Texas Computer Science 2nd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Texas Computer Science 2nd Grade Course 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 and collaboration, providing students with a solid base in computer science concepts and digital literacy.

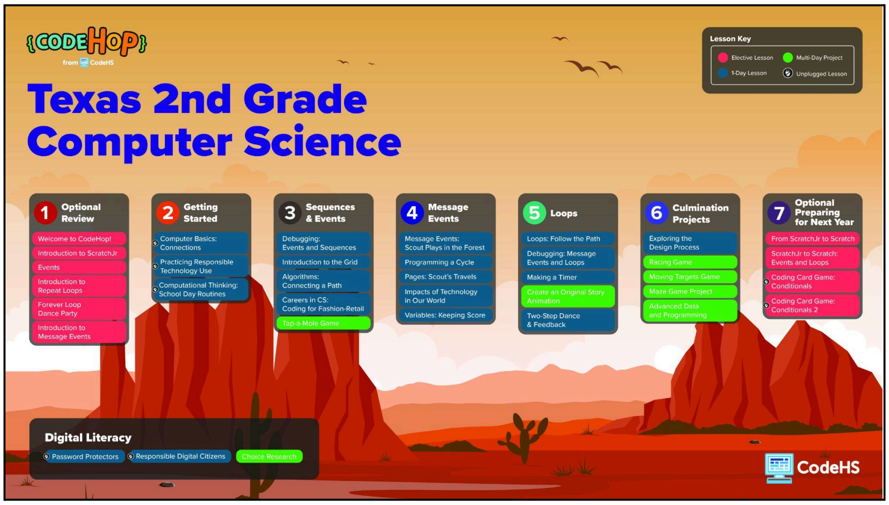
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts and build confidence over time.

The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. Required lessons are labeled with the specific standards they address to support planning and compliance.

Programming Environment: Students will write and run programs that are saved in the CodeHop platform. 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/26371/overview



Texas 2nd Grade Computer Science

Lesson Key
- 🟥 Elective Lesson
- 🟩 Multi-Day Project
- 🟦 1-Day Lesson
- 🔌 Unplugged Lesson

1 Optional Review
- Welcome to CodeHop!
- Introduction to ScratchJr
- Events
- Introduction to Repeat Loops
- Forever Loop Dance Party
- Introduction to Message Events

2 Getting Started
- 🔌 Computer Basics: Connections
- 🔌 Practicing Responsible Technology Use
- 🔌 Computational Thinking: School Day Routines

3 Sequences & Events
- Debugging: Events and Sequences
- Introduction to the Grid
- Algorithms: Connecting a Path
- Careers in CS: Coding for Fashion-Retail
- Tap-a-Mole Game

4 Message Events
- Message Events: Scout Plays in the Forest
- Programming a Cycle
- Pages: Scout's Travels
- Impacts of Technology in Our World
- Variables: Keeping Score

5 Loops
- Loops: Follow the Path
- Debugging: Message Events and Loops
- Making a Timer
- Create an Original Story Animation
- Two-Step Dance & Feedback

6 Culmination Projects
- Exploring the Design Process
- Racing Game
- Moving Targets Game
- Maze Game Project
- Advanced Data and Programming

7 Optional Preparing for Next Year
- From ScratchJr to Scratch
- ScratchJr to Scratch: Events and Loops
- 🔌 Coding Card Game: Conditionals
- 🔌 Coding Card Game: Conditionals 2

Digital Literacy
- 🔌 Password Protectors
- 🔌 Responsible Digital Citizens
- Choice Research

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

Course Breakdown

Optional Review

This optional review unit offers extra practice for students who need reinforcement or a refresher on key programming concepts. It includes introductory lessons and creative activities that revisit sequences, events, loops, and message blocks in a flexible and engaging way.

Objectives / Topics Covered	● Review core programming concepts like events, loops, and message events. ● Strengthen comfort navigating the programming interface. ● Build confidence using basic blocks to create interactive programs. ● Provide flexible support for reteaching or extending learning.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Introductory lesson to help students log in and explore the CodeHS Playground; ideal as a warm-up or standalone activity. **Introduction to Programming** ● Navigate the interface and create a scene with characters and background. **Events** ● Use multiple event blocks to make characters move, react, or interact based on user input. **Introduction to Repeat Loops** ● Learn how to use repeat loops to make actions happen more than once in a row. **Forever Loop Dance Party** ● Create a fun animation where characters continuously dance using the “repeat forever” loop. **Introduction to Message Events** ● Program a relay-style animation where characters take turns moving using send and receive message blocks.

Unit 1: Getting Started (3 lessons)

Students will learn how computers work, how to use them responsibly, and how to think like a computer scientist by recognizing patterns and breaking down tasks from their daily routines.

Objectives / Topics Covered	● Identify the parts of a computer and how they work together. ● Practice safe and responsible technology use. ● Apply computational thinking to real-life routines.
Lessons	**Computer Basics: Connections** ● Explore computer parts, how they function, and what to do when something doesn’t work. **Practicing Responsible Technology Use** ● Learn and demonstrate safe, respectful, and responsible tech use. **Computational Thinking: School Day Routines** ● Break down school day routines using patterns, sequencing, and problem-solving skills.

Unit 2: Sequences & Events (6 lessons)

Students will strengthen their understanding of sequences, events, and debugging by creating programs with movement, interactivity, and creativity—including games and real-world applications.

Objectives /	● Debug code by identifying and fixing sequence or event errors.

Topics Covered	● Use the grid to control character positioning. ● Build and adjust simple algorithms based on design constraints. ● Understand coding applications in fashion design. ● Create an interactive game using event-based programming.
Lessons	**Debugging: Events and Sequences** ● Find and fix errors in sample code to improve program outcomes. **Introduction to the Grid** ● Use the grid to move characters precisely on the stage. **Algorithms: Connecting a Path** ● Write and adjust step-by-step instructions to connect characters on the screen. **Careers in CS: Coding for Fashion-Retail** ● Learn how coding is used in fashion, then animate a custom fashion character. **Tap-a-Mole Game (2 part lesson)** ● Design a game using events where users tap characters to score points.

Unit 3: Message Events (4 lessons)

Students will use message events to control program flow, model real-world cycles, and build interactive, multi-page experiences. They'll also explore the impact of technology and use variables to track scores in a program.

Objectives / Topics Covered	● Use message events to sequence and organize actions in a program. ● Model cycles and multi-page navigation with messages. ● Demonstrate technology's impact through programming. ● Use variables to keep score in interactive programs.
Lessons	**Message Events: Scout Plays in the Forest** ● Control a program's flow using messages to trigger actions. **Programming a Cycle** ● Use messages to represent and loop through a natural or repeating cycle. **Pages: Scout's Travels** ● Help Scout travel between scenes by combining message events and page transitions. **Impacts of Technology in Our World** ● Create a project that shows a positive or negative impact of technology. **Variables: Keeping Score** ● Program a simple score-tracking system using a variable.

Unit 4: Loops (6 lessons)

Students will explore how loops simplify patterns in code, practice debugging, build timers, and create animated stories. They'll also learn to revise programs using peer feedback and give proper credit.

Objectives / Topics Covered	● Identify and use loops to repeat actions and patterns. ● Debug code involving message events and loops. ● Use loops with timing and motion blocks. ● Create and revise animations with feedback and attribution.
Lessons	**Loops: Follow the Path** ● Identify repeated patterns and use loops to move characters efficiently. **Debugging: Message Events and Loops** ● Find and fix coding errors involving loops and message events. **Making a Timer** ● Use loops, wait, and turn blocks to create two timers with different speeds. **Create an Original Story Animation (2 part lesson)**

	● Design and animate a unique story using characters, loops, and events. **Two-Step Dance & Feedback** ● Code a dance, revise it with peer feedback, and give credit to collaborators.

Unit 5: Culmination Projects (12 lessons)

In this unit, students apply their coding skills to design interactive games and data-driven projects. They'll use sequences, events, loops, and messages while exploring the design process and refining their work with feedback.

Objectives / Topics Covered	● Apply sequences, events, loops, and messages in original projects. ● Design interactive games and animations. ● Collect and present data visually using programming. ● Revise programs based on peer feedback and design thinking.
Lessons	**Exploring the Design Process** ● Apply all steps of the design process independently to create and improve a program that includes loops to solve a real-world problem from a user's perspective. **Racing Game (2 part lesson)** ● Create a racing game using events, loops, and messages to control character actions. **Moving Targets Game (3 part lesson)** ● Program a game with moving targets that uses sequences, events, and pages. **Maze Game Project (3 part lesson)** ● Design a maze game and revise it using feedback while applying key programming concepts. **Advanced Data and Programming (3 part lesson)** ● Conduct a survey, analyze the data, and create a program to share findings visually.

Unit 6: Digital Literacy (4 lessons)

Students will learn how to protect their personal information, act responsibly online, and conduct research using reliable sources to communicate their findings.

Objectives / Topics Covered	● Understand the importance of usernames and passwords. ● Practice responsible online behavior and digital citizenship. ● Research a topic, assess sources, and share results visually.
Lessons	**Password Protectors** ● Learn why usernames and passwords matter and how to keep them secure. **Responsible Digital Citizens** ● Explore digital footprints, cyberbullying, and how to report unsafe online behavior. **Choice Research** ● Research a self-selected topic using reliable sources and create a visual program to share findings.

Optional Preparing for Next Year

These optional lessons help students build on what they've learned by applying conditionals in unplugged activities and transitioning from ScratchJr to Scratch programming.

Objectives / Topics Covered	● Use conditionals to create step-by-step instructions. ● Collaborate to solve maze challenges with logic. ● Explore the Scratch interface and basic programming tools. ● Create simple Scratch programs using events and loops.

Lessons	**Coding Card Game: Conditionals** ● Use conditionals to guide Scout through a maze using a card-based activity. **Coding Card Game: Conditionals 2** ● Continue practicing conditional logic with new maze challenges. **From ScratchJr to Scratch** ● Learn the basics of Scratch by building a simple program from scratch. **ScratchJr to Scratch: Events and Loops** ● Create a Scratch program that includes an event and a repeating action using a loop.

Texas Computer Science 2nd Grade Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
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.
All of these resources and more are found on the **Elementary Resources Page**.	