Utah Interdisciplinary Computer Science First Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

This Utah Interdisciplinary First 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, 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.

Assessments: Some units conclude with a brief three-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.

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/21098/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 foundational programming concepts. Through interactive activities and projects, students learn how programs work by creating scenes, using events, and building sequences.

Objectives / Topics Covered	● Programming basics ● Events and sequences ● Digital responsibility ● Program creation ● Computational thinking
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to Programming** (1.AP.1) ● Create a scene with characters. **Events** (1.AP.2) ● Use events to start actions. **Sequences: Digital Responsibilities** (1.AP.1) ● Create sequences about digital responsibility. **Getting Started With Coding Quiz (3 questions)**

Coding in Math (9 lessons)

Students use coding to explore mathematical concepts including number sense, operations, patterns, measurement, and geometry. Through interactive programs and problem-solving activities, students represent mathematical thinking using computational thinking skills.

Objectives / Topics Covered	● Number sense ● Operations and algebraic thinking ● Patterns and relationships ● Measurement and time ● Geometry ● Problem solving
Lessons	**Place Value: Ones and Tens** (1.DA.1) ● Model groups of ones, fives, and tens. **Grid: Solving Mazes** (1.AP.1) ● Program a character through a maze. **Place Value: Adding Up to 20** (1.DA.1) ● Decompose numbers into tens and ones. **Greater Than and Less Than: Two-Digit Numbers** (1.DA.1) ● Compare two-digit numbers. **Story Problems: Add and Subtract within 20** (1.AP.1) ● Create a scene for a story problem. **Algebraic Thinking: Find an Unknown Number Up to 10 in a Number Story** (1.CT.1) ● Solve for an unknown number. **Patterns and Music** (1.AP.3) ● Create musical patterns with code. **Telling Time** (1.AP.2) ● Create an analog and digital clock. **Combining Shapes** (1.AP.1) ● Create composite shapes in a scene.

	Divide Shapes into Equal Parts (1.AP.1) ● Partition shapes into equal parts.

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, investigate patterns, and represent real-world phenomena.

Objectives / Topics Covered	● Light and sound ● Earth and space patterns ● Life cycles and adaptations ● Cause-and-effect relationships ● Scientific modeling ● Computational thinking
Lessons	**Light and Shadows** (1.AP.2) ● Model sources and effects of light. **Sound and Pitch** (1.AP.2) ● Model sound and vibration. **Sun and Moon, Day and Night** (1.AP.3) ● Model patterns of day and night. **Phases of the Moon** (1.AP.2) ● Model the phases of the Moon. **Adaptations and Survival: Camouflage** (1.IC.1) ● Explain how camouflage helps survival. **Animal Life Cycles** (1.AP.2) ● Model the butterfly life cycle.

Coding in ELA (5 lessons)

Students use coding to strengthen literacy skills through reading, writing, storytelling, and communication. Through interactive projects, students connect language arts concepts with creativity and computational thinking.

Objectives / Topics Covered	● Phonics and word recognition ● Sentence construction ● Storytelling and communication ● Creative expression ● Problem solving ● Computational thinking
Lessons	**Phonics: Digraphs** (1.NI.1) ● Create an interactive digraphs game. **Build a Sentence** (1.AP.2) ● Create sentences with events and audio. **Original Story Animations - Personal Story** (2 classes | 1.IC.1) ● Animate an original personal story. **Careers in CS: Litter Free Communities** (1.IC.1) ● Explore how technology solves community problems.

Coding in Social Studies (3 lessons)

Students use coding to explore responsibility, maps, and economic choices. Through interactive projects, students represent community concepts while developing sequencing, events, and computational thinking skills.

Objectives / Topics Covered	● Responsibility and citizenship ● Maps and directions ● Wants and needs ● Decision making ● Computational thinking
Lessons	**Our Responsibilities** (1.IC.1) ● Explain responsibilities at school and home. **Create a Map** (1.AP.1) ● Program a character to follow a map. **Economic Choices** (1.AP.2) ● Show choices between wants and needs.

More Coding Lessons & Projects (12 lessons)

Students expand their coding skills through creative projects, games, and design challenges. Lessons provide opportunities to practice the design process, loops, events, debugging, variables, data, and interactive program design.

Objectives / Topics Covered	● Program design ● Sequences and loops ● Events and interactions ● Debugging ● Variables and data ● Game design
Lessons	**Practicing the Design Process** (1.CT.1) ● Use the design process to solve a problem. **Hide and Seek Game** (1.CT.1) ● Create an interactive hide-and-seek game. **Grow and Shrink Blocks in Motion** (1.AP.3) ● Change character size during movement. **Introduction to Repeat Loops** (1.AP.3, 1.CT.2) ● Use loops to repeat code. **Loops: Predator and Prey** (2 classes | 1.AP.3) ● Program animal interactions with loops. **Forever Loop Dance Party** (1.AP.3) ● Create repeating dance animations. **Message Events: Simon Says** (1.AP.3) ● Send messages between characters. **Debugging** (1.AP.2) ● Find and fix bugs in sequences. **Variables: Storing Data with Symbols** (1.AP.1) ● Model data storage with variables. **Basic Data and Programming** (1.DA.1, 1.DA.2) ● Present collected data with code.

	Seasonal Project (Summer): Pop the Balloons (1.AP.3) ● Build a balloon popping game.

Digital Literacy (7 lessons)

Students develop foundational technology skills and learn how digital tools help people communicate, learn, and stay safe online. Lessons introduce computer systems, digital citizenship, data, and research while building confidence using technology.

Objectives / Topics Covered	● Computer systems ● Digital citizenship ● Online safety ● Data and information ● Research skills ● Responsible technology use
Lessons	**Computer Basics: Exploration** (1.CS.2) ● Explore parts and functions of a computer. **Keeping Information Safe** (1.NI.2) ● Identify private and personal information. **Responsible Digital Citizens** (1.IC.1) ● Practice safe and responsible online behavior. **Data Storage and Files Practice** (1.DA.1) ● Explore how computers store information. **Guided Research** (2 classes | 1.IC.2) ● Find and share information from sources. **Digital Literacy Quiz (3 questions)**

Artificial Intelligence (3 lessons)

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

Objectives / Topics Covered	● Patterns and classification ● Data and decision making ● Technology and society ● Problem solving
Lessons	**How AI Helps Us** ● Explore how AI helps people solve problems. **Sorting with Charts** (1.DA.1) ● Use charts to classify objects. **Machine Learning: What is a Blorg?** (1.DA.2) ● Explore how AI learns from examples. **Artificial Intelligence Quiz (3 questions)**

Utah Interdisciplinary Computer Science First Grade 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**.	