



Tennessee 2nd Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Tennessee 2nd 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/26971/overview>.



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

Course Breakdown

Unit 1: Optional Review (3 weeks)

This optional unit gives students a refresher on foundational programming concepts and navigation skills. Lessons reinforce the use of events, loops, and messages through simple, creative projects.

Objectives / Topics Covered	- Log in and navigate the CodeHop Playground. - Review key programming blocks, including events, loops, and messages. - Reinforce foundational animation and sequencing concepts.
Lessons	Welcome to CodeHop! (15 minute lesson) - Log in and navigate the CodeHop Playground. Introduction to Programming - Navigate the CodeHopJr interface to create a scene with characters. Forever Loop Dance Party - Create a sequence using a “repeat forever” loop to make characters repeat actions. Introduction to Message Events - Program a relay race that uses messages to cause characters to interact.

Unit 2: Getting Started (4 weeks)

In this unit, students explore the fundamentals of computer systems and practice responsible technology use. They are also introduced to conditionals and computational thinking skills through unplugged and classroom-based activities.

Objectives / Topics Covered	- Explain the basic parts of a computer system and how they work together. - Demonstrate safe and responsible technology use. - Use conditionals to guide a character through a maze. - Apply computational thinking to break down and sequence daily tasks.
Lessons	Computer Basics: Connections - Identify input, output, hardware, and software, and explain how they work together. Practicing Responsible Technology Use (2.NI.1, 2.IC.1) - Demonstrate ways to use technology safely and responsibly. Coding Card Game: Conditionals (2.DA.2) - Work together to create a sequence of instructions with conditionals to move Scout through a maze. Computational Thinking: School Day Routines - Use computational thinking to identify patterns and sequence school day tasks.

Unit 3: Sequences & Events (2 weeks)

This unit reinforces algorithms and events by using blocks to create and debug interactive programs. Students apply sequencing and problem-solving strategies to move characters on the stage.

Objectives / Topics Covered	- Create and adjust simple algorithms for character movement. - Identify and correct errors in event-based code.
Lessons	Debugging: Events and Sequences - Find and fix errors in provided code. Algorithms: Connecting a Path (2.AT.2) - Create and adjust simple algorithms to move characters based on their size, shape, and starting position.

Unit 4: Message Events (2 weeks)

In this unit, students deepen their understanding of how characters interact in a program using message events. Students will explore the use of messages to model cycles and navigate between pages.

Objectives / Topics Covered	• Use message events to model real-world processes and cycles. • Program characters to communicate using messages across multiple pages.
Lessons	Programming a Cycle (2.FC.4) • Use message events to model a cycle. Pages: Scout's Travels (2.FC.4) • Use messages to help Scout travel between pages in a program.

Unit 5: Loops (2 weeks)

In this unit, students learn how loops can repeat actions in a program. They use wait and turn blocks to control timing and revise programs collaboratively using feedback and attribution.

Objectives / Topics Covered	• Use loops to repeat actions with varying speeds. • Collaborate on projects using peer feedback and give proper attribution.
Lessons	Making a Timer (2.FC.4) • Use loops, wait blocks, and turn blocks to create and compare two timers with different speeds. Two-Step Dance & Feedback (2.FC.2, 2.NI.2) • Create a program and revise it based on peer feedback while giving attribution to a peer who helped improve the work.

Unit 6: Culmination Projects (4 weeks)

In this unit, students apply programming concepts they've learned throughout the year to design, build, and improve creative projects. They use the design process to solve problems, incorporate multiple types of blocks, and reflect on feedback.

Objectives / Topics Covered	• Apply the steps of the design process to a project. • Program interactive games and animations using sequences, events, loops, and messages. • Revise programs based on peer or teacher feedback.
Lessons	Exploring the Design Process (2.FC.4, 2.AT.1, 2.AT.2) • Use the design process to plan, create, and improve a program with loops that models a solution to a simple real-world problem. Maze Game Project (3-part lesson) • Create and explore multiple ways to program an interactive game using events, messages, loops, and sequences, and revise the program based on feedback.

Unit 7: Digital Literacy (9 weeks)

In this unit, students explore digital responsibility, privacy, and the use of technology to collect and represent information. They investigate ways to stay safe online, interpret data, and use visual tools like spreadsheets and programming to communicate findings.

Objectives / Topics	• Explain safe and responsible digital behavior, including the use of passwords and reporting concerns.
------------------------	---

Covered	• Use spreadsheets to review and present simple datasets. • Identify trends in data and use programs to represent patterns and predictions. • Conduct simple research to answer a question and present findings visually.
Lessons	Password Protectors (2.NI.1, 2.IC.1) • Understand the importance of usernames and passwords and demonstrate strategies to keep login information safe. Responsible Digital Citizens (2.NI.1, 2.IC.1) • Explain what it means to be a responsible digital citizen, including understanding digital footprints, discussing cyberbullying, and knowing how to report concerns. Exploring Spreadsheets (2.FC.1, 2.AT.2, 2.DA.3) • Use spreadsheet software to review and present data on class pets and favorite colors. Choice Research (2 classes 2.FC.3, 2.NI.2, 2.NI.3) • Collect and assess sources to answer a research question and communicate findings visually. Data Patterns and Predictions (2.DA.1, 2.DA.2, 2.DA.3) • Identify and describe patterns and trends in data visualizations, then create a program using events to communicate patterns, trends, and predictions from a given data set. Advanced Data and Programming (3 classes 2.FC.3, 2.DA.1, 2.DA.2, 2.DA.3, 2.NI.3) • Develop an investigative question, collect data using a survey, and create a program to present the data visually.

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 math concepts such as story problems, shapes, and place value through programming. • Use sequences, loops, and message events to simulate scientific phenomena. • Apply computer science to literacy and storytelling. • Explore real-world community and technology connections through interactive projects.
Lessons	Story Problems: Add and Subtract within 100 (2.FC.4, 2.AT.2) • Create a program to model a math story problem. Place Value: Ones, Tens, and Hundreds (2.FC.4) • Connect a digit's place in a number to its value and create an interactive program that uses events. Identify Shapes by Attributes (2.AT.3) • Create a program to draw and identify shapes with specified attributes. Greater Than and Less Than: Three-Digit Numbers • Use events to create a program that tells if a number is greater than, less than, or equal to another number. Designing Solutions from Nature (2.FC.3, 2.AT.2) • Create a program using events that show how humans survive in their environments by mimicking plants and animals. Properties of Matter (2.FC.4, 2.AT.3) • Use conditionals to program a robot to make accurate decisions and to group different types of matter according to their properties. Seed Dispersal (2.AT.1) • Create a program using message events and loops to model how an animal can help disperse seeds. Preventing Erosion (2.AT.2) • Create a program to compare multiple solutions designed to slow or prevent wind or water from changing the shape of the land.

	Punctuation: Write a Great Sentence! (2.FC.4) • Create sequences with loops to write sentences with correct punctuation and spacing. Original Story Animations - Fiction Story (2 classes 2.AT.1) • Develop an original story and create a program to animate a story. Careers in CS: Coding for Fashion-Retail (2.FC.4) • Explain how coding helps create and improve fashion designs and create a program to design and animate a fashion character. Communities Modify Their Environment (2.FC.4) • Create a program that shows how people modify their environment in a community.
--	--

2nd 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.
These resources and more are found on the CodeHop Resources Page .	