Utah Interdisciplinary Computer Science Third Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

This Utah Interdisciplinary Third 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.

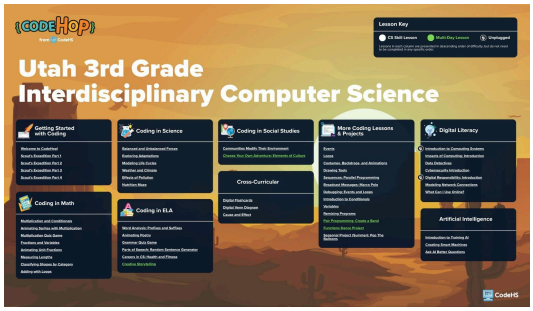
Each lesson includes the Utah Computer Science standard(s) it addresses. Some standards are intentionally reinforced across multiple lessons, while others are addressed within a single lesson. Standards are identified next to lesson titles throughout the syllabus to support standards alignment and instructional 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/21100/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 a sequential story-driven unit, students learn to create programs using events, loops, motion, and animation while developing computational thinking skills.

Objectives / Topics Covered	● Programming basics ● Events and loops ● Animation
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Scout's Expedition Part 1** (3.AP.3, 3.CT.1) ● Use basic Scratch commands to move and talk. **Scout's Expedition Part 2** (3.AP.3, 3.CT.1) ● Add sprites and animate a story sequence. **Scout's Expedition Part 3** (3.AP.3, 3.CT.1) ● Animate a Scout story using loops, events, and looks. **Scout's Expedition Part 4** (3.AP.3, 3.CT.1) ● Continue the Scout story with motion, events, and looks.. **Getting Started with Coding Quiz (5 questions)**

Coding in Math (8 lessons)

Students use coding to explore mathematical concepts including multiplication, fractions, measurement, geometry, and problem solving. Through interactive programs and games, students model mathematical thinking using variables, conditionals, loops, and computational thinking skills.

Objectives / Topics Covered	● Multiplication ● Fractions ● Measurement ● Geometry
Lessons	**Multiplication and Conditionals** (3.AP.4) ● Use conditionals to review multiplication. **Animating Sprites with Multiplication** (3.AP.3) ● Animate sprites using multiplication. **Multiplication Quiz Game** (2 classes | 3.AP.4, 3.AP.5) ● Create a multiplication quiz game. **Fractions and Variables** (3.AP.5) ● Represent fractions using variables. **Animating Unit Fractions** (3.AP.3) ● Animate fractions on a number line. **Measuring Lengths** (3.AP.2) ● Create an interactive measurement game. **Classifying Shapes by Category** (3.AP.2) ● Classify quadrilaterals by their properties. **Adding with Loops** (3.AP.3) ● Use loops to add multi-digit numbers.

Coding in Science (6 lessons)

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

Objectives / Topics Covered	● Forces and motion ● Adaptations and survival ● Life cycles ● Weather and climate ● Environmental impact ● Scientific modeling
Lessons	**Balanced and Unbalanced Forces** (3.AP.4, 3.AP.5) ● Model balanced and unbalanced forces. **Exploring Adaptations** (3.AP.2) ● Show how adaptations support survival. **Modeling Life Cycles** (3.AP.2) ● Model the stages of a frog's life cycle. **Weather and Climate** (3.DA.2) ● Predict weather using climate data. **Effects of Pollution** (3.AP.4) ● Explain how actions affect pollution. **Nutrition Maze** (3.AP.4) ● Create an interactive nutrition maze.

Coding in ELA (6 lessons)

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

Objectives / Topics Covered	● Word analysis ● Grammar and language ● Poetry ● Storytelling ● Creative expression
Lessons	**Word Analysis: Prefixes and Suffixes** (3.AP.2) ● Identify and explain word meanings. **Animating Poetry** (3.AP.2, 3.IC.1) ● Create an animated poetry reading. **Grammar Quiz Game** (3.AP.4) ● Create a grammar review game. **Parts of Speech: Random Sentence Generator** (3.AP.5) ● Generate random sentences with code. **Careers in CS: Health and Fitness** (3.IC.1) ● Explore technology in health and fitness. **Creative Storytelling** (2 classes | 3.AP.2, 3.IC.1) ● Plan and animate an original story.

Coding in Social Studies (2 lessons)

Students use coding to explore communities, culture, and human interaction with the environment. Through interactive projects and games, students represent social studies concepts while developing computational thinking skills.

Objectives / Topics Covered	• Communities • Culture • Human-environment interaction • Decision making
Lessons	**Communities Adapt to & Modify Their Environment** (3.AP.2) • Animate community adaptation or modification. **Choose Your Own Path: Elements of Culture** (2 classes | 3.IC.1) • Build a cultural choose-your-own-path game.

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** (3.AP.2) • Create interactive flashcards for any subject. **Digital Venn Diagram** (3.AP.2) • Compare ideas with an interactive Venn diagram. **Cause and Effect** (3.AP.4) • Model cause-and-effect relationships with code.

More Coding Lessons & Projects (15 lessons)

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

Objectives / Topics Covered	• Events and interactions • Loops and repetition • Variables and functions • Debugging • Game development
Lessons	**Events** (3.AP.2) • Create a program using events. **Loops** (3.AP.3) • Use loops to repeat instructions. **Costumes, Backdrops, and Animations** (3.AP.2) • Create animated sprites and backdrops.

	Drawing Tools (3.AP.2) ● Design custom sprites and backdrops. **Sequences: Parallel Programming** (3.AP.3) ● Create programs with parallel sequences. **Broadcast Messages: Marco Polo** (3.AP.2) ● Trigger actions using messages. **Debugging: Events and Loops** (3.AP.6) ● Find and fix coding errors. **Introduction to Conditionals** (3.AP.4) ● Create programs using conditionals. **Variables** (3.AP.5) ● Store and change data with variables. **Remixing Programs** (3.IC.2) ● Remix programs and give credit. **Pair Programming: Create a Band** (2 classes | 3.IC.3) ● Collaborate to create a digital band. **Functions Dance Project** (2 classes | 3.AP.7) ● Create and use custom functions. **Seasonal Project (Summer): Pop The Balloons** (3.AP.4, 3.AP.5) ● Build a balloon popping 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, cybersecurity, and evaluating information.

Objectives / Topics Covered	● Computing systems ● Impacts of technology ● Data analysis ● Cybersecurity ● Networks ● Research and attribution
Lessons	**Introduction to Computing Systems** (3.CS.1) ● Identify computing system components. **Impacts of Computing: Introduction** (3.IC.1) ● Explore how technology influences society. **Data Detectives** (3.DA.2) ● Analyze data and draw conclusions. **Cybersecurity Introduction** (3.NI.2) ● Identify online threats and protections. **Digital Responsibility: Getting Started** (3.IC.2) ● Practice responsible technology use. **Modeling Network Connections** (3.NI.1) ● Model how data travels across networks. **What Can I Use Online?** (3.IC.2) ● Research information and cite sources. **Digital Literacy Quiz (5 questions)**

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. **Artificial Intelligence Quiz (5 questions)**

Utah Interdisciplinary Computer Science Third 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**.	