Loops** ● Find and fix (debug) message events and loop errors in the provided code.

Unit 5: Culmination Projects (10 lessons)

In this Culmination Projects unit, students will apply their full range of coding skills to design and build creative, interactive programs. Through projects like a racing game, a moving target game, and a multi-page adventure story, they'll demonstrate their understanding of events, loops, sequences, messages, and more, culminating in a comprehensive review of the coding concepts they've learned.

Objectives / Topics Covered	● Use a variety of coding blocks and explain their functions within a program ● Create an interactive racing game using events, loops, and messages ● Build a moving target game using sequences, events, and pages ● Design a story-based, multi-page adventure game using learned computer science skills
Lessons	**Code Block Review (2 lessons)** ● Use a variety of coding blocks in a program and explain their function within the program. **Racing Game (2 lessons)** ● Create an interactive racing game with events, loops, and messages. **Moving Targets Game (3 lessons)** ● Create a moving target game using sequences, events, and pages. **Design an Adventure Game (3 lessons)** ● Create a story-based, multi-page game using the Computer Science skills they have learned.

Unit 6: Preparing for Next Year (3 lessons)

Students will begin using variables to track information like scores and transition from CodeHopJr to the more advanced CodeHop Blocks. They'll build simple programs in the CodeHop interface using events, loops, and variables to expand their coding skills and project complexity.

Objectives / Topics Covered	● Create a program that simulates keeping score using a variable ● Navigate the basic interface to build a simple program ● Use events and loops to create an interactive program
Lessons	**Variables: Keeping Score** ● Create a program that simulates keeping score using a variable **From CodeHopJr to CodeHop Blocks: Part 2** ● Create a program that uses an event and a loop. ● **From CodeHopJr to CodeHop Blocks** ● Navigate the basic interface to create a simple program.

Unit 7: Digital Literacy (8 lessons)

Students will build foundational computer knowledge and explore the role of technology in daily life. They'll learn how computers work, how to stay safe online, communicate research through programming, and examine the growing impact of artificial intelligence on problem-solving, jobs, and decision-making.

Objectives / Topics Covered	● Understand what a computer is. ● Communicate research findings through a programming project ● Explain how AI assistants answer questions and how AI impacts problem-solving and jobs. ● Explain how AI uses data, and compare human vs. AI tasks ● Understand the importance of usernames and passwords and how to

	protect login information ● Explain responsible digital citizenship and digital footprints.
Lessons	**Computer Basics: Connections** ● Learn what a computer is, how we use it, and what to do when it doesn't work. They will also be able to identify input, output, hardware, and software, and explain how they work together. **Managing Data Storage and Files** ● Recognize that computers store data as files and model how data is collected and stored. **Choice Research Project (2 lessons)** ● Communicate research findings through programming. **How AI Helps Us** ● Explain how AI assistants answer questions by observing a demonstration, and describe how AI helps people by solving problems and changing jobs. **What Can AI Do?** ● Identify tools that use AI, explain that AI uses data to learn and make decisions, and compare tasks that are better suited for humans versus AI. **Password Protectors** ● Understand the importance of usernames and passwords and demonstrate strategies to keep login information safe. **Responsible Digital Citizens** ● Explain what it means to be a responsible digital citizen, including understanding digital footprints and knowing how to report concerns.

Unit 8: Optional Preparing for Next Year (2 lessons)

In this section, students will collaborate to create sequences of instructions using conditionals to help Scout navigate through a maze.

Objectives / Topics Covered	● Create a sequence of instructions using conditionals to move a character through a maze ● Collaborate with peers to apply conditional logic in problem-solving scenarios
Lessons	**Coding Card Game: Conditionals** ● Work together to create a sequence of instructions with conditionals to move Scout through a maze. **Coding Card Game: Conditionals 2** ● Work together to create a sequence of instructions with conditionals to move Scout through a maze.

Nevada Computer Science and Integrated Technology: 2nd Grade

Course Supplemental Materials

Resources	Description
Parent Welcome Letter (Spanish)	Send this letter home to introduce families 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 (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**.	