Idaho Computer Science 1st Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

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

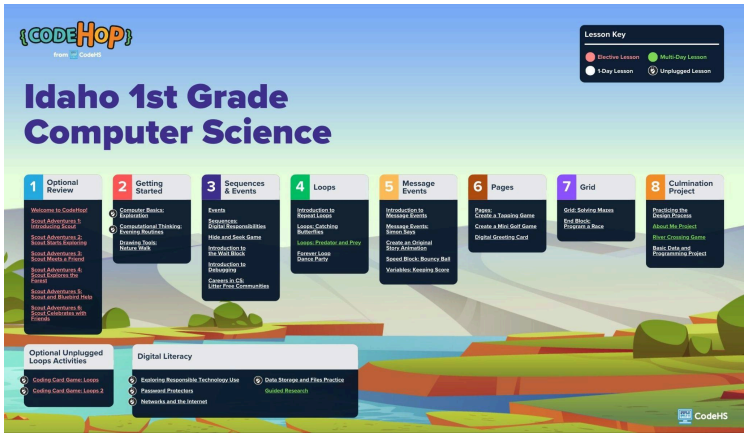
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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**, with each lesson approximately 45 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 **ScratchJr** embedded and saved in the CodeHS 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/26465/overview



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

Course Breakdown

Optional Review

This optional module supports students in revisiting key skills such as logging in, navigating ScratchJr, and reviewing basic programming concepts introduced in previous lessons.

Objectives / Topics Covered	● Refresh familiarity with the CodeHop Playground and login process. ● Revisit character, page, and background creation in ScratchJr. ● Reinforce sequences and motion block usage in a narrative structure.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and navigate the Playground. Can be used as a warm-up or review. **Scout Adventures 1-6** ● Follow Scout the Squirrel through a six-part interactive story that helps students build foundational ScratchJr skills.

Unit 1: Getting Started (3 lessons)

This unit reestablishes foundational computer knowledge and encourages students to apply computational thinking in everyday contexts.

Objectives / Topics Covered	● Identify computer components and functions. ● Use computational thinking to break down routines. ● Explore digital drawing tools.
Lessons	**Computer Basics: Exploration** ● Learn how computers work and understand hardware, software, input, and output. **Computational Thinking: Evening Routines** ● Apply pattern recognition and sequencing to daily tasks. **Drawing Tools: Nature Walk** ● Use ScratchJr's drawing tools to create a nature-inspired scene.

Unit 2: Sequences & Events (6 lessons)

This unit focuses on sequencing and events to create interactive stories, animations, and games while developing foundational problem-solving skills.

Objectives / Topics Covered	● Use sequences to program step-by-step actions. ● Utilize various event blocks to trigger responses. ● Practice debugging and responsible digital behavior.
Lessons	**Events** ● Explain events and use multiple event types in a single program. **Sequences: Digital Responsibilities** ● Create a program demonstrating responsible online behavior. **Hide and Seek Game** ● Use the hide block to make an interactive game. **Introduction to the Wait Block** ● Add pauses to character actions using the wait block. **Introduction to Debugging** ● Identify and correct mistakes in sequences. **Careers in CS: Litter Free Communities** ● Explore real-world applications of computer science in solving community problems.

Optional Unplugged Loops Activities

This optional unplugged unit gives students a hands-on way to practice programming concepts without using devices. By working together to move Scout through a maze using coding cards, students reinforce sequencing and looping skills in a fun, collaborative setting.

Objectives / Topics Covered	● Practice creating and following sequences of instructions without a screen. ● Use loops to simplify repeated actions in a physical coding activity. ● Build teamwork and communication skills while applying coding logic.
Lessons	**Coding Card Game: Loops** ● Work together to guide Scout through a maze using cards that represent movement and repeat loops. **Coding Card Game: Loops 2** ● Tackle a new maze challenge by building efficient sequences with loops to help Scout reach the goal.

Unit 3: Loops (5 lessons)

Students implement loops in ScratchJr to simplify code, add rhythm, and create animations that repeat.

Objectives / Topics Covered	● Use different types of loops to repeat code. ● Combine loops with motion and event blocks. ● Develop games and animations with dynamic elements.
Lessons	**Introduction to Repeat Loops** ● Use repeat loops to make a character perform an action multiple times in a row. **Loops: Catching Butterflies** ● Animate butterflies using show/hide and repeat loops. **Loops: Predator and Prey (2 part lesson)** ● Use loops, events, and sequences to simulate animal interactions. **Forever Loop Dance Party** ● Use the "repeat forever" loop to create a continuous dance party animation.

Unit 4: Message Events (4 lessons)

Students learn to communicate between characters using message blocks for complex, interactive storytelling.

Objectives / Topics Covered	● Use message blocks to coordinate character actions. ● Build multi-character interactions.
Lessons	**Introduction to Message Events** ● Program a relay race using message events. **Message Events: Simon Says** ● Create a “Simon Says” game where one character triggers multiple others.

Unit 5: Pages (3 lessons)

In this unit, students use ScratchJr’s page navigation tools to build interactive, multi-scene projects. They will apply looping, messaging, and event blocks to create simple games and animated greetings.

Objectives / Topics Covered	● Use the “go to page” block to move between pages. ● Combine messages, loops, and events for interactivity. ● Create personalized animations and games with multiple scenes.
Lessons	**Pages: Create a Tapping Game**

	● Design an interactive game that switches between pages when a character is tapped. **Create a Mini Golf Game** ● Students will be able to use messages and loops to create a mini golf game. **Digital Greeting Card** ● Students will be able to create a digital greeting card with events and loops.

Unit 6: Grid (2 lessons)

This unit introduces students to spatial thinking using ScratchJr's grid and "end" block features. Students will learn how to control movement precisely and complete animations with clear stopping points.

Objectives / Topics Covered	● Use the grid to guide character movement through designed mazes. ● Apply the "end" block to complete animations and programs. ● Develop logic for timing and movement in programmed sequences.
Lessons	**Grid: Solving Mazes** ● Design a maze and use grid-based movement to guide a character from start to finish. **End Block: Program a Race** ● Program a race between characters and use the "end" block to stop the animation at the finish line.

Unit 7: Culmination Projects (6 lessons)

In this unit, students apply the design process to create original projects demonstrating problem-solving, personal expression, and technical skill.

Objectives / Topics Covered	● Plan, create, and revise ScratchJr projects. ● Use sequences, events, loops, and variables. ● Model real-world or imaginative scenarios.
Lessons	**Practicing the Design Process** ● Use the design process to solve a simple real-world problem. **About Me Project (2 part lesson)** ● Design a program that shares personal characteristics, interests, and favorite things. **River Crossing Game (2 part lesson)** ● Create an obstacle game with difficulty controlled by speed blocks. **Basic Data and Programming** ● Collect and present data through a visual program.

Unit 8: Digital Literacy (5 lessons)

Students will develop foundational digital skills by learning how to create presentations, protect personal information, understand file storage, and conduct guided research.

Objectives / Topics Covered	● Create a simple presentation using digital tools. ● Understand how data is stored and organized as files. ● Learn strategies to keep usernames and passwords secure. ● Research a topic and share findings using a digital program.
Lessons	**Exploring Responsible Technology Use** ● Understand how to use technology safely and responsibly. **Password Protectors** ● Understand the role of usernames and passwords and how to keep them safe. **Networks and the Internet** ● Explain how networks work and how messages are shared online.

	Data Storage and Files Practice ● Learn how computers store information and explore how files hold data. **Guided Research (2 part lesson)** ● Find information on a topic and create a visual program to present research.

Idaho Computer Science 1st 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.
All of these resources and more are found on the **Elementary Resources Page**.	