Utah Interdisciplinary Computer Science Fourth Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Interdisciplinary Computer Science Fourth Grade Course introduces students to foundational programming and computational thinking concepts through Scratch, 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 Scratch. 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 Scratch 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/21101/overview

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

Course Breakdown

Getting Started with Coding

Students are introduced to the CodeHop Playground and build early programming skills in Scratch. They will explore event blocks, loops, and conditionals while completing short, engaging projects.

Objectives / Topics Covered	● Navigating CodeHop and Scratch ● Using event blocks and loops in sequences ● Applying conditionals in interactive programs
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Events: Dot in Space** ● Create a program using multiple event types. **Loops: Catch the Ball** ● Use two types of loops to make a simple game. **Scout's Quest: Conditionals** ● Create a program using if/then conditionals.

Coding in Math (9 lessons)

Students connect math concepts to Scratch programming through classification, multiplication, division, fractions, and data visualization.

Objectives / Topics Covered	● Classifying shapes by lines and angles ● Using loops, conditionals, and variables in math programs ● Creating math-based games and visualizations
Lessons	**Classifying Shapes Using Lines and Angles** ● Categorize shapes and add code comments. **Animating Sprites with Factors** ● Use factors, loops, and waits to animate sprites. **Multi-digit Multiplication and Conditionals** ● Review multiplication with if/then logic. **Division and Conditionals** ● Build an interactive division game using conditionals. **Multiplying Fractions by Whole Numbers** ● Demonstrate fraction multiplication with animations. **Programming and Data Project** ● Collect and present data in an interactive program. **Naming Numbers Game** ● Use variables and operators to match number forms. **House Design with Area and Perimeter (2 part lesson)** ● Design a floor plan using functions, area, and perimeter.

Coding in Science (7 lessons)

Students create Scratch programs to model science concepts such as energy, cycles, environmental responses, and waves.

Objectives / Topics Covered	● Modeling heat, light, and sound ● Representing scientific cycles and processes ● Using variables and loops for simulations

Lessons	**Exploring Heat** ● Show how sunlight affects objects on Earth. **Exploring the Water Cycle** ● Animate the water cycle using broadcast messages. **Sensing and Responding to the Environment** ● Model animal sensory responses with events. **Plant and Animal Cells** ● Create interactive diagrams of cell structures. **How We See: Light Reflection** ● Use conditionals to model light reflection into the eye. **Sound Frequency & Amplitude** ● Use variables to adjust sound and vibration strength. **Wave Generator** ● Draw wave patterns with variables and loops.

Coding in ELA (6 lessons)

Students build literacy and creativity skills by programming animations, interactive readings, and games.

Objectives / Topics Covered	● Animating poetry and narratives ● Creating interactive nonfiction presentations ● Practicing grammar through coding
Lessons	**Animating Poetry** ● Animate and narrate a poem with visual and literary meaning. **Creative Storytelling (2 part lesson)** ● Plan and animate an original story. **Nonfiction Animated Recordings** ● Create animated readings of nonfiction texts. **Grammar Quiz Game** ● Build a conditional-based grammar quiz. **Mad Libs Project** ● Use lists to create a Mad Libs game.

Coding in Social Studies (2 lessons)

Students design interactive programs that demonstrate knowledge of geography, civics, and history.

Objectives / Topics Covered	● Presenting state-specific information ● Modeling civic responsibilities ● Reviewing with digital flashcards ● Integrating micro:bit inputs into projects
Lessons	**State Project** ● Create an interactive program with state facts. **Rights and Responsibilities** ● Simulate voting to demonstrate civic roles. **Digital Flashcards** ● Use broadcast messages for subject review. **Utah Timeline with the BBC micro:bit®** ● Create a micro:bit-powered interactive timeline.

More Coding Lessons & Projects (15 lessons)

Students explore advanced Scratch techniques, conditionals, coordinate systems, variables, and game mechanics.

Objectives / Topics Covered	• Debugging programs and creating algorithms • Using conditionals and coordinate grids • Building complex games and creative apps
Lessons	**Creating Algorithms** • Compare multiple algorithms for a task. **Debugging: Mazes** • Identify and fix errors in maze programs. **Conditionals: Color Sense** • Program actions triggered by colors. **The Coordinate Plane** • Make an animation using the coordinate grid. **Conditionals: Flying Bird** • Build a bird game using conditionals. **Complex Conditionals: Chase the Star** • Use if/then/else in an interactive game. **Scout's Quest: Variables** • Track points with variables. **Pong Game (2 part lesson)** • Make a Pong game with scorekeeping. **Create a Drawing App** • Program a drawing tool with inputs and conditionals. **Snake Game** • Build a snake game using variables and clones. **Designing Solutions for Accessibility (2 part lesson)** • Improve game accessibility through redesign. **Seasonal Project (Spring): Create a Chase Game!** • Make a seasonal chase game with loops and conditionals. **Scratch Digital Greeting Card** • Create a greeting card with loops and events.

Digital Literacy (7 lessons)

Students explore computing systems, online responsibility, networks, AI, and accessibility design.

Objectives / Topics Covered	• Understanding hardware, software, and networks • Promoting positivity and safe practices online • Creating AI-based and accessibility-focused projects
Lessons	**Exploring Computing Systems** • Identify computer parts and troubleshoot. **Computational Thinking: Design a School** • Apply computational thinking to school design. **Internet Positivity** • Develop a digital code of conduct for positive online interactions. **Research: Informational Programs** • Create a PSA using research on healthy habits. **Impacts of Computing: Exploration** • Explore how computing has changed communities and evolved over time. **Program an AI Chatbot** • Build a chatbot using lists. **Networks, Packets, and the Internet** • Model Internet communication with packets. **3D Design: Keyboard Accommodations** • Use Tinkercad to design an accessible keyboard.

Utah Interdisciplinary Computer Science Fourth 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.
Design-Your-Own-Lesson Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch graphic organizer templates. Designed with adaptability and ease of use in mind, these interactive tools transform any subject into an engaging, hands-on learning experience.
These resources and more are found on the **CodeHop Resources Page**.	