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 and computational thinking through a block-based programming environment. Students apply coding concepts across mathematics, science, English language arts (ELA), and social studies while developing problem-solving and critical thinking skills.

Learning Environment: This teacher-led course includes ready-to-use lesson plans that follow a consistent instructional format: Review, Guided Practice, Independent Practice, Extension, and Reflection. The course begins with the Computational Thinking module, followed by the Getting Started with Coding unit. After completing these introductory units, teachers may teach lessons from the mathematics, science, English language arts (ELA), social studies, Cross-Curricular, and More Coding sections in the order that best supports their instructional goals. The Cross-Curricular section includes coding lessons that can be integrated into any subject area. Lessons within each section progress in approximate order of complexity, and spiral review opportunities help reinforce previously learned concepts.

Programming Environment: Students write and run programs within the CodeHop platform using a block-based programming environment.

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/21114/overview



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

Course Breakdown

Getting Started with Coding

In this introductory unit, students become familiar with the CodeHop platform and the basics of programming. 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	● Navigating the CodeHop platform ● Introduction to CS vocabulary ● Developing and evaluating algorithms ● Using conditionals ● Debugging programs
Lessons	**Welcome to CodeHop!** ● Learn how to log in and use the CodeHop Playground. **Introduction to Computer Science** ● Define important computer science vocabulary and create a simple program. **Creating Algorithms** ● Program multiple algorithms and assess which one best meets needs. **Conditionals: Underwater Exploration** ● Create a program that uses conditionals. **Debugging: Make a Pizza** ● Decompose a program to debug and make the program run as intended. **Getting Started with Programming Quiz (5 questions)**

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	● Using variables and operators ● Creating unit rate converters ● Programming algebraic equations ● Solving equations ● Using positive and negative numbers in context
Lessons	**Rates and Unit Rates** ● Create a unit rate converter using variables and operators. **Evaluate Algebraic Equations** ● Create a program that uses operators and conditionals to check and solve real-world algebraic equations. **Positive and Negative Numbers** ● Use positive and negative numbers to represent quantities in a real-world context.

Coding in Science

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

Objectives / Topics Covered	● Using comparison operators ● Modeling Earth's rotation ● Calculating planetary scale and properties ● Modeling atomic structures and molecules

	• Simulating ecosystems and population effects • Adjusting variables for state of matter changes
Lessons	**Day and Night** • Use comparison operators and conditionals to show how the Earth's rotation causes day and night. **Scale of Planets in the Solar System** • Create a program to calculate and model the scale and properties of planets in the Solar System. **Atoms and Molecules** • Use conditionals and broadcast events to create a model of atom and molecule combinations. **Balanced Ecosystems** • Simulate an ecosystem and demonstrate how changing components affects populations. **Physical Changes** • Write a program that uses variables to adjust the temperature of a sprite to change between solid, liquid, and gas.

Coding in ELA

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

Objectives / Topics Covered	• Creating informational text relationships • Multimedia book reports • Creative storytelling with events and sequences • Improving game functionality
Lessons	**Connecting Ideas in Informational Texts** • Demonstrate the relationships between key ideas or events from an informational text. **Multimedia Book Report** • Create a program that conveys information about a book. **Creative Storytelling** • Plan and animate a story using events and sequences. **Careers in CS: Mobile Apps** • Analyze and improve a game by modifying code and adding new functionality.

Coding in Social Studies

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

Objectives / Topics Covered	• Understanding the impact of geography on civilizations • Creating and using digital flashcards for study
Lessons	**Geographic Effects on Early Civilizations** • Explain why physical geography affected the development of early civilizations using broadcast messages.

Cross-Curricular

Objectives / Topics Covered	● Creating interactive study tools ● Comparing data with Venn diagrams ● Using events and conditionals for cause-and-effect relationships
Lessons	**Digital Flashcards** ● Build a set of interactive flashcards using broadcast messages. **Digital Venn Diagram** ● Program an interactive Venn diagram to compare two things. **Cause and Effect** ● Create a program that demonstrates cause-and-effect relationships using events and conditionals.

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	● Using clones in games ● Creating and using functions with inputs ● Classes and objects ● Developing digital pets ● Comprehensive game design ● Design thinking for app creation
Lessons	**Clones in Games** ● Use clones to program an endless runner game. **Functions: About Me** ● Create and use a function with input in a program. **Functions: Paint by Numbers** ● Create and use a function that takes input in a program. **Flower Garden Functions Project** ● Create a function in a program to draw multiple flowers. **Classes and Objects in Games** ● Create an interactive game using classes, objects, and randomizers. **Digital Pet Project** ● Create a digital pet using events, conditionals, variables, and broadcasts. **Game Design Project** ● Design and create a game using multiple programming skills such as loops and variables. **Design an App** ● Use the design thinking process to design an app that solves a user's 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 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**.	