1st Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The 1st Grade Computer Science 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.

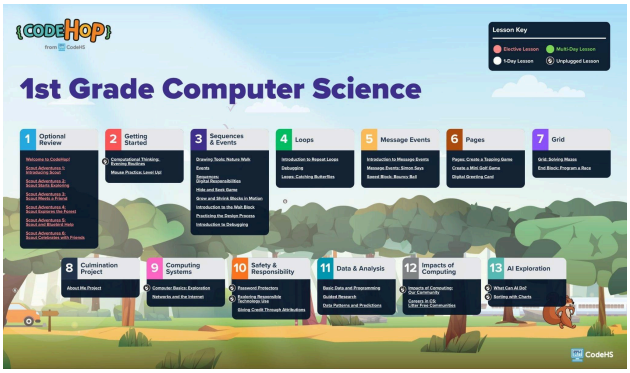
The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet CSTA computer science standards and are clearly identified within the syllabus.

Standards Alignment Note: Lessons that list standards in parentheses next to the title are required to fully meet CSTA computer science standards. Lessons without a standards tag support spiral review, practice, or enrichment.

Programming Environment: Students will write and run programs that are saved in students' accounts. 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/28413/overview



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

Course Breakdown

Unit 1: Optional Review (7 lessons)

These optional lessons are available for teachers to use as review or warm-up activities at the start of 1st grade. They are not counted toward the 36-lesson course total and can be used selectively based on student readiness.

Objectives / Topics Covered	● Review foundational platform navigation and mouse skills. ● Revisit block-based programming concepts from Kindergarten.
Lessons	**Welcome to CodeHop!** ● Learn how to log in and use the CodeHop Playground. This short introductory lesson can be used on its own or right before a full lesson. **Scout Adventures 1: Introducing Scout** ● Explore the programming interface and add characters. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a page to a program. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters in a program. **Scout Adventures 4: Scout Explores the Forest** ● Explore and use motion blocks to move characters around the stage in a program. **Scout Adventures 5: Scout and Bluebird Help** ● Build a sequence of motion blocks to move characters around the stage to collect objects. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene by adding characters, pages, backgrounds, and sequences of motion blocks with events.

Unit 2: Getting Started (2 lessons)

In this module, students build on foundational technology skills from Kindergarten. Through an unplugged computational thinking activity and leveled mouse practice, students strengthen their readiness for 1st grade programming.

Objectives / Topics Covered	● Apply computational thinking skills to break down real-world routines into algorithms. ● Demonstrate advanced mouse skills including precision clicking, dragging, and scrolling.
Lessons	**Computational Thinking: Evening Routines (Unplugged)** (E1-ALG-PS-01) ● Use computational thinking concepts to identify patterns, break down tasks, sequence steps, and simplify processes in their evening routines. **Mouse Practice: Level Up!** ● This library of mouse activities should be completed after the lesson, Mouse Practice.

Unit 3: Sequences & Events (8 lessons)

In this module, students deepen their understanding of sequences and events in programming. Through a variety of projects, including drawing, games, and animations, students learn to use event blocks, control timing, apply the design process, and debug their programs.

Objectives / Topics Covered	● Use event blocks to trigger actions and create interactive programs. ● Apply sequencing skills to build animations and interactive scenes. ● Explore drawing tools and size-change blocks for creative programming. ● Use wait blocks to control timing in a program. ● Apply the design process to plan and build a project. ● Identify and fix errors using debugging strategies.
Lessons	**Drawing Tools: Nature Walk** ● Use drawing tools to create a nature-walk scene. **Events** (E1-PRO-RD-06) ● Explain what an event is in programming and use multiple event blocks in a program. **Sequences: Digital Responsibilities** (E1-PRO-PD-04) ● Use sequences to program characters and explain how to be responsible online. **Hide and Seek Game** ● Use the hide block to program an interactive game of hide-and-seek. **Grow and Shrink Blocks in Motion** ● Create a program using motion blocks and grow and shrink blocks to change the size of characters. **Introduction to the Wait Block** ● Use "wait" blocks to cause characters to pause in a program. **Practicing the Design Process** (E1-ALG-IM-03) ● Use the design process to plan, create, and revise a programming project that models a solution to a simple real-world problem. **Introduction to Debugging** (E1-PRO-TR-07) ● Find and correct bugs in sequences.

Unit 4: Loops (3 lessons)

In this module, students learn how repeat loops automate sequences in programs, making code more efficient. Students also continue developing debugging skills by applying them to loop-based programs.

Objectives / Topics Covered	● Explain how repeat loops automate a sequence of instructions. ● Use loop blocks to simplify programs by replacing repeated sequences. ● Debug programs that use loops and sequences.
Lessons	**Introduction to Repeat Loops** ● Use repeat loops to run a section of code multiple times. **Debugging** (E1-PRO-TR-07) ● Describe what bugs are and find and correct bugs in sequences. **Loops: Catching Butterflies** ● Use "show" and "hide" blocks and loops to create a butterfly-catching game.

Unit 5: Message Events (3 lessons)

In this module, students explore message blocks, which allow characters to communicate and coordinate actions across a program. Students apply message events to build interactive games and animations.

Objectives / Topics Covered	● Explain how message blocks allow sprites to send and receive signals. ● Use message events to coordinate actions between multiple characters. ● Control character movement and speed to create dynamic programs.

Lessons	**Introduction to Message Events** (E1-PRO-RD-06) ● Program a relay race that uses messages to cause characters to interact. **Message Events: Simon Says** (E1-PRO-RD-06) ● Use message events to make one character communicate to many characters in a program. **Speed Block: Bouncy Ball** ● Use speed blocks and messages to program a character to move at different speeds.

Unit 6: Pages (3 lessons)

In this module, students create multi-page programs, applying page navigation to design interactive games and projects. Students combine events, sequences, and page blocks to build more complex programming experiences.

Objectives / Topics Covered	● Create programs with multiple pages using navigation blocks. ● Apply programming skills to design interactive games spanning multiple pages. ● Combine events, pages, and creative elements to build a digital project.
Lessons	**Pages: Create a Tapping Game** ● Create a game that moves from one page to the next using "go to page" blocks. **Create a Mini Golf Game** ● Use messages and loops to create a mini golf game. **Digital Greeting Card** ● Create a digital greeting card with events and loops.

Unit 7: Grid (2 lessons)

In this module, students use grid-based programming to navigate characters through structured environments. Students apply directional commands and control blocks to solve maze challenges and create a programmed race.

Objectives / Topics Covered	● Use directional commands to navigate a character through a grid-based environment. ● Apply end blocks to control when a program concludes. ● Solve problems using structured, step-by-step programming logic.
Lessons	**Grid: Solving Mazes** (E1-PRO-PD-04) ● Design a maze and use the grid to program a character to move through the maze. **End Block: Program a Race** ● Complete a project to program a race and use the "end" block in an animation.

Unit 8: Culmination Project (1 lesson)

In this module, students apply all programming skills learned throughout the year to create a multi-page 'About Me' project. This capstone project allows students to showcase their creativity and demonstrate mastery of CodeHop concepts.

Objectives / Topics Covered	● Apply sequencing, events, loops, pages, and design skills in a single project. ● Create a multi-page program that communicates personal information and interests. ● Reflect on the design process and evaluate their own programming work.
Lessons	**About Me Project** (2 classes | E1-PRO-PD-04) ● Plan a program to describe their characteristics and interests.

Unit 9: Computing Systems (2 lessons)

In this module, students explore how computers work and how devices communicate through networks. Students identify hardware and software components and learn how the Internet enables people to share information.

Objectives / Topics Covered	● Identify hardware and software components and explain their functions. ● Distinguish between input and output devices with real-world examples. ● Explain what a network is and how the Internet transmits information.
Lessons	**Computer Basics: Exploration (Unplugged)** (E1-SYS-HW-11) ● Learn what a computer is, how we use it, and what to do when it doesn't work. They will be able to identify input, output, hardware, and software. **Networks and the Internet** (E1-SOC-CE-16) ● Explain what a network is and how people communicate over networks and the Internet. They will model how messages are communicated using the Internet.

Unit 10: Safety & Responsibility (3 lessons)

In this module, students explore online safety, responsible technology use, and digital ethics. Through unplugged activities and online projects, students learn how to protect their information and give appropriate credit for others' work.

Objectives / Topics Covered	● Create and evaluate strong passwords to protect personal accounts. ● Demonstrate responsible and respectful technology use. ● Give credit to others by using proper attributions in digital projects.
Lessons	**Password Protectors (Unplugged)** (E1-SYS-SE-12) ● Understand the importance of usernames and passwords and demonstrate strategies to keep login information safe. **Exploring Responsible Technology Use (Unplugged)** (E1-SYS-IM-13) ● Explain ways to use technology safely and responsibly. **Giving Credit Through Attributions** ● Explain why giving credit for another person's ideas and creations is important. They will reuse parts of a story to make a new story and give attribution to the original creator.

Unit 11: Data & Analysis (3 lessons)

In this module, students collect, organize, and analyze data while connecting data literacy to programming. Students conduct guided research and use programs to present their findings, learning how to interpret data and make predictions.

Objectives / Topics Covered	● Collect and organize data and create a program to present it visually. ● Use reliable sources to research a topic and communicate findings through programming. ● Identify patterns in data and use them to make predictions.
Lessons	**Basic Data and Programming** (E1-DAT-DC-08, E1-DAT-DI-09) ● Collect data, organize it into categories, and create a program to present the data visually, connecting data literacy with coding skills. **Guided Research** (2 classes) ● Use reliable online sources to research a chosen topic and create a multi-page program to communicate findings visually — completed over two class periods. **Data Patterns and Predictions** (E1-PRO-VD-05, E1-DAT-IM-10) ● Analyze data sets to identify patterns and use those patterns to make predictions, applying computational thinking to real-world information.

Unit 12: Impacts of Computing (2 lessons)

In this module, students examine how computing shapes communities and careers. Through unplugged activities and programming projects, students explore real-world applications of computer science and its positive impact on the world around them.

Objectives / Topics Covered	• Describe how computing and technology affect daily life and communities. • Explain how computer science careers contribute to solving real-world problems.
Lessons	**Impacts of Computing: Our Community (Unplugged)** (E1-SOC-HI-14, E1-SOC-CE-16) • Identify examples of computing devices in their daily lives and explain how technology has changed the way people live, work, and play. **Careers in CS: Litter Free Communities** (E1-SOC-CE-16) • Explain how computer science can help solve community problems and create a program to collect and sort litter.

Unit 13: AI Exploration (2 lessons)

In this module, students take their first steps into artificial intelligence. Through unplugged activities, students explore what AI can and cannot do, and learn how AI systems sort and classify information using patterns in data.

Objectives / Topics Covered	• Identify examples of what AI can and cannot do in everyday life. • Explain how AI systems use patterns in data to sort and classify information.
Lessons	**What Can AI Do? (Unplugged)** (E1-SOC-HU-15) • Identify tools that use AI, explain that AI uses data to learn and make decisions, and compare tasks that are better suited for humans versus AI. **Sorting with Charts (Unplugged)** (E1-ALG-ML-02) • Recognize patterns, use a decision tree to classify objects, and create a chart to show how patterns help computers make decisions.

K-1 Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to their new computer science curriculum.
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.
These resources and more are found on the **CodeHop Resources Page**.	