Texas Computer Science 1st Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Texas Computer Science 1st 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.

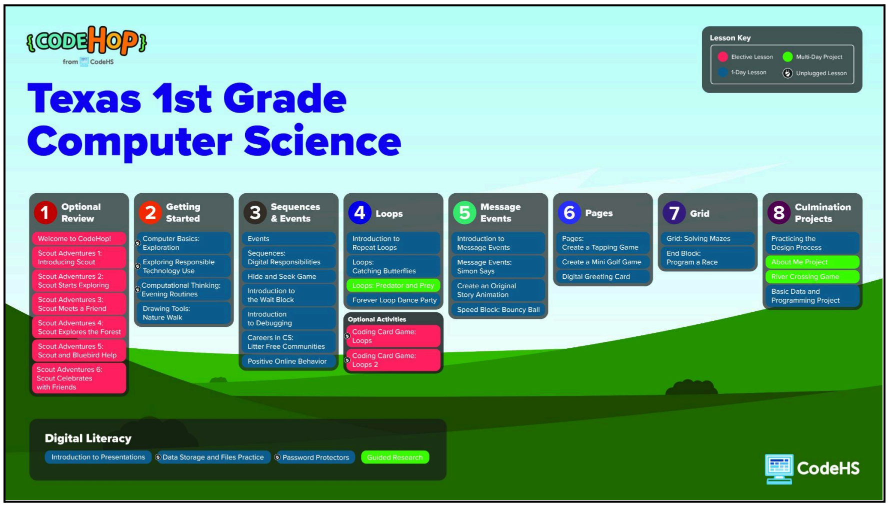
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts 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/26370/overview



Texas 1st Grade Computer Science

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

1 Optional Review
- Welcome to CodeHop!
- Scout Adventures 1: Introducing Scout
- Scout Adventures 2: Scout Starts Exploring
- Scout Adventures 3: Scout Meets a Friend
- Scout Adventures 4: Scout Explores the Forest
- Scout Adventures 5: Scout and Bluebird Help
- Scout Adventures 6: Scout Celebrates with Friends

2 Getting Started
- ✋ Computer Basics: Exploration
- ✋ Exploring Responsible Technology Use
- ✋ Computational Thinking: Evening Routines
- Drawing Tools: Nature Walk

3 Sequences & Events
- Events
- Sequences: Digital Responsibilities
- Hide and Seek Game
- Introduction to the Wait Block
- Introduction to Debugging
- Careers in CS: Litter Free Communities
- Positive Online Behavior

4 Loops
- Introduction to Repeat Loops
- Loops: Catching Butterflies
- Loops: Predator and Prey
- Forever Loop Dance Party

Optional Activities
- ✋ Coding Card Game: Loops
- ✋ Coding Card Game: Loops 2

5 Message Events
- Introduction to Message Events
- Message Events: Simon Says
- Create an Original Story Animation
- Speed Block: Bouncy Ball

6 Pages
- Pages: Create a Tapping Game
- Create a Mini Golf Game
- Digital Greeting Card

7 Grid
- Grid: Solving Mazes
- End Block: Program a Race

8 Culmination Projects
- Practicing the Design Process
- About Me Project
- River Crossing Game
- Basic Data and Programming Project

Digital Literacy
- Introduction to Presentations
- ✋ Data Storage and Files Practice
- ✋ Password Protectors
- Guided Research

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

Course Breakdown

Optional Review

This optional review unit is designed to support students who need more time exploring programming or who would benefit from additional practice before or after completing core lessons. The Scout Adventures lessons offer a sequential, story-based experience to reinforce key skills in a fun and engaging way.

Objectives / Topics Covered	● Review core programming skills including sequences, motion, events, and page transitions. ● Strengthen confidence using the programming interface through a story-based project. ● Provide flexible options for differentiation, reinforcement, or introductory practice.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Introductory lesson to help students log in and explore the Playground; can be used before a full lesson or on its own. **Scout Adventures 1-6** ● Follow Scout the Squirrel through a six-part interactive story that helps students build foundational programming skills.

Unit 1: Getting Started (4 lessons)

In this unit, students will explore the basics of how computers work and how to use them responsibly. They'll practice computational thinking through real-life routines and use creative tools to design a digital nature scene.

Objectives / Topics Covered	● Understand what a computer is and how it is used. ● Identify and categorize computer components as input, output, hardware, or software. ● Recognize safe and responsible technology use. ● Apply computational thinking strategies to everyday tasks.
Lessons	**Computer Basics: Exploration** ● Identify and categorize input, output, hardware, and software components of a computer. **Exploring Responsible Technology Use** ● Explain ways to use technology safely and responsibly. **Computational Thinking: Evening Routines** ● Break down everyday routines into steps while identifying patterns, sequences, and simplifications. **Drawing Tools: Nature Walk** ● Create a nature-walk scene using drawing tools to customize characters and backgrounds.

Unit 2: Sequences & Events (7 lessons)

In this unit, students explore how to use sequences and events to control the behavior of characters. They will build animated scenes, games, and interactive projects while learning how to respond to events, pause actions with wait blocks, and debug errors. The module also connects computer science to real-world problem-solving and digital responsibility.

Objectives / Topics Covered	● Understand and apply the concept of sequences in programming. ● Use event blocks to trigger actions in response to user interaction. ● Create interactive programs with visual effects, timing, and size changes. ● Identify and correct errors through debugging. ● Connect computer science skills to real-world issues and careers.

Lessons	**Events** ● Use event blocks like green flag, tap, and message to trigger actions in a program. **Sequences: Digital Responsibilities** ● Program a sequence of actions while exploring ways to stay safe and respectful online. **Hide and Seek Game** ● Build an interactive hide-and-seek game using the hide block and tap events. **Introduction to the Wait Block** ● Add pauses between actions using the wait block to better control timing in a program. **Introduction to Debugging** ● Find and fix bugs in a program where actions are out of order or not working as expected. **Careers in CS: Litter Free Communities** ● Explore how computer science can help solve environmental challenges and create a program to sort and clean up virtual litter. **Positive Online Behavior** ● Use programming to show examples of safe and unsafe choices when interacting online.

Optional Unplugged Loops Activities

This optional unplugged unit gives students a hands-on way to practice programming concepts without using devices. By working together to move Scout through a maze using coding cards, students reinforce sequencing and looping skills in a fun, collaborative setting.

Objectives / Topics Covered	● Practice creating and following sequences of instructions without a screen. ● Use loops to simplify repeated actions in a physical coding activity. ● Build teamwork and communication skills while applying coding logic.
Lessons	**Coding Card Game: Loops** ● Work together to guide Scout through a maze using cards that represent movement and repeat loops. **Coding Card Game: Loops 2** ● Tackle a new maze challenge by building efficient sequences with loops to help Scout reach the goal.

Unit 3: Loops (5 lessons)

In this unit, students will learn how to make characters repeat actions in their programs using loops. They'll use repeat and forever loops to build fun animations, games, and stories that include repeating movements and patterns. By the end of the module, students will be able to recognize when and how to use loops to make their code shorter and more efficient.

Objectives / Topics Covered	● Understand the concept of loops and how they control repetition in programs. ● Use repeat and forever loops to streamline sequences of actions. ● Combine loops with events and sequences to create interactive and dynamic programs. ● Build animations and games that demonstrate repeating patterns of behavior.
Lessons	**Introduction to Repeat Loops** ● Use repeat loops to make a character perform an action multiple times in a row. **Loops: Catching Butterflies** ● Build a butterfly-catching game using show, hide, and repeat blocks to animate the gameplay. **Loops: Predator and Prey (2 part lesson)**

	● Program two animal characters to interact using sequences, events, and repeat loops. **Forever Loop Dance Party** ● Create a dance sequence using the “repeat forever” loop to animate continuous movement.

Unit 4: Message Events (4 lessons)

Students will explore how characters can communicate in a program using message events. They’ll create interactive projects like a relay race, a Simon Says game, and an animated story, while learning to control timing and speed.

Objectives / Topics Covered	● Use message events to trigger actions between characters. ● Program one-to-one and one-to-many communication in animations. ● Create interactive stories using messaging and sequencing. ● Use speed blocks to control how fast characters move.
Lessons	**Introduction to Message Events** ● Create a relay race where characters take turns moving by sending and receiving messages. **Message Events: Simon Says** ● Use message events to have one character control the actions of several others in a game of Simon Says. **Create an Original Story Animation** ● Design and animate an original story that includes sequences, events, and messages. **Speed Block: Bouncy Ball** ● Use speed and message blocks to make a character move at different speeds in a bouncing animation.

Unit 5: Pages (3 lessons)

In this unit, students will learn how to use the “go to page” block to create multi-page projects. They’ll design games and animations that move from one scene to another, building on their knowledge of events and loops while adding creativity through digital storytelling and design.

Objectives / Topics Covered	● Use the “go to page” block to move between scenes in a project. ● Combine pages, loops, and message events to create interactive and animated experiences. ● Apply creativity to design digital games and greeting cards with smooth transitions.
Lessons	**Pages: Create a Tapping Game** ● Design an interactive game that switches between pages when a character is tapped. **Create a Mini Golf Game** ● Use message events and loops to program a mini golf game that moves across multiple pages. **Digital Greeting Card** ● Build a digital greeting card that uses loops and events to animate a message across different pages.

Unit 6: Grid (4 lessons)

In this unit, students will learn how to use the grid to help plan and control character movement. They’ll design mazes, create animations, and explore how to end actions using the “end” block, all while practicing spatial reasoning and precise programming.

Objectives / Topics Covered	● Use the grid to guide character movement accurately. ● Plan paths and animations using grid positions. ● Apply the “end” block to stop actions at the right time in a program. ● Build projects that combine creativity with structured movement.
Lessons	**Grid: Solving Mazes** ● Design a maze and use grid-based movement to guide a character from start to finish. **End Block: Program a Race** ● Program a race between characters and use the “end” block to stop the animation at the finish line.

Unit 7: Culmination Projects (6 lessons)

In this unit, students will apply everything they’ve learned throughout the course to create original projects that showcase their coding skills and creativity. From sharing personal interests to building games and displaying data, these culminating activities give students a chance to express themselves and demonstrate mastery of key programming concepts.

Objectives / Topics Covered	● Combine sequences, events, motion, and visual elements in creative ways. ● Apply concepts like speed, data, and interactivity to make programs more engaging. ● Reflect personal interests, challenges, and learning through coding.
Lessons	**Practicing the Design Process** ● Use the design process to plan, create, and revise a project that models a solution to a simple real-world problem, with some teacher support. **About Me Project (2 part lesson)** ● Design a program that shares personal characteristics, interests, and favorite things. **River Crossing Game (2 part lesson)** ● Create a game with obstacles and use speed blocks to adjust difficulty as characters cross a river. **Basic Data and Programming** ● Collect simple data and create a program to visually present the results through animation or scenes.

Unit 8: Digital Literacy (5 lessons)

Students will develop foundational digital skills by learning how to create presentations, protect personal information, understand file storage, and conduct guided research.

Objectives / Topics Covered	● Create a simple presentation using digital tools. ● Understand how data is stored and organized as files. ● Learn strategies to keep usernames and passwords secure. ● Research a topic and share findings using a digital program.
Lessons	**Introduction to Presentations** ● Make a simple slide presentation about favorite things using presentation software. **Data Storage and Files Practice** ● Learn how computers store information and explore how files hold data. **Password Protectors** ● Understand the role of usernames and passwords and how to keep them safe. **Guided Research (2 part lesson)** ● Find information on a topic and create a visual program to present research.

Texas Computer Science 1st 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**.	