Kindergarten Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Kindergarten 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 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, each approximately 30-45 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 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/20681/overview?lang=en.

A clickable PDF can be found at https://codehs.com/K5CSRoadmaps.

Course Breakdown

Unit 1: Unplugged Exploration (7 weeks)

In this optional unit, students explore foundational computer science concepts through unplugged activities that introduce sequencing, events, loops, and computational thinking in a hands-on, screen-free way.

Objectives / Topics Covered	● Create and follow step-by-step sequences to solve problems or complete tasks. ● Understand events as triggers for actions. ● Use loops to repeat simple instructions. ● Apply computational thinking to real-world routines. ● Transition into coding with unplugged programming practice.
Lessons	**Sequences (Unplugged)** ● Create a step-by-step sequence of instructions by choreographing a dance routine. **Coding Card Game: Sequences** ● Work collaboratively to guide Scout through a maze by arranging a sequence of instruction cards. **Coding Card Game: Sequences 2** ● Continue developing sequencing skills through a new maze challenge using coding cards. **Acting with Events** ● Act out real-world examples of how events trigger actions, laying the foundation for event-based programming. **Computational Thinking: Morning Routines** ● Break down a morning routine into steps using computational thinking skills like sequencing and pattern recognition. **Unplugged Introduction to Programming: Part 1** ● Practice creating and sequencing simple programs without a device, preparing for digital programming work. **Unplugged Introduction to Programming: Part 2** ● Learn how loops can be used to repeat sequences in unplugged activities that simulate programming behavior.

Unit 2: Programming Exploration (6 weeks)

In this unit, students begin programming through a story-based adventure with Scout, where they learn to navigate the interface, build sequences, animate scenes, and incorporate events.

Objectives / Topics Covered	● Navigate the programming interface and add, edit, or delete characters and backgrounds. ● Use motion and event blocks to animate characters and build sequences. ● Create multi-scene stories by adding pages. ● Develop storytelling and problem-solving skills.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Log in and explore the CodeHop Playground as an introduction or warm-up before diving into a full project. **Scout Adventures 1: Introducing Scout** ● Explore the programming interface and add Scout or other characters to the stage. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and new pages to expand the story world with programming. **Scout Adventures 3: Scout Meets a Friend** ● Modify and delete characters to continue shaping the animated story. **Scout Adventures 4: Scout Explores the Forest** ● Use motion blocks to animate Scout as the character moves around the stage. **Scout Adventures 5: Scout and Bluebird Help**

	● Build a sequence of motion blocks to help characters move and collect objects. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene by combining characters, backgrounds, pages, motion blocks, and events in a complete story.

Unit 3: Sequences and Events (9 weeks)

Students use programming to build animated scenes and interactive stories by combining sequences, events, visual effects, and creative tools—while also connecting coding skills to real-world applications like fashion design.

Objectives / Topics Covered	● Create programs using sequences, events, and visual effect blocks. ● Customize sprites and scenes using painting tools. ● Use grow, shrink, show, and hide blocks to animate changes. ● Explore how coding is used in creative careers like fashion and design.
Lessons	**Drawing Tools: Fairy Tale Painting** ● Use painting tools to create and decorate a fairy-tale scene. **Introduction to Events** ● Create a program that uses different types of event blocks to trigger actions. **Sequences: Snowball Fight** ● Build a program using multiple sequences to animate a snowball fight scene. **Events: Submarine Sequences** ● Combine event and motion blocks to program an underwater animation. **Weather and Seasons** ● Use sequences in a program to display the weather during different seasons. **Introduction to Show and Hide Blocks** ● Use “show” and “hide” blocks in a sequence to make characters appear and disappear. **Introduction to Grow and Shrink Blocks (2-part lesson)** ● Create a program that uses “grow” and “shrink” blocks to change the size of characters. **Growing Garden** ● Model how a garden grows using sequences that include grow, shrink, and hide blocks.

Unit 4: Pages (3 weeks)

In this unit, students learn how to create multi-page programs and use navigation blocks to move between scenes, building more complex and interactive stories.

Objectives / Topics Covered	● Create programs that span multiple pages. ● Add storytelling elements by linking scenes. ● Use the “go to page” block to navigate between pages in an organized sequence.
Lessons	**Introduction to Pages** ● Create a program with multiple pages to extend a story or activity across scenes. **Pages: Dragon Story** ● Design a multi-page program that tells a simple dragon-themed story. **Using the Go To Page Block** ● Use the “go to page” block to program smooth transitions between pages in an interactive project.

Unit 5: Block Exploration (9 weeks)

In this unit, students explore a variety of blocks—including speed, sound, and motion—to build interactive and personalized animations, games, and problem-solving projects.

Objectives / Topics Covered	● Use speed and sound blocks to animate and personalize characters. ● Debug simple programs that use events and motion blocks. ● Apply the design process to solve a problem through animation. ● Build interactive projects and games using a variety of blocks.
Lessons	**Introduction to Speed Blocks** ● Create a program that uses different speed blocks to animate how fast characters move. **Animal Sounds** ● Use “say” or “sound” blocks to make a character speak or make animal noises. **Transportation Speeds and Sounds** ● Program different vehicles using “say,” “sound,” and speed blocks to simulate movement and noise. **All About Me! (2-part lesson)** ● Create a program that shares favorite things using character speech and visuals. **Debugging: Events and Motion** ● Find and fix errors in a program that uses motion and event blocks. **Bowling Game (2-part lesson)** ● Apply coding skills to create a simple interactive bowling game. **Introduction to the Design Process** ● Follow the design process to plan, build, and improve an animation that solves a simple problem.

Unit 6: Loops (4 weeks)

In this unit, students explore how loops can be used in programming to simplify code and create repeating actions, using both repeat and forever loops in engaging animation projects.

Objectives / Topics Covered	● Use loops to repeat instructions and simplify programs. ● Differentiate between repeat loops and forever loops. ● Apply loops to animated scenes and interactive projects.
Lessons	**Loops** ● Create a program using loops and explain how they help repeat code efficiently. **Loops: Frog and Rabbit** ● Use loops to animate Frog and Rabbit characters repeating movements. **Forever Loops: Fireworks** ● Build a continuous animation using forever loops to create a fireworks display. **Loops: Play in the Park** ● Use both repeat and forever loops to animate playful scenes in a park setting.

Unit 7: Culmination Projects (4 weeks)

In this unit, students apply their understanding of programming concepts by building creative projects that showcase interactive storytelling and game design.

Objectives / Topics Covered	● Use sequences, events, and loops to build interactive scenes and games. ● Apply creative thinking and programming skills in culminating projects. ● Demonstrate mastery of learned concepts through original design and animation.
Lessons	**Chicken Crossing Game (2-part lesson)** ● Program an interactive player character and adjust difficulty settings in a simple game. **Wildlife Scene Project (2-part lesson)** ● Create a dynamic wildlife scene using sequences, events, and loops to animate animals and their environment.

Unit 8: Digital Literacy (5 weeks)

In this supplemental unit, students build foundational digital literacy by exploring how computers work, practicing basic input skills, learning responsible technology use, and understanding how AI makes decisions using data.

Objectives / Topics Covered	• Identify computer components and how to use them. • Practice using a mouse and keyboard effectively. • Understand responsible and safe technology use, including Acceptable Use Policies. • Explore how AI uses data to make decisions through simple decision trees.
Lessons	**Computer Basics: Introduction** • Learn what a computer is, how we use it, how to handle basic issues, and identify input, output, hardware, and software components. **Mouse Practice** • Practice clicking and dragging in multiple games to strengthen mouse control and accuracy. **Keyboard Introduction** • Use letters, numbers, and basic keyboard functions to complete simple digital tasks. **Introduction to Responsible Technology Use** • Learn how to use technology safely and responsibly, including understanding school-appropriate use policies. **Sorting with Decision Trees** • Explore how AI makes decisions by using data and building a simple decision tree to sort items based on defined rules.

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**.	