Utah Interdisciplinary Computer Science Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Interdisciplinary Computer Science Kindergarten 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/21097/overview

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

Course Breakdown

Getting Started with Coding

Students are introduced to the coding platform, interface navigation, and core programming actions. Through a sequential story-driven unit, they learn to add, modify, and animate characters and backgrounds while building confidence in sequencing commands.

Objectives / Topics Covered	● Navigating the programming environment ● Adding, modifying, and deleting characters ● Using motion blocks to move sprites ● Creating sequences of commands ● Adding backgrounds, pages, and events
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Scout Adventures 1: Introducing Scout** ● Explore the interface and add characters. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a new page. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters. **Scout Adventures 4: Scout Explores the Forest** ● Use motion blocks to move characters. **Scout Adventures 5: Scout and Bluebird Help** ● Build sequences to collect objects. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene using multiple pages, characters, and events.

Coding in Math (7 lessons)

Students apply programming to represent and solve math concepts, from basic operations to patterns and place value. They use events, sequences, loops, and interactive elements to visualize mathematical thinking.

Objectives / Topics Covered	● Representing addition, subtraction, and comparisons ● Creating interactive math problems ● Visualizing place value and decomposition ● Programming shapes and patterns
Lessons	**Decompose Numbers Up to 10** ● Use events to show number decomposition. **Story Problems: Add and Subtract within 10** ● Animate addition or subtraction problems. **Finding an Unknown in a Word Problem** ● Program number stories with missing values. **Greater Than and Less Than: Single-Digit Numbers** ● Create a comparison program. **Creating Shapes** ● Program shapes to respond to user interaction and combine into complex shapes. **Place Value: Ones and Tens** ● Animate counting by ones, fives, and tens. **Patterns and Music** ● Use sequences and loops to create musical patterns.

Coding in Science (6 lessons)

Students explore science topics through coding, modeling real-world phenomena such as weather, motion, and the needs of living things.

Objectives / Topics Covered	● Classifying living and nonliving things ● Modeling needs of organisms and survival ● Representing weather patterns and motion types ● Comparing organisms based on characteristics
Lessons	**Weather and Seasons** ● Animate seasonal weather changes. **Types of Motion** ● Program animations for different motion types. **Living and Nonliving** ● Classify and animate living vs. nonliving. **How Living Things Survive** ● Illustrate survival needs. **Needs of Plants** ● Model plant needs with events and messages. **Comparing Organisms** ● Group animals by characteristics.

Coding in ELA (4 lessons)

Students integrate programming into literacy skills by coding phonics games, sequencing story events, and animating original stories.

Objectives / Topics Covered	● Phonics and letter-sound connections ● Sentence construction ● Sequencing story events ● Story animation and narration
Lessons	**Phonics: Letter Sounds** ● Create an interactive phonics game. **Build a Sentence** ● Program sentences that can be read aloud. **Storytelling Animations** ● Retell a story in sequence. **Original Story Animations – Personal Story** ● Animate a personal story with narration.

Coding in Social Studies (2 lessons)

Students use coding to represent social studies concepts such as roles in the community and spatial awareness.

Objectives / Topics Covered	● Identifying authority figures and their roles ● Using cardinal directions and maps
Lessons	**Who Keeps Us Safe?** ● Animate an authority figure's role. **Cardinal Directions** ● Create an interactive compass rose.

More Coding Lessons & Projects (10 lessons)

Students explore new programming blocks, game mechanics, and creative tools for independent or group projects.

Objectives / Topics Covered	● Using new block types (grow/shrink, show/hide) ● Creating multipage programs ● Implementing loops and interactive game elements
Lessons	**Coding Card Game: Sequences** ● Work together to write sequence instructions. **Coding Card Game: Sequences 2** ● Continue practicing with new sequences. **Drawing Tools: Fairy Tale Painting** ● Use painting tools to create scenes. **Introduction to Grow and Shrink Blocks** ● Change sprite sizes dynamically. **Introduction to Show and Hide Blocks** ● Make sprites appear/disappear. **Introduction to Pages** ● Create multi-page programs. **Loops: Frog and Rabbit** ● Use loops to repeat code. **Forever Loops: Fireworks** ● Repeat sequences indefinitely. **All About Me!** ● Create a personal interactive project. **Chicken Crossing Game (2 part lesson)** ● Program an interactive player game.

Digital Literacy (7 lessons)

Students develop foundational computer and technology skills, including basic operations, data handling, and digital citizenship.

Objectives / Topics Covered	● Computer hardware, software, and basic troubleshooting ● Mouse and keyboard skills ● Data collection, analysis, and AI decision-making ● Networks, research, and digital responsibility
Lessons	**Computer Basics: Introduction** ● Learn parts of a computer and basic functions. **Mouse Practice** ● Practice clicking and dragging. **Keyboard Introduction** ● Learn letters, numbers, and functions. **What Can Data Tell Us?** ● Collect and analyze simple data. **Sorting with Decision Trees** ● Explore AI decision-making. **Using Networks to Connect** ● Learn how networks share information. **Introduction to Research (2 part lesson)** ● Find and present information from sources.

Utah Interdisciplinary Computer Science Kindergarten 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**.	