Utah Interdisciplinary Computer Science Fifth Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

This Utah Interdisciplinary Fifth 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.

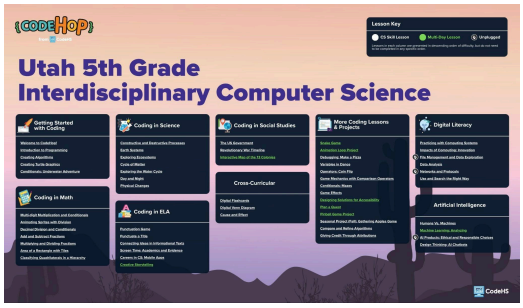
Each lesson includes the Utah Computer Science standard(s) it addresses. Some standards are intentionally reinforced across multiple lessons, while others are addressed within a single lesson. Standards are identified next to lesson titles throughout the syllabus to support standards alignment and instructional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

Assessments: Some units conclude with a brief five-question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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 are introduced to the coding platform and core computer science concepts through interactive programming activities. Lessons focus on algorithms, conditionals, loops, and problem solving while building a foundation for more advanced programming projects.

Objectives / Topics Covered	● Algorithms ● Loops ● Conditionals
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to Computer Science** (5.CS.1) ● Define key CS terms and build a simple Scratch program. **Creating Algorithms** (5.AP.1, 5.CT.1) ● Program multiple algorithms and assess their efficiency. **Creating Turtle Graphics** (5.AP.3) ● Use the pen tool and loops to create turtle graphics. **Conditionals: Underwater Exploration** (5.AP.4) ● Build an underwater scene using conditionals to trigger actions. **Getting Started with Coding Quiz (5 questions)**

Coding in Math (9 lessons)

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

Objectives / Topics Covered	● Operations and algebraic thinking ● Fractions ● Geometry ● Measurement ● Data analysis
Lessons	**Multi-digit Multiplication and Conditionals** (5.AP.4) ● Use conditionals to solve multiplication problems. **Animating Sprites with Division** (5.AP.3) ● Animate sprites using division. **Decimal Division and Conditionals** (5.AP.4) ● Create an interactive division game. **Add and Subtract Fractions** (5.AP.2, 5.AP.4) ● Create a fractions quiz game. **Multiplying and Dividing Fractions** (5.AP.2) ● Model fraction operations with code. **Area of a Rectangle with Tiles** (5.AP.3, 5.AP.5) ● Model fractional area using variables. **Classifying Quadrilaterals in a Hierarchy** (5.AP.2) ● Classify quadrilaterals by attributes. **Inquiry Project: Survey Bar Graph** (2 classes | 5.DA.1, 5.DA.2) ● Collect and visualize survey data.

Coding in Science (9 lessons)

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

Objectives / Topics Covered	• Earth systems • Ecosystems • Cycles in nature • Energy transfer • Properties of matter • Scientific modeling
Lessons	**Constructive and Destructive Processes** (5.AP.2) • Model changes to Earth's surface. **Earth Systems** (5.AP.2) • Show interactions within Earth systems. **Exploring Ecosystems** (5.AP.2, 5.AP.4) • Model energy flow in ecosystems. **Cycle of Matter** (5.AP.2) • Create an animated matter cycle model. **Exploring the Water Cycle** (5.AP.2) • Animate the stages of the water cycle. **Day and Night** (5.AP.4) • Model Earth's rotation using conditionals. **Physical Changes** (5.AP.4, 5.AP.5) • Model changes between states of matter.

Coding in ELA (7 lessons)

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

Objectives / Topics Covered	• Reading comprehension • Language and grammar • Informational text
Lessons	**Punctuation Game** (5.AP.4) • Create a game using punctuation rules. **Punctuate a Title** (5.AP.4) • Create a title punctuation challenge. **Connecting Ideas in Informational Texts** (5.AP.2) • Show relationships between key ideas. **Screen Time: Academics and Evidence** (5.IC.1) • Evaluate screen time using evidence. **Careers in CS: Mobile Apps** (5.AP.2, 5.AP.4) • Improve a game by adding new features. **Creative Storytelling** (2 classes | 5.AP.2, 5.IC.1) • Plan and animate an original story.

Coding in Social Studies (4 lessons)

Students use coding to explore government, history, and geography concepts. Through interactive projects, students model historical events, civic processes, and geographic information while developing computational thinking skills.

Objectives / Topics Covered	• Government and civics • US history • Geography • Timelines • Maps
Lessons	**The US Government** (5.AP.4) • Use conditionals to model how a bill becomes law. **Revolutionary War Timeline** (5.AP.2, 5.AP.4, 5.AP.5) • Build a timeline with inputs, events, and variables. **Interactive Map of the 13 Colonies** (5.AP.2, 5.AP.4, 5.AP.5, 5.CT.1) • Make a map with events, variables, and modular code design.

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** (5.CT.1) • Create interactive flashcards for any subject. **Digital Venn Diagram** (5.CT.1) • Compare ideas with an interactive Venn diagram. **Cause and Effect** (5.CT.1) • Model cause-and-effect relationships with code.

More Coding Lessons & Projects (19 lessons)

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

Objectives / Topics Covered	• Loop-based animations and interactive games • Debugging and program decomposition • Designing full games and apps with multiple programming concepts
Lessons	**Snake Game** (2 classes | 5.AP.5) - Create a snake game with variables. **Animation Loops Project** (2 classes | 5.AP.3) - Create a multi-scene animation. **Debugging: Make a Pizza** (5.AP.6) - Find and fix coding errors.

	Variables in Dance (5.AP.5) - Control animations using variables. **Operators: Coin Flip** (5.AP.5) - Create a coin flip simulation. **Game Mechanics with Comparison Operators** (5.AP.4) - Add game mechanics with operators. **Conditionals: Mazes** (5.AP.4) - Create a maze using conditionals. **Game Effects** (5.IC.2) - Improve a game using feedback. **Designing Solutions for Accessibility** (2 classes | 5.CT.1, 5.IC.1) - Redesign a game for accessibility. **Plan a Quest** (2 classes | 5.CT.1) - Plan and decompose a quest program. **Pinball Game Project** (3 classes | 5.AP.3, 5.AP.4, 5.AP.5) - Design and build a pinball game. **Seasonal Project (Fall): Gathering Apples Game** (5.AP.3, 5.AP.4, 5.AP.5) - Create a fall-themed collecting game. **Compare and Refine Algorithms** (5.AP.1, 5.CT.1) - Compare and improve algorithms. **Giving Credit Through Attributions** (5.IC.2) - Credit original creators appropriately.

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, data analysis, cybersecurity, and evaluating information.

Objectives / Topics Covered	● Computing systems ● Impacts of technology ● Data analysis ● Networks and communication ● Digital citizenship ● Research and attribution
Lessons	**Practicing with Computing Systems** (5.CS.1) ● Identify computing system components and problems. **Impacts of Computing: Innovation** (5.IC.1) ● Explore how technology influences society. **File Management and Data Exploration** (5.DA.1) ● Explore digital storage and file types. **Data Analysts** (5.DA.2) ● Analyze data and make predictions. **Network Connections** (5.NI.1) ● Model communication across networks. **Use and Search the Right Way** (5.IC.2) ● Research information and cite sources. **Digital Literacy Quiz (5 questions)**

Artificial Intelligence (6 lessons)

Students explore foundational artificial intelligence concepts through hands-on activities and real-world examples. Lessons examine how AI systems learn from data, make predictions, and impact society while encouraging responsible and ethical use of emerging technologies.

Objectives / Topics Covered	• Machine learning • Data and prediction • Emerging technologies • Ethics and responsibility
Lessons	**Humans Vs. Machines** (5.IC.1) • Compare human and computer problem solving. **Machine Learning: Analyzing** (2 classes | 5.DA.3) • Train and analyze an AI model. **AI Products: Ethical and Responsible Choices** (5.IC.1) • Evaluate the impacts of AI products. **Design Thinking: AI Chatbots** (2 classes | 5.CT.1, 5.DA.3) • Design and train an AI chatbot. **Artificial Intelligence Quiz (5 questions)**

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 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**.	