



Tennessee 4th Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Tennessee 4th 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 a 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 teachers can weave 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/26973/overview>.



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

Course Breakdown

Optional Review (2 weeks)

This optional unit reviews foundational programming skills and introduces students to the CodeHop platform. Students practice navigation, basic coding, and coordinate-based positioning.

Objectives / Topics Covered	• Log in and navigate the CodeHop Playground. • Define and use basic computer science terms. • Create simple programs using the coordinate plane.
Lessons	Welcome to CodeHop! (4.FC.2) • Log in and navigate the CodeHop Playground. Introduction to Computer Science • Define computer science vocabulary and create a simple program. The Coordinate Plane • Create an open-ended animation using the coordinate plane.

Unit 1: Getting Started (3 weeks)

This unit introduces students to computing systems, responsible technology use, and computational thinking through real-world and creative applications.

Objectives / Topics Covered	• Identify parts of the computing system and describe hardware/software problems. • Explain how to act responsibly and spread positivity online. • Use computational thinking to break down and design a school layout.
Lessons	Exploring Computing Systems (4.FC.1, 4.IC.3) • Identify parts of the computing system and identify simple hardware and software problems. Internet Positivity (4.NI.1) • Explain how actions online can spread positivity and create a responsible conduct code. Computational Thinking: Design a School • Use computational thinking to design a school.

Unit 2: Sequences, Events, & Loops (5 weeks)

This unit helps students understand how sequences, events, and loops are used in programming by creating algorithms, collaborating on projects, and debugging code.

Objectives / Topics Covered	• Program algorithms and choose the best solution. • Collaborate to design interactive programs using events. • Use and compare different types of loops. • Decompose and debug programs.
Lessons	Creating Algorithms • Program multiple algorithms and assess which one best meets their needs. Pair Programming: Create a Band (2 classes 4.FC.1, 4.AT.1, 4.PC.1) • Collaborate through pair programming to design and code a band using keyboard inputs. Loops: Catch the Ball • Use two types of loops to create a simple game. Debugging: Mazes (4.PC.1) • Decompose a program to debug and make the program run as intended.

Unit 3: Conditionals & Operators (2 weeks)

This unit introduces students to conditional logic and the use of operators in programming by designing interactive projects that react to user input and decision-making structures.

Objectives / Topics Covered	• Create programs using basic conditional blocks. • Apply “if/then/else” logic to create responsive programs. • Use events to trigger condition-based actions.
Lessons	Conditionals: Underwater Exploration • Create a program that uses conditionals. Complex Conditionals: Chase the Star • Explain what an “if/then/else” conditional is and use it in a program.

Unit 4: Variables, Clones, & Functions (5 weeks)

This unit deepens students’ understanding of how variables, clones, and functions are used in programs to manage information, control repeated elements, and organize reusable code.

Objectives / Topics Covered	• Create and use variables to store and update data. • Use clones to duplicate sprites and simplify animation. • Build functions with different input types to trigger unique behaviors.
Lessons	Pong Game (2 classes 4.AT.1, 4.PC.1) • Create and use variables to keep score in an interactive pong game. Introduction to Clones • Create an animation using clones and investigate the limitations of the program. Scout's Quest: Functions with Boolean Inputs • Create a function including a boolean input to perform different actions based on whether a password is correct. Scout's Quest: Functions with Number Inputs • Create a drawing using functions with number inputs.

Unit 5: Culmination Projects (4 weeks)

This unit allows students to apply the programming skills they’ve developed throughout the course by completing creative and open-ended projects that emphasize user-centered design and real-world application.

Objectives / Topics Covered	• Apply programming concepts like variables, operators, and conditionals to real-world projects. • Use design thinking to improve program usability and accessibility. • Design and test original programs using iterative development.
Lessons	Code Tunes (2 classes 4.AT.1, 4.PC.1) • Use variables, operators, and conditionals to create a custom music player. Designing Solutions for Accessibility (2 classes 4.FC.1, 4.AT.1, 4.AT.2, 4.IC.2, 4.IC.3) • Use the design thinking process to redesign a game that improves accessibility and usability for diverse users.

Unit 6: Digital Literacy (5 weeks)

This unit introduces students to digital citizenship concepts including safe online behavior, research strategies, data collection, and how to securely communicate and store information using cryptographic concepts.

Objectives / Topics Covered	• Examine sources and communicate findings through digital programs. • Collect and visualize data. • Practice basic encryption techniques and understand digital security concepts.
-----------------------------------	---

Lessons	Research: Informational Programs (2 classes 4.FC.1, 4.FC.2, 4.NI.2) - Examine resources and create a PSA on healthy sleep habits. Programming and Data Project (2 classes 4.FC.1, 4.DA.1, 4.DA.2) - Develop a question, collect and analyze data, and present findings with an interactive program. Scout's Cryptography Escape Room (4.AT.3) - Solve cipher puzzles and learn about cryptography and information protection.
---------	--

Unit 7: Interdisciplinary Computer Science (11 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	- Use conditionals to model and solve real-world math problems. - Create programs that simulate scientific processes and systems. - Develop storytelling and ELA-based animations. - Explore how technology and culture intersect in creative projects.
Lessons	Multi-digit Multiplication and Conditionals - Use if/then conditionals to review multiplication with multi-digit factors. Multiplying Fractions by Whole Numbers (4.AT.1) - Create an interactive activity to demonstrate multiplying fractions by whole numbers. House Design with Area and Perimeter (2 classes 4.AT.1) - Calculate and use the area and perimeter of a room to create a house design using functions. Exploring Heat (4.FC.3) - Use events in a program to communicate information about how heat energy from the sun affects objects on earth. Exploring the Water Cycle (4.FC.3) - Explain the phases of the water cycle using broadcast message events. Creative Storytelling (2-part lesson) - Plan and animate a story using events and sequences. Nonfiction Animated Recordings - Use events to create a clear, animated reading of a nonfiction text. Choose Your Own Path: Elements of Culture (2 classes 4.FC.1, 4.AT.1, 4.AT.2, 4.NI.1) - Identify elements of culture as you create a cultural choose-your-own-path game.

4th - 5th 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 .	