Texas Computer Science 4th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Texas Computer Science 4th 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.

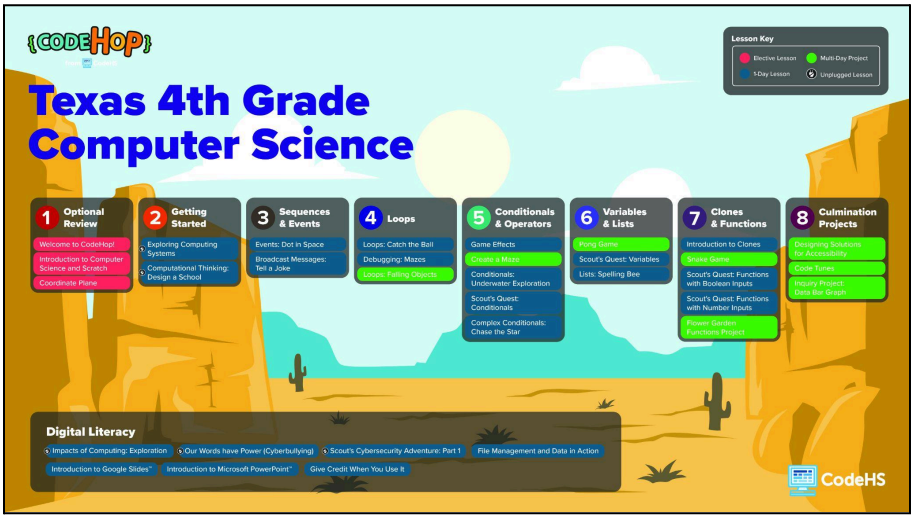
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts 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/26376/overview



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

Course Breakdown

Optional Review

This optional review unit helps students refresh foundational computer science skills and get reacquainted with programming. It includes lessons on logging in, reviewing vocabulary, and practicing programming with coordinates and simple animations.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Review key computer science terms and programming basics. ● Use the coordinate plane to control sprite movement.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Practice logging in and exploring the Playground before starting a full lesson. **Introduction to Computer Science** ● Review basic computer science vocabulary and create a simple program. **The Coordinate Plane** ● Use the coordinate plane to design an open-ended animation.

Unit 1: Getting Started (2 lessons)

Students will explore the parts of a computing system and apply computational thinking to creatively solve problems and design solutions.

Objectives / Topics Covered	● Identify parts of a computing system and troubleshoot simple issues. ● Apply computational thinking to plan and solve design challenges.
Lessons	**Exploring Computing System** ● Identify hardware and software parts of a computer and spot simple problems. **Computational Thinking: Design a School** ● Use computational thinking to design and plan the layout of a school.

Unit 2: Sequences & Events (2 lessons)

Students will build interactive programs using event blocks and broadcast messages to control the flow of actions between sprites.

Objectives / Topics Covered	● Use multiple types of event blocks to trigger actions. ● Coordinate sprite interactions using broadcast messages.
Lessons	**Events: Dot in Space** ● Create a program using different types of event blocks to control actions. **Broadcast Messages: Tell a Joke** ● Use broadcast messages to program two sprites to perform a knock knock joke.

Unit 3: Loops (4 lessons)

Students will explore different types of loops, use them to build simple games, and practice debugging by breaking down and fixing programs.

Objectives /	● Use multiple types of loops to simplify and control repeated actions.

Topics Covered	● Debug programs by decomposing and correcting code. ● Compare loop algorithms and their effectiveness in different scenarios.
Lessons	**Loops: Catch the Ball** ● Use repeat and forever loops to build a simple ball-catching game. **Debugging: Mazes** ● Decompose and fix a maze program to make it run as expected. **Loops: Falling Objects (2 day lesson)** ● Use different loops to animate falling objects and compare their behavior.

Unit 4: Conditionals & Operators (6 lessons)

Students will use conditionals and operators to make decisions in their programs, create interactive games, and respond to peer feedback to enhance their projects.

Objectives / Topics Covered	● Use if/then conditionals and operators to control actions based on specific conditions. ● Apply conditionals in game logic to enhance interactivity. ● Create custom visuals like mazes and game effects. ● Improve programs through peer feedback and reflection.
Lessons	**Game Effects** ● Modify an existing game by adding creative effects and making improvements based on peer feedback. **Create a Maze (2 day lesson)** ● Design a custom maze backdrop and program Scout to navigate through it using logic and movement. **Conditionals: Underwater Exploration** ● Build an underwater scene that uses conditionals to control character behavior and responses. **Scout's Quest: Conditionals** ● Use if/then conditionals to build a program that responds to different inputs or scenarios. **Complex Conditionals: If/Then/Else Chase the Star** ● Use if/then conditionals to build a program that responds to different inputs or scenarios.

Unit 5: Variables & Lists (4 lessons)

Students will build interactive programs that use variables to track information and lists to manage sets of data. They'll apply these concepts through games and skill-building challenges.

Objectives / Topics Covered	● Create and use variables to store and update values. ● Use lists to manage and display collections of information. ● Apply variables and lists in game-based projects.
Lessons	**Pong Game (2 day lesson)** ● Create a pong game that uses variables to keep score. **Scout's Quest: Variables** ● Practice creating and updating variables to track points in a program. **Lists: Spelling Bee** ● Build a spelling bee game that uses a list of words for interactive play.

Unit 6: Clones & Functions (7 lessons)

In this unit, students explore advanced programming concepts like cloning and functions with inputs. They'll use these tools to create dynamic projects—from animations and games to creative drawings—while developing efficient, reusable code structures and continuing skill review through Scout's Quest.

Objectives / Topics Covered	● Create and use clones to animate or duplicate sprites in a program. ● Build functions with Boolean and number inputs to control program behavior. ● Use variables and clones to design interactive games. ● Apply functions to automate and simplify complex drawings or repeated tasks.
Lessons	**Introduction to Clones** ● Create an animation using clones and explore how duplicate sprites can be used efficiently—and where they may have limitations. **Snake Game (2 day lesson)** ● Build a version of the classic Snake game using clones and variables to track movement and score. **Scout's Quest: Functions with Boolean Inputs** ● Create a function with a true/false input to determine actions, like checking a password. **Scout's Quest: Functions with Number Inputs** ● Use number inputs in functions to create a program that draws with specific values. **Flower Garden Functions Project (2 day lesson)** ● Use a function to draw multiple flowers in a scene, making the code cleaner and more efficient through reuse.

Unit 7: Culmination Projects (5 lessons)

In this unit, students apply their programming knowledge and design skills to create meaningful, original projects. They'll solve real-world problems, personalize interactive tools, and present data-driven findings.

Objectives / Topics Covered	● Apply design thinking to develop accessible technology solutions. ● Use variables, operators, and conditionals in creative programs. ● Conduct an investigation and display results using data visualization.
Lessons	**Designing Solutions for Accessibility** ● Use design thinking to build a program that improves access or inclusion. **Code Tunes (2 day lesson)** ● Create a custom music player using variables, conditionals, and operators. **Inquiry Project: Data Bar Graph (2 day lesson)** ● Follow the inquiry process and display investigation results in a modified bar graph program.

Texas Computer Science 4th 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.
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.
All of these resources and more are found on the **Elementary Resources Page**.	