Utah Interdisciplinary Computer Science Sixth Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Interdisciplinary Computer Science Sixth 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/21103/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, foundational computer science vocabulary, and essential programming concepts. They will learn to write algorithms, use conditionals, and debug programs in Scratch.

Objectives / Topics Covered	● Navigating CodeHop and Scratch ● Understanding CS vocabulary and programming basics ● Writing and comparing algorithms ● Using conditionals for decision-making ● Debugging programs through decomposition
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to Computer Science and Scratch** ● Define CS vocabulary and create a simple program. **Creating Algorithms** ● Write multiple algorithms and compare their efficiency. **Conditionals: Underwater Exploration** ● Use if/then logic in an underwater scene. **Debugging: Make a Pizza** ● Decompose a program and fix errors to make it work correctly.

Coding in Math (4 lessons)

Students connect math skills to programming by building interactive fraction models, converters, equations, and number representations in Scratch.

Objectives / Topics Covered	● Multiplying and dividing fractions ● Creating unit rate converters ● Writing and solving algebraic equations ● Representing positive and negative numbers
Lessons	**Multiplying and Dividing Fractions** ● Show the relationship between multiplying and dividing fractions in word problems. **Rates and Unit Rates** ● Build a converter using variables and operators. **Evaluate Algebraic Equations** ● Use operators and conditionals to write and solve real-world equations. **Positive and Negative Numbers** ● Represent quantities with positive and negative numbers in an interactive program.

Coding in Science (3 lessons)

Students model science concepts through interactive programs that include scale models, chemical structures, and ecosystem simulations.

Objectives / Topics Covered	● Calculating planetary scale and properties ● Modeling atomic and molecular structures ● Simulating balanced ecosystems and population changes
Lessons	**Scale of Planets in the Solar System** ● Calculate and model planetary scale and properties. **Atoms and Molecules**

	● Use conditionals and broadcasts to model atom combinations. **Balanced Ecosystems** ● Simulate ecosystems and demonstrate how changes affect populations.

Coding in ELA (1 lesson)

Students combine reading comprehension and programming by creating interactive multimedia book reports.

Objectives / Topics Covered	● Summarizing and presenting books using multimedia elements
Lessons	**Multimedia Book Report** ● Create a program that conveys information about a book.

Coding in Social Studies (1 lesson)

Students use programming to explore how geography influences the development of civilizations.

Objectives / Topics Covered	● Explaining geographic effects on early civilizations
Lessons	**Geographic Effects on Early Civilizations** ● Use broadcast messages to explain how geography shaped early civilizations.

More Coding Lessons & Projects (10 lessons)

Students extend their programming skills through collaborative projects, game development, and advanced programming concepts like functions, clones, and classes.

Objectives / Topics Covered	● Collaborating in pair programming ● Using variables, functions, and clones ● Designing games and interactive experiences ● Applying object-oriented concepts
Lessons	**Game Effects** ● Enhance a game with engaging effects and peer feedback updates. **Pair Programming: Create a Band** ● Collaborate to design and code a musical band with keyboard inputs. **Variables in Dance** ● Control pitch and dance speeds with variables. **Clones in Games** ● Program an endless runner game using clones. **Functions: About Me** ● Create and use a function with input to share information. **Functions: Paint by Numbers** ● Build a function that takes input to create art. **Flower Garden Functions Project** ● Write a function to draw multiple flowers. **Classes and Objects in Games** ● Create a game using classes, objects, and randomizers. **Digital Pet Project** ● Build a digital pet with events, conditionals, variables, and broadcasts. **Design an App**

	• Use design thinking to create an app that solves a real-world need.

Utah Interdisciplinary Computer Science Sixth 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**.	