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 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 teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts and build confidence over time.

The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. All lessons are labeled with the specific standards they address to support planning and compliance.

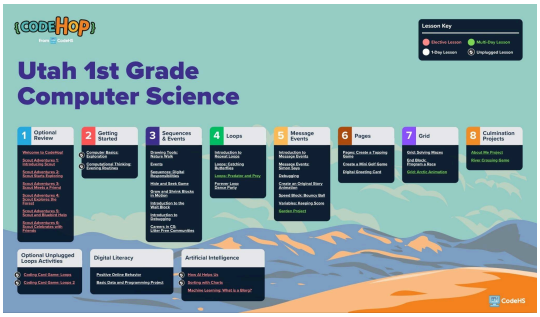
Standards Alignment Note: Standards addressed in each lesson are listed in parentheses after the lesson title. Standards may be reinforced across multiple lessons, and full coverage of state computer science standards is achieved through the course as a whole.

Programming Environment: Students will write and run programs that are saved in students' accounts. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Assessments: Some units conclude with a brief three-question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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 the coding platform 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	● Review core programming skills including sequences, motion, events, and page transitions. ● Strengthen confidence using the programming interface through a story-based project. ● Provide flexible options for differentiation, reinforcement, or introductory practice.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Introductory lesson to help students log in and explore the Playground; can be used before a full lesson or on its own. **Scout Adventures 1-6** (1.AP.3) ● Follow Scout the Squirrel through a six-part interactive story that helps students build foundational skills.

Unit 1: Getting Started (2 lessons)

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

Objectives / Topics Covered	● Understand what a computer is and how it is used. ● Identify and categorize computer components as input, output, hardware, or software. ● Apply computational thinking strategies to everyday tasks.
Lessons	**Computer Basics: Exploration** (1.CS.1, 1.CS.2) ● Identify and categorize input, output, hardware, and software components of a computer. **Computational Thinking: Evening Routines** (1.AP.2, 1.CT.1) ● Break down everyday routines into steps while identifying patterns, sequences, and simplifications. **Getting Started Quiz (3 questions)**

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	● 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. ● Connect computer science skills to real-world issues and careers.
Lessons	**Drawing Tools: Nature Walk** (1.AP.3) ● Create a nature-walk scene using drawing tools to customize characters and

	backgrounds. **Events** (1.AP.2, 1.AP.3, 1.CT.1) ● Use event blocks like green flag, tap, and message to trigger actions in a program. **Sequences: Digital Responsibilities** (1.IC.1) ● Program a sequence of actions while exploring ways to stay safe and respectful online. **Hide and Seek Game** (1.CT.1) ● Build an interactive hide-and-seek game using the hide block and tap events. **Grow and Shrink Blocks in Motion** (1.AP.3) ● Animate characters with motion and use grow/shrink blocks to change their size dynamically. **Introduction to the Wait Block** (1.AP.3) ● Add pauses between actions using the wait block to better control timing in a program. **Introduction to Debugging** (1.AP.2) ● Find and fix bugs in a program where actions are out of order or not working as expected. **Careers in CS: Litter Free Communities** (1.AP.2, 1.AP.3, 1.CT.1) ● Explore how computer science can help solve environmental challenges and create a program to sort and clean up virtual litter. **Sequences & Events Quiz (3 questions)**

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** (1.AP.3, 1.CT.1) ● Work together to guide Scout through a maze using cards that represent movement and repeat loops. **Coding Card Game: Loops 2** (1.AP.3, 1.CT.1) ● Tackle a new maze challenge by building efficient sequences with loops to help Scout reach the goal.

Unit 3: Loops (5 lessons)

In this unit, students will learn how to make characters repeat actions in their 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	● 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. ● Build animations and games that demonstrate repeating patterns of behavior.
Lessons	**Introduction to Repeat Loops** (1.AP.2, 1.AP.3, 1.CT.1) ● Use repeat loops to make a character perform an action multiple times in a row. **Loops: Catching Butterflies** (1.AP.3)

	• Build a butterfly-catching game using show, hide, and repeat blocks to animate the gameplay. **Loops: Predator and Prey** (2 classes | 1.AP.3) • Program two animal characters to interact using sequences, events, and repeat loops. **Forever Loop Dance Party** (1.AP.2, 1.AP.3, 1.CT.1) • Create a dance sequence using the "repeat forever" loop to animate continuous movement. **Loops Quiz (3 questions)**

Unit 4: Message Events (8 lessons)

In this unit, students will learn how to make characters talk to each other in their 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	• 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	**Introduction to Message Events** (1.AP.3, 1.CT.1) • Create a relay race where characters take turns moving by sending and receiving messages. **Message Events: Simon Says** (1.AP.2, 1.AP.3, 1.CT.1) • Use message events to have one character control the actions of several others in a game of Simon Says. **Debugging** (1.AP.2) • Find and fix bugs that cause problems in a sequence of code. **Create an Original Story Animation** (1.AP.3) • Design and animate an original story that includes sequences, events, and messages. **Speed Block: Bouncy Ball** (1.AP.2, 1.AP.3, 1.CT.1) • Use speed and message blocks to make a character move at different speeds in a bouncing animation. **Variables: Keeping Score** (1.AP.1) • Build a simple program that keeps track of points using a score variable. **Garden Project** (2 classes | 1.AP.2, 1.AP.3, 1.CT.1) • Animate a seed growing into a plant using message events along with grow, shrink, hide, and show blocks. **Message Events Quiz (3 questions)**

Unit 5: Pages (3 lessons)

In this unit, students will learn how to use the "go to page" block 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	• Use the "go to page" block to move between scenes in a project. • Combine pages, loops, and message events to create interactive and animated experiences. • Apply creativity to design digital games and greeting cards with smooth transitions.
Lessons	**Pages: Create a Tapping Game** (1.AP.3, 1.CT.1)

	● Design an interactive game that switches between pages when a character is tapped. **Create a Mini Golf Game** (1.AP.3) ● Use message events and loops to program a mini golf game that moves across multiple pages. **Digital Greeting Card** (1.AP.3) ● Build a digital greeting card that uses loops and events to animate a message across different pages. **Pages Quiz (3 questions)**

Unit 6: Grid (4 lessons)

In this unit, students will learn how to use the grid 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	● Use the 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	**Grid: Solving Mazes** (1.AP.2, 1.AP.3, 1.CT.1) ● Design a maze and use grid-based movement to guide a character from start to finish. **End Block: Program a Race** (1.AP.3) ● Program a race between characters and use the "end" block to stop the animation at the finish line. **Grid: Arctic Animation** (2 classes | 1.AP.3, 1.CT.2) ● Create a winter-themed animation by programming character movement with the help of the grid. **Grid Quiz (3 questions)**

Unit 7: Culmination Projects (4 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 programming concepts.

Objectives / Topics Covered	● 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	**About Me Project** (2 classes | 1.AP.2, 1.AP.3, 1.CT.1) ● Design a program that shares personal characteristics, interests, and favorite things. **River Crossing Game** (2 classes | 1.AP.3) ● Create a game with obstacles and use speed blocks to adjust difficulty as characters cross a river.

Digital Literacy (2 lessons)

In this unit, students explore how to use technology safely and responsibly while learning how data can be collected and represented in a program. They practice recognizing positive online behavior and discover how computers can organize and display information in meaningful ways.

Objectives / Topics Covered	• Identify positive and responsible online behavior. • Understand how data can be collected and represented in a program. • Recognize how technology can be used safely in everyday life.
Lessons	**Positive Online Behavior** (1.IC.1) • Create a program that demonstrates examples of positive and responsible online behavior. **Basic Data and Programming** (1.AP.2, 1.DA.1, 1.DA.2) • Collect data and create a program that visually represents the results. **Digital Literacy Quiz (3 questions)**

Optional: Artificial Intelligence (3 lessons)

In this optional unit, students explore how artificial intelligence uses patterns and information to help people solve problems. Through hands-on activities and simple examples, students discover how AI can sort information, recognize patterns, and learn from examples.

Objectives / Topics Covered	• Describe ways artificial intelligence helps people in everyday life. • Recognize that AI uses patterns and information to make decisions. • Explore how AI learns by using examples to recognize new information.
Lessons	**How AI Helps Us** (1.IC.1) • Explore how AI helps people answer questions and solve problems. **Sorting with Charts** (1.DA.2) • Sort objects into groups and create a chart to show patterns. **Machine Learning: What is a Blorg?** (1.DA.2) • Explore how AI learns by identifying a new type of alien. **Artificial Intelligence Quiz (3 questions)**

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 &	This guides students in engaging with computational thinking concepts,

Computational Thinking (Spanish)	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 **CodeHop Resources Page**.	