Utah Computer Science
Kindergarten Course Syllabus
One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Computer Science Kindergarten Course introduces students to foundational programming concepts through block-based programming language. Students will develop computational thinking and problem-solving skills while learning to create interactive projects, animations, and games. This course emphasizes creativity and collaboration, providing students with a solid base in computer science concepts and digital literacy.

Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts and build confidence over time.

The course includes 36 lessons, each approximately 30 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. All lessons are labeled with the specific standards they address to support planning and compliance.

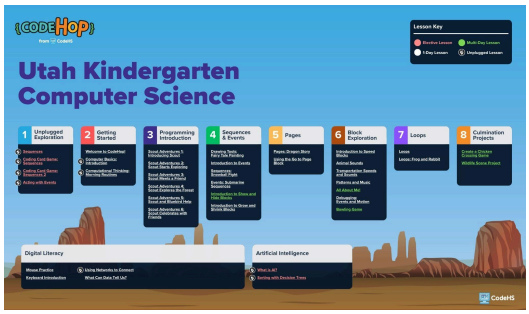
Standards Alignment Note: Standards addressed in each lesson are listed in parentheses after the lesson title. Standards may be reinforced across multiple lessons, and full coverage of state computer science standards is achieved through the course as a whole.

Programming Environment: Students will write and run programs that are saved in students' accounts. 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 two-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/25600/overview



A clickable PDF can be found at https://codehs.com/UT-CSRoadmaps

Course Breakdown

Optional Unplugged Exploration

In this unplugged unit, students explore the basics of coding without using a device. Through hands-on activities, movement, and teamwork, they learn how to create step-by-step instructions (sequences) and discover how events can trigger actions—laying the foundation for future programming with fun, active experiences.

Objectives / Topics Covered	● Understand and create simple sequences of instructions. ● Learn how an event can cause an action to happen. ● Practice following and giving directions through movement and games. ● Collaborate with classmates to solve challenges using logic and sequencing.
Lessons	**Sequences** (K.AP.1, K.CT.1) ● Create a step-by-step sequence of dance moves to understand how order matters in instructions. **Coding Card Game: Sequences** (K.AP.1) ● Work in teams to guide Scout through a maze using sequence coding cards. **Coding Card Game: Sequences 2** (K.AP.1) ● Tackle a new maze using sequence cards to build and follow a plan together. **Acting with Events** (K.AP.1) ● Act out how specific events (like clapping or jumping) can trigger a character or person to take action.

Unit 1: Getting Started (2 lessons)

In this introductory unit, students will begin developing basic computer skills and explore foundational concepts in computer science. Through hands-on practice and familiar routines, they'll learn how computers work, how to use a mouse and keyboard, and how to think like a computer scientist by recognizing patterns and sequencing steps.

Objectives / Topics Covered	● Learn how to log in and navigate the Playground. ● Understand what a computer is and how its parts work together. ● Practice using a mouse and keyboard confidently. ● Apply computational thinking to everyday routines by identifying patterns and sequencing steps.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Introductory lesson to help students log in and explore the Playground; perfect as a warm-up or standalone activity. **Computer Basics: Introduction** (K.CS.1) ● Learn what a computer is, how it's used, and how input, output, hardware, and software work together. **Computational Thinking: Morning Routines** (K.AP.1) ● Break down a morning routine into steps and recognize patterns using computational thinking strategies. **Getting Started Quiz (2 questions)**

Unit 2: Programming Introduction (6 lessons)

In this story-driven unit, students follow Scout the Squirrel on a series of adventures while learning how to use the programming environment. Students explore characters, backgrounds, pages, motion, sequences, and events as they build foundational programming skills through storytelling and creativity.

Objectives / Topics Covered	● Navigate the programming environment. ● Add, delete, and modify characters, backgrounds, and pages. ● Use motion blocks to create sequences. ● Use events to trigger actions. ● Build confidence through storytelling and creative projects.
Lessons	**Scout Adventures 1: Introducing Scout** (K.CS.1) ● Explore the coding interface and add characters to begin Scout's journey. **Scout Adventures 2: Scout Starts Exploring** (K.CS.1) ● Add backgrounds and a new page as Scout begins to explore the world around them. **Scout Adventures 3: Scout Meets a Friend** (K.CS.1) ● Delete and modify characters as Scout meets a new friend during the adventure. **Scout Adventures 4: Scout Explores the Forest** (K.CS.1) ● Use motion blocks to move characters around the stage as Scout explores the forest. **Scout Adventures 5: Scout and Bluebird Help** (K.CS.1) ● Build a sequence of motion blocks to help Scout and Bluebird collect items. **Scout Adventures 6: Scout Celebrates with Friends** (K.CS.1) ● Create a celebration scene by combining pages, characters, backgrounds, and motion blocks with events. **Programming Introduction Quiz (2 questions)**

Unit 3: Sequences & Events (7 lessons)

In this unit, students build foundational programming skills by exploring how to create sequences of actions and use events to trigger those actions. They'll experiment with fun tools like show/hide and grow/shrink blocks while creating interactive animations inspired by fairy tales, snowball fights, and underwater adventures. They'll also be introduced to data by collecting and organizing real-world information.

Objectives / Topics Covered	● Understand and create sequences of actions using motion and event blocks. ● Use events to trigger animations and interactions in a program. ● Practice visual creativity using drawing and design tools. ● Learn how to make characters appear, disappear, grow, and shrink within a sequence. ● Collect and organize simple data and use it in a program.
Lessons	**Drawing Tools: Fairy Tale Painting** (K.AP.1) ● Use the painting tools to create a custom fairy-tale scene with characters and a background. **Introduction to Events** (K.AP.1) ● Create a simple program that uses different types of events to control how and when characters move. **Sequences: Snowball Fight** (K.AP.1) ● Build a snowball fight animation using multiple sequences to make characters move and interact. **Events: Submarine Sequences** (K.AP.1, K.CT.1) ● Use event and motion blocks to program a character's movement in an animated underwater scene. **Introduction to Show and Hide Blocks** (2 classes | K.AP.1) ● Add show and hide blocks to make characters appear and disappear within a program. **Introduction to Grow and Shrink Blocks** (K.AP.1) ● Create an animated scene where characters change size using grow and shrink blocks. **Sequences & Events Quiz (2 questions)**

Unit 4: Pages (2 lessons)

In this unit, students learn how to create multi-page projects. By telling stories and building activities that span across different scenes, students use the “go to page” block to control navigation and enhance the storytelling experience.

Objectives / Topics Covered	● Create multi-page programs with different scenes. ● Use the “go to page” block to switch between pages within a project. ● Develop storytelling skills through sequencing and scene transitions.
Lessons	**Pages: Dragon Story** (K.AP.1) ● Create a multi-page story featuring a dragon, with different actions and scenes on each page. **Using the Go To Page Block** (K.AP.1, K.CT.1) ● Build a program that switches between pages using the “go to page” block for interactive navigation. **Pages Quiz (2 questions)**

Unit 5: Block Exploration (9 lessons)

In this unit, students explore a variety of blocks to expand their programming skills. They'll use speed, sound, and say blocks, create musical patterns, tell stories about themselves, and build fun, interactive projects like games—while also learning how to fix problems in their code.

Objectives / Topics Covered	● Use speed, sound, and say blocks to enhance animations and interactivity. ● Create sequences and loops to build patterns and music. ● Practice debugging to identify and fix errors in code. ● Apply coding skills creatively through games and personal storytelling.
Lessons	**Introduction to Speed Blocks** (K.AP.1, K.CT.1) ● Animate characters at different speeds using speed blocks. **Animal Sounds** (K.AP.1) ● Use say or sound blocks to make animals speak or produce sounds. **Transportation Speeds and Sounds** (K.AP.1, K.CT.1) ● Combine sound and speed blocks to animate different types of transportation. **Patterns and Music** (K.NI.1) ● Use loops and sequences to create repeating patterns in a musical program. **All About Me!** (2 classes | K.AP.1, K.CT.1) ● Design a program that shares favorite things and personal interests. **Debugging: Events and Motion** (K.CT.1) ● Find and fix bugs in a program that uses event and motion blocks. **Bowling Game** (2 classes | K.AP.1) ● Create an interactive bowling game using blocks learned throughout the course. **Block Exploration Quiz (2 questions)**

Unit 6: Loops (2 lessons)

In this unit, students learn how to use loops to repeat actions in their programs. They’ll discover how loops help simplify code and make animations more efficient while creating fun, repeatable patterns with characters like a frog and a rabbit.

Objectives / Topics Covered	● Understand what loops are and how they are used to repeat actions. ● Create programs that use loops to simplify sequences. ● Practice recognizing patterns that can be repeated in code.

Lessons	**Loops** (K.AP.1, K.CT.1) ● Create a program using loops and explain how they repeat parts of the code. **Loops: Frog and Rabbit** (K.AP.1, K.CT.1) ● Use loops to animate a frog and rabbit repeating actions in a fun, interactive program. **Loops Quiz (2 questions)**

Unit 7: Culmination Projects (4 lessons)

In this unit, students apply the coding skills they've learned throughout the course to create original projects. By combining sequences, events, and loops, they will design interactive animations and games that reflect their creativity and growing mastery of programming.

Objectives / Topics Covered	● Apply sequences, events, and loops in original projects. ● Design interactive programs that include movement, timing, and user input. ● Adjust difficulty and interactivity through coding decisions. ● Express creativity through animation and storytelling.
Lessons	**Chicken Crossing Game** (2 classes | K.AP.1) ● Create an interactive game with a player character and adjust the difficulty using speed and timing. **Wildlife Scene Project** (2 classes | K.AP.1, K.CT.1) ● Design a wildlife animation using events, sequences, and loops to bring the scene to life.

Unit 8: Digital Literacy (4 lessons)

In this unit students learn how people and computers connect and share information using networks, both with and without wires.

Objectives / Topics Covered	● Describe how networks help people and devices share information. ● Understand the difference between wired and wireless connections.
Lesson	**Mouse Practice** (K.CS.1) ● Practice mouse skills like clicking and dragging through fun and interactive games. **Keyboard Introduction** (K.CS.1) ● Explore the keyboard by identifying and using letters, numbers, and simple function keys. **Using Networks to Connect** (K.NI.1) ● Learn how people and devices share information through wired and wireless networks, and explore examples of technology that helps us stay connected. **What Can Data Tell Us?** (K.DA.1) ● Collect, organize, and analyze data about how students get to school, then represent that information in a program. **Digital Literacy Quiz (2 questions)**

Optional: Artificial Intelligence (2 lessons)

In this unit, students are introduced to the basics of artificial intelligence by exploring how smart machines use data to make decisions. Through familiar real-world examples and hands-on sorting activities, students begin to understand how AI differs from regular tools and how simple rules help AI organize information.

Objectives / Topics Covered	• Identify examples of artificial intelligence in everyday life. • Describe how AI is different from a regular tool. • Understand that AI uses data and rules to make decisions. • Create and use a simple decision tree to sort items.
Lesson	**What is AI?** • Identify examples of AI in everyday life and explain one way a smart machine is different from a regular tool. **Sorting with Decision Trees** • Create a simple decision tree to sort items and explore how AI uses data and rules to make decisions. **Artificial Intelligence Quiz (2 questions)**

Utah Computer Science Kindergarten Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
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.
All of these resources and more are found on the **CodeHop Resources Page**.	