



Oklahoma Computer Science: Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Oklahoma Computer Science: Kindergarten 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 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 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 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/OK_K/overview



A clickable PDF can be found at <https://codehs.com/OK-K-5Roadmaps>

Course Breakdown

Optional Unplugged Exploration

Students explore foundational coding concepts away from the computer by building step-by-step sequences of instructions and acting out how events trigger actions.

Objectives / Topics Covered	• Create step-by-step sequences of instructions • Work together to solve mazes using coding cards • Act out how events trigger actions
Lessons	Sequences (Unplugged) • Create a sequence of step-by-step instructions for a dance. Coding Card Game: Sequences • Work together to build a sequence of instructions that moves Scout through a maze. Coding Card Game: Sequences 2 • Practice building longer maze sequences with the coding card game. Acting with Events • Act out how an event can trigger an action.

Unit 1: Getting Started (5 lessons)

Students learn to log in and use a mouse and keyboard, discover how computers work, and build habits for safe, responsible technology use and computational thinking.

Objectives / Topics Covered	• Use a mouse and keyboard effectively • Identify input, output, hardware, and software • Use technology safely and responsibly • Apply computational thinking to everyday routines
Lessons	Welcome to CodeHop! • Learn to log in and explore the CodeHopJr Playground. Mouse Practice (K.CS.HS.01) • Demonstrate mouse skills by clicking and dragging in games. Keyboard Introduction (K.CS.HS.01) • Use the letters, numbers, and basic functions of the keyboard. Computer Basics: Introduction (K.CS.HS.01, K.CS.T.01, K.IC.C.01) • Learn what a computer is and identify input, output, hardware, and software. Introduction to Responsible Technology Use (K.CS.D.01, K.IC.SI.01) • Identify ways to use technology safely and responsibly, including an Acceptable Use Policy. Computational Thinking: Morning Routines (K.AP.A.01) • Break down morning routines into patterns and step-by-step sequences.

Unit 2: Programming Exploration (6 lessons)

Through a story-driven adventure with Scout, students explore the programming interface and learn to add, modify, and move characters, backgrounds, and pages.

Objectives / Topics Covered	• Navigate the block programming interface • Add and modify characters, backgrounds, and pages • Build sequences of motion blocks to move characters
Lessons	Scout Adventures 1: Introducing Scout (K.CS.D.01) • Explore the programming interface and add characters.

	Scout Adventures 2: Scout Starts Exploring (K.CS.D.01) - Add backgrounds and a page to a program. Scout Adventures 3: Scout Meets a Friend (K.CS.D.01) - Delete and modify characters in a program. Scout Adventures 4: Scout Explores the Forest (K.CS.D.01) - Use motion blocks to move characters around the stage. Scout Adventures 5: Scout and Bluebird Help (K.CS.D.01) - Build a sequence of motion blocks to collect objects on the stage. Scout Adventures 6: Scout Celebrates with Friends (K.CS.D.01) - Create a celebration scene using characters, pages, backgrounds, and motion sequences with events.
--	--

Unit 3: Sequences and Events (7 lessons)

Students create programs that use sequences and events, explore drawing and character blocks, and collect and analyze simple data.

Objectives / Topics Covered	- Create programs that use events to trigger actions - Build programs with multiple sequences - Collect, organize, and analyze simple data - Use blocks to change how characters appear
Lessons	Drawing Tools: Fairy Tale Painting - Use painting tools to create a fairy-tale scene. Introduction to Events (K.AP.A.01, K.AP.C.01, K.AP.PD.03) - Create a program using different types of events. Sequences: Snowball Fight (K.AP.PD.03) - Create a program using multiple sequences. What Can Data Tell Us? (K.DA.CVT.01, K.DA.IM.01) - Collect, organize, and analyze data about school transportation. Introduction to Show and Hide Blocks (2 classes K.AP.A.01, K.AP.C.01) - Use “show” and “hide” blocks in a sequence to make characters appear and disappear. Introduction to Grow and Shrink Blocks - Change the size of characters using “grow” and “shrink” blocks.

Unit 4: Pages (3 lessons)

Students create programs with multiple pages and learn to move between them using the “go to page” block.

Objectives / Topics Covered	- Create programs with multiple pages - Use the “go to page” block to switch between pages - Tell stories that span multiple pages
Lessons	Introduction to Pages (K.AP.C.01, K.AP.PD.03) - Create a program with multiple pages. Pages: Dragon Story (K.AP.C.01, K.AP.PD.01, K.AP.PD.03) - Build a multi-page dragon story program. Using the Go To Page Block (K.AP.A.01, K.AP.PD.03) - Use the “go to page” block to switch from page to page in an activity.

Unit 5: Block Exploration (5 lessons)

Students expand their block vocabulary with speed and sound blocks, create a program all about themselves, and practice finding and fixing bugs.

Objectives	Animate characters using speed blocks
------------	---

/ Topics Covered	• Program characters to speak using “say” and “sound” blocks • Find and correct bugs in sequences
Lessons	Introduction to Speed Blocks (K.AP.A.01, K.AP.C.01, K.AP.PD.03) • Animate characters at different speeds using speed blocks. Animal Sounds (K.AP.A.01, K.AP.C.01) • Program a character to speak using “say” or “sound” blocks. All About Me! (2 classes K.AP.A.01, K.AP.C.01, K.AP.PD.01) • Create a program that shares information about their favorites. Debugging: Events and Motion (K.AP.PD.03, K.AP.PD.04) • Find and correct bugs in sequences.

Unit 6: Loops (3 lessons)

Students learn how loops are used to repeat code and use standard and forever loops to animate their programs.

Objectives / Topics Covered	• Explain how loops are used to repeat code • Create programs using loops • Use forever loops to create continuous animations
Lessons	Loops (K.AP.A.01, K.AP.C.01, K.AP.PD.03) • Create a program using loops and explain how loops repeat code. Loops: Frog and Rabbit (K.AP.C.01) • Use loops to repeat code in a program. Forever Loops: Fireworks (K.AP.C.01) • Create firework sequences that repeat while the program runs.

Unit 7: Culmination Project (2 lessons)

Students bring together everything they have learned to plan and build an interactive wildlife scene.

Objectives / Topics Covered	• Combine events, sequences, and loops in one project • Plan and build an interactive scene • Share and reflect on a finished project
Lessons	Wildlife Scene Project (2 classes K.AP.A.01, K.AP.C.01, K.AP.PD.03) • Create a wildlife scene using events, sequences, and loops.

Unit 8: Digital Literacy (5 lessons)

Students learn how to keep information safe, research using digital sources, understand how computers store data, and describe how networks connect people and devices.

Objectives / Topics Covered	• Identify private and personal information • Find information using research sources • Model how computers store data as files • Describe how networks connect people and devices
Lessons	Keeping Information Safe (K.NI.C.01) • Identify private and personal information. Introduction to Research (2 classes K.AP.PD.02) • Find information using research sources and create a program to share it visually. Introduction to Data Storage and Files (K.AP.V.01, K.DA.S.01) • Recognize that computers store data as files and model how data is collected and stored.

	Using Networks to Connect (K.NI.NCO.01) Describe how people and devices share information using networks, with and without wires.
--	--

Oklahoma Computer Science: Kindergarten 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 .	