Utah Interdisciplinary Computer Science Sixth Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

This Utah Interdisciplinary Sixth Grade Computer Science course introduces computer science concepts through hands-on coding experiences and interdisciplinary connections. Students develop computational thinking, problem-solving, creativity, and collaboration skills while exploring concepts across mathematics, science, social studies, and English language arts.

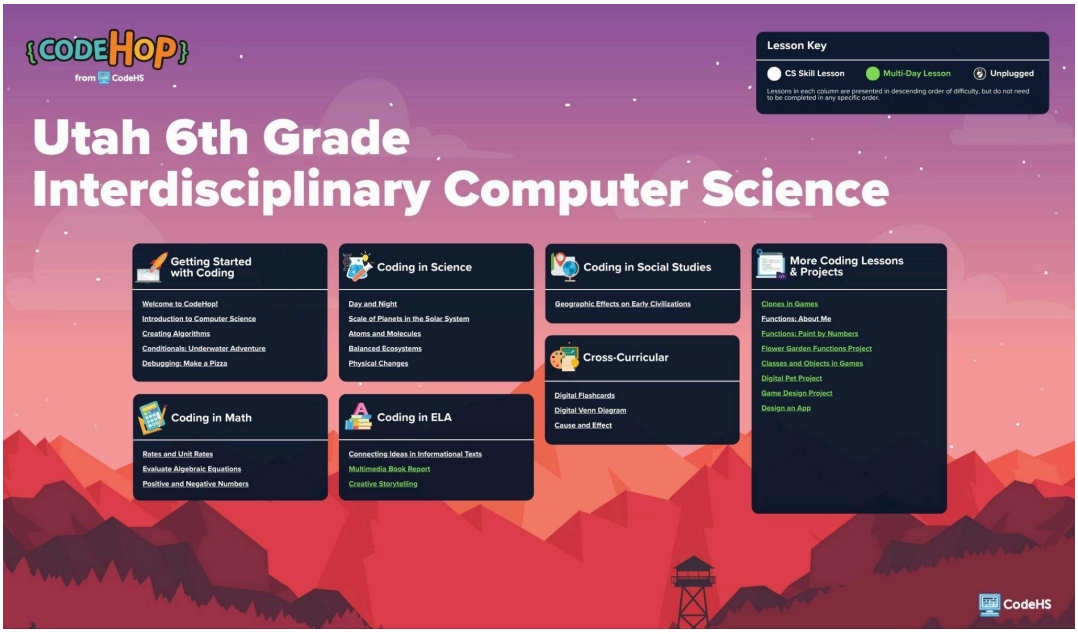
The course is organized into instructional strands, including Getting Started with Programming, Coding in Math, Coding in Science, Coding in English Language Arts, Coding in Social Studies, Cross-Curricular, and More Coding Projects. This organization supports flexible implementation and provides multiple opportunities to integrate computer science concepts throughout the school year.

Students create and explore programs using a block-based programming environment. Through guided and independent activities, students learn programming concepts while applying their learning in meaningful academic contexts.

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 computer science concepts and foundational programming skills through interactive coding activities. Lessons focus on algorithms, conditionals, debugging, and problem solving while building confidence in creating and improving programs.

Objectives / Topics Covered	• Algorithms • Conditionals • Debugging • Problem solv
Lessons	**Welcome to CodeHop! (15 minute lesson)** • Learn how to log in and use the CodeHop Playground. **Introduction to Computer Science** • 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 (3 lessons)

Students use coding to explore mathematical concepts including rates, algebraic reasoning, and integers. Through interactive programs, students apply computational thinking to model mathematical ideas and solve real-world problems.

Objectives / Topics Covered	• Ratios and rates • Algebraic reasoning • Positive and negative numbers • Problem solving • Variables and operators
Lessons	**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 (5 lessons)

Students use coding to investigate earth science, life science, and physical science concepts. Through simulations and models, students explore systems, patterns, and relationships in the natural world while applying computational thinking skills.

Objectives / Topics Covered	• Earth and space science • Life science • Physical science • Systems and interactions

Lessons	**Day and Night** ● Model Earth's rotation and day-night cycles. **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. **Physical Changes** ● Model changes in states of matter.

Coding in ELA (4 lesson)

Students use coding to strengthen literacy skills through reading, writing, communication, and creative expression. Through interactive projects, students analyze texts, communicate ideas, and create digital stories using computational thinking.

Objectives / Topics Covered	● Reading comprehension ● Informational texts ● Communication ● Creative expression ● Storytelling
Lessons	**Connecting Ideas in Informational Texts** - Show relationships between key ideas. **Multimedia Book Report** ● Create a program that conveys information about a book. **Creative Storytelling** ● Plan and animate an original story.

Coding in Social Studies (1 lesson)

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

Objectives / Topics Covered	● Early civilizations ● Cause and effect ● Human-environment interaction
Lessons	**Geographic Effects on Early Civilizations** ● Use broadcast messages to explain how geography shaped early civilizations.

Cross-Curricular (3 lessons)

Students apply coding skills across multiple subject areas through flexible projects that support comparison, analysis, communication, and creative expression. These lessons can be adapted to reinforce concepts from a variety of academic disciplines.

Objectives / Topics Covered	● Communication ● Comparison and analysis ● Cause and effect

Lessons	**Digital Flashcards** ● Create interactive flashcards for any subject. **Digital Venn Diagram** ● Compare ideas with an interactive Venn diagram. **Cause and Effect** ● Model cause-and-effect relationships with code.

More Coding Lessons & Projects (10 lessons)

Students expand their coding skills through creative projects, games, and design challenges. Lessons provide opportunities to apply programming concepts including functions, classes, objects, variables, conditionals, and game design while creating interactive programs and apps.

Objectives / Topics Covered	● Functions ● Classes and objects ● Variables and conditionals ● Game design ● App design
Lessons	**Clones in Games** ● Create a game using clones. **Functions: About Me** ● Create and use functions with input. **Functions: Paint by Numbers** ● Create functions that accept input. **Flower Garden Functions Project** ● Use functions to create drawings. **Classes and Objects in Games** ● Create a game using classes and objects. **Digital Pet Project** ● Build an interactive digital pet. **Game Design Project** ● Design and create a game. **Design an App** ● Design an app to solve a problem.

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 Templates	Empower your students to explore and express their knowledge creatively with our versatile 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**.	