



Louisiana Computer Science: Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Louisiana Computer Science: Kindergarten 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 27 required class sessions across five standards-aligned units, plus optional lessons in math and science, for a total of roughly 36 class sessions of about 30 minutes each — a full year of instruction when taught once per week. The units follow the categories of the Louisiana computer science standards, and teaching Units 1 through 5, Computing Systems through Impacts of Computing, is what is required to fully meet those standards. The optional Thinking Across Disciplines lessons in math and science can be taught in any order and integrated throughout the school year as needed.

Standards Alignment Note: Lessons that address a specific standard list it in bold in parentheses after the lesson title. 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.

Assessments: Some units conclude with a brief 2 question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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/29031/overview>.



A clickable PDF can be found at <http://codehs.com/LA-CSRoadmaps>.

Course Breakdown

Optional Review

An optional introductory lesson that helps students get comfortable logging in and moving around the CodeHopJr Playground. It can be taught on its own or used as a warm-up right before a full lesson.

Objectives / Topics Covered	• Log in to a CodeHop account • Navigate the CodeHopJr Playground • Explore block-based coding tools
Lessons	Welcome to CodeHop! • Log in and explore the CodeHopJr Playground before beginning the course.

Unit 1: Computing Systems (1 lesson)

Students are introduced to what a computer is and how people use one. They identify hardware and software, describe how computing devices help with everyday tasks, and learn what to do when a device stops working.

Objectives / Topics Covered	• Identify hardware and software • Describe how people use computing devices for tasks • Distinguish input from output • Recognize a basic hardware problem
Lessons	Computer Basics: Introduction (E.CS.1A, E.CS.1B, E.CS.1C, E.CS.2A) • Identify the parts of a computer and describe what to do when a device does not work. Computing Systems Quiz (2 questions)

Unit 2: Networks and the Internet (2 lessons)

Students explore how people and devices connect to share information across a network, with and without wires. They also begin telling apart private information from information that is safe to share.

Objectives / Topics Covered	• Define a network and describe its purpose • Explain how devices connect with and without wires • Distinguish private information from public information • Describe how usernames and passwords protect information
Lessons	Using Networks to Connect (E.NI.1A) • Describe how people and devices connect and share information over a network. Keeping Information Safe (E.NI.2A, E.NI.2B) • Identify which personal information should stay private and why. Networks and the Internet Quiz (2 questions)

Unit 3: Data and Analysis (4 lessons)

Students collect, sort, and organize data to answer questions about the world around them. They also learn that computers store data as files, and practice finding information through simple research.

Objectives / Topics Covered	• Collect and organize data into categories • Record and analyze data in a table or chart • Recognize that computers store data as files • Find information using research sources
-----------------------------------	--

Lessons	What Can Data Tell Us? (E.DA.1A, E.DA.2A, E.DA.2B, E.DA.4A, E.DA.4B, E.DA.5B) - Collect, sort, and analyze data about how classmates travel to school. Introduction to Data Storage and Files (E.DA.3A, E.DA.3B) - Model how a computer saves, names, and reopens a data file. Introduction to Research (2 classes E.DA.1B, E.AP.4D) - Find information using research sources and build a program to share what they learned. Data and Analysis Quiz (2 questions)
---------	---

Unit 4: Algorithms and Programming (17 lessons)

Students build the core programming skills of the course, moving from acting out algorithms unplugged to writing block-based programs. They work with events, sequences, loops, speed blocks, and variables, and practice finding and fixing bugs along the way.

Objectives / Topics Covered	- Create and follow step-by-step algorithms - Program with events, sequences, and loops - Find and fix bugs in a program - Plan a program using the design process
Lessons	Computational Thinking: Morning Routines (E.AP.3B, E.AP.4E) - Break a daily routine into ordered steps and spot the patterns inside it. Acting with Events (E.AP.1B) - Act out how an event triggers an action. Introduction to Events - Build a program that responds to several different types of events. Events: Submarine Sequences - Combine event and motion blocks to animate an underwater scene. Sequences (Unplugged) (E.AP.4C) - Write a sequence of step-by-step instructions for a class dance. Sequences: Snowball Fight - Program multiple sequences that run together in a single scene. Debugging: Events and Motion (E.AP.4B) - Find and correct bugs in a sequence. Introduction to Speed Blocks - Animate characters using different speed blocks. Transportation Speeds and Sounds - Program modes of transportation using speed blocks paired with say and sound blocks. Bowling Game (2 classes) - Apply coding skills to build an interactive bowling game. Loops (E.AP.2A) - Use loops to repeat code and explain why repeating code is useful. Loops: Frog and Rabbit - Repeat code with loops to animate two characters. What Is a Variable? (E.AP.1A) - Model how programs store and change data using variables. All About Me! (2 classes) - Create a program that tells classmates about their favorite things. Introduction to the Design Process (E.AP.3A, E.AP.4A, E.AP.4F) - Follow the steps of the design process to solve a simple problem with an animation. Algorithms and Programming Quiz (2 questions)

Unit 5: Impacts of Computing (3 lessons)

Students consider how computing devices shape daily life and what it means to use them responsibly. They practice recognizing kind and unkind online behavior and learn the rules that keep them safe at school.

Objectives / Topics Covered	- Identify computing devices used in daily life - Explain how technology has changed how people live and work - Recognize cyberbullying and know who to tell - Describe safe and responsible technology use
Lessons	Impacts of Computing: Daily Life (E.IC.1A) - Identify everyday computing devices and how they change the way people live and work. Being Kind Online (E.IC.2A, E.IC.2B) - Tell the difference between kind and unkind online messages, including cyberbullying. Introduction to Responsible Technology Use (E.IC.3A, E.IC.3B, E.IC.3C) - Discuss school technology rules and what it means to follow an Acceptable Use Policy. Impacts of Computing Quiz (2 questions)

Thinking Across Disciplines - Math (Optional)

Optional lessons that apply coding to kindergarten math content, programming animations and interactive scenes that model shapes, place value, comparison, and number decomposition. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	- Program shapes with specified attributes - Model addition and subtraction story problems - Represent groups of ones, fives, and tens - Compare single-digit numbers using events
Lessons	Identify Shapes by Attributes (E.DA.5A) - Draw and identify shapes by their attributes in a program. Story Problems: Add and Subtract within 10 - Build a scene that acts out an addition or subtraction story problem. Place Value: Ones and Tens - Animate groups of one, five, and ten to model place value. Greater Than and Less Than: Single-Digit Numbers - Use events to program a comparison of two numbers. Creating Shapes - Program shapes that respond to taps, then combine them into a more complex shape. Decompose Numbers Up to 10 - Create an interactive program that breaks numbers apart in different ways.

Thinking Across Disciplines - Science (Optional)

Optional lessons that use programming to explore kindergarten science concepts, animating living and nonliving things, seasonal weather, and the ways organisms meet their needs. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	- Classify items as living or nonliving - Program animations using sequences - Describe weather across the seasons - Explain how living things survive in their environment
Lessons	Living and Nonliving - Classify items as living or nonliving using tap events to trigger a character. Weather and Seasons - Program an animation showing the weather during different seasons. How Living Things Survive - Illustrate and explain how living things meet their needs in their environment.

Louisiana 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 .	