



Utah Interdisciplinary Computer Science Sixth Grade Course Syllabus

Partial Year for Elementary School

Course Overview and Goals

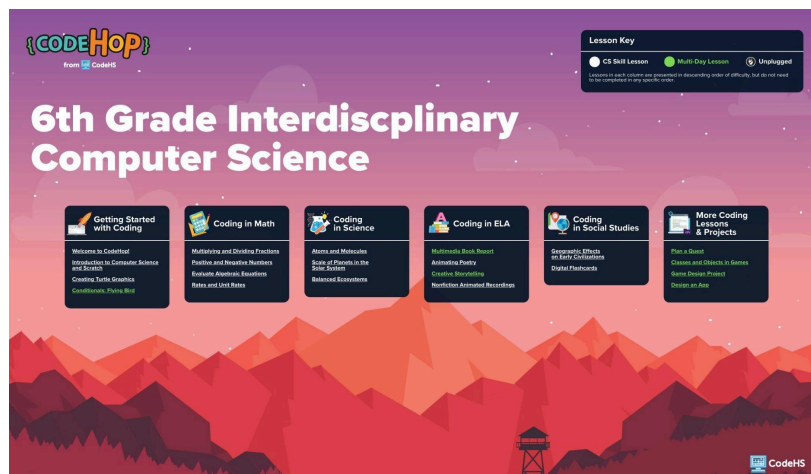
The Interdisciplinary Computer Science Sixth Grade Course introduces students to foundational programming concepts through Scratch, a block-based programming language designed for learners new to coding. This course serves as an introductory and transitional library of lessons, providing flexible opportunities for students to build coding skills while applying them across math, science, English language arts (ELA), and social studies. Through these cross-curricular projects, students reinforce problem-solving and critical thinking abilities in each subject area. As students gain confidence, they are encouraged to continue their learning in the **Mix and Match Middle School Computer Science Course**, which offers more advanced programming topics and project options.

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 Programming unit to establish basic programming skills, followed by subject-area coding lessons that can be taught in any order. Topics are arranged in approximate order of complexity to support flexible pacing. Teachers can choose activities that align with their instructional goals, while spiral review helps reinforce key concepts. The “**I do, we do, you do**” instructional model ensures a gradual release of responsibility, building confidence and independence as students learn to program.

Programming Environment: Students will write and run programs in **Scratch** within the CodeHS platform. This environment supports hands-on programming, enabling students to design, test, and debug interactive projects in a user-friendly space.

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/21114/overview>



A clickable PDF can be found at <https://codehs.com/InterCSRoadmaps>

Course Breakdown

Getting Started with Programming

In this introductory unit, students become familiar with the CodeHop platform and the basics of programming in Scratch. They will explore essential vocabulary, learn how to navigate the interface, and begin creating interactive programs using simple sequences, loops, and conditionals.

Objectives / Topics Covered	<ul style="list-style-type: none">• Logging in and using the CodeHop Playground• Introduction to Scratch interface and vocabulary• Creating graphics with pen tools• Using conditionals in games
Lessons	Welcome to CodeHop! (15 minute lesson) <ul style="list-style-type: none">• Learn how to log in and use the CodeHop Playground. Introduction to Computer Science and Scratch <ul style="list-style-type: none">• Define CS vocabulary and create a simple program. Creating Turtle Graphics <ul style="list-style-type: none">• Students use the pen tool in Scratch to create looping turtle graphics. Conditionals: Flying Bird <ul style="list-style-type: none">• Students apply different types of conditionals to program a Flying Bird game.

Coding in Math

Students connect mathematical concepts to programming by building projects that model problem-solving strategies, calculations, and relationships between numbers.

Objectives / Topics Covered	<ul style="list-style-type: none">• Multiplying and dividing fractions• Positive and negative numbers in context• Writing, checking, and solving algebraic equations• Calculating rates and unit rates
Lessons	Multiplying and Dividing Fractions <ul style="list-style-type: none">• Show the relationship between multiplying and dividing fractions in word problems. Rates and Unit Rates <ul style="list-style-type: none">• Build a converter using variables and operators. Evaluate Algebraic Equations <ul style="list-style-type: none">• Use operators and conditionals to write and solve real-world equations. Positive and Negative Numbers <ul style="list-style-type: none">• Represent quantities with positive and negative numbers in an interactive program.

Coding in Science

This unit uses programming to model scientific concepts, from atomic structures to planetary scales and ecosystems.

Objectives / Topics Covered	<ul style="list-style-type: none">• Calculating planetary scale and properties• Modeling atomic and molecular structures• Simulating balanced ecosystems and population changes
Lessons	Scale of Planets in the Solar System <ul style="list-style-type: none">• Calculate and model planetary scale and properties. Atoms and Molecules <ul style="list-style-type: none">• Use conditionals and broadcasts to model atom combinations.

	Balanced Ecosystems <ul style="list-style-type: none"> • Simulate ecosystems and demonstrate how changes affect populations.
--	--

Coding in ELA

Students combine reading comprehension and programming by creating interactive multimedia book reports.

Objectives / Topics Covered	<ul style="list-style-type: none"> • Multimedia presentations of literature • Analyzing and animating poetry • Creative storytelling and nonfiction readings
Lessons	Multimedia Book Report <ul style="list-style-type: none"> • Create a program that conveys information about a book. Animating Poetry <ul style="list-style-type: none"> • Animate a reading of a poem, exploring both literal and metaphorical meanings. Creative Storytelling <ul style="list-style-type: none"> • Plan and animate an original story using events and sequences. Nonfiction Animated Recordings <ul style="list-style-type: none"> • Create a clear, animated reading of nonfiction text.

Coding in Social Studies

Students use programming to explore how geography influences the development of civilizations.

Objectives / Topics Covered	<ul style="list-style-type: none"> • Understanding the impact of geography on civilizations • Creating and using digital flashcards for study
Lessons	Geographic Effects on Early Civilizations <ul style="list-style-type: none"> • Use broadcast messages to explain how geography shaped early civilizations. Digital Flashcards <ul style="list-style-type: none"> • Build a set of interactive flashcards using broadcast messages.

More Coding Lessons & Projects

In this final unit, students apply skills from throughout the course in creative, open-ended projects, including game design, object-oriented programming, and app creation.

Objectives / Topics Covered	<ul style="list-style-type: none"> • Project planning and decomposition • Classes and objects in programming • Game design and mechanics • App design using the design thinking process
Lessons	Plan a Quest <ul style="list-style-type: none"> • Plan and break down the steps for creating a quest program. Classes and Objects in Games <ul style="list-style-type: none"> • Create a game using classes, objects, and randomizers. Game Design Project <ul style="list-style-type: none"> • Design and build a game using multiple programming skills such as loops, conditionals, and variables. Design an App <ul style="list-style-type: none"> • Use design thinking to create an app that solves a real-world need.

Utah Interdisciplinary Computer Science Sixth 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 Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch 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 .	