Texas Computer Science Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Texas Computer Science Kindergarten Course introduces students to foundational programming concepts through a 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.

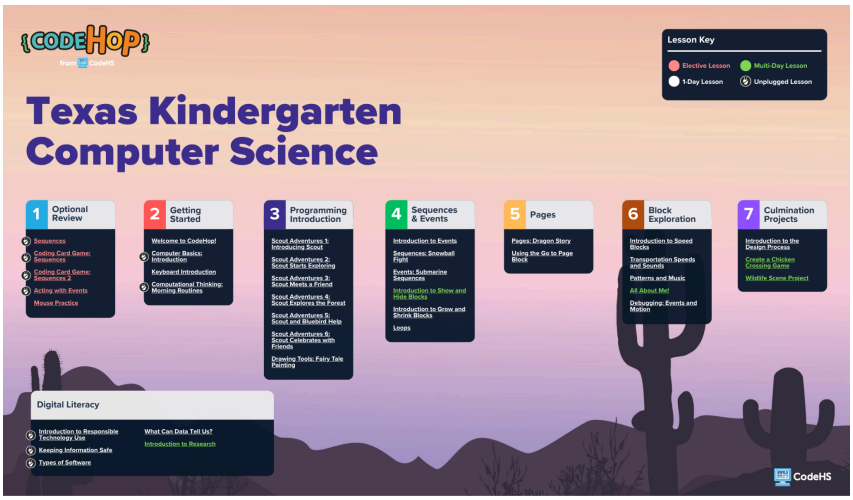
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts 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. Required lessons are labeled with the specific standards they address to support planning and compliance.

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.

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/26368/overview



A clickable PDF can be found at https://codehs.com/TX-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** ● Create a step-by-step sequence of dance moves to understand how order matters in instructions. **Coding Card Game: Sequences** ● Work in teams to guide Scout through a maze using sequence coding cards. **Coding Card Game: Sequences 2** ● Tackle a new maze using sequence cards to build and follow a plan together. **Acting with Events** ● Act out how specific events (like clapping or jumping) can trigger a character or person to take action. **Mouse Practice** ● Practice mouse skills like clicking and dragging through fun and interactive games.

Unit 1: Getting Started (3 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.8.B)** ● Learn what a computer is, how it's used, and how input, output, hardware, and software work together. **Keyboard Introduction (K.8.E)** ● Explore the keyboard by identifying and using letters, numbers, and simple function keys. **Computational Thinking: Morning Routines (K.1.A, K.1.C, K.3.A)** ● Break down a morning routine into steps and recognize patterns using computational thinking strategies.

Unit 2: Programming Introduction (7 lessons)

In this story-driven unit, students will follow Scout the Squirrel on an adventure through the programming world. As they explore the interface, add and modify elements, and animate characters, students will develop foundational coding skills like sequencing, motion, and events—building their confidence through storytelling and creativity.

Objectives / Topics Covered	● Learn to navigate the programming interface and tools. ● Add, delete, and modify characters and backgrounds. ● Use motion blocks and create sequences of actions. ● Introduce event blocks to trigger movement and interactions. ● Build confidence in coding through storytelling and visual programming.
Lessons	**Scout Adventures 1: Introducing Scout** ● Explore the interface and add characters to begin Scout's journey. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a new page as Scout begins to explore the world around them. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters as Scout meets a new friend during the adventure. **Scout Adventures 4: Scout Explores the Forest** ● Use motion blocks to move characters around the stage as Scout explores the forest. **Scout Adventures 5: Scout and Bluebird Help** ● Build a sequence of motion blocks to help Scout and Bluebird collect items along their path. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene by combining pages, characters, backgrounds, and motion blocks with events. **Drawing Tools: Fairy Tale Painting** ● Use the painting tools to create a custom fairy-tale scene with characters and a background.

Unit 3: Sequences, Events & Loops (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 snowball fights and underwater adventures.

Objectives / Topics Covered	● Understand and create sequences of actions using motion and event blocks. ● Use events to trigger animations and interactions in a program. ● 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	**Introduction to Events** ● Create a simple program that uses different types of events to control how and when characters move. **Sequences: Snowball Fight (K.2)** ● Build a snowball fight animation using multiple sequences to make characters move and interact. **Events: Submarine Sequences (K.2)** ● Use event and motion blocks to program a character's movement in an animated underwater scene. **Introduction to Show and Hide Blocks (2 classes)** ● Add show and hide blocks to make characters appear and disappear within a program. **Introduction to Grow and Shrink Blocks** ● Create an animated scene where characters change size using grow and shrink blocks. **Loops** ● Use loops and explain how loops are used to repeat code.

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** ● Create a multi-page story featuring a dragon, with different actions and scenes on each page. **Using the Go To Page Block** ● Build a program that switches between pages using the “go to page” block for interactive navigation.

Unit 5: Block Exploration (6 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 personal storytelling.
Lessons	**Introduction to Speed Blocks** ● Animate characters at different speeds using speed blocks. **Transportation Speeds and Sounds** ● Combine sound and speed blocks to animate different types of transportation. **Patterns and Music (K.1.B)** ● Use loops and sequences to create repeating patterns in a musical program. **All About Me! (2 classes)** ● Design a program that shares favorite things and personal interests. **Debugging: Events and Motion** ● Find and fix bugs in a program that uses event and motion blocks.

Unit 6: Culmination Projects (5 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	**Introduction to the Design Process (K.3.B)** ● Identify and participate in the steps of the design process with adult guidance to solve a simple, authentic problem through animation. **Chicken Crossing Game (2 classes)**

	• Create an interactive game with a player character and adjust the difficulty using speed and timing. **Wildlife Scene Project (2 classes)** • Design a wildlife animation using events, sequences, and loops to bring the scene to life.

Unit 8: Digital Literacy (6 lessons)

In this unit, students will learn how to use technology safely, responsibly, and effectively. They'll explore topics like protecting personal information, choosing the right software, understanding basic data, and conducting simple online research.

Objectives / Topics Covered	• Use technology responsibly and follow an Acceptable Use Policy (AUP). • Identify and protect private and personal information. • Select appropriate software tools for different computing tasks. • Collect, organize, and analyze simple data. • Research a topic using digital sources and share their findings visually.
Lesson	**Introduction to Responsible Technology Use (K.5, K.6.A)** • Learn how to use technology responsibly, follow classroom tech rules, and understand what an Acceptable Use Policy means. **Keeping Information Safe (K.7.A, K.7.B)** • Identify personal and private information and discuss strategies to stay safe when using technology. **Types of Software (K.8.A, K.8.C)** • Explore different software tools (like word processors, presentation software, and spreadsheets) and decide which to use for specific tasks. **What Can Data Tell Us? (K.4.A)** • Collect and organize class data about school transportation, then analyze it using charts or graphs to discover patterns. **Introduction to Research (K.4.B, K.6.B)** • Use safe research tools to find information on a topic and create a simple visual program to present what was learned.

Texas 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 **Elementary Resources Page**.	