Utah Interdisciplinary Computer Science Second Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Interdisciplinary Computer Science Second 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/21099/overview

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

Course Breakdown

Getting Started with Coding

Students begin by exploring the CodeHop Playground and the ScratchJr environment. They practice basic navigation, creating scenes, and identifying common coding errors.

Objectives / Topics Covered	● Navigating ScratchJr and CodeHop ● Adding and moving characters ● Understanding and fixing errors in sequences and events
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. **Debugging: Events and Sequences** ● Identify mistakes in event and sequence code, then fix them to make the program run correctly.

Coding in Math (10 lessons)

Students apply programming to math concepts including place value, shapes, algebraic thinking, and problem solving. Programs model and solve problems visually.

Objectives / Topics Covered	● Using grids and coordinates ● Representing addition, subtraction, and place value ● Identifying and drawing shapes ● Modeling money and time concepts
Lessons	**Introduction to the Grid** ● Move characters precisely using grid coordinates. **Counting with Mazes** ● Design a maze that requires a specific number of steps to solve. **Story Problems: Add and Subtract within 100** ● Create a program to model a story problem. **Telling Time** ● Build an analog clock and display time in analog and digital forms. **Create Your Own Story Problem** ● Write and animate a unique math word problem. **Identify Shapes by Attributes** ● Draw and label shapes based on specific properties. **Place Value: Ones, Tens, and Hundreds** ● Connect digits to their place values using interactive animations. **Events and Money** ● Create a program that states the value of different coin sets. **Algebraic Thinking: Finding a Two-Digit Unknown** ● Visualize and solve for unknown numbers in story problems. **3D Shapes** ● Draw 3D shapes and make a quiz game about their attributes.

Coding in Science (6 lessons)

Students use coding to model scientific concepts from physical, earth, and life sciences, showing cause-and-effect and problem-solving solutions.

Objectives / Topics Covered	● Classifying matter ● Modeling environmental changes ● Representing life science adaptations and life cycles ● Designing human solutions inspired by nature
Lessons	**Properties of Matter** ● Use conditionals to sort matter by properties. **Designing Solutions from Nature** ● Show how humans mimic animals or plants to solve problems. **Changing Landforms** ● Animate how volcanic eruptions change the Earth's surface. **Preventing Erosion** ● Compare solutions for preventing erosion. **Changes in the Environment** ● Model causes and effects of environmental changes. **Seed Dispersal** ● Use message events and loops to show how animals spread seeds.

Coding in ELA (3 lessons)

Programming is used to build literacy skills including sentence structure, punctuation, and creative storytelling.

Objectives / Topics Covered	● Writing and reading sentences ● Using punctuation correctly ● Planning and animating stories
Lessons	**Build a Sentence** ● Program sentences that can be read aloud. **Storytelling Animations** ● Retell a story in sequence. **Original Story Animations – Fiction Story** ● Create and animate a fictional story from start to finish.

Coding in Social Studies (2 lessons)

Students code projects that explore how communities interact with and change their environment, as well as the role of technology in the world.

Objectives / Topics Covered	● Modeling community changes ● Understanding technology's impact
Lessons	**Communities Modify Their Environment** ● Animate examples of people changing their surroundings. **Impacts of Technology in Our World** ● Show positive and negative ways technology influences society.

More Coding Lessons & Projects (13 lessons)

Students expand their coding skills with advanced events, loops, message passing, and game design.

Objectives / Topics Covered	● Using loops, events, and pages for complex programs ● Designing interactive games ● Debugging multi-step programs

	● Using variables to track data
Lessons	**Tap-a-Mole Game** ● Program a game using events. **Message Events: Scout Plays in the Forest** ● Use message events to sequence multiple actions. **Pages: Scout's Travels** ● Move between pages using messages. **Loops: Follow the Path** ● Identify and use patterns in loops. **Debugging: Message Events and Loops** ● Find and fix mistakes in complex programs. **Design an Adventure Game (2 part lesson)** ● Build a multi-page, story-driven game. **Moving Targets Game (2 part lesson)** ● Program moving targets with sequences and events. **Variables: Keeping Score** ● Use a variable to track score in a game. **Seasonal Project (Fall): Gathering Apples Game** ● Create a fall-themed collecting game. **Digital Greeting Card** ● Make an animated greeting card using events and loops. **Seasonal Project (Spring): Create a Chase Game** ● Build a chase-style game with loops and events.

Digital Literacy (11 lessons)

Students build foundational technology skills, understand online safety, and explore data and AI concepts.

Objectives / Topics Covered	● Computer systems and connections ● Safe passwords and privacy ● Data storage and visualization ● Network communication ● AI pattern recognition
Lessons	**Computer Basics: Connections** ● Learn hardware, software, and how systems work together. **Password Protectors** ● Create strategies for safe and secure login information. **Managing Data Storage and Files** ● Understand how files are stored and organized. **Exploring Computer Networks** ● Model how networks connect devices and share information. **Machine Learning: AutoDraw** ● Learn how AI recognizes and suggests drawings. **Data Patterns and Predictions** ● Identify patterns in data and create a visual representation. **Advanced Data and Programming (2 part lesson)** ● Collect data from surveys and present it visually. **Giving Credit Through Attributions** ● Learn to credit original creators when reusing ideas. **Choice Research (2 part lesson)** ● Research a topic, assess sources, and share findings visually.

Utah Interdisciplinary Computer Science Second Grade Course Supplemental Materials

Resources	Description
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to their new computer science curriculum.
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.
These resources and more are found on the **CodeHop Resources Page**.	