Utah Interdisciplinary Computer Science Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

This Utah Interdisciplinary Kindergarten 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, 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 developmentally appropriate block-based programming environment. Through guided and independent activities, students learn foundational computer science concepts while applying their learning in meaningful academic contexts.

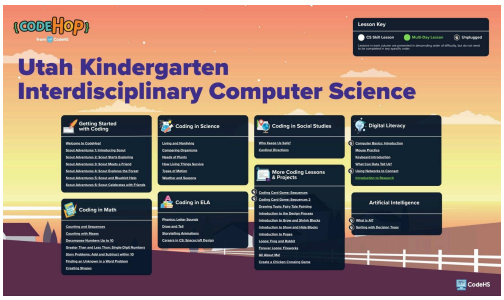
Each lesson includes the Utah Computer Science standard(s) it addresses. Some standards are intentionally reinforced across multiple lessons, while others are addressed within a single lesson. Standards are identified next to lesson titles throughout the syllabus to support standards alignment and instructional 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 two-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/21097/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, interface navigation, and core programming actions. Through a sequential story-driven unit, they learn to add, modify, and animate characters and backgrounds while building confidence in sequencing commands.

Objectives / Topics Covered	● Navigating the programming environment ● Adding, modifying, and deleting characters ● Using motion blocks to move sprites ● Creating sequences of commands ● Adding backgrounds, pages, and events
Lessons	**Welcome to CodeHop!** ● Learn how to log in and use CodeHop. **Scout Adventures 1: Introducing Scout** (K.CS.1) ● Explore the interface and add characters. **Scout Adventures 2: Scout Starts Exploring** (K.CS.1) ● Add backgrounds and a new page. **Scout Adventures 3: Scout Meets a Friend** (K.CS.1) ● Delete and modify characters. **Scout Adventures 4: Scout Explores the Forest** (K.CS.1) ● Use motion blocks to move characters. **Scout Adventures 5: Scout and Bluebird Help** (K.CS.1) ● Build sequences to collect objects. **Scout Adventures 6: Scout Celebrates with Friends** (K.CS.1) ● Create a celebration scene with events. **Getting Started with Coding Quiz (2 questions)**

Coding in Math (7 lessons)

Students use coding to explore foundational math concepts including counting, number relationships, shapes, and problem solving. Through interactive programs and unplugged activities, students represent mathematical thinking using sequences, events, and computational thinking skills.

Objectives / Topics Covered	● Counting and sequencing ● Number relationships ● Addition and subtraction ● Problem solving ● Shapes and patterns ● Computational thinking through coding
Lessons	**Counting and Sequences** (K.AP.1) ● Write a sequence to solve a maze. **Counting with Mazes** (K.AP.1) ● Design a maze using counting skills. **Decompose Numbers Up to 10** (K.CT.1) ● Create an interactive number decomposition program. **Greater Than and Less Than: Single-Digit Numbers** (K.DA.1) ● Compare numbers using events. **Story Problems: Add and Subtract within 10** (K.AP.1) ● Create a scene for a story problem. **Finding an Unknown in a Word Problem** (K.CT.1) ● Program a number story with missing values.

	Creating Shapes (K.AP.1) ● Program shapes to respond to user interaction.

Coding in Science (6 lessons)

Students use coding to explore life science, earth science, and physical science concepts. Through interactive programs and animations, students model scientific ideas, make observations, and represent real-world phenomena.

Objectives / Topics Covered	● Observation and classification ● Living things and environments ● Motion and weather ● Patterns and relationships ● Scientific modeling ● Computational thinking
Lessons	**Weather and Seasons** (K.DA.1) ● Animate weather changes across seasons. **Types of Motion** (K.AP.1) ● Program animations that show different types of motion. **Living and Nonliving** (K.DA.1) ● Classify living and nonliving things. **How Living Things Survive** (K.IC.1) ● Illustrate how living things survive. **Needs of Plants** (K.AP.2) ● Model the needs of plants. **Comparing Organisms** (K.DA.1) ● Group organisms by shared characteristics.

Coding in ELA (4 lessons)

Students use coding to strengthen literacy skills through storytelling, communication, and creative expression. Through interactive projects, students connect reading, writing, speaking, and listening to computational thinking.

Objectives / Topics Covered	● Storytelling and sequencing ● Communication and expression ● Phonics and language development ● Computational thinking
Lessons	**Phonics: Letter Sounds** (K.NI.1) ● Create an interactive phonics game. **Draw and Tell** (K.IC.1) ● Create a picture and tell about it. **Storytelling Animations** (K.AP.1) ● Retell a story in sequence. **Careers in CS: Spacecraft Design** (K.IC.1) ● Explore technology careers through coding.

Coding in Social Studies (2 lessons)

Students use coding to explore community roles and spatial thinking. Through interactive projects, students represent people, places, and directions while developing computational thinking skills.

Objectives / Topics Covered	• Community roles • Maps and directions • Spatial reasoning • Computational thinking
Lessons	**Who Keeps Us Safe?** (K.IC.1) • Animate an authority figure's role. **Cardinal Directions** (K.AP.1) • Create an interactive compass rose.

More Coding Lessons & Projects (13 lessons)

Students expand their coding skills through creative projects, games, and new programming concepts. Lessons provide opportunities to practice sequencing, loops, interactivity, and program design while creating original projects.

Objectives / Topics Covered	• Creative expression • Sequences and loops • Interactive projects • Game design • Computational thinking
Lessons	**Coding Card Game: Sequences** (K.AP.1) • Create sequences to solve challenges. **Coding Card Game: Sequences 2** (K.AP.1) • Practice building longer sequences. **Drawing Tools: Fairy Tale Painting** • Create a scene using drawing tools. **Introduction to the Design Process** (K.IC.1) • Use the design process to solve a problem. **Introduction to Grow and Shrink Blocks** (K.AP.1) • Change character sizes with code. **Introduction to Show and Hide Blocks** (2 classes | K.AP.1) • Make characters appear and disappear. **Introduction to Pages** (K.AP.1) • Create programs with multiple pages. **Loops: Frog and Rabbit** (K.AP.2) • Use loops to repeat code. **Forever Loops: Fireworks** (K.AP.2) • Create repeating animations. **All About Me!** (2 classes | K.IC.1) • Create a project about yourself. **Chicken Crossing Game** (2 classes | K.AP.2) • Build an interactive crossing game.

Digital Literacy (7 lessons)

Students develop foundational technology skills and learn how digital tools help people communicate, learn, and solve problems. Lessons introduce computers, data, networks, and research while building confidence using technology.

Objectives / Topics	• Computer systems • Digital tools

Covered	• Data and information • Networks • Research skills • Digital citizenship
Lessons	**Computer Basics: Introduction** (K.CS.2) • Identify parts and functions of a computer. **Mouse Practice** • Practice clicking and dragging. **Keyboard Introduction** • Use letters, numbers, and keyboard functions. **What Can Data Tell Us?** (K.DA.1) • Collect and analyze data. **Using Networks to Connect** (K.NI.1) • Explore how networks share information. **Introduction to Research** (2 classes | K.IC.1) • Find and share information from sources. **Digital Literacy Quiz (2 questions)**

Artificial Intelligence (2 lessons)

Students are introduced to foundational artificial intelligence concepts through age-appropriate activities and discussions. Lessons explore how AI uses data, follows rules, and helps people solve problems in everyday life.

Objectives / Topics Covered	• Data and decision making • Patterns and rules • Technology in daily life
Lessons	**What is AI?** • Identify examples of AI in everyday life. **Sorting with Decision Trees** (K.DA.1) • Use a decision tree to sort items. **Artificial Intelligence Quiz (2 questions)**

Utah Interdisciplinary Computer Science Kindergarten Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
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.
All of these resources and more are found on the **CodeHop Resources Page**.	