



Utah Computer Science 1st Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Utah 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.

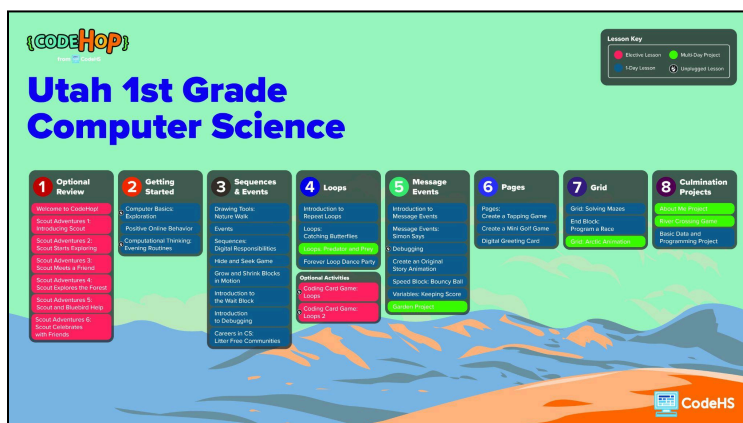
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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/25764/overview>



A clickable PDF can be found at <https://codehs.com/UT-CSRoadmaps>

Course Breakdown

Optional Review

This optional review unit is designed to support students who need more time exploring ScratchJr or who would benefit from additional practice before or after completing core lessons. The Scout Adventures lessons offer a sequential, story-based experience to reinforce key skills in a fun and engaging way.

Objectives / Topics Covered	<ul style="list-style-type: none">● Review core ScratchJr skills including sequences, motion, events, and page transitions.● Strengthen confidence using the ScratchJr interface through a story-based project.● Provide flexible options for differentiation, reinforcement, or introductory practice.
Lessons	<p>Welcome to CodeHop! (15 minute lesson)</p> <ul style="list-style-type: none">● Introductory lesson to help students log in and explore the Playground; can be used before a full lesson or on its own. <p>Scout Adventures 1-6</p> <ul style="list-style-type: none">● Follow Scout the Squirrel through a six-part interactive story that helps students build foundational ScratchJr skills.

Unit 1: Getting Started (3 lessons)

In this unit, students will explore the basics of computing by learning the functions of computers and their components, practice identifying positive and negative online behaviors through ScratchJr, and apply foundational computational thinking skills like pattern recognition, sequencing, and task decomposition to real-life routines.

Objectives / Topics Covered	<ul style="list-style-type: none">● Understand what a computer is and how it is used.● Identify and categorize computer components as input, output, hardware, or software.● Recognize appropriate and inappropriate online behavior.● Apply computational thinking strategies to everyday tasks.
Lessons	<p>Computer Basics: Exploration</p> <ul style="list-style-type: none">● Identify and categorize input, output, hardware, and software components of a computer. <p>Positive Online Behavior</p> <ul style="list-style-type: none">● Use ScratchJr to show examples of safe and unsafe choices when interacting online. <p>Computational Thinking: Evening Routines</p> <ul style="list-style-type: none">● Break down everyday routines into steps while identifying patterns, sequences, and simplifications.

Unit 2: Sequences & Events (8 lessons)

In this unit, students explore how to use sequences and events to control the behavior of characters. They will build animated scenes, games, and interactive projects while learning how to respond to events, pause actions with wait blocks, and debug errors. The module also connects computer science to real-world problem-solving and digital responsibility.

Objectives / Topics Covered	<ul style="list-style-type: none">● Understand and apply the concept of sequences in programming.● Use event blocks to trigger actions in response to user interaction.● Create interactive programs with visual effects, timing, and size changes.● Identify and correct errors through debugging.
-----------------------------	--

	<ul style="list-style-type: none"> ● Connect computer science skills to real-world issues and careers.
Lessons	<p>Drawing Tools: Nature Walk</p> <ul style="list-style-type: none"> ● Create a nature-walk scene using ScratchJr's drawing tools to customize characters and backgrounds. <p>Events</p> <ul style="list-style-type: none"> ● Use event blocks like green flag, tap, and message to trigger actions in a ScratchJr program. <p>Sequences: Digital Responsibilities</p> <ul style="list-style-type: none"> ● Program a sequence of actions while exploring ways to stay safe and respectful online. <p>Hide and Seek Game</p> <ul style="list-style-type: none"> ● Build an interactive hide-and-seek game using the hide block and tap events. <p>Grow and Shrink Blocks in Motion</p> <ul style="list-style-type: none"> ● Animate characters with motion and use grow/shrink blocks to change their size dynamically. <p>Introduction to the Wait Block</p> <ul style="list-style-type: none"> ● Add pauses between actions using the wait block to better control timing in a program. <p>Introduction to Debugging</p> <ul style="list-style-type: none"> ● Find and fix bugs in a program where actions are out of order or not working as expected. <p>Careers in CS: Litter Free Communities</p> <ul style="list-style-type: none"> ● Explore how computer science can help solve environmental challenges and create a program to sort and clean up virtual litter.

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	<ul style="list-style-type: none"> ● 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	<p>Coding Card Game: Loops</p> <ul style="list-style-type: none"> ● Work together to guide Scout through a maze using cards that represent movement and repeat loops. <p>Coding Card Game: Loops 2</p> <ul style="list-style-type: none"> ● Tackle a new maze challenge by building efficient sequences with loops to help Scout reach the goal.

Unit 3: Loops (2 lessons)

In this unit, students will learn how to make characters repeat actions in their ScratchJr programs using loops. They'll use repeat and forever loops to build fun animations, games, and stories that include repeating movements and patterns. By the end of the module, students will be able to recognize when and how to use loops to make their code shorter and more efficient.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Understand the concept of loops and how they control repetition in programs. ● Use repeat and forever loops to streamline sequences of actions. ● Combine loops with events and sequences to create interactive and dynamic programs.
-----------------------------	---

	<ul style="list-style-type: none"> ● Build animations and games that demonstrate repeating patterns of behavior.
Lessons	<p>Introduction to Repeat Loops</p> <ul style="list-style-type: none"> ● Use repeat loops to make a character perform an action multiple times in a row. <p>Loops: Catching Butterflies</p> <ul style="list-style-type: none"> ● Build a butterfly-catching game using show, hide, and repeat blocks to animate the gameplay. <p>Loops: Predator and Prey (2 part lesson)</p> <ul style="list-style-type: none"> ● Program two animal characters to interact using sequences, events, and repeat loops. <p>Forever Loop Dance Party</p> <ul style="list-style-type: none"> ● Create a dance sequence using the “repeat forever” loop to animate continuous movement.

Unit 4: Message Events (8 lessons)

In this unit, students will learn how to make characters talk to each other in their ScratchJr programs using message events. They’ll explore how to send and receive messages to create interactions between characters, bring stories to life, and build fun, interactive projects. Students will also work on debugging, explore speed and variables, and finish the module by creating their own animated story or growing garden scene.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Understand how message events work to trigger interactions between characters. ● Use “send” and “receive” blocks to create cause-and-effect relationships in programs. ● Debug programs with mistakes and improve logical flow. ● Explore how to change speed and use variables like score. ● Build original projects that combine creativity with coding concepts.
Lessons	<p>Introduction to Message Events</p> <ul style="list-style-type: none"> ● Create a relay race where characters take turns moving by sending and receiving messages. <p>Message Events: Simon Says</p> <ul style="list-style-type: none"> ● Use message events to have one character control the actions of several others in a game of Simon Says. <p>Debugging</p> <ul style="list-style-type: none"> ● Find and fix bugs that cause problems in a sequence of code. <p>Create an Original Story Animation</p> <ul style="list-style-type: none"> ● Design and animate an original story that includes sequences, events, and messages. <p>Speed Block: Bouncy Ball</p> <ul style="list-style-type: none"> ● Use speed and message blocks to make a character move at different speeds in a bouncing animation. <p>Variables: Keeping Score</p> <ul style="list-style-type: none"> ● Build a simple program that keeps track of points using a score variable. <p>Garden Project (2 part lesson)</p> <ul style="list-style-type: none"> ● Animate a seed growing into a plant using message events along with grow, shrink, hide, and show blocks.

Unit 5: Pages (3 lessons)

In this unit, students will learn how to use the “go to page” block in ScratchJr to create multi-page projects. They’ll design games and animations that move from one scene to another, building on their knowledge of events and loops while adding creativity through digital storytelling and design.

Objectives / Topics Covered	<ul style="list-style-type: none"> ● Use the “go to page” block to move between scenes in a ScratchJr project. ● Combine pages, loops, and message events to create interactive and animated experiences.
-----------------------------	---

	<ul style="list-style-type: none"> • Apply creativity to design digital games and greeting cards with smooth transitions.
Lessons	<p>Pages: Create a Tapping Game</p> <ul style="list-style-type: none"> • Design an interactive game that switches between pages when a character is tapped. <p>Create a Mini Golf Game</p> <ul style="list-style-type: none"> • Use message events and loops to program a mini golf game that moves across multiple pages. <p>Digital Greeting Card</p> <ul style="list-style-type: none"> • Build a digital greeting card that uses loops and events to animate a message across different pages.

Unit 6: Grid (4 lessons)

In this unit, students will learn how to use the grid in ScratchJr to help plan and control character movement. They'll design mazes, create animations, and explore how to end actions using the "end" block, all while practicing spatial reasoning and precise programming.

Objectives / Topics Covered	<ul style="list-style-type: none"> • Use the ScratchJr grid to guide character movement accurately. • Plan paths and animations using grid positions. • Apply the "end" block to stop actions at the right time in a program. • Build projects that combine creativity with structured movement.
Lessons	<p>Grid: Solving Mazes</p> <ul style="list-style-type: none"> • Design a maze and use grid-based movement to guide a character from start to finish. <p>End Block: Program a Race</p> <ul style="list-style-type: none"> • Program a race between characters and use the "end" block to stop the animation at the finish line. <p>Grid: Arctic Animation (2 part lesson)</p> <ul style="list-style-type: none"> • Create a winter-themed animation by programming character movement with the help of the grid.

Unit 7: Culmination Projects (5 lessons)

In this unit, students will apply everything they've learned throughout the course to create original projects that showcase their coding skills and creativity. From sharing personal interests to building games and displaying data, these culminating activities give students a chance to express themselves and demonstrate mastery of key ScratchJr concepts.

Objectives / Topics Covered	<ul style="list-style-type: none"> • Combine sequences, events, motion, and visual elements in creative ways. • Apply concepts like speed, data, and interactivity to make programs more engaging. • Reflect personal interests, challenges, and learning through coding.
Lessons	<p>About Me Project (2 part lesson)</p> <ul style="list-style-type: none"> • Design a program that shares personal characteristics, interests, and favorite things. <p>River Crossing Game (2 part lesson)</p> <ul style="list-style-type: none"> • Create a game with obstacles and use speed blocks to adjust difficulty as characters cross a river. <p>Basic Data and Programming</p> <ul style="list-style-type: none"> • Collect simple data and create a program to visually present the results through animation or scenes.

Utah 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 .	