



Tennessee Kindergarten Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Tennessee Kindergarten Computer Science** 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, cross-curricular integration, and digital literacy.

Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans structured as **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are delivered in an “I do, we do, you do” format, supporting gradual release of responsibility and fostering confidence as students learn.

Unique to Tennessee: The pathway is structured as an approximate 36-hour sequence that balances computer science skill development, interdisciplinary integration, and digital literacy. Core CS modules spiral across grade levels, while interdisciplinary lessons are organized in subject-area menus (Math, Science, ELA, and Social Studies) that are meant to be woven in based on local instructional needs. Digital literacy lessons are embedded within the sequence, ensuring students also build essential technology and digital citizenship skills alongside programming.

The full course includes 36 lessons, each approximately 45 minutes, offering a complete school year if taught once per week.

Programming Environment: Students will write and run programs in students’ accounts. 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/26969/overview>.



A clickable PDF can be found at <https://codehs.com/TN-CSRoadmaps>

Course Breakdown

Unit 1: Optional Unplugged Exploration (2 weeks)

In this optional unit, students develop foundational programming and problem-solving skills through unplugged activities. They will learn to sequence actions and understand how events can trigger responses in a system.

Objectives / Topics Covered	• Create step-by-step sequences of instructions. • Understand and model how events trigger actions.
Lessons	Sequences (Unplugged) • Create a sequence of step-by-step instructions for a dance. Acting with Events • Act out how an event can trigger an action.

Unit 2: Getting Started (5 weeks)

Students are introduced to the basics of computing and the CodeHop platform. They will build essential skills for navigating hardware and software, using input devices like the mouse and keyboard, and understanding responsible technology use. Students also begin applying computational thinking strategies to everyday tasks.

Objectives / Topics Covered	• Identify basic components and functions of a computer system. • Practice using input devices. • Understand and model responsible technology use. • Apply computational thinking to sequence real-world tasks.
Lessons	Welcome to CodeHop! (15 minute lesson) • Log in and navigate the CodeHop Playground. Computer Basics: Introduction (K.FC.1) • Identify what a computer is, how we use it, and what to do when it doesn't work. Mouse Practice • Practice mouse skills by dragging and clicking in interactive games. Keyboard Introduction (K.FC.1) • Use letters, numbers, and basic keyboard functions effectively. Introduction to Responsible Technology Use • Recognize safe and responsible ways to use technology, including following an Acceptable Use Policy. Computational Thinking: Morning Routines (K.AT.1) • Use computational thinking to recognize patterns, sequence steps, and break down tasks in a morning routine.

Unit 3: Programming Exploration (4 weeks)

Students begin using the CodeHopJr programming environment through a sequence of lessons featuring the character Scout. They explore the interface, add and modify characters and backgrounds, and use motion blocks to animate scenes.

Objectives / Topics Covered	• Navigate the CodeHopJr interface. • Add and modify characters, backgrounds, and pages. • Use motion blocks to animate characters. • Build simple sequences of code to tell a story.
Lessons	Scout Adventures 1: Introducing Scout (K.AT.1) • Explore the CodeHopJr interface and add characters to the stage.

	Scout Adventures 2: Scout Starts Exploring (K.AT.1) - Add backgrounds and create a new page. Scout Adventures 3: Scout Meets a Friend (K.AT.1) - Delete and modify characters in a project. Scout Adventures 4: Scout Explores the Forest (K.AT.1) - Use motion blocks to move characters around the stage.
--	--

Unit 4: Sequences & Events (3 weeks)

Students learn to create programs that use ordered steps and event blocks. They build skills in sequencing actions and triggering responses to events in interactive animations.

Objectives / Topics Covered	- Create programs using sequential code. - Use events to trigger actions in a program. - Animate characters using motion and event blocks. - Build an understanding of how order and timing affect outcomes in a program.
Lessons	Introduction to Events (K.AT.1) - Create a program using different types of event blocks. Sequences: Snowball Fight (K.AT.1) - Build an animated snowball fight using sequences. Events: Submarine Sequences (K.AT.1) - Program an underwater animation using motion and event blocks.

Unit 5: Block Exploration (4 weeks)

In this unit, students explore CodeHopJr blocks to build fluency with core features like page navigation, sound, speed, and basic debugging. They combine these blocks to create animated stories and simple interactive scenes.

Objectives / Topics Covered	- Use page navigation blocks to build multi-page programs. - Program movement and sound effects using speed and sound blocks. - Debug event and motion sequences in a program. - Use loops to repeat code in a program.
Lessons	Using the Go To Page Block (K.AT.1) - Create a program using the “go to page” block to move between pages. Transportation Speeds and Sounds (K.AT.1) - Program different vehicles with sound and speed blocks. Debugging: Events and Motion (K.AT.1) - Identify and fix bugs in sequences involving motion and events. Loops: Frog and Rabbit (K.AT.1) - Use loops to program repetitive movement in an animated scene.

Unit 6: Culmination Projects (3 weeks)

In this unit, students apply their foundational coding skills to complete creative projects using the design process. They solve problems and tell stories using events, loops, and sequences.

Objectives / Topics Covered	- Follow the design process to solve a problem through animation. - Combine sequences, loops, and events in a final project.
Lessons	Introduction to the Design Process (K.FC.2) - Identify and participate in steps of the design process to solve a simple problem. Wildlife Scene Project (2 classes K.AT.1) - Create a wildlife animation using events, sequences, and loops.

Unit 7: Digital Literacy (5 weeks)

In this unit, students explore how to stay safe and kind online, protect personal information, and use data and research to answer questions and share ideas through programs.

Objectives / Topics Covered	• Collect and analyze data to answer questions. • Communicate information through research-based programs. • Recognize private and personal information. • Identify and respond to unkind online behavior.
Lessons	Introduction to Research (2 classes K.FC.2, K.DA.1) • Find information using research sources and create a program to share findings visually. What Can Data Tell Us? (K.FC.2, K.DA.1) • Collect and analyze data about school transportation using a visual program. Keeping Information Safe (K.NI.1) • Identify private and personal information that should be protected online. Being Kind Online (K.AT.1) • Recognize kind and unkind messages to help identify and respond to cyberbullying.

Unit 8: Interdisciplinary Computer Science (13 weeks)

This unit provides interdisciplinary lessons that reinforce core concepts across math, science, ELA, music, and social studies. These are for flexible use throughout the year; within each subject area, lessons are listed in order of increasing coding complexity.

Objectives / Topics Covered	• Represent math concepts such as operations, comparisons, and patterns with code. • Retell and build sentences through storytelling and phonics activities. • Model science and social studies ideas through interactive animations. • Practice sequencing, loops, events, and message-based programming.
Lessons	Decompose Numbers Up to 10 (K.AT.1) • Create an interactive program using events to visualize decomposing numbers. Story Problems: Add and Subtract within 10 (K.AT.1) • Animate a scene that models an addition or subtraction story problem. Greater Than and Less Than: Single-Digit Numbers (K.AT.1) • Use events to compare numbers and display greater than, less than, or equal to. Patterns and Music (K.AT.1) • Create musical patterns using sequences and loops. Living and Nonliving (K.AT.1) • Classify items and use tap events to animate living or nonliving actions. Weather and Seasons (K.AT.1) • Program an animation showing weather changes across seasons. Needs of Plants (K.AT.1) • Model what plants need using events, messages, and character actions. Phonics: Letter Sounds (K.AT.1) • Create a phonics game using “on tap” and “record audio” blocks. Build a Sentence (K.AT.1) • Use events to build and read sentences in an interactive program. Storytelling Animations (K.AT.1) • Retell a story in sequence through an animated program. All About Me! (2 classes K.AT.1) • Create a program that shares information about your favorites. Who Keeps Us Safe? (K.AT.1) • Animate a scene that explains the role of an authority figure.

K-1 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.
These resources and more are found on the CodeHop Resources Page .	