



CodeHS

Texas Computer Science 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Texas Computer Science 5th 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/26377/overview>



Course Breakdown

Optional Review

This optional review unit helps students refresh key computer science skills and prepare for the year ahead. Through Scratch-based activities, students will review vocabulary, practice programming on the coordinate plane, and customize their projects using drawing tools.

Objectives / Topics Covered	<ul style="list-style-type: none">• Log in and navigate the Playground.• Review core computer science vocabulary and Scratch basics.• Practice using the coordinate plane for animation.• Customize sprites and scenes with Scratch's drawing tools.
Lessons	<p>Welcome to CodeHS!</p> <ul style="list-style-type: none">• Practice logging in and exploring the Playground before starting a full lesson. <p>Introduction to Computer Science and Scratch</p> <ul style="list-style-type: none">• Review basic computer science vocabulary and create a simple Scratch program. <p>The Coordinate Plane</p> <ul style="list-style-type: none">• Use the coordinate plane in Scratch to design an open-ended animation. <p>Drawing Tools: Sea Creatures</p> <ul style="list-style-type: none">• Use built-in Scratch drawing tools to create and personalize custom sprites and backgrounds.

Unit 1: Getting Started (2 lessons)

Students will explore how computing systems work and apply computational thinking to design creative, real-world solutions.

Objectives / Topics Covered	<ul style="list-style-type: none">• Identify computing system parts and troubleshoot basic issues.• Use computational thinking to plan and solve design challenges.
Lessons	<p>Practicing Computing Systems</p> <ul style="list-style-type: none">• Identify hardware and software components and recognize simple problems in a computing system. <p>Computational Thinking: Design a Neighborhood</p> <ul style="list-style-type: none">• Apply computational thinking to plan and design a neighborhood layout.

Unit 2: Sequences, Events & Loops (4 lessons)

Students will deepen their understanding of core programming concepts by using events, refining algorithms, debugging programs, and creating loop-based graphics.

Objectives / Topics Covered	<ul style="list-style-type: none">• Use events to trigger actions in a program.• Compare and improve algorithms for efficiency.• Debug by breaking down code into manageable parts.• Create looping patterns using Scratch's pen tool.
Lessons	<p>Events: Traveling with Scout</p> <ul style="list-style-type: none">• Use event blocks to trigger character actions in a travel-themed program. <p>Compare and Refine Algorithms</p> <ul style="list-style-type: none">• Create multiple solutions to a problem, then compare and improve them to find the most efficient algorithm. <p>Debugging: Make a Pizza</p> <ul style="list-style-type: none">• Break down and fix a pizza-making program to ensure it runs correctly.

	Creating Turtle Graphics <ul style="list-style-type: none"> Use loops and the pen tool in Scratch to draw repeated, artistic patterns.
--	--

Unit 3: Loops (2 lessons)

In this unit, students apply loops to create repeated designs using Scratch's pen tool, while also learning the importance of giving credit when using or remixing digital content. They'll combine creativity with responsible digital citizenship and efficient coding practices.

Objectives / Topics Covered	<ul style="list-style-type: none"> Use loops to create repeating patterns and graphics with Scratch's pen tool. Understand and apply the concept of attribution when using or sharing digital work. Create original or remixed projects while practicing ethical digital behavior.
Lessons	Creating Turtle Graphics <ul style="list-style-type: none"> Use repeat loops and the pen tool in Scratch to draw looping patterns similar to turtle graphics. Giving Credit Through Attributions <ul style="list-style-type: none"> Learn how to appropriately credit original creators when remixing programs or using shared images online.

Unit 4: Conditionals & Operators (4 lessons)

Students will explore how to make decisions in their programs using conditionals and operators. They'll plan and build interactive projects that respond to different outcomes.

Objectives / Topics Covered	<ul style="list-style-type: none"> Use conditionals to control program flow. Decompose and plan the steps for a multi-part program. Apply operators and variables in programs to create random or logical outcomes.
Lessons	Conditionals: Mazes <ul style="list-style-type: none"> Create a maze program that uses conditionals to navigate paths. Plan a Quest (2 day lesson) <ul style="list-style-type: none"> Break down and plan the steps needed to build a quest-style program. Operators: Coin Flip <ul style="list-style-type: none"> Use operators and variables to simulate a random coin flip in a program.

Unit 5: Variables & Lists (3 lessons)

In this unit, students deepen their understanding of how variables and lists can be used to manage and manipulate information in Scratch programs. Through creative projects like dance animations and shopping simulations, they apply variables, lists, and operators to create more dynamic and interactive experiences.

Objectives / Topics Covered	<ul style="list-style-type: none"> Use variables to store and control values like speed and pitch. Create and manage lists to store multiple related items. Combine variables, lists, and operators to power interactive simulations. Build projects that reflect real-world problem-solving using data structures.
Lessons	Variables in Dance <ul style="list-style-type: none"> Use variables to change dance speed and pitch, creating a dynamic and musical animation. Lists: Shopping with Scout (2 day lesson) <ul style="list-style-type: none"> Build a shopping simulator that uses lists to track items and variables and operators to calculate prices.

Unit 6: Clones (3 lessons)

Students will learn how to use clones to create multiple copies of sprites in a program. They'll apply this concept to build dynamic games like a throwing game and an endless runner.

Objectives / Topics Covered	<ul style="list-style-type: none">• Use clones to create repeated elements in a program.• Build interactive games using clones.• Understand the benefits of using clones in game design.
Lessons	Clones: Throwing Acorns Game <ul style="list-style-type: none">• Create a game where acorns are cloned and thrown toward targets. Clones in Games (2 day lesson) <ul style="list-style-type: none">• Program an endless runner game using clones and explain their usefulness for repeating actions or objects.

Unit 7: Culmination Projects (11 lessons)

Students will synthesize their programming and design skills to create original projects, including games, apps, and data visualizations. They'll apply design thinking, game design principles, and the inquiry process to guide their work.

Objectives / Topics Covered	<ul style="list-style-type: none">• Apply programming concepts to build original games.• Use design thinking to develop app ideas that meet user needs.• Conduct investigations and present findings using a line graph.
Lessons	Pinball Game Project (3 day lesson) <ul style="list-style-type: none">• Design and build a pinball game using key game mechanics and Scratch features. Game Design Project (3 day lesson) <ul style="list-style-type: none">• Create a custom game using loops, conditionals, variables, and other programming concepts. Design an App (3 day lesson) <ul style="list-style-type: none">• Use design thinking to plan an app that solves a real-world problem. Inquiry Project: Line Graph (2 day lesson) <ul style="list-style-type: none">• Investigate a question and modify a program to present the results as a line graph.

Unit 8: Digital Literacy (9 lessons)

Students will explore how computing innovations, online collaboration, networks, and cybersecurity shape their digital lives. They'll build skills in productivity tools and learn how to research, organize, and present information responsibly.

Objectives / Topics Covered	<ul style="list-style-type: none">• Understand how computing innovations and networks impact the world.• Practice safe, responsible, and secure online behavior.• Use collaboration and communication tools effectively.• Organize, visualize, and present data using productivity software.• Conduct and present research using digital tools.
Lessons	Impacts of Computing: Innovation <ul style="list-style-type: none">• Explore how computing innovations have changed the way we live and work. Networks and Protocols <ul style="list-style-type: none">• Learn how data is transferred through networks and compare WiFi, wired, and cellular connections. Standing Up to Cyberbullying <ul style="list-style-type: none">• Discuss how to recognize, respond to, and report cyberbullying. Online Collaboration <ul style="list-style-type: none">• Explore how people collaborate both online and offline, and how working with

	<p>others brings diverse perspectives.</p> <p>Scout's Cybersecurity Adventure: Part 2</p> <ul style="list-style-type: none"> Practice secure online habits and learn how tools and technologies protect personal information. <p>File Management & Data Storage</p> <ul style="list-style-type: none"> Understand how different types of data are stored and managed on devices. <p>Introduction to Microsoft Excel™/Google Sheets™</p> <ul style="list-style-type: none"> Enter, organize, and visualize data in a spreadsheet. <p>Productivity Software Inquiry Project</p> <ul style="list-style-type: none"> Use documents, spreadsheets, and slides to organize research and share findings with others.
--	---

Texas Computer Science 5th 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 .	