Utah Interdisciplinary Computer Science Second Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

This Utah Interdisciplinary Second 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.

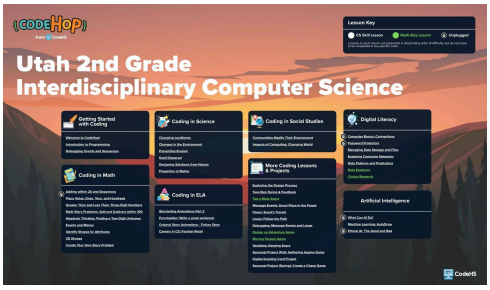
Each lesson includes the Utah Computer Science standard(s) it addresses. Some standards are intentionally reinforced across multiple lessons, while others are addressed within a single lesson. Standards are identified next to lesson titles throughout the syllabus to support standards alignment and instructional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

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 four-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/21099/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, students learn how programs work by creating projects, using events, and debugging code.

Objectives / Topics Covered	● Programming basics ● Events and sequences ● Debugging ● Program creation
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Introduction to Programming** (2.AP.1) ● Create a scene with characters. **Debugging: Events and Sequences** (2.AP.4) ● Find and fix coding errors. **Getting Started with Coding Quiz (4 questions)**

Coding in Math (9 lessons)

Students use coding to explore mathematical concepts including place value, operations, geometry, money, and problem solving. Through interactive programs and computational thinking activities, students model mathematical ideas and communicate their reasoning.

Objectives / Topics Covered	● Place value ● Operations and algebraic thinking ● Money ● Geometry ● Problem solving
Lessons	**Adding within 20 and Sequences (Unplugged)** (2.AP.1) ● Write sequences to solve mazes. **Place Value: Ones, Tens, and Hundreds** (2.DA.1) ● Model place value with events. **Greater Than and Less Than: Three-Digit Numbers** (2.DA.1) ● Compare three-digit numbers. **Story Problems: Add and Subtract within 100** (2.AP.1) ● Model a math story problem. **Algebraic Thinking: Finding a Two-Digit Unknown** (2.CT.1) ● Solve for an unknown number. **Events and Money** (2.AP.2) ● Calculate the value of coins. **Identify Shapes by Attributes** (2.AP.1) ● Identify and draw shapes by attributes. **3D Shapes** (2.AP.1) ● Create a quiz about 3D shapes. **Create Your Own Story Problem** (2.AP.2) ● Design and solve a story problem.

Coding in Science (6 lessons)

Students use coding to investigate earth science, life science, and physical science concepts. Through simulations and interactive programs, students model natural processes, analyze patterns, and explore how living things and environments change over time.

Objectives / Topics Covered	• Earth's surface • Environmental change • Plant and animal adaptations • Life cycles • Properties of matter • Scientific modeling
Lessons	**Changing Landforms** (2.AP.1) • Model how landforms change over time. **Changes in the Environment** (2.AP.2) • Animate environmental changes and causes. **Preventing Erosion** (2.CT.1) • Compare solutions that prevent erosion. **Seed Dispersal** (2.AP.2) • Model how seeds are dispersed. **Designing Solutions from Nature** (2.IC.1) • Show how nature inspires solutions. **Properties of Matter** (2.AP.5) • Sort matter based on its properties.

Coding in ELA (5 lessons)

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

Objectives / Topics Covered	• Storytelling • Writing and punctuation • Communication • Creative expression
Lessons	**Storytelling Animations Part 2** (2.AP.1) • Retell a story in sequence. **Punctuation: Write a Great Sentence!** (2.AP.3) • Write sentences with correct punctuation. **Original Story Animations - Fiction Story** (2 classes | 2.IC.1) • Animate an original fiction story. **Careers in CS: Fashion-Retail** (2.IC.1) • Explore coding in fashion design.

Coding in Social Studies (2 lessons)

Students use coding to explore how people interact with their communities and how technology impacts daily life. Through interactive projects, students represent real-world ideas while developing computational thinking skills.

Objectives / Topics Covered	• Communities • Human-environment interaction • Technology and society • Cause and effect
Lessons	**Communities Modify Their Environment** (2.IC.1) • Show how communities change their environment. **Impacts of Technology in Our World** (2.IC.1) • Explain how technology changes daily life.

More Coding Lessons & Projects (16 lessons)

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

Objectives / Topics Covered	• Events and messages • Loops and repetition • Debugging • Variables and data • Game development
Lessons	**Exploring the Design Process** (2.CT.1) • Use the design process to improve a program. **Two-Step Dance & Feedback** (2.IC.2) • Revise a program using peer feedback. **Tap-a-Mole Game** (2.AP.2) • Create an interactive game with events. **Message Events: Scout Plays in the Forest** (2.AP.2) • Use messages to control program actions. **Pages: Scout's Travels** (2.AP.2) • Navigate between pages with messages. **Loops: Follow the Path** (2.AP.3) • Create patterns using loops. **Debugging: Message Events and Loops** (2.AP.4) • Find and fix coding errors. **Design an Adventure Game** (2 classes | 2.AP.1, 2.AP.2) • Create a multi-page adventure game. **Moving Targets Game** (3 classes | 2.AP.2) • Build a moving target game. **Variables: Keeping Score** (2.AP.5) • Track score using a variable. **Seasonal Project (Fall): Gathering Apples Game** (2.AP.3) • Create a fall-themed collecting game. **Digital Greeting Card** (2.AP.2) • Create an interactive greeting card. **Seasonal Project (Spring): Create a Chase Game** (2.AP.3) • Create a spring-themed chase game.

Digital Literacy (11 lessons)

Students develop technology skills and digital citizenship practices while learning how computers, networks, data, and research tools support communication and problem solving. Lessons build confidence using technology safely, responsibly, and effectively.

Objectives / Topics Covered	• Computer systems • Digital citizenship • Online safety • Networks and communication • Data and information • Research skills
Lessons	**Computer Basics: Connections** (2.CS.2) • Explore how computer parts work together. **Password Protectors** (2.NI.2) • Create strong and secure passwords. **Managing Data Storage and Files** (2.DA.1) • Explore how computers store information. **Exploring Computer Networks** (2.NI.1) • Model how networks share information. **Data Patterns and Predictions** (2.DA.2) • Identify patterns and make predictions from data. **Data Explorers** (2 classes | 2.DA.1, 2.DA.2) • Collect and present survey data. **Choice Research** (2 classes | 2.IC.2) • Research a topic and share findings. **Digital Literacy Quiz (4 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, recognizes patterns, and supports decision making in everyday life.

Objectives / Topics Covered	• Patterns and classification • Data and decision making • Technology and society • Problem solving
Lessons	**What Can AI Do?** • Explore how AI uses data to make decisions. **Machine Learning: AutoDraw** (2.DA.2) • Explore how AI recognizes patterns in drawings. **Artificial Intelligence Quiz (4 questions)**

Utah Interdisciplinary Computer Science Second 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.
These resources and more are found on the **CodeHop Resources Page**.	