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 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.

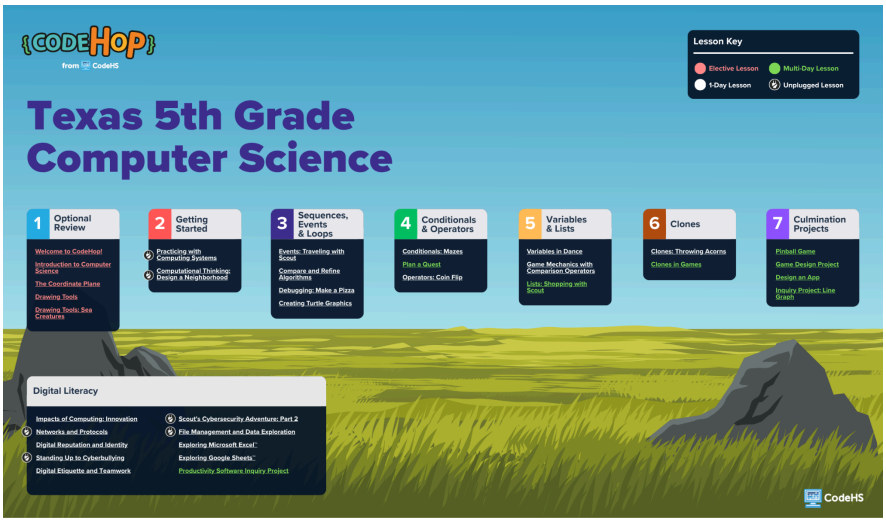
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts 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/26377/overview



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

Course Breakdown

Optional Review

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

Objectives / Topics Covered	● Log in and navigate the Playground. ● Review core computer science vocabulary and programming basics. ● Practice using the coordinate plane for animation. ● Customize sprites and scenes with drawing tools.
Lessons	**Welcome to CodeHop!** ● 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. **Drawing Tools: Sea Creatures** ● Use 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	● Identify computing system parts and troubleshoot basic issues. ● Use computational thinking to plan and solve design challenges.
Lessons	**Practicing with Computing Systems (5.9.A, 5.12.E)** ● Identify hardware and software components and recognize simple problems in a computing system. **Design a Neighborhood (5.1.A, 5.1.B)** ● 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	● 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 the pen tool.
Lessons	**Events: Traveling with Scout** ● Use event blocks to trigger character actions in a travel-themed program. **Compare and Refine Algorithms (5.1.D)** ● Create multiple solutions to a problem, then compare and improve them to find the most efficient algorithm. **Debugging: Make a Pizza** ● Break down and fix a pizza-making program to ensure it runs correctly. **Creating Turtle Graphics**

	● Use loops and the pen tool to draw repeated, artistic patterns.

Unit 3: 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	● 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** ● Create a maze program that uses conditionals to navigate paths. **Plan a Quest (2 classes)** ● Break down and plan the steps needed to build a quest-style program. **Operators: Coin Flip** ● Use operators and variables to simulate a random coin flip in a program.

Unit 4: 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 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	● 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 (5.2.A)** ● Use variables to change dance speed and pitch, creating a dynamic and musical animation. **Lists: Shopping with Scout (2 classes)** ● Build a shopping simulator that uses lists to track items and variables and operators to calculate prices.

Unit 5: 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	● 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** ● Create a game where acorns are cloned and thrown toward targets. **Clones in Games (2 classes)** ● Program an endless runner game using clones and explain their usefulness for repeating actions or objects.

Unit 6: 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	● 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 classes)** ● Design and build a pinball game using key game mechanics and programming features. **Game Design Project (3 classes)** ● Create a custom game using loops, conditionals, variables, and other programming concepts. **Design an App (3 classes | 5.1.C, 5.2.B, 5.3.A, 5.3.B)** ● Use design thinking to plan an app that solves a real-world problem. **Inquiry Project: Line Graph (2 classes | 5.5.A, 5.6.A, 5.7.A)** ● Investigate a question and modify a program to present the results as a line graph.

Unit 7: 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	● 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 (5.4.A)** ● Explore how computing innovations have changed the way we live and work. **Networks and Protocols (5.12.A)** ● Learn how data is transferred through networks and compare WiFi, wired, and cellular connections. **Digital Reputation and Identity (5.8.A)** ● Analyze how online actions affect digital reputation and explain how to protect personal data and digital security. **Standing Up to Cyberbullying (5.10.C)** ● Discuss how to recognize, respond to, and report cyberbullying. **Digital Etiquette and Teamwork (5.8.B, 5.8.C)** ● Demonstrate proper digital etiquette when communicating in an online community. **Scout's Cybersecurity Adventure: Part 2 (5.10.A, 5.10.B)** ● Practice secure online habits and learn how tools and technologies protect personal information. **File Management and Data Exploration (5.11.A, 5.12.B)** ● Understand how different types of data are stored and managed on devices. **Exploring Microsoft Excel™/Google Sheets™ (5.5.A)** ● Enter, organize, and visualize data in a spreadsheet. **Productivity Software Inquiry Project (5.11.B, 5.12.D)** ● 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 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**.	