Utah Interdisciplinary Computer Science Third Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Interdisciplinary Computer Science Third Grade Course introduces students to foundational programming and computational thinking concepts through Scratch, a block-based programming language. Students explore computer science skills while engaging in lessons that integrate coding into math, science, English language arts (ELA), and social studies. This course emphasizes creativity, collaboration, and real-world connections, providing students with a strong foundation in both academic content and computer science.

Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: Review, Guided Practice, Independent Practice, Extension, and Reflection. The course begins with a Getting Started with Coding unit to establish familiarity with basic programming and Scratch. After that, subject-area lessons can be taught in any order, with topics listed in approximate order of complexity to support flexible planning and pacing.

The course includes more than 36 lessons, allowing teachers to select the subject-area lessons that best align with their instructional goals and schedule. Lessons are built with spiral review to reinforce key concepts and culminate in engaging projects to showcase student understanding. The "I do, we do, you do" instructional model ensures a gradual release of responsibility, building confidence and independence as students learn to program. Optional coding extension projects and digital literacy lessons are included to provide comprehensive coverage of the Utah Computer Science Standards.

Programming Environment: Students will write and run programs in Scratch embedded and saved in the CodeHS 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/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 CodeHop Playground and begin the sequential Scout's Scratch Expedition series. They will learn to navigate Scratch, build simple sequences, and animate stories.

Objectives / Topics Covered	● Navigating CodeHop and Scratch ● Using motion, looks, events, and loops in sequences ● Creating animated stories based on a guided narrative
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. **Scout's Scratch Expedition Part 1** ● Use basic Scratch commands to move and talk. **Scout's Scratch Expedition Part 2** ● Add sprites and animate a story sequence. **Scout's Scratch Expedition Part 3** ● Animate a Scout story using loops, events, and looks. **Scout's Scratch Expedition Part 4** ● Continue the Scout story with motion, events, and looks..

Coding in Math (7 lessons)

Students integrate math concepts with Scratch programming to create interactive projects that practice classification, measurement, operations, fractions, and variables.

Objectives / Topics Covered	● Classifying and measuring shapes ● Modeling addition, multiplication, and fractions ● Using loops, conditionals, and variables in math-based animations
Lessons	**Classifying Shapes by Category** ● Classify quadrilaterals by properties. **Measuring Lengths** ● Create an interactive ruler and plot measurements. **Adding with Loops** ● Use loops to add multi-digit numbers. **Animating Sprites with Multiplication** ● Animate sprite movement using multiplication and wait blocks. **Multiplication and Conditionals** ● Review multiplication with if/then blocks. **Animating Unit Fractions** ● Show repeated addition of unit fractions. **Fractions and Variables** ● Represent fractions on a number line with variables and conditionals.

Coding in Science (6 lessons)

Students program interactive models to explore life, earth, and physical science topics, applying events, loops, and conditionals.

Objectives / Topics Covered	● Showing animal adaptations and climate patterns ● Modeling life cycles, nutrition, and classification ● Simulating forces and interpreting fossil evidence

Lessons	**Exploring Adaptations** ● Show how animals adapt to survive. **Weather and Climate** ● Use data to model typical weather. **Modeling Life Cycles** ● Create a frog life cycle animation. **Nutrition Maze** ● Build a game to encourage healthy food choices. **Classifying Rocks** ● Sort rocks with if/then statements. **Fossils and Past Environments** ● Show how fossils reveal past environments. **Balanced and Unbalanced Forces** ● Model how forces affect movement.

Coding in ELA (6 lessons)

Students create interactive programs that build literacy skills through sentence construction, storytelling, and literary analysis.

Objectives / Topics Covered	● Generating random sentences ● Animating poetry readings ● Story planning and animation ● Building games to practice punctuation
Lessons	**Parts of Speech: Random Sentence Generator** ● Create random sentences with nouns, verbs, adjectives, and adverbs. **Animating Poetry** ● Animate and narrate a poem with visual and literary meaning. **Creative Storytelling (2 part lesson)** ● Plan and animate an original story. **Nonfiction Animated Recordings** ● Create an animated reading of nonfiction text. **Punctuation Game** ● Build a game to add punctuation using if/then/else blocks.

Coding in Social Studies (2 lessons)

Students use programming to represent how cultures adapt, interact, and share information.

Objectives / Topics Covered	● Showing environmental adaptation and modification ● Exploring cultural elements through interactive games ● Using digital flashcards for review
Lessons	**Communities Adapt to & Modify Their Environment** ● Animate community adaptation or modification. **Choose Your Own Path: Elements of Culture** ● Build a cultural choose-your-own-path game. **Digital Flashcards** ● Use broadcast messages to make study tools.

More Coding Lessons & Projects (17 lessons)

Students practice advanced Scratch features including custom graphics, parallel programming, loops, debugging, and game design.

Objectives / Topics Covered	● Creating animations and custom art ● Using events, broadcast messages, and loops effectively ● Debugging and remixing projects ● Designing platform and seasonal games ● Using variables and functions
Lessons	**Costumes, Backdrops, and Animations** ● Animate sprites with interactive backdrops. **Scratch Drawing Tools** ● Customize sprites and backgrounds. **Sequences: Parallel Programming** ● Create multiple sequences running at once. **Events** ● Trigger actions using events. **Broadcast Messages: Marco Polo** ● Communicate between sprites with broadcast. **Loops** ● Repeat instructions with loops. **Remixing Programs** ● Remix an existing project and give credit. **Debugging: Events and Loops** ● Find and fix errors in a program. **Introduction to Conditionals** ● Use if/then logic in Scratch. **Variables** ● Create and update variables in a program. **Platform Game Design (2 part lesson)** ● Build a platform game with movement, loops, and game logic. **Pair Programming: Create a Band (2 part lesson)** ● Collaborate to code a band using keyboard inputs. **Functions Dance Project (2 part lesson)** ● Make functions to choreograph a dance sequence. **Seasonal Project (Summer): Pop the Balloons** ● Make a summer-themed game with loops, conditionals, and variables.

Digital Literacy (7 lessons)

Students learn computer fundamentals, digital citizenship, network concepts, and how to give proper credit online.

Objectives / Topics Covered	● Identifying computer components and solving problems ● Creating strong credentials and safe online habits ● Understanding digital identity and responsible online use ● Modeling how networks work ● Researching and attributing online sources
Lessons	**Introduction to Computing Systems** ● Learn parts of a computer and solve simple issues. **Strong Usernames and Passwords** ● Create and protect secure login information. **Digital Identity** ● Connect real-world and online identity and promote a positive footprint. **What Can I Use Online?** ● Search and give proper attribution for online sources. **Impacts of Computing: Introduction** ● Explore how computing has changed communities and evolved over time.

	Technology Timeline ● Build an interactive timeline showing changes in music player technology. **Modeling Network Connections** ● Show how messages move between devices in a network.

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.