Utah Interdisciplinary Computer Science
Fifth Grade Course Syllabus
One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Interdisciplinary Computer Science Fifth 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/21102/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 learning foundational computer science concepts in Scratch. They practice creating algorithms, drawing with code, and using conditionals in interactive projects.

Objectives / Topics Covered	● Navigating CodeHop and Scratch ● Understanding CS vocabulary and basic programming structures ● Writing and evaluating algorithms ● Creating graphics and animations with loops and pen tools ● Using conditionals for decision-making in programs
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to Computer Science and Scratch** ● Define key CS terms and build a simple Scratch program. **Creating Algorithms** ● Program multiple algorithms and assess their efficiency. **Creating Turtle Graphics** ● Use the pen tool and loops to create turtle graphics. **Conditionals: Underwater Exploration** ● Build an underwater scene using conditionals to trigger actions.

Coding in Math (8 lessons)

Students integrate math concepts with Scratch programming to build interactive visualizations, quizzes, and calculators.

Objectives / Topics Covered	● Animating division, multiplication, and fractions ● Classifying shapes by properties and hierarchy ● Using variables, operators, and conditionals for math applications ● Modeling area and measurement conversions
Lessons	**Animating Sprites with Division** ● Animate sprites with loops and wait blocks to show division. **Classifying Quadrilaterals in a Hierarchy** ● Categorize quadrilaterals and document with comments. **Decimal Division and Conditionals** ● Create an interactive decimal division game with levels. **Decimal Multiplication and Conditionals** ● Use conditionals to review decimal multiplication. **Unit Converter** ● Program a measurement converter with variables and conditionals. **Multiplication Quiz Game** ● Build a quiz game with variables, operators, loops, and conditionals. **Add and Subtract Fractions** ● Create a fractions quiz using broadcast messages and patterns. **Area of a Rectangle with Tiles** ● Model fractional area using loops and variables.

Coding in Science (9 lessons)

Students use programming to model scientific concepts, cycles, and processes, incorporating animation, conditionals, and data representation.

Objectives / Topics Covered	● Modeling earth science, ecosystems, and environmental effects ● Simulating day/night and planetary scale ● Representing cycles of matter and physical changes
Lessons	**Constructive and Destructive Processes** ● Animate volcanic changes to Earth's surface. **Exploring Ecosystems** ● Model the flow of energy in an ecosystem. **Animal Classification** ● Create a sorting program for animal groups. **Earth Systems** ● Show wind and rain interactions with the geosphere. **Cycle of Matter** ● Animate matter cycles using events and messages. **Effects of Pollution** ● Use conditionals to demonstrate how actions affect water pollution. **Day and Night** ● Show Earth's rotation causing day/night using comparison operators. **Physical Changes** ● Use variables and conditionals to simulate phase changes of matter. **Scale of Planets in the Solar System** ● Calculate and display planetary scale and properties.

Coding in ELA (5 lessons)

Students apply programming to enhance reading, writing, and analysis skills through interactive games and animations.

Objectives / Topics Covered	● Animating poetry and narratives ● Creating nonfiction presentations ● Practicing punctuation rules through interactive challenges
Lessons	**Animating Poetry** ● Animate and narrate poetry for meaning and tone. **Nonfiction Animated Recordings** ● Build an animated reading of nonfiction text. **Creative Storytelling (2 part lesson)** ● Plan and animate an original story with events and sequences. **Punctuate a Title** ● Make a game that teaches title punctuation using conditionals.

Coding in Social Studies (4 lessons)

Students use programming to represent historical processes, government structures, and geography.

Objectives / Topics Covered	● Modeling how laws are made ● Creating interactive timelines and maps ● Building study tools with digital flashcards
Lessons	**The US Government** ● Use conditionals to model how a bill becomes law. **Digital Flashcards** ● Create interactive flashcards with broadcast messages. **Revolutionary War Timeline** ● Build a timeline with inputs, events, and variables.

	Interactive Map of the 13 Colonies ● Make a map with events, variables, and modular code design.

More Coding Lessons & Projects (18 lessons)

Students strengthen programming skills through creative projects, debugging challenges, and game design

Objectives / Topics Covered	● Loop-based animations and interactive games ● Debugging and program decomposition ● Designing full games and apps with multiple programming concepts
Lessons	**Animation Loops Project (2 part lesson)** ● Create multi-scene animations using repeat loops. **Debugging: Make a Pizza** ● Decompose and fix errors in a program. **Game Mechanics with Comparison Operators** ● Add endgame mechanics using comparisons. **Variables in Dance** ● Use variables to control music pitch and dance speed. **Operators: Coin Flip** ● Make a coin flip simulation with variables and operators. **Conditionals: Mazes** ● Create a maze game using conditional logic. **Plan a Quest** ● Break down and design a quest game. **Pinball Game Project (2 part lesson)** ● Build a functional pinball game with key mechanics. **Game Design Project (2 part lesson)** ● Design and code a game using loops, variables, and conditionals. **Design an App (2 part lesson)** ● Use design thinking to create a problem-solving app. **Seasonal Project (Fall): Gathering Apples Game** ● Create a fall-themed collecting game.

Digital Literacy (7 lessons)

Students develop skills in computing systems, online safety, collaboration, and data visualization.

Objectives / Topics Covered	● Troubleshooting computer hardware and software ● Understanding cyberbullying and respectful communication ● Working with digital data and networks ● Collaborating online and giving attribution
Lessons	**Practicing with Computing Systems** ● Identify computer parts and solve basic problems. **Our Words Have Power (Cyberbullying)** ● Learn about cyberbullying and being an upstander. **File Management and Data Exploration** ● Compare file sizes and storage locations. **Networks and Protocols** ● Understand how protocols enable data transfer. **Using Digital Tools to Create Line Graphs** ● Turn data into line graphs to support claims. **3D Design: Keyboard Accommodations** ● Design an accessible keyboard in Tinkercad®. **Online Collaboration**

	• Explore the benefits and diversity of collaborative work. **Giving Credit Through Attributions** • Attribute sources when creating or remixing programs.

Utah Interdisciplinary Computer Science Fifth 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**.	