Utah Interdisciplinary Computer Science Fourth Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

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

The course is organized into instructional strands, including Getting Started with Coding, Coding in Math, Coding in Science, Coding in English Language Arts, Coding in Social Studies, Cross-Curricular Connections, More Coding Lessons and Projects, Digital Literacy, and optional Artificial Intelligence (AI) lessons. This organization supports flexible implementation and provides multiple opportunities to integrate computer science concepts throughout the school year.

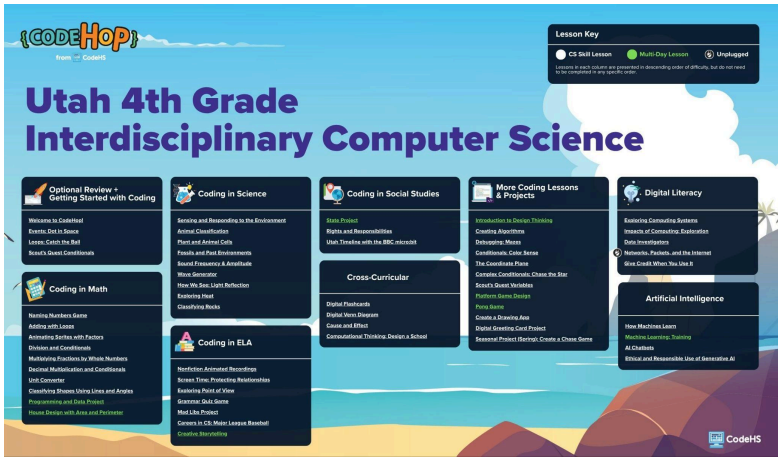
Students create and explore programs using a block-based programming environment. Through guided and independent activities, students learn programming concepts such as events, loops, conditionals, variables, functions, and data while applying their learning in meaningful academic contexts.

Each lesson includes the Utah Computer Science standard(s) it addresses. Some standards are intentionally reinforced across multiple lessons, while others are addressed within a single lesson. Standards are identified next to lesson titles throughout the syllabus to support standards alignment and instructional planning.

Programming Environment: Students will write and run programs that are saved in the CodeHop 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 (4 lessons)

Students are introduced to the coding platform and foundational programming concepts. Through interactive projects, students learn how events, loops, and conditionals control program behavior while developing computational thinking and problem-solving skills.

Objectives / Topics Covered	● Events ● Loops ● Conditionals
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Events: Dot in Space** (4.AP.2, 4.AP.3, 4.AP.4) ● Create a program using multiple event types. **Loops: Catch the Ball** (4.AP.2, 4.AP.4) ● Use two types of loops to make a simple game. **Scout's Quest: Conditionals** (4.AP.2) ● Create a program using if/then conditionals.

Coding in Math (12 lessons)

Students use coding to explore mathematical concepts including multiplication, division, fractions, decimals, measurement, geometry, and data. Through interactive programs and projects, students apply computational thinking to model mathematical ideas and solve problems.

Objectives / Topics Covered	● Operations and algebraic thinking ● Fractions and decimals ● Measurement and Geometry ● Data analysis
Lessons	**Naming Numbers Game** (4.AP.2, 4.AP.4) ● Create a game that identifies numbers. **Adding with Loops** (4.AP.2) ● Use loops to model addition. **Animating Sprites with Factors** (4.AP.2) ● Animate sprites using factors. **Division and Conditionals** (4.AP.2) ● Use conditionals to solve division problems. **Multiplying Fractions by Whole Numbers** (4.AP.2, 4.AP.3, 4.AP.4) ● Model fraction multiplication with code. **Decimal Multiplication and Conditionals** (4.AP.2) ● Use conditionals to solve decimal problems. **Unit Converter** (4.AP.2) ● Create an interactive unit converter. **Classifying Shapes Using Lines and Angles** (4.AP.4) ● Classify shapes by geometric properties. **Programming and Data Project** (2 classes | 4.DA.1) ● Analyze and present data with code. **House Design with Area and Perimeter** (2 classes | 4.AP.2) ● Design a house using area and perimeter.

Coding in Science (7 lessons)

Students use coding to investigate life science, earth science, and physical science concepts. Through simulations, models, and interactive programs, students explore systems, patterns, energy, and relationships in the natural world.

Objectives / Topics Covered	• Life science • Earth science • Physical science • Energy and waves • Scientific modeling
Lessons	**Sensing and Responding to the Environment** (4.AP.2) • Model how organisms respond to their environment. **Animal Classification** (4.AP.2) • Classify animals by their characteristics. **Plant and Animal Cells** (4.AP.2) • Explore the structures of cells. **Fossils and Past Environments** (4.AP.2) • Investigate fossils and ancient environments. **Sound Frequency & Amplitude** (4.AP.2, 4.AP.4) • Model sound waves and their properties. **Wave Generator** (4.AP.2) • Create and explore wave patterns. **How We See: Light Reflection** (4.AP.2) • Model how light reflects. **Exploring Heat** (4.AP.2) • Investigate heat transfer and energy. **Classifying Rocks** (4.AP.2, 4.AP.4) • Classify rocks by their properties.

Coding in ELA (6 lessons)

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

Objectives / Topics Covered	• Reading comprehension • Language and grammar • Point of view • Communication • Creative expression
Lessons	**Nonfiction Animated Recordings** (4.AP.2, 4.AP.4) • Create an animated nonfiction presentation. **Screen Time: Protecting Relationships** (4.AP.2, 4.IC.1) • Explore healthy technology habits. **Exploring Point of View** (4.AP.2) • Compare different points of view. **Grammar Quiz Game** (4.AP.2, 4.AP.3, 4.AP.4) • Create an interactive grammar quiz.

	Mad Libs Project (4.AP.2, 4.AP.3) ● Generate stories with random inputs. **Careers in CS: Major League Baseball** (4.AP.2, 4.AP.3) ● Explore technology in professional sports. **Creative Storytelling** (2 classes I 4.AP.2) ● Plan and animate an original story.

Coding in Social Studies (3 lessons)

Students use coding to explore government, citizenship, and state history. Through interactive projects, students represent historical events, civic responsibilities, and geographic concepts while developing computational thinking skills.

Objectives / Topics Covered	● Government and citizenship ● Rights and responsibilities ● State history
Lessons	**State Project** (4.AP.2) ● Create an interactive program with state facts. **Rights and Responsibilities** (4.AP.2, 4.AP.4) ● Simulate voting to demonstrate civic roles. **Utah Timeline with the BBC micro:bit®** (4.CS.1, 4.AP.2) ● Create a micro:bit-powered interactive timeline.

Cross-Curricular (4 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** (4.AP.2, 4.AP.4) ● Create interactive flashcards for any subject. **Digital Venn Diagram** (4.AP.2) ● Compare ideas with an interactive Venn diagram. **Cause and Effect** (4.AP.4) ● Model cause-and-effect relationships with code. **Computational Thinking: Design a School** (4.CT.1) ● Apply computational thinking to school design.

More Coding Lessons & Projects (15 lessons)

Students expand their coding skills through creative projects, games, and design challenges. Lessons provide opportunities to apply programming concepts including algorithms, debugging, conditionals, variables, and app design while creating interactive programs and games.

Objectives / Topics	● Algorithms ● Debugging

Covered	• Conditionals • Variables • App and game design
Lessons	**Introduction to Design Thinking** (2 classes | 4.AP.2, 4.AP.3, 4.AP.4, 4.AP.5, 4.IC.2) • Apply the design thinking process. **Creating Algorithms** (4.AP.1, 4.AP.2) • Create and test algorithms. **Debugging: Mazes** (4.AP.4) • Find and fix coding errors. **Conditionals: Color Sense** (4.AP.2) • Create programs using conditionals. **The Coordinate Plane** (4.AP.4) • Explore coordinates through coding. **Complex Conditionals: Chase the Star** (4.AP.2) • Use complex conditionals in a game. **Scout's Quest Variables** (4.AP.2, 4.AP.3, 4.AP.4) • Store and update data with variables. **Platform Game Design** (2 classes | 4.AP.2, 4.AP.3, 4.AP.4) • Design a platform-style game. **Pong Game** (2 classes | 4.AP.2, 4.AP.3, 4.AP.4) • Create an interactive Pong game. **Create a Drawing App** (4.CS.1, 4.AP.2, 4.AP.3) • Build a simple drawing application. **Digital Greeting Card Project** (4.AP.2, 4.AP.4) • Create an interactive greeting card. **Seasonal Project (Spring): Create a Chase Game** (4.AP.2) • Create a spring-themed chase game.

Digital Literacy (7 lessons)

Students develop technology skills and digital citizenship practices while learning how computing systems, networks, data, and online resources support communication and problem solving. Lessons emphasize responsible technology use, digital ownership, and understanding how information moves through technology systems.

Objectives / Topics Covered	• Computing systems • Impacts of technology • Data analysis • Networks and the Internet • Digital citizenship • Research and attribution
Lessons	**Exploring Computing Systems** (4.CS.1) • Explore how hardware and software work together. **Impacts of Computing: Exploration** (4.IC.1) • Examine how technology influences society. **Data Investigators** (4.DA.2) • Analyze data and communicate findings. **Networks, Packets, and the Internet** (2 classes | 4.NI.1)

	• Model how information travels online. **Give Credit When You Use It** (4.AP.3) • Practice citing and crediting sources.

Artificial Intelligence (3 lessons)
Students explore foundational artificial intelligence concepts through hands-on activities and discussions. Lessons examine how AI uses data, learns from examples, and assists people in solving problems and making decisions.

Objectives / Topics Covered	• Machine learning • Data and decision making • Emerging technologies • Responsible AI use
Lessons	**Introduction to Training AI** (3.DA.3) • Explore how AI learns from data. **Creating Smart Machines** (3.IC.1) • Examine how emerging technologies are designed. **Ask AI Better Questions** (3.IC.2) • Write effective prompts for AI tools.

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 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**.	