



Tennessee 3rd Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Tennessee 3rd 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.

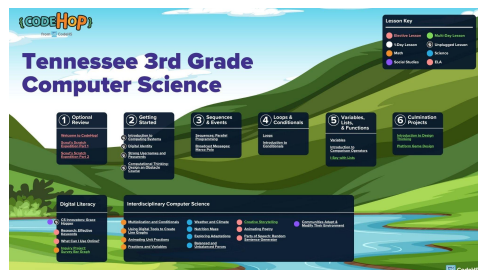
Unique to Tennessee: The pathway is structured as an approximate 36-hour sequence that balances computer science skill development, interdisciplinary integration, and digital literacy. Core CS modules spiral across grade levels, while interdisciplinary lessons are organized in subject-area menus (Math, Science, ELA, and Social Studies) that 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/26972/overview>.



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

Course Breakdown

Optional Review (2 weeks)

In this unit, students are introduced to the CodeHop platform and begin programming using story-driven activities. They explore foundational coding concepts including sequences, events, and sprite interactions.

Objectives / Topics Covered	• Navigate the CodeHop interface and use basic programming blocks. • Create sequences that animate sprites and tell stories. • Practice programming movement, dialogue, and sprite behavior.
Lessons	Welcome to CodeHop! (15 minute lesson) • Log in and navigate the CodeHop Playground. Scout's Programming Expedition Part 1 • Use basic commands to program a sprite to move and talk. Scout's Programming Expedition Part 2 • Add sprites and create a sequence to animate a story.

Unit 1: Getting Started (4 weeks)

In this unit, students explore the parts of computing systems and practice safe, responsible use of technology. They also begin developing their computational thinking skills through unplugged problem solving.

Objectives / Topics Covered	• Identify hardware and software components of computing systems. • Practice safe behaviors and strong password creation. • Connect offline and online identity. • Use computational thinking to design solutions.
Lessons	Introduction to Computing Systems • Identify parts of the computing system and solve simple hardware and software problems. Digital Identity (3.NI.1) • Connect offline and online identity and describe how actions shape a digital footprint. Strong Usernames and Passwords (3.NI.1) • Develop a strong username and password and explain why password strength matters. Computational Thinking: Design an Obstacle Course (3.AT.1, 3.AT.2, 3.PC.2) • Use computational thinking to design an obstacle course.

Unit 2: Sequences & Events (2 weeks)

In this unit, students deepen their understanding of sequences and events by building programs that include simultaneous actions and sprite communication.

Objectives / Topics Covered	• Create parallel sequences using event blocks. • Trigger actions between sprites using broadcast messages.
Lessons	Sequences: Parallel Programming (3.AT.2, 3.PC.2) • Create a program using parallel sequences. Broadcast Messages: Marco Polo • Use broadcast messages to trigger action between sprites in a program.

Unit 3: Loops & Conditionals (2 weeks)

In this unit, students explore how loops and conditionals can control the flow of a program and make animations or interactions more efficient and responsive.

Objectives / Topics Covered	• Use loops to repeat one or more instructions. • Create programs using if/then conditional logic.
Lessons	Loops • Use loops to repeat one or more instructions. Introduction to Conditionals • Create a program that uses if/then blocks to make decisions.

Unit 4: Variables, Lists, & Functions (4 weeks)

In this unit, students explore how variables store values, how comparison operators influence decisions, and how lists and data structures support interactive programs.

Objectives / Topics Covered	• Create and modify variables in programs. • Use comparison operators to make decisions. • Build interactive programs using lists.
Lessons	Variables • Create and change the value of a variable in a program. Introduction to Comparison Operators (3.PC.3) • Use comparison operators in if/else blocks with numbers and variables. I-Spy with Lists (2 classes 3.PC.3) • Create an I-Spy activity using variables and lists.

Unit 5: Culmination Projects (4 weeks)

This unit allows students to apply foundational programming concepts in creative, culminating projects that integrate design thinking, feedback, and storytelling.

Objectives / Topics Covered	• Apply loops, conditionals, and events in a game or story-based project. • Use the design thinking process to solve user-centered problems.
Lessons	Introduction to Design Thinking (2 classes 3.AT.3, 3.PC.3) • Use the design thinking process to make a tool accessible to more users. Platform Game Design (2 classes 3.AT.3, 3.PC.2) • Create a platform game using keyboard events, loops, and conditionals, and improve the program through debugging.

Unit 6: Digital Literacy (5 weeks)

Students explore how to search, evaluate, and share information online responsibly. They build foundational research skills, learn about key figures in computing, and represent their findings with code and data visualizations.

Objectives / Topics Covered	• Explore key figures in computer science history. • Use effective search strategies and evaluate online sources. • Practice digital citizenship by understanding attribution and data representation.
Lessons	CS Innovators: Grace Hopper (3.PC.2) • Explain Grace Hopper's contributions to computing and use binary code to decode mystery words.

	Research: Effective Keywords (3.FC.1, 3.DA.2, 3.NI.2) - Evaluate search keywords and determine if a source is credible and relevant. What Can I Use Online? (3.FC.1, 3.DA.2, 3.NI.2) - Search for information and give proper attribution to sources. Inquiry Project: Survey Bar Graph (2-part lesson) - Follow the inquiry process and modify a program to display data from a class investigation.
--	---

Unit 7: 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	- Apply computer science skills to explore and reinforce math, science, and ELA concepts. - Create interactive programs using loops, conditionals, variables, and events. - Model scientific processes and illustrate literacy and social studies ideas.
Lessons	Multiplication and Conditionals - Create a program that uses "if/then" conditional blocks to review multiplication. Animating Unit Fractions (3.PC.1, 3.PC.3) - Use loops in a program to animate repeated addition of unit fractions on a number line. Fractions and Variables (3.PC.1, 3.PC.3) - Represent fractions on a number line using variables and conditionals. Using Digital Tools to Create Line Graphs (3.FC.2, 3.FC.3, 3.DA.1) - Examine a table of information and convert the values into a data visualization (line graph) that supports a claim. Weather and Climate (3.FC.2, 3.FC.3) - Use climate data and event blocks to predict and demonstrate typical weather conditions for a specific month. Exploring Adaptations - Explain how adaptations help animals survive in their habitats by creating an interactive program using events. Nutrition Maze (3.PC.2) - Describe the benefits of healthy foods and use conditionals to create an interactive nutrition maze game. Balanced and Unbalanced Forces (3.PC.3) - Describe how balanced and unbalanced forces impact an object's speed and model these forces using conditionals and variables. Parts of Speech: Random Sentence Generator - Generate random numbers to create simple sentences with nouns, adjectives, verbs, and adverbs. Animating Poetry - Break a poem into sections to understand literal and metaphorical meanings and create an animated reading of a poem. Creative Storytelling (2 classes 3.FC.2, 3.FC.3, 3.AT.1) - Plan and animate a story using events and sequences. Communities Adapt to & Modify Their Environment - Use click events to create a scene that shows how communities adapt to or modify their environments.

3rd 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 .	