Kindergarten Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Kindergarten Computer Science Course Syllabus 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.

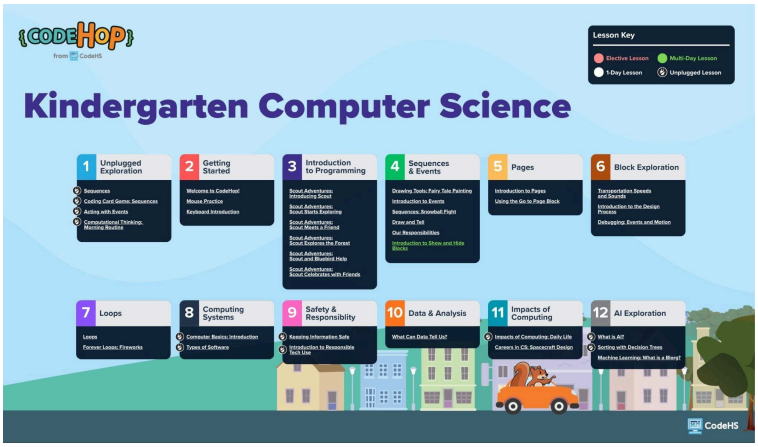
The course includes 36 lessons, each approximately 30 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet CSTA computer science standards and are clearly identified within the syllabus.

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/28412/overview



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

Course Breakdown

Unit 1: Unplugged Exploration (4 lessons)

In this module, students explore foundational computer science concepts away from the screen. Through hands-on, movement-based activities, students learn about sequences, algorithms, and events, building a concrete understanding of how computers follow instructions before transitioning to digital coding.

Objectives / Topics Covered	● Define a sequence as a series of steps in order. ● Describe an algorithm as a set of instructions for completing a task. ● Act out events and responses to understand cause-and-effect in programming. ● Apply computational thinking to real-world scenarios like a morning routine.
Lessons	**Sequences (Unplugged)** (EK-PRO-PD-04, EK-ALG-PS-01, EK-ALG-IM-03) ● Create a sequence of step-by-step instructions for a dance. **Coding Card Game: Sequences (Unplugged)** (EK-PRO-PD-04) ● Work together to create a sequence of instructions to move Scout through a maze. **Acting with Events (Unplugged)** ● Act out how an event can trigger an action. **Computational Thinking: Morning Routine (Unplugged)** (EK-ALG-PS-01) ● Use computational thinking concepts to identify patterns, break down tasks, sequence steps, and simplify processes in their morning routines.

Unit 2: Getting Started (3 lessons)

In this module, students are introduced to the CodeHop programming environment and develop essential foundational skills for using a computer. Students learn to log in, navigate the platform, and build mouse and keyboard confidence needed for all future lessons.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground independently. ● Demonstrate basic mouse skills such as clicking, dragging, and pointing. ● Use letters, numbers, and essential keyboard functions.
Lessons	**Welcome to CodeHop! (optional)** ● 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. **Mouse Practice** ● Demonstrate mouse skills by dragging and clicking with the mouse in multiple games. **Keyboard Introduction** ● Use the letters, numbers, and basic functions of the keyboard effectively.

Unit 3: Introduction to Programming (6 lessons)

In this module, students are introduced to block-based programming through the Scout Adventures series. Following Scout the dog and friends, students learn to sequence commands, use motion blocks, and build simple programs while developing a love of coding through story-driven lessons.

Objectives / Topics Covered	● Navigate the CodeHop programming environment to create and run programs. ● Sequence blocks to control a character's movement and actions. ● Create programs featuring characters, backgrounds, and simple interactions. ● Demonstrate an understanding of sequences and events through programming.
Lessons	**Scout Adventures: Introducing Scout** ● Explore the programming interface and add characters. **Scout Adventures: Scout Starts Exploring**

	● Add backgrounds and a page to a program. **Scout Adventures: Scout Meets a Friend** ● Delete and modify characters in a program. **Scout Adventures: Scout Explores the Forest** ● Explore and use motion blocks to move characters around the stage in a program. **Scout Adventures: Scout and Bluebird Help** (EK-PRO-PD-04) ● Build a sequence of motion blocks to move characters around the stage to collect objects. **Scout Adventures: Scout Celebrates with Friends** (EK-PRO-PD-04) ● Create a celebration scene by adding characters, pages, backgrounds, and sequences of motion blocks with events.

Unit 4: Sequences & Events (6 lessons)

In this module, students expand their programming skills by exploring sequences, events, and interactive features. Students create projects involving animation, storytelling, and drawing tools, while also learning about digital citizenship and responsible online behavior.

Objectives / Topics Covered	● Use event blocks to trigger actions in a program. ● Apply sequence skills to create animations and interactive scenes. ● Use drawing tools to create artwork within a program. ● Understand and apply responsible digital behavior. ● Use show and hide blocks to create interactive program elements.
Lessons	**Drawing Tools: Fairy Tale Painting** ● Use painting tools to create a fairy-tale scene. **Introduction to Events** ● Create a program using different types of events. **Sequences: Snowball Fight** (EK-PRO-RD-06) ● Create a program using multiple sequences. **Draw and Tell** (EK-PRO-PD-04) ● Create a short scene in CodeHopJr and write/dictate a sentence about the scene they created. **Our Responsibilities** (EK-PRO-VD-05) ● Use sequences to program two characters to explain how to be responsible in school and at home. **Introduction to Show and Hide Blocks** (2 classes) ● Use "show" and "hide" blocks in a sequence to make characters appear and disappear.

Unit 5: Pages (2 lessons)

In this module, students learn to create multi-page programs and use navigation blocks to move between them.

Objectives / Topics Covered	● Create programs with multiple pages and use navigation blocks to move between them.
Lessons	**Introduction to Pages** ● Create a program with multiple pages. **Using the Go to Page Block** ● Create a program including a "go to page" block to switch from page to page in an activity.

Unit 6: Block Exploration (3 lessons)

In this module, students explore new block types including sound, motion, and design, while also applying debugging strategies to find and fix errors in their programs.

Objectives / Topics Covered	● Explore a variety of block types to add sound, motion, and interactivity. ● Apply the design process to plan and create a program. ● Identify and fix errors in programs using debugging strategies.
Lessons	**Transportation Speeds and Sounds** ● Use "say" or "sound" blocks and speed blocks to program modes of transportation. **Introduction to the Design Process** (EK-ALG-IM-03) ● Identify and participate in the steps of the design process with guidance to solve a simple problem through a programming animation. **Debugging: Events and Motion** (EK-PRO-TR-07) ● Find and correct bugs in sequences.

Unit 7: Loops (2 lessons)

In this module, students are introduced to loops as a way to repeat actions in a program. Students learn how loops save time and effort by automating repetitive steps, and use forever loops to create ongoing animations.

Objectives / Topics Covered	● Explain that a loop repeats a set of instructions a specified number of times. ● Use loop blocks to simplify programs by replacing repeated sequences. ● Apply forever loops to create continuous animations.
Lessons	**Loops** ● Create a program using loops and explain how loops are used to repeat code. **Forever Loops: Fireworks** ● Create sequences that are repeated while the program runs.

Unit 8: Computing Systems (2 lessons)

In this module, students explore how computers work and the difference between hardware and software.

Objectives / Topics Covered	● Identify basic hardware components of a computer (e.g., keyboard, mouse, screen). ● Distinguish between hardware and software with everyday examples.
Lessons	**Computer Basics: Introduction (Unplugged)** (EK-SYS-HW-11) ● Learn what a computer is, how we use it, and what to do when it doesn't work. They will also be able to identify input, output, hardware, and software. **Types of Software (Unplugged)** (EK-SYS-HW-11) ● Compare and select appropriate software applications to complete different computing tasks.

Unit 9: Safety & Responsibility (2 lessons)

In this module, students learn how to stay safe and responsible when using technology.

Objectives / Topics Covered	● Recognize the importance of keeping personal information private online. ● Demonstrate responsible and respectful technology use.
Lessons	**Keeping Information Safe (Unplugged)** (EK-SYS-SE-12) ● Identify private and personal information. **Introduction to Responsible Tech Use (Unplugged)** (EK-SYS-IM-13)

	● Identify ways to use technology safely and responsibly, including understanding an Acceptable Use Policy (AUP).

Unit 10: Data & Analysis (1 lesson)

In this module, students explore how data is collected and used.

Objectives / Topics Covered	● Collect, organize, and interpret simple data sets.
Lessons	**What Can Data Tell Us?** (EK-DAT-DC-08, EK-DAT-DI-09) ● Collect, organize, and analyze data about school transportation.

Unit 11: Impacts of Computing (2 lessons)

In this module, students examine the impact of computing on daily life and careers.

Objectives / Topics Covered	● Describe how computing affects daily life and the world around us. ● Explore computer science careers and their impact on communities.
Lessons	**Impacts of Computing: Daily Life (Unplugged)** (EK-SOC-HI-14, EK-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: Spacecraft Design** (EK-SOC-HU-15) ● Explain the advantages of using technology in different careers and create a space program to animate an astronaut and a spacecraft.

Unit 12: AI Exploration (3 lessons)

In this module, students take their first steps into artificial intelligence, learning how AI systems recognize patterns and are trained to make decisions.

Objectives / Topics Covered	● Explain what artificial intelligence is and how it learns from examples. ● Demonstrate how machine learning systems are trained to classify information.
Lessons	**What is AI?** ● Identify two examples of AI in their daily lives and explain one way a smart machine is different from a regular tool. **Sorting with Decision Trees (Unplugged)** (EK-ALG-ML-02, EK-DAT-IM-10) ● Explain how AI uses data to learn and make decisions and create a simple decision tree to sort items based on rules. **Machine Learning: What is a Blorg?** (EK-ALG-ML-02) ● Explain how AI can learn information by being trained to identify an alien.

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**.	