Utah Interdisciplinary Computer Science First Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Interdisciplinary Computer Science First Grade Course introduces students to foundational programming and computational thinking concepts through ScratchJr, a block-based programming language. Students explore computer science skills while engaging in lessons that integrate coding into math, science, English language arts (ELA), and social studies. This course emphasizes creativity, collaboration, and real-world connections, providing students with a strong foundation in both academic content and computer science.

Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: Review, Guided Practice, Independent Practice, Extension, and Reflection. The course begins with a Getting Started with Coding unit to establish familiarity with basic programming and ScratchJr. After that, subject-area lessons can be taught in any order, with topics listed in approximate order of complexity to support flexible planning and pacing.

The course includes more than 36 lessons, allowing teachers to select the subject-area lessons that best align with their instructional goals and schedule. Lessons are built with spiral review to reinforce key concepts and culminate in engaging projects to showcase student understanding. The “I do, we do, you do” instructional model ensures a gradual release of responsibility, building confidence and independence as students learn to program. Optional coding extension projects and digital literacy lessons are included to provide comprehensive coverage of the Utah Computer Science Standards.

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/21098/overview

A clickable PDF can be found at https://codehs.com/UT-IDCSRoadmaps

Course Breakdown

Getting Started with Coding

Students are introduced to the ScratchJr programming environment and learn basic coding concepts such as events and sequences. Early lessons focus on navigation, adding characters, and understanding cause-and-effect in programming.

Objectives / Topics Covered	● Navigating ScratchJr ● Creating and animating characters ● Understanding events and sequences ● Practicing responsible digital behavior
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to ScratchJr** ● Explore the ScratchJr interface and create a simple scene by adding and moving characters. **Events** ● Learn what an event is in programming and use multiple types of event blocks to control actions. **Sequences: Digital Responsibilities** ● Create a program using sequences while also learning about safe and responsible online behavior.

Coding in Math (9 lessons)

Students apply coding to math concepts such as addition, place value, geometry, and algebraic thinking. Programs model mathematical ideas through animation and interactive elements.

Objectives / Topics Covered	● Representing and solving math problems ● Modeling place value and number comparisons ● Creating shapes and composite figures ● Using the grid to navigate mazes
Lessons	**Adding within 20 and Sequences (Unplugged** ● Write sequences to guide a character through a maze while counting to 20. **Divide Shapes into Equal Parts** ● Animate partitioning of circles and rectangles into equal shares. **Combining Shapes** ● Create composite shapes, then animate them with event and motion blocks. **Grid: Solving Mazes** ● Design a maze and program a character to navigate it using grid coordinates. **Finding an Unknown in a Word Problem** ● Animate a number story to solve for an unknown value. **Story Problems: Add and Subtract within 20** ● Create a scene to model addition or subtraction problems. **Algebraic Thinking: Find an Unknown Number Up to 10 in a Number Story** ● Build an interactive program to solve for unknowns in number stories. **Place Value: Adding Up to 20** ● Show how to break apart numbers into tens and ones using animations. **Greater Than and Less Than: Two-Digit Numbers** ● Create a program that compares two-digit numbers and states if one is greater, less, or equal.

Coding in Science (6 lessons)

Students use ScratchJr to model and explain science topics, from light and sound to animal adaptations and life cycles.

Objectives / Topics Covered	● Modeling physical science concepts ● Representing earth and space science patterns ● Showing life science processes and adaptations
Lessons	**Adaptations and Survival: Camouflage** ● Animate how animals use camouflage to avoid predators or catch prey. **Light and Shadows** ● Show sources of light and how light creates shadows. **Sound and Pitch** ● Model the relationship between vibration and pitch. **Sun and Moon, Day and Night** ● Use loops to represent daily celestial movement. **Phases of the Moon** ● Use message events to animate the Moon's phases. **Animal Life Cycles** ● Show the butterfly life cycle with message events for each stage.

Coding in ELA (5 lessons)

Coding projects reinforce literacy skills such as sentence building, phonics, punctuation, and storytelling.

Objectives / Topics Covered	● Building and reading sentences ● Practicing phonics with digraphs ● Using correct punctuation ● Retelling and creating stories
Lessons	**Build a Sentence** ● Program an interactive activity to write and read sentences aloud. **Phonics: Digraphs** ● Create a game where players match sounds to digraphs. **Punctuation: Write a Great Sentence!** ● Use loops and sequences to add punctuation correctly. **Storytelling Animations Part 2** ● Animate the retelling of a story's sequence. **Original Story Animations – Fiction Story** ● Develop and animate an original fictional story.

Coding in Social Studies (3 lessons)

Students explore community roles, mapping, and economic decision-making through coding.

Objectives / Topics Covered	● Identifying responsibilities and roles ● Reading and creating maps ● Understanding wants and needs
Lessons	**Our Responsibilities** ● Animate scenarios showing responsible behavior in school and at home. **Create a Map** ● Design a map and program a character to travel it.

	Economic Choices ● Use message events to illustrate making choices between wants and needs.

More Coding Lessons & Projects (11 lessons)

Students expand their programming skills with advanced blocks, interactive games, and debugging.

Objectives / Topics Covered	● Using grow/shrink, show/hide, and loop blocks ● Programming interactive games ● Sending messages between characters ● Understanding and using variables and data
Lessons	**Hide and Seek Game** ● Use the hide block to create an interactive game. **Grow and Shrink Blocks in Motion** ● Change a character's size while it moves. **Introduction to Repeat Loops** ● Repeat a set of commands multiple times. **Loops: Predator and Prey (2 part lesson)** ● Program two characters to interact in a looped chase. **Forever Loop Dance Party** ● Make characters perform repeating dance moves. **Message Events: Simon Says** ● Send messages from one character to many others. **Debugging** ● Identify and fix errors in sequences. **Variables: Storing Data with Symbols** ● Show how variables store information in programs. **Basic Data and Programming** ● Collect and display data in a visual program. **Seasonal Project (Summer): Pop the Balloons** ● Make a summer game with events and loops.

Digital Literacy (7 lessons)

Students learn foundational computing skills and digital citizenship, with an introduction to AI concepts.

Objectives / Topics Covered	● Basic computer operations ● Safe and responsible technology use ● Understanding data storage and files ● Research skills ● Introductory AI concepts
Lessons	**Computer Basics: Exploration** ● Learn about hardware, software, and simple troubleshooting. **Keeping Information Safe** ● Identify personal and private information. **Responsible Digital Citizens** ● Understand digital footprints, cyberbullying, and reporting. **Data Storage and Files Practice** ● See how data is stored as files. **Guided Research (2 part lesson)** ● Find reliable information and present it visually. **Machine Learning: What is a Blorg?** ● Explore how AI recognizes patterns in data.

Utah Interdisciplinary Computer Science First 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 **CodeHop Resources Page**.	