Tennessee 1st Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Tennessee 1st Grade 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/26970/overview.

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

Course Breakdown

Unit 1: Optional Review (5 weeks)

This optional unit gives students a refresher on foundational coding and navigation skills using CodeHopJr and unplugged activities.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Reinforce sequencing through unplugged activities. ● Practice CodeHopJr skills like adding characters, pages, and movement. ● Strengthen foundational block-based programming concepts.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Log in and navigate the CodeHop Playground. **Coding Card Game: Sequences** ● Create a sequence of instructions to move Scout through a maze. **Scout Adventures 1: Introducing Scout** ● Add characters to the stage. **Scout Adventures 2: Scout Starts Exploring** ● Add a background and create a new page. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters in a program. **Scout Adventures 4: Scout Explores the Forest** ● Program characters to move using motion blocks.

Unit 2: Getting Started (4 weeks)

This unit introduces students to foundational computing concepts including hardware, software, responsible technology use, and basic digital citizenship. Students also explore computational thinking through real-life routines.

Objectives / Topics Covered	● Identify basic computer parts and their functions. ● Understand the difference between hardware, software, input, and output. ● Practice responsible and safe technology use. ● Apply computational thinking to real-world routines.
Lessons	**Computer Basics: Exploration (1.FC.2)** ● Identify computer parts and explain how we use them to complete tasks. **Exploring Responsible Technology Use (1.NI.1)** ● Explain ways to use technology safely and responsibly. **Positive Online Behavior (1.NI.1)** ● Identify appropriate and inappropriate online behavior. **Computational Thinking: Evening Routines (1.AT.2)** ● Break down real-life routines using computational thinking.

Unit 3: Sequences & Events (2 weeks)

In this unit, students deepen their understanding of sequencing and events by building interactive programs that include hiding, showing, and debugging. They explore how blocks work together to control when and how actions happen.

Objectives / Topics Covered	● Use the hide block to create an interactive game. ● Identify and fix bugs in simple sequences.

Lessons	**Hide and Seek Game (1.FC.2)** ● Use the hide block to program an interactive game of hide-and-seek. **Introduction to Debugging (1.FC.2)** ● Find and correct bugs in sequences.

Unit 4: Loops & Message Events (4 weeks)

In this unit, students deepen their understanding of loops and message events to program more interactive behaviors. They explore character interactions, speed control, and communication between sprites.

Objectives / Topics Covered	● Use repeat loops to simplify and extend programs. ● Program character interactions using events and loops. ● Use message events to communicate between characters. ● Control animation speed using speed blocks and messages.
Lessons	**Loops: Predator and Prey (2 classes | 1.FC.2)** ● Use events, sequences, and loops to program two animal characters to interact with each other. **Message Events: Simon Says (1.FC.2)** ● Use message events to make one character communicate to many characters in a program. **Speed Block: Bouncy Ball (1.FC.2)** ● Use speed blocks and messages to program a character to move at different speeds.

Unit 5: Pages & Grid (2 weeks)

In this unit, students explore how to build more complex projects by using multiple pages and the grid system to navigate and program spatial relationships.

Objectives / Topics Covered	● Create interactive programs with multiple pages. ● Use “go to page” blocks for navigation. ● Apply grid-based logic to move characters through a maze.
Lessons	**Pages: Create a Tapping Game (1.FC.2)** ● Create a game that moves from one page to the next using “go to page” blocks. **Grid: Solving Mazes (1.FC.2)** ● Design a maze and use the grid to program a character to move through the maze.

Unit 6: Culmination Projects (4 weeks)

In this unit, students apply the programming and problem-solving skills they’ve learned throughout the course to create animated stories and interactive projects.

Objectives / Topics Covered	● Apply the design process to plan and revise a project. ● Create a multipage animation using messages, loops, and sequences.
Lessons	**Practicing the Design Process (1.AT.3)** ● Use the design process to plan, create, and revise a project that models a solution to a simple real-world problem. **Space Travel Project (3 classes | 1.FC.2)** ● Create and explore ways to program a multipage story using messages, loops, and sequences.

Unit 7: Digital Literacy (7 weeks)

In this unit, students explore digital tools and concepts such as software selection, spreadsheets, data visualization, attribution, and basic AI through engaging and age-appropriate activities.

Objectives / Topics Covered	● Compare and choose appropriate software for simple computing tasks. ● Use spreadsheets to organize and analyze data. ● Practice safe, responsible, and ethical digital behavior. ● Model how computers use data to make decisions.
Lessons	**Types of Software (1.FC.1)** ● Compare and select appropriate software applications to complete different computing tasks. **Introduction to Spreadsheets (1.FC.1)** ● Use spreadsheet software to review and present data on class pets and favorite seasons. **Guided Research (2 classes | 1.AT.1, 1.AT.2, 1.DA.1)** ● Find information using research sources and create a program to communicate findings visually. **Basic Data and Programming (1.DA.1, 1.DA.2)** ● Collect data and create a program to present the data visually. **Giving Credit Through Attributions (1.NI.1)** ● Reuse parts of a story to make a new one and give credit to the original creator. **Sorting with Decision Trees (1.DA.2)** ● Create a simple decision tree to sort items and explain how AI uses data to learn.

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	● Model and explain math concepts such as place value, telling time, and dividing shapes. ● Create animations that support understanding of science concepts like day/night and sound. ● Reinforce reading, writing, and phonics through interactive story-based programs.
Lessons	**Place Value: Adding Up to 20 (1.FC.2)** ● Use events to illustrate how to decompose a two-digit number into tens and ones. **Story Problems: Add and Subtract within 20 (1.FC.2)** ● Use events to create a scene that represents an addition or subtraction story problem. **Telling Time (1.FC.2)** ● Use sequences and events to create an analog clock and display time in digital and analog forms. **Divide Shapes into Equal Parts (1.FC.2)** ● Create an animation to partition circles and rectangles into equal parts. **Sun and Moon, Day and Night (1.FC.2)** ● Use loops to model the movements of the sun and moon and show the pattern of day and night. **Sound and Pitch (1.FC.2)** ● Create and use animation to model sound and pitch and explain the relationship between sound and vibration. **Animal Life Cycles (1.FC.2)** ● Use message events to model the butterfly life cycle. **Phonics: Digraphs (1.FC.2)** ● Create a phonics program with digraphs using events and recordings. **Build a Sentence (1.FC.2)** ● Create an interactive program that uses events to write sentences and then read them aloud.

	Original Story Animations - Personal Story (2 classes | 1.FC.2) ● Design a program to animate an original story and describe the program development process. **Careers in CS: Litter Free Communities (1.FC.2)** ● Explain how computer science can help solve community problems and create a program to collect and sort litter. **Impacts of Technology in Our World (1.NI.1)** ● Create a program to demonstrate how technology impacts our world.

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**.	