Texas Computer Science 3rd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Texas Computer Science 3rd Grade Course** introduces students to foundational programming concepts through **Scratch**, 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.

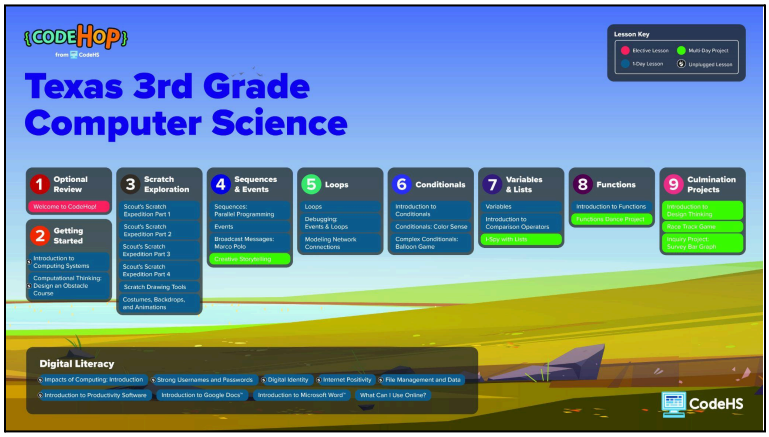
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging projects to showcase student understanding.

The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. Teachers can adapt the content to fit their schedule and instructional needs. The concepts taught in this course spiral across grade levels, ensuring that students can revisit and build upon their understanding year after year, even if all lessons are not completed within a single year. The course includes a total of 36 **lessons**, with each lesson approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week. Optional digital literacy lessons are also available to complement the programming curriculum with non-programming computer and technology skills.

Programming Environment: Students will write and run programs in **Scratch** embedded and saved in the CodeHS 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/26375/overview



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

Course Breakdown

Optional Review

This brief review lesson helps students get comfortable logging in and navigating the Playground.

Objectives / Topics Covered	● Log in and explore basic site features to prepare for upcoming lessons.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Practice logging in and exploring the Playground before starting a full lesson.

Unit 1: Getting Started (2 lessons)

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

Objectives / Topics Covered	● Identify hardware and software components of a computing system. ● Troubleshoot simple computing issues. ● Use computational thinking to design and plan a solution.
Lessons	**Introduction to Computing Systems** ● Learn the parts of a computing system and how to solve basic hardware and software problems. **Computational Thinking: Design a School** ● Apply computational thinking skills to design and map out a school layout.

Unit 2: Scratch Exploration (6 lessons)

Students will follow a story-driven sequence to build skills in Scratch, learning how to animate characters, use events and loops, and customize visuals through drawing tools, costumes, and backdrops.

Objectives / Topics Covered	● Use Scratch to sequence movements and dialogue. ● Animate stories using motion, looks, events, and loops. ● Create and customize sprites, costumes, and backdrops. ● Build interactive animations with visual storytelling.
Lessons	**Scout's Scratch Expedition Part 1** ● Program a sprite to move and talk using basic Scratch commands. **Scout's Scratch Expedition Part 2** ● Add sprites and create a sequence to animate part of the story. **Scout's Scratch Expedition Part 3** ● Animate a Scout story using loops, events, and motion blocks. **Scout's Scratch Expedition Part 4** ● Continue the animated story using events, looks, and motion for more interaction. **Scratch Drawing Tools** ● Customize sprites and backdrops using Scratch's drawing tools. **Costumes, Backdrops, and Animations** ● Animate sprites and make interactive scenes with costumes and changing backdrops.

Unit 3: Sequences & Events (4 lessons)

Students will learn how to build programs that run multiple sequences at once, use events to trigger actions, and create interactive stories by combining these concepts creatively.

Objectives / Topics Covered	● Create programs using parallel sequences. ● Use events and broadcast messages to control sprite interactions. ● Combine sequences and events to tell animated stories.
Lessons	**Sequences: Parallel Programming** ● Create a program where multiple sequences run at the same time. **Events** ● Use event blocks to start actions when keys are pressed or sprites are clicked. **Broadcast Messages: Marco Polo** ● Send and receive messages between sprites to coordinate actions. **Creative Storytelling (2 day lesson)** ● Plan and animate a story using sequences and event-based interactions.

Unit 4: Loops (3 lessons)

Students will explore how loops repeat instructions to simplify code, practice debugging with events and loops, and model how data travels through a network using programming.

Objectives / Topics Covered	● Understand and use loops to repeat actions in Scratch. ● Debug programs involving events and loops. ● Model how data moves through a network using sprite interactions.
Lessons	**Loops** ● Learn how loops repeat instructions and use them in a Scratch project. **Debugging: Events and Loops** ● Break down and fix a program with issues involving loops and events. **Modeling Network Connections** ● Create a program that shows how messages or data move between devices on a network.

Unit 5: Conditionals (3 lessons)

Students will learn how conditionals are used in programming to make decisions. They'll explore simple and complex conditionals to create interactive, logic-based programs.

Objectives / Topics Covered	● Understand and use basic if/then conditionals. ● Apply conditionals to control program behavior. ● Use if/then/else structures to create more complex logic.
Lessons	**Introduction to Conditionals** ● Learn what conditionals are and use if/then blocks in a program. **Conditionals: Color Sense** ● Use conditionals to trigger actions when specific colors are touched in a program. **Complex Conditionals: Balloon Game** ● Create a game using if/then/else blocks to respond to different outcomes.

Unit 6: Variables & Lists (4 lessons)

Students will explore how variables and lists are used to store and manage information in a program. They'll apply comparison operators and build interactive projects that respond to data.

Objectives / Topics Covered	● Understand and use variables to store changing values. ● Use comparison operators in conditional statements. ● Create programs that use lists and variables to track and respond to user input.

Lessons	**Variables** ● Learn what a variable is and how to create and change its value in Scratch. **Introduction to Comparison Operators** ● Use comparison operators with numbers and variables in if/else logic. **I-Spy with Lists (2 day lesson)** ● Build an interactive I-Spy game using lists and variables to track items and responses.

Unit 7: Functions (3 lessons)

Students will learn how to simplify and organize their code by creating functions. They'll apply this concept in creative ways, including programming a dance routine.

Objectives / Topics Covered	● Understand what a function is and how it's used in programming. ● Create and call functions to organize repeated actions. ● Use functions to structure a sequence of events in a project.
Lessons	**Introduction to Functions** ● Learn how to define and use functions to simplify and organize code. **Functions Dance Project (2 day lesson)** ● Create a dance program by writing functions for dance moves and sequencing them with music.

Unit 8: Culmination Project (5 lessons)

Students will apply their coding skills through creative, open-ended projects. Using the design thinking process, they'll plan, build, and improve programs that combine conditionals, variables, lists, and data.

Objectives / Topics Covered	● Apply design thinking to create and refine programs. ● Use conditionals, variables, and lists in a custom racing game. ● Follow the inquiry process and visualize data in a modified program.
Lessons	**Introduction to Design Thinking** ● Use the design thinking process to brainstorm, build, and improve a program. **Race Track Game (2 day lesson)** ● Design a race car and track, then program a racing game that includes scorekeeping with conditionals, variables, and lists. **Inquiry Project: Survey Bar Graph (2 day lesson)** ● Follow the inquiry process to investigate a topic and modify a Scratch program to display survey results using a bar graph.

Unit 9: Digital Literacy (7 lessons)

Students will explore how computing impacts communities, learn how to stay safe and positive online, and build practical skills using productivity tools. They'll also practice safe online research and responsible digital behavior.

Objectives / Topics Covered	● Understand the societal impact of computing and digital footprints. ● Create strong usernames and passwords for online safety. ● Promote positive behavior and responsibility online. ● Compare and use different types of productivity software. ● Conduct safe, cited online research.
Lessons	**Impacts of Computing: Introduction** ● Explore how computing has changed communities and evolved over time. **Strong Usernames and Passwords** ● Create secure login information and understand how it protects personal data.

	Digital Identity ● Connect real-world identity to online behavior and build a positive digital footprint. **Internet Positivity** ● Learn how online actions affect others and create a responsible behavior code. **File Management and Data** ● Explore how different types of digital data take up varying amounts of space and where that data can be stored. **Introduction to Google Docs™/Microsoft Word™** ● Learn to edit and format text using a word processing application. **What Can I Use Online?** ● Practice safe online research, use keywords effectively, and cite sources properly.

Texas Computer Science 3rd 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 Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch 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**.	